GBASE®

GBase 8s 管理员参考

www.gbase.cn

GBase 8s 管理员参考，南大通用数据技术股份有限公司

版权声明

免责声明

通讯方式

南大通用数据技术股份有限公司

天津市高新区开华道22号普天创新产业园东塔20-23层

电话：400-013-9696　　邮箱：info@gbase.cn

商标声明

目　　录

1 GBase 8s 管理员参考

这些主题包括对 GBase 8s 配置参数、sysmaster 数据库中系统监视接口（SMI）表、诸如 gadmin 和 gstat 之类数据库服务器实用程序的句法、逻辑日志记录、磁盘结构、事件警报，以及无编号错误消息的详尽描述。

这些是下列用户感兴趣的主题：

- 数据库管理员
- 系统管理员
- 性能工程师

编写这些主题时假设您已具有下列背景：

- 计算机、操作系统和操作系统提供的实用程序的知识
- 对关系数据库有一定使用经验，或对数据库概念有所了解
- 有数据库服务器管理、操作系统管理或网络管理方面的相关经验

2 配置和监视 GBase 8s

2.1 数据库配置参数

在初始化期间， GBase 8s 数据库服务器使用一个称为 onconfig 文件的配置文件。这个文件包含缺省配置参数值。您可以更改这些参数值来提高实例或数据库的性能和其他特性。ONCONFIG 环境变量指定您的 onconfig 文件。

2.1.1 onconfig 文件的格式

当要新增或修改 onconfig 文件的内容时,必须遵循该文档的编译规则。

对参数的描述和其可以赋的值都阐述在 onconfig.std 文件的注释中。

以下显示了参数行的语法：

```
PARAMETER_NAME parameter_value comments
```

onconfig文件内容的格式规定如下：

- 一行只能命名一个参数。
- 注释行以 # 为标志并位于行的开头。
- onconfig 文件的最大行限制为 512 字节，超出此限制的行将被截断，而且可能导致配置问题。
- 参数名、参数值和可选注释之间必须放置空格（制表符和/或空格）。请不要再参数值内部使用任何制表符或空格。在参数值和空格之后的字符串都将被视为注释，不论这些字符串的开头是否以 # 标记。
- onconfig 文件中的参数及其值是区分大小写的，参数名总是大写的，如果值项使用大写字母描述的，那么您必须使用大写（例如： NETTYPE 参数 的 CPU 值）。
- 大多数参数只能有一个有效的项。在 onconfig 文件中，如果一些参数有多项，系统将默认第一项有效。 然而，一些参数可以有多个项。 另一些参数，例如 VPCLASS 可以出现多次。
- 未识别的参数可以被保存，但是会被忽略并且不会报错。
- 复制但忽略未识别的参数，且不会报错。

提示：如果您在 onconfig 模板文件中运行一个像 grep 的程序,那么指定新行字符（^）只返回配置参数的名称和值。若没有新行字符，参数的描述也将会被返回。

例如：下面的命令不仅返回配置参数的描述，也返回它的值：

```
grep "MSGPATH" onconfig
# MSGPATH          - The path of the IDS message log file
MSGPATH $GBASEDBTDIR/tmp/online.log
```

然而，下面的命令只返回配置参数的值：

```
grep "^MSGPATH" onconfig
MSGPATH $GBASEDBTDIR /tmp/online.log
```

环境变量的约定

在环境变量适用的任何配置参数中，您可以输入一个环境变量作为值。例如：对于 DB DBSPACETEMP 配置参数您可以指定以下的环境变量来代替您数据库服务器的名称。

```
DBSPACETEMP        $MY_DBSPACETEMP
```

重要： 如果您输入一个环境变量作为值，您必须在任何可执行程序或读取 onconfig 文件的实用程序的环境中设置该环境变量。读取 onconfig 文件的实用程序包括 oninit、gcheck、gbackuprestore、gtape、glogdump 和 archecker。

更改配置文件

您可以更改数据库服务器的 onconfig 文件，来定制服务器功能或者调整数据库行为。缺省情况下，onconfig 文件位于 GBASEDBTDIR/conf/ 目录中。ONCONFIG 环境变量指定 onconfig 文件的名称和位置。

您可以通过下列任何一种方法更改 onconfig 文件：

- 您可以使用 GBase OpenAdmin Tool (OAT) for GBase 8s 监视和更新您的配置。OAT 提供配置参数值的建议来优化您的数据库服务器配置。
- 您可以使用文本编辑器更改配置参数值。在下一次数据库服务器关闭并重启之后，这些变更生效。
- 通过运行 gadmin -wf 来永久地更新配置参数，或者通过运行 gadmin -wm 命令来更新内存中的配置参数，您可以动态地更改许多配置参数的值，而无需重启数据库服务器。
- 通过运行 genoncfg 实用程序，您可以生成一个 onconfig 文件，根据您的估算来优化连接、磁盘空间，以及 CPU 使用情况设置。
- 您可以成组地导出、导入和修改配置参数：
  - 使用 gadmin -we 命令将当前配置的一个快照导出到一个文件。这个结果快照可以再归档，用作一个配置文件，或者导入到另一个正在运行的实例。
  - 使用 gadmin -wi 命令从一个先前导出的文件导入可调整的配置参数。忽略这个文件中那些不能动态地调整的配置参数。
- 您可以使用 SQL 管理 API 命令来修改、重置、导出和导入一个配置文件。
  - 使用 admin() 或 task() 函数的 **modify config** 参数来改变一个配置参数的值。

- 使用 admin() 或 task() 函数的 **export config** 和 **import config** 参数来导出或导入一个文件，这个文件包含一个或多个可动态地调整的配置参数。
- 使用 admin() 或 task() 函数的 **reset config** 或 **reset config all** 参数来将一个配置参数或全部配置参数恢复为 onconfig 文件中的值。

通过运行 onconfig_diff 实用程序，您可以比较两个 onconfig 文件。

显示 onconfig 文件中的设置

您可以使用若干个工具来显示 onconfig 文件中的设置。

请使用下列工具之一来显示 onconfig 文件中的设置：

- 使用一个文本编辑器打开 onconfig 文件。
- 使用 gstat -c 命令，或使用 GBase OpenAdmin Tool (OAT) for GBase 8s 查看 onconfig 文件的内容。
- 通过运行 gstat -g cfg 命令查看配置参数列表及其当前值。如果动态地更新配置参数，则当前值与 onconfig 文件中的永久值不同。

您可以使用 gstat -g cfg 命令的附加选项，来只显示那些动态变更了的配置参数，或显示关于全部配置参数的附加信息。

2.1.2 onconfig 门户：按功能性类别划分配置参数

这一节中的信息列出 UNIX™ onconfig 文件中的那些配置参数。

类别列表

要使用这一节，您首先要从如下列表中选定适当的类别，然后随着链接转到该类别的配置参数。这些类别的罗列顺序与 onconfig 文件中的顺序相同。不在 onconfig 文件中而您可以添加到 onconfig 文件中去的那些参数，罗列在 表 57 中。

- root dbspace 配置参数
- 物理日志配置参数
- 逻辑日志配置参数
- 长事务配置参数
- 服务器消息文件配置参数
- tblspace 配置参数
- 临时 dbspace 和 sbspace 配置参数
- dbspace 和 sbspace 配置参数
- 系统配置参数
- 网络配置参数
- CPU 相关的配置参数
- 自动调整配置参数
- AIO 和清除程序相关的配置参数

- 锁相关的配置参数
- 共享内存配置参数
- 检查点和系统锁配置参数
- 转换防护配置参数
- 事务相关的配置参数
- gtape 磁带设备配置参数
- gtape 逻辑日志磁带设备配置参数
- 备份和恢复配置参数
- 主存储管理器配置参数
- 数据字典高速缓存配置参数
- 数据分布配置参数
- 用户定义例程（UDR）配置参数
- SQL 语句高速缓存配置参数
- 操作系统会话相关的配置参数
- 索引相关的配置参数
- 并行数据库查询（PDQ）配置参数
- 优化器配置参数
- 扫描配置参数
- SQL 跟踪配置参数
- 安全配置参数
- 基于标签的访问控制配置参数
- 内建字符数据类型配置参数
- 顺序高速缓存配置参数
- 高可用性和 Enterprise Replication 安全配置参数
- Enterprise Replication 配置参数
- 高可用性集群配置参数
- 逻辑恢复配置参数
- 诊断转储配置参数
- 报警程序配置参数
- 技术支持配置参数
- 字符处理配置参数
- 统计配置参数
- 用户映射配置参数
- 存储提供配置参数
- 数据库对象的自动定位
- 缺省转义字符配置参数
- Websphere MQ 服务器配置参数
- 非 root 用户服务器安装配置参数
- 低内存配置参数
- GSKit 配置参数
- 连接参数
- Java 配置参数
- 缓冲池和 LRU 调整配置参数
- 附加参数

root dbspace 配置参数

使用下列配置参数来配置 root dbspace.

表 1. Root dbspace 配置参数

配置参数	参考
ROOTNAME 配置参数	root dbspace 名称。
ROOTPATH 配置参数	root dbspace 的路径。
ROOTOFFSET 配置参数	root dbspace 的偏移量。
ROOTSIZE 配置参数	root dbspace 的大小。
MIRROR 配置参数	启用或禁用镜像。
MIRRORPATH 配置参数	镜像 root dbspace 的路径。
MIRROROFFSET 配置参数	镜像 root dbspace 的偏移量。

物理日志配置参数

使用下列配置参数来配置物理日志。

表 2. 物理日志配置参数

配置参数	参考
PHYSFILE 配置参数	物理日志的大小。
PLOG_OVERFLOW_PATH 配置参数	物理日志文件的溢出目录。
PHYSBUFF 配置参数	物理日志缓冲区的大小。

逻辑日志配置参数

使用下列配置参数来配置逻辑日志。

表 3. 逻辑日志配置参数

配置参数	参考
LOGFILES 配置参数	逻辑日志文件的数量。
LOGSIZE 配置参数	每个逻辑日志文件的大小。
DYNAMIC_LOGS 配置参数	动态日志分配的类型。
LOGBUFF 配置参数	逻辑日志缓冲区的大小。

长事务配置参数

使用下列配置参数来控制长事务回滚的时间。

表 4. 长事务配置参数

配置参数	参考
LTXHWM 配置参数	在一个长事务回滚之前可以填满逻辑日志文件的百分比。
LTXEHWM 配置参数	为了使一个长事务排他地使用日志，在服务器挂起其他活动之前可以填满逻辑日志文件的百分比。

服务器消息文件配置参数

使用下列配置参数来配置服务器消息文件。

表 5. 服务器消息文件配置参数

配置参数	参考
MSGPATH 配置参数	消息文件的路径。
CONSOLE 配置参数	控制台消息文件的路径。

tblspace 配置参数

使用下列配置参数来在 root dbspace 中配置 tblspace。

表 6. tblspace 配置参数

配置参数	参考
TBLTBLFIRST 配置参数	tblspace tblspace 的第一个 extent 的大小。
TBLTBLNEXT 配置参数	tblspace tblspace 的下一个 extent 的大小。
TBLSPACE_STATS 配置参数	启用或禁用 tblspace 统计。

临时 dbspace 和 sbspace 配置参数

使用下列配置参数来配置缺省的临时 dbspace 和 sbspace。

表 7. 临时 dbspace 和 sbspace 配置参数

配置参数	参考
DBSPACETEMP 配置参数	临时对象的 dbspace 列表。
SBSPACETEMP 配置参数	临时智能大对象的 sbspace 列表。

dbspace 和 sbspace 配置参数

使用下列配置参数来配置缺省的 dbspace 和 sbspace。

表 8. 缺省的 dbspace 和 sbspace 配置参数

配置参数	参考
SBSPACENAME 配置参数	存储智能大对象的缺省 sbspace。
SYSSBSPACENAME 配置参数	系统统计的缺省 sbspace。
ONDBSPACEDOWN 配置参数	指定当一个 dbspace 不可用时服务器的行为。

系统配置参数

使用下列配置参数来设置服务器实例信息。

表 9. 系统配置参数

配置参数	参考
SERVERNUM 配置参数	数据库服务器实例的唯一 ID。
FULL_DISK_INIT 配置参数	防止一个现有服务器实例的意外磁盘重新初始化。

网络配置参数

使用下列配置参数来配置网络。

表 10. 网络配置参数

配置参数	参考
NETTYPE 配置参数	一个特定协议的轮询线程配置。
LISTEN_TIMEOUT 配置参数	数据库服务器等待一次连接的时间。
MAX_INCOMPLETE_CONNECTIONS 配置参数	不完整连接的最大数目。
FASTPOLL 配置参数	启用或禁用快速轮询。
NUMFDSERVERS 配置参数	对于 UNIX 上的网络连接，使用 NUMFDSERVERS 配置参数来指定某些轮询线程的最大数目，这些线程处理正在 VP 之间移动的网络连接。
NS_CACHE 配置参数	定义单个条目的最大保留时间，条目位于主机名/IP 地址高速缓存、服务高速缓存、用户高速缓存和组高速缓存之中。

CPU 相关的配置参数

使用下列配置参数来配置 CPU 虚拟处理器。

表 11. CPU 虚拟处理器配置参数

配置参数	参考

配置参数	参考
MULTIPROCESSOR 配置参数	设置为 1 支持多个 CPU VP。
VPCLASS 配置参数	定义每一个 CPU 虚拟处理器类的特性
VP_MEMORY_CACHE_KB 配置参数	CPU 虚拟处理器的专用内存块数量。
SINGLE_CPU_VP 配置参数	设置为 0 来启动用户定义 CPU VP，或为单个 CPU VP 设置为 1。

自动调整配置参数

使用下列配置参数来自动地调整数据库服务器的配置。

表 12. CPU 自动调整配置参数

配置参数	参考
AUTO_TUNE 配置参数	启用或禁用全部自动调整配置参数，在 onconfig 文件里没有这些参数的值。
AUTO_LRU_TUNING 配置参数	启用或禁用 LRU 队列的自动调整。
AUTO_AIOVPS 配置参数	启用或禁用 AIO 虚拟处理器的自动管理。
AUTO_CKPTS 配置参数	启用或禁用自动检查点。
AUTO_REPREPARE 配置参数	启用或禁用自动地重新优化存储过程和重新准备已准备就绪的语句。
AUTO_STAT_MODE 配置参数	为您的系统启用或禁用选择性地更新统计模式。
AUTO_READAHEAD 配置参数	更改查询的自动预读模式或者禁用或启用查询的自动预读。

AIO 和清除程序相关的配置参数

使用下列配置参数来配置 AIO 虚拟处理器和缓冲区清除程序。

表 13. AIO 和缓冲区清除程序配置参数

配置参数	参考
VPCLASS 配置参数	配置 AIO 虚拟处理器。
CLEANERS 配置参数	页清除程序线程的数目。
DIRECT_IO 配置参数（UNIX）	指定是否使用直接 I/O。

锁相关的配置参数

使用下列配置参数来设置锁行为。

表 14. 锁配置参数

配置参数	参考
LOCKS 配置参数	启动时锁的初始数目。
DEF_TABLE_LOCKMODE 配置参数	缺省的表锁模式。

共享内存配置参数

使用下列配置参数来配置共享内存。

表 15. 共享内存配置参数

配置参数	参考
RESIDENT 配置参数	控制共享内存是否驻留。
SHMBASE 配置参数	共享内存基本地址。不要更改该值。
SHMVIRTSIZE 配置参数	共享内存虚拟部分的初始大小，以 KB 为单位。
SHMADD 配置参数	虚拟共享内存部分的大小。
EXTSHMADD 配置参数	用户定义例程和 DataBlade 的每一个虚拟扩展共享内存部分的大小，这些例程运行在用户定义虚拟处理器中。
SHMTOTAL 配置参数	数据库服务器共享内存的最大数量。
SHMVIRT_ALLOCSEG 配置参数	控制添加一个内存部分的时间。
SHMNOACCESS 配置参数	罗列服务器不能访问的那些共享内存地址。

检查点和系统锁配置参数

使用下列配置参数来配置检查点、恢复时间目标和系统阻塞时间。

表 16. 检查点、恢复时间目标和系统阻塞时间配置参数

配置参数	参考
CKPTINTVL 配置参数	如果需要一个检查点，那么检查频率如何。
RTO_SERVER_RESTART 配置参数	故障后重启的恢复时间目标。
BLOCKTIMEOUT 配置参数	系统阻塞的时间量。

转换防护配置参数

在升级到新服务器版本期间，使用下列配置参数来控制 GBase 8s 使用的信息。

表 17. 转换防护配置参数

配置参数	参考
CONVERSION_GUARD 配置参数	如果在升级期间发生错误，指定是停止升级还是继续升级。

配置参数	参考
RESTORE_POINT_DIR 配置参数	当启用 CONVERSION_GUARD 时，在一次失败的升级期间，将路径名指定到一个放置恢复点文件的空目录。

事务相关的配置参数

使用下列配置参数来控制分布式事务。

表 18. 分布式事务配置参数

配置参数	参考
TXTIMEOUT 配置参数	分布式事务超时期间。
DEADLOCK_TIMEOUT 配置参数	在一个分布式事务中等待锁的最大时间量。
HETERO_COMMIT 配置参数	为使用 EGM 网关的事务启用或禁用异类提交。

gtape 磁带设备配置参数

使用下列参数来配置用 gtape 实用程序备份的磁带设备。

表 19. 磁带设备配置参数

配置参数	参考
TAPEDEV 配置参数	用于备份的磁带设备。
TAPEBLK 配置参数	磁带块大小。
TAPESIZE 配置参数	在一备份磁带上放置的最大数据量。

gtape 逻辑日志磁带设备配置参数

使用下列配置参数来通过 gtape 实用程序配置逻辑日志的磁带设备。

表 20. 逻辑日志磁带设备配置参数

配置参数	参考
LTAPEDEV 配置参数	用于逻辑日志备份的磁带设备。
LTAPEBLK 配置参数	逻辑日志备份的磁带块大小。
LTAPESIZE 配置参数	在一逻辑日志备份磁带上放置的最大数据量。

备份和恢复配置参数

使用下列配置参数来通过 ON-Bar 实用程序控制备份和恢复。如未作其他指定，在《GBase 8s 备份与恢复指南》中记录这些配置参数。

表 21. ON-Bar 配置参数

配置参数	参考
BAR_ACT_LOG 配置参数	ON-Bar 活动日志文件的位置。
BAR_DEBUG_LOG 配置参数	ON-Bar 调试日志文件的位置。
BAR_DEBUG 配置参数	ON-Bar 的调试级别。
BAR_MAX_BACKUP 配置参数	一备份中使用的备份线程数目。
BAR_RETRY 配置参数	再次尝试备份或恢复的次数。
BAR_NB_XPORT_COUNT 配置参数	每一备份进程使用的数据缓冲区数目。
BAR_XFER_BUF_SIZE 配置参数	每个数据缓冲区的大小。
RESTARTABLE_RESTORE 配置参数	失败后启用 ON-Bar 来继续备份。
BAR_PROGRESS_FREQ 配置参数	将进展消息放置到活动日志中的频率。
BAR_BSALIB_PATH 配置参数	ON-Bar 和存储管理器的共享库路径。
BACKUP_FILTER 配置参数	备份期间使用的过滤程序路径。
RESTORE_FILTER 配置参数	恢复期间使用的过滤程序路径。
BAR_PERFORMANCE 配置参数	报告的 ON-Bar 性能统计类型。
BAR_CKPTSEC_TIMEOUT 配置参数	在辅助服务器中等待完成归档检查点的秒数。

主存储管理器配置参数

使用下列配置参数来配置 GBase 8s 主存储管理器。

表 22. GBase 8s 主存储管理器配置参数

配置参数	参考
PSM ACT_LOG 配置参数	如果您不想在 ON-Bar 活动日志中包括日志信息的话，指定 GBase 8s 主存储管理器活动日志的位置。
PSM DEBUG_LOG 配置参数	如果您不想在 ON-Bar 调试日志中包括日志信息的话，指定 GBase 8s 主存储管理器调试日志的位置。
PSM DEBUG 配置参数	如果您想使用一个与 ON-Bar 所用调试级别不同的级别的话，指定打印在 GBase 8s 主存储管理器调试日志中的信息量。
PSM CATALOG 配置参数	指定包含 GBase 8s 主存储管理器目录表的目录的完整路径。
PSM DBS POOL 配置参数	指定 GBase 8s 主存储管理器放置备份和恢复 dbspace 数据池的名称。
PSM LOG POOL 配置参数	指定 GBase 8s 主存储管理器放置备份和恢复日志数据池的名称。

数据字典高速缓存配置参数

使用下列配置参数来配置数据字典高速缓存。

表 23. 数据字典高速缓存配置参数

配置参数	参考
DD_HASHSIZE 配置参数	数据字典高速缓存中散列存储区的数目。
DD_HASHMAX 配置参数	每个散列存储区中表的最大数目。

数据分布配置参数

使用下列配置参数来配置数据分布池。

表 24. 数据分布配置参数

配置参数	参考
DS_HASHSIZE 配置参数	在数据分布高速缓存和其他高速缓存中的散列存储区数目。
DS_POOLSIZE 配置参数	在数据分布高速缓存和其他高速缓存中的最大条目数。

用户定义例程（UDR）配置参数

使用下列配置参数来配置 UDR。

表 25. UDR 配置参数

配置参数	参考
PC_HASHSIZE 配置参数	在 UDR 高速缓存中的散列存储区数目。
PC_POOLSIZE 配置参数	在 UDR 高速缓存中的最大条目数目。
PRELOAD_DLL_FILE 配置参数	在服务器启动时加载的 C UDR 共享库路径名。

SQL 语句高速缓存配置参数

使用下列配置参数来配置 SQL 语句高速缓存。

表 26. SQL 语句高速缓存配置参数

配置参数	参考
STMT_CACHE 配置参数	控制 SQL 语句高速缓存。
STMT_CACHE_HITS 配置参数	在高速缓存之前运行 SQL 语句的次数。
STMT_CACHE_SIZE 配置参数	SQL 语句高速缓存的大小。
STMT_CACHE_NOLIMIT 配置参数	控制 SQL 语句高速缓存的额外内存消耗。
STMT_CACHE_NUMPOOL 配置参数	SQL 语句高速缓存的池数。

操作系统会话相关的配置参数

使用下列配置参数来配置操作系统和会话特征。

表 27. 操作系统和会话配置参数

配置参数	参考
USEOSTIME 配置参数	SQL 语句的计时精度。
STACKSIZE 配置参数	会话堆栈的大小。
ALLOW_NEWLINE 配置参数	在 SQL 语句中是否允许嵌入换行符。
USELASTCOMMITTED 配置参数	控制 committed read 隔离级别。

索引相关的配置参数

使用下列配置参数来配置索引特征。

表 28. 索引配置参数

配置参数	参考
FILLFACTOR 配置参数	索引页充满百分比。
MAX_FILL_DATA_PAGES 配置参数	如果数据页有可变长度行，启用或禁用尽可能填满数据页。
BTSCANNER 配置参数	配置 B-tree 扫描程序线程。
ONLIDX_MAXMEM 配置参数	pre-image 和 updator 日志池的内存数量。

并行数据库查询（PDQ）配置参数

使用下列配置参数来配置 PDQ。

表 29. PDQ 配置参数

配置参数	参考
MAX_PDQPRIORITY 配置参数	单一查询的最大资源百分比。
DS_MAX_QUERIES 配置参数	并发决策支持查询的最大数。
DS_TOTAL_MEMORY 配置参数	决策支持内存的最大量。
DS_MAX_SCANS 配置参数	决策支持扫描的最大数。
DS_NONPDQ_QUERY_MEM 配置参数	非 PDQ 查询内存的数量。
DATASKIP 配置参数	在处理查询时是否跳过一个 dbspace。

优化器配置参数

使用下列配置参数来影响查询执行优化器计划和伪指令。

表 30. 优化器配置参数

配置参数	参考
OPTCOMPIND 配置参数	控制优化器如何确定最佳查询路径。
DIRECTIVES 配置参数	启用或禁用 inline 优化器伪指令。
EXT_DIRECTIVES 配置参数	启用或禁用外部伪指令。
OPT_GOAL 配置参数	控制如何优化进行最快检索。
IFX_FOLDVIEW 配置参数	启用或禁用折叠视图。
STATCHANGE 配置参数	给一个更改阈的全局百分比指定一个正整数，来确定需要更新的数据分布统计。
USTLOW_SAMPLE 配置参数	当您以 LOW 模式运行 UPDATE STATISTICS 语句时，启用或者禁用基于抽样生成索引统计。

扫描配置参数

使用下列配置参数来设置预读行为。

表 31. 扫描配置参数

配置参数	参考
BATCHEDREAD_TABLE 配置参数	在压缩表上，在行大于一页的表上，以及 VARCHAR, LVARCHAR 和 NVARCHAR 数据表上启用或禁用轻扫描。
BATCHEDREAD_INDEX 配置参数	启用优化器来执行索引的轻扫描。

SQL 跟踪配置参数

使用下列配置参数来设置 SQL 跟踪。

表 32. SQL 跟踪配置参数

配置参数	参考
EXPLAIN_STAT 配置参数	启用或禁用在说明输出文件中包括查询统计。
SQLTRACE 配置参数	配置 SQL 跟踪。

安全配置参数

使用下列配置参数来配置安全选项。

表 33. 安全配置参数

配置参数	参考
DBCREATE_PERMISSION 配置参数	指定可以创建数据库的用户。

DB_LIBRARY_PATH 配置参数	指定 UDR 或 UDT 共享库的位置。
IFX_EXTEND_ROLE 配置参数	控制如何指定哪些用户可以注册外部例程。
SECURITY_LOCALCONNECTION 配置参数	数据库服务器是否检查本地连接的安全。
UNSECURE_ONSTAT 配置参数	非 DBSA 用户是否可以运行 **gstat** 命令。
ADMIN_USER_MODE_WITH_DBSA 配置参数	控制谁可以管理模式连接到服务器。
ADMIN_MODE_USERS 配置参数	罗列那些可以管理模式连接的用户。
SSL_KEYSTORE_LABEL 配置参数	SSL 标签。
TLS_VERSION 配置参数	指定网络连接的传输层安全（TLS）版本。

基于标签的访问控制配置参数

使用下列配置参数来配置基于标签的访问控制(LBAC)高速缓存。在 *GBase 8s 安全指南* 中记录这些配置参数。

表 34. LBAC 配置参数

配置参数	参考
PLCY_POOLSIZE 配置参数	在 LBAC 安全信息高速缓存中的散列存储区数目。
PLCY_HASHSIZE 配置参数	在 LBAC 安全信息高速缓存的每一个散列存储区中的最大条目数。
USRC_POOLSIZE 配置参数	在 LBAC 凭证内存高速缓存中的散列存储区数目。
USRC_HASHSIZE 配置参数	在 LBAC 凭证内存高速缓存的每一个散列存储区中的最大条目数。

内建字符数据类型配置参数

使用下列配置参数来配置内建字符数据类型。

表 35. 内建字符数据类型配置参数

配置参数	参考
SQL_LOGICAL_CHAR 配置参数	启用或禁用在内建字符数据类型声明中的大小规范的扩展。

顺序高速缓存配置参数

使用下列配置参数来配置顺序高速缓存。

表 36. 顺序高速缓存数据类型配置参数

配置参数	参考

SEQ_CACHE_SIZE 配置参数	指定那些在顺序高速缓存中可以有预分配值的顺序对象的最大数目。

高可用性和 Enterprise Replication 安全配置参数

使用下列配置参数来配置高可用性集群和 Enterprise Replication 安全。

表 37. 高可用性和 Enterprise Replication 安全配置参数

配置参数	参考
ENCRYPT_HDR 配置参数	启用或禁用 HDR 加密。
ENCRYPT_SMX 配置参数	SDS 或 RSS 服务器的加密级别。
ENCRYPT_CDR 配置参数	Enterprise Replication 的加密级别。 请参阅《GBase 8s 性能指南》。
ENCRYPT_CIPHERS 配置参数	罗列加密算法和模式。
ENCRYPT_MAC 配置参数	消息认证代码（MAC）的级别。
ENCRYPT_MACFILE 配置参数	MAC 密钥文件的路径。
ENCRYPT_SWITCH 配置参数	切换加密算法和密钥的频率。

Enterprise Replication 配置参数

使用下列配置参数来配置 Enterprise Replication （ER）。在《GBase 8s Enterprise Replication 指南》中记录这些配置参数。

表 38. Enterprise Replication 配置参数

配置参数	参考
CDR_EVALTHREADS 配置参数	求值程序线程的数目。
CDR_DSLOCKWAIT 配置参数	数据同步线程等待数据库锁的时间量。
CDR_QUEUEMEM 配置参数	发送和接收队列的最大内存量。
CDR_NIFCOMPRESS 配置参数	网络接口压缩级别。
CDR_SERIAL 配置参数	serial 列的增量大小和起始值。
CDR_DBSPACE 配置参数	**syscdr** 数据库的 dbspace 名称。
CDR_QDATA_SBSPACE 配置参数	GBase 8s spooled 事务的 sbspace 名。
CDR_SUPPRESS_ATSRISWARN 配置参数	在 ATS 和 RIS 文件中不显示数据同步报警和错误。
CDR_DELAY_PURGE_DTC 配置参数	保留删除表的时间量。
CDR_LOG_LAG_ACTION 配置参数	在数据库服务器即将改写逻辑日志，而

配置参数	参考
	Enterprise Replication 尚未处理该日志的情况下，采取的行动。
CDR_LOG_STAGING_MAXSIZE 配置参数	Enterprise Replication 用于 GBase 8s stage 日志文件的最大空间量。
CDR_MAX_DYNAMIC_LOGS 配置参数	Enterprise Replication 可以在一个会话中发出的动态日志请求的最大数。
GRIDCOPY_DIR 配置参数	ifx_grid_copy 过程使用的缺省目录。
CDR_TSINSTANCEID 配置参数	复制的时间序列实例的唯一标识符。
CDR_MAX_FLUSH_SIZE 配置参数	在日志清空到磁盘之前应用的最大事务数。
CDR_AUTO_DISCOVER 配置参数	允许通过 cdr autoconfig serv、安装助手或 gclone 实用程序自动配置 Enterprise Replication。
CDR_MEM 配置参数	指定 Enterprise Replication 的内存池分配方法。

高可用性集群配置参数

使用下列配置参数来配置高可用性集群。

表 39. 高可用性集群配置参数

配置参数	参考
DRAUTO 配置参数	控制主服务器的自动故障转移。
DRINTERVAL 配置参数	缓冲区清空的最大间隔。
HDR_TXN_SCOPE 配置参数	在客户端应用程序、主服务器和 HDR 辅助服务器之间调整事务同步。
DRTIMEOUT 配置参数	网络超时期间。
DRLOSTFOUND 配置参数	HDR 失而复得文件的路径。
DRIDXAUTO 配置参数	启用或禁用自动索引修理。
HA_ALIAS 配置参数	高可用性集群的服务器别名。
HA_FOC_ORDER 配置参数	定义连接管理器使用的单个故障转移规则。
LOG_INDEX_BUILDS 配置参数	启用或禁用索引页日志。
SDS_ENABLE 配置参数	启用或禁用 SD 辅助服务器。
SDS_TIMEOUT 配置参数	主服务器等待 SD 辅助服务器响应的时间。
SDS_TEMPDBS 配置参数	SD 辅助服务器使用的临时 dbspace。
SDS_ALTERNATE 配置参数	在高可用性集群中的主服务器和 SD 辅助服务器之间

配置参数	参考
	通信的替代方法。
SDS_PAGING 配置参数	GBase 8s SD 辅助 paging 文件的路径。
SDS_LOGCHECK 配置参数	主服务器是否产生日志活动，以及允许或防止主服务器的故障转移。
UPDATABLE_SECONDARY 配置参数	辅助服务器是否接受客户端的更新、插入或删除操作。
FAILOVER_CALLBACK 配置参数	当辅助服务器引起向标准服务器或主服务器的切换时调用的程序。
TEMPTAB_NOLOG 配置参数	临时表的缺省日志模式。
DELAY_APPLY 配置参数	在 RS 辅助服务器上应用事务的延迟时间。
STOP_APPLY 配置参数	在 RS 辅助服务器上停止应用事务。
LOG_STAGING_DIR 配置参数	stage 日志文件的目录。
RSS_FLOW_CONTROL 配置参数	启用 RS 辅助服务器的流控制。
FAILOVER_TX_TIMEOUT 配置参数	启用或禁用故障转移期间的事务存活行为。
ENABLE_SNAPSHOT_COPY 配置参数	能否通过 gclone 实用程序克隆服务器实例。
SMX_COMPRESS 配置参数	在从源数据库服务器向目标数据库服务器发送数据时，数据库服务器使用的压缩级别。
SMX_PING_INTERVAL 配置参数	超时间隔的秒数。
SMX_PING_RETRY 配置参数	在辅助服务器关闭到主服务器的 SMX 连接之前的超时间隔数。
CLUSTER_TXN_SCOPE 配置参数	控制事务提交可以返回到客户端应用程序的时间。

逻辑恢复配置参数

使用下列配置参数来设置逻辑恢复线程。

表 40. 逻辑恢复配置参数

配置参数	参考
ON_RECVRY_THREADS 配置参数	在热恢复期间并行运行的逻辑恢复线程数。
OFF_RECVRY_THREADS 配置参数	在冷恢复中为了快速恢复使用的逻辑恢复线程数。

诊断转储配置参数

使用下列配置参数来控制诊断转储信息。

表 41. 诊断配置参数

配置参数	参考

DUMPDIR 配置参数	断言失败诊断文件的位置。
DUMPSHMEM 配置参数（UNIX）	控制共享内存转储。
DUMPGCORE 配置参数（UNIX）	启用或禁用数据库服务器是否把核心转出到 gcore 文件。
DUMPCORE 配置参数（UNIX）	启用或禁用是否在断言失败之后数据库服务器转储核心。
DUMPCNT 配置参数（UNIX）	一个会话的共享内存转储最大数目。

报警程序配置参数

使用下列配置参数来配置报警程序。

表 42. 报警程序配置参数

配置参数	参考
ALARMPROGRAM 配置参数	显示事件报警的报警程序。
ALRM_ALL_EVENTS 配置参数	是否为全部时间运行报警程序。
STORAGE_FULL_ALARM 配置参数	当存储空间变满或者在页或 extent 之外运行分区时，产生消息和时间的频率。
SYSALARMPROGRAM 配置参数	断言失败之后触发的系统报警程序。

技术支持配置参数

技术支持使用并自动设置下列配置参数。

表 43. 技术支持配置参数

配置参数	参考
RAS_PLOG_SPEED	为支持保留。
RAS_LLOG_SPEED	为支持保留。

字符处理配置参数

如果字符是语言环境有效的，使用下列配置参数来控制是否检查 GBase 8s 。

表 44. 字符处理配置参数

配置参数	参考
EILSEQ_COMPAT_MODE 配置参数	启用或禁用字符有效性检查。

统计配置参数

使用下列配置参数来控制队列和等待统计的收集。

表 45. 队列和等待统计配置参数

配置参数	参考
QSTATS 配置参数	启用或禁用收集队列统计。
WSTATS 配置参数	启用或禁用收集等待统计。

用户映射配置参数

使用该配置参数来控制用户映射。

表 46. 用户映射.

配置参数	描述
USERMAPPING 配置参数（UNIX，Linux）	映射用户能否连接到 GBase 8s ，如果可以，映射能否有管理权限。

存储提供配置参数

使用下列配置参数来控制信息，在现有的存储空间（dbspace、临时 dbspace、sbspace、临时 sbspace 或 blobspace）中需要更多空间时，该信息使服务器能够动态地扩展或添加一个 chunk。

表 47. 存储提供配置参数

配置参数	参考
SP_AUTOEXPAND 配置参数	在一个存储空间中启用或禁用 chunk 的自动创建或扩展。
SP_THRESHOLD 配置参数	定义在一个存储空间中可以存在的最小可用 KB 数。
SP_WAITTIME 配置参数	指定一个线程在返回一个“空间不足”错误之前等待存储池扩展的最大秒数。

数据库对象的自动定位

使用下列配置参数来启用自动定位和分片存储。

表 48. 自动定位配置参数

配置参数	参考
AUTOLOCATE 配置参数	启用数据库和表的自动定位以及表的自动分片存储。

缺省转义字符配置参数

按需要使用下列配置参数。

表 49. 缺省转义字符配置参数

配置参数	参考
DEFAULTESCCHAR 配置参数	指定一个缺省转义字符。

Websphere MQ 服务器配置参数

使用下列配置参数来配置 MQ 消息的数据库服务器。在《*GBase 8s 数据库扩展用户指南*》里记录这些配置参数。

表 50. MQ 配置参数

配置参数	参考
MQSERVER 配置参数	定义一个通道，指定 GBase WebSphere® MQ 服务器的位置，并指定使用的通信方法。
MQCHLLIB 配置参数	指定到目录的路径，该目录包含 GBase WebSphere MQ 客户端通道定义表。
MQCHLTAB 配置参数	指定 GBase WebSphere MQ 客户端通道定义表的名称。

非 root 用户服务器安装配置参数

使用下列配置参数进行非 root 服务器安装。

表 51. 非 root 用户服务器安装

配置参数	参考
REMOTE_SERVER_CFG 配置参数	指定罗列远程主机的文件名，数据库服务器计算机信任这些主机。
REMOTE_USERS_CFG 配置参数	指定罗列可信任用户名的文件名，这些用户在远程主机上。
S6_USE_REMOTE_SERVER_CFG 配置参数	指定在可信任网络环境中的用于认证安全服务器连接的文件。

低内存配置参数

使用下列配置参数来管理低内存

表 52. 低内存配置参数

配置参数	参考
LOW_MEMORY_RESERVE 配置参数	保留特定数量的内存，以供需要执行关键活动并且服务器可用内存有限时使用。
LOW_MEMORY_MGR 配置参数	更改达到内存限制时服务器的缺省行为。

GSKit 配置参数

使用该参数来设置 GBase Global Security Kit (GSKit) 版本。

表 53. GSKit

配置参数	描述
GSKIT_VERSION 配置参数	指定数据库服务器使用的 GBase Global Security Kit (GSKit) 版本。

连接参数

使用下列参数来管理连接。

表 54. 连接配置参数.

配置参数	描述
GBASEDBTCONRETRY 配置参数	指定在初始连接尝试失败后数据库服务器可以进行的连接尝试次数。通过 GBASEDBTCONTIME 配置参数，指定 CONNECT 语句试图连接到数据库服务器的频率。
GBASEDBTCONTIME 配置参数	指定 CONNECT 语句尝试建立到数据库服务器连接的秒数。通过 GBASEDBTRETRY 配置参数，指定 CONNECT 试图连接到数据库服务器的频率。

Java™ 配置参数

使用下列配置参数来配置 Java 虚拟处理器。在《 *J/Foundation 开发者指南*》中记录这些配置参数。

表 55. Java 配置参数

配置参数	参考
VPCLASS	配置一个 Java 虚拟处理器类。
JVPPROPFILE	Java VP 特性文件。
JVPLOGFILE	Java VP 日志文件。
JVPARGS	配置 Java VM。
JVPCLASSPATH	Java 类路径。

缓冲池和 LRU 调整配置参数

使用下列配置参数来配置缓冲池和调整 LRU 队列。

表 56. 缓冲池和 LRU 调整配置参数

配置参数	参考
BUFFERPOOL 配置参数	配置缓冲池。

附加参数

有些配置参数不在 onconfig.std 文件中。如有必要，您可以将这些参数添加到 onconfig 文件中。

表 57. onconfig.std 文件中没有的参数

配置参数	参考
AUTO_TUNE_SERVER_SIZE 配置参数	根据预期的用户数，设置数据库服务器的大小。 如果您在安装期间创建服务器，这个参数设置在 onconfig 文件中。
AUTO_LLOG 配置参数	在指定的 dbspace 中自动地添加逻辑日志，来提高性能并限定逻辑日志文件总计的大小。 如果您在安装期间创建服务器，该参数设置在 onconfig 文件中。
CDR_APPLY 配置参数	指定数据同步线程的最小数目和最大数目。
CDR_ENV 配置参数	设置一些特定的 Enterprise Replication 环境变量。
CHECKALLDOMAINSFORUSER 配置参数	指定在 Windows™ 网络环境中数据库服务器如何查找用户名。
DISABLE_B162428_XA_FIX 配置参数	指定在回滚操作之后是否释放全局事务。
DRDA_COMMBUFFSIZE 配置参数	指定 DRDA 通信缓冲区的大小。
IFX_XA_UNIQUEXID_IN_DATABASE 配置参数	在同一个数据库服务器实例中，启用事务管理器来使用相同的 XID 表达不同数据库上的全局事务。
LIMITNUMSESSIONS 配置参数	指定可以连接到数据库服务器的最大会话数。
MSG_DATE 配置参数	在打印到 online 日志的消息开始出插入一个日期戳记。
NET_IO_TIMEOUT_ALARM 配置参数	如果网络写操作受阻 30 分钟或更长时间，则发送通知。
PN_STAGEBLOB_THRESHOLD 配置参数	为轮转法分片中的 BYTE 和 TEXT 数据保留空间。
SESSION_LIMIT_LOCKS 配置参数	在单一会话中允许那些没有管理权限的用户使用锁的最大数目。

2.1.3 ADMIN_MODE_USERS 配置参数

除了用户 gbasedbt 和 DBSA 组成员之外，ADMIN_MODE_USERS 配置参数指定您想在管理模式下访问数据库服务器的用户列表。

onconfig.std 值

未设置。只有用户 gbasedbt 和 DBSA 组成员可以在管理模式下访问 GBase 8s 。

分隔符

用逗号分隔用户名，比如：Karin,Sarah,Andrew，字符串最长 127 字节。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当通过运行 gadmin -wf 命令来动态地重置 onconfig 文件中的值时。

当通过运行 gadmin -wm 命令来重置内存中的值时。

用法

永久保存 ADMIN_MODE_USERS 配置参数中的用户列表。您可以使用 gadmin -wm 或 gadmin -wf 命令来移除用户。

在数据库运行时，使用 gadmin -j -U 命令来允许一个或多个用户在管理模式下访问数据库服务器。

您必须将 ADMIN_USER_MODE_WITH_DBSA 配置参数设置为 1 以使得罗列在 ADMIN_MODE_USERS 配置参数中的用户能够在管理模式下连接到数据库服务器。

2.1.4 ADMIN_USER_MODE_WITH_DBSA 配置参数

除了 gbasedbt 之外，ADMIN_USER_MODE_WITH_DBSA 配置参数指定哪些用户可在管理模式下连接到数据库服务器。

onconfig.std 值

未设置。只有用户 gbasedbt 可在管理模式下连接到数据库服务器。

值

0 = 只有用户 gbasedbt 可在管理模式下连接

1 = 如果未设置 ADMIN_USER_MODE 配置参数，则下列用户可在管理模式下连接：

- 用户 gbasedbt
- DBSA 组成员

如果 ADMIN_USER_MODE 配置参数设置为一个或多个用户名的列表，那么下列用户可在管理模式下连接：

- 用户 gbasedbt
- 组列表中包含 gbasedbt 组的那些用户（仅 UNIX™）
- DBSA 组成员
- 罗列在 ADMIN_MODE_USERS 配置参数中的管理用户

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.5 ALARMPROGRAM 配置参数

使用 ALARMPROGRAM 配置参数来指定 alarmprogram 文件的完整路径名，该文件处理事件报警并控制逻辑日志备份。

onconfig.std 值

在 UNIX™ 上：$GBASEDBTDIR/etc/alarmprogram.sh

在 Windows™ 上：%GBASEDBTDIR%\etc\alarmprogram.bat

如未出现

在 UNIX 上：$GBASEDBTDIR/etc/no_log.sh

在 Windows 上：%GBASEDBTDIR%\etc\no_log.bat

值

pathname = alarmprogram 文件的完整路径名。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

您可设置 ALRM_ALL_EVENTS 配置参数，指定是对于在 MSGPATH 中保存日志的全部事件还是只对于特定的值得通知的事件（事件严重性大于 1）来运行 ALARMPROGRAM 配置参数。如果不存在 ALARMPROGRAM 配置参数指定的脚本，则由缺省报警处理器、 no_log.sh 或 no_log.bat 取而代之。 在正确的脚本就位后，更新 ALARMPROGRAM 配置参数的值来指定该脚本。您可在服务器 online 时通过使用 gadmin -wm 命令完成该更新。

提供下列样例脚本。

表 1. 样例脚本

脚本名称 (UNIX)	脚本名称 (Windows)	描述
log_full.sh	log_full.bat	ALARMPROGRAM 设置为 log_full.sh 或 log_full.bat，当数据库服务器发出日志已满事件

脚本名称（UNIX）	脚本名称（Windows）	描述
		报警时，自动地备份逻辑日志。 您可修改该脚本，并设置到 onconfig 文件中 ALARMPROGRAM 的完整路径。
no_log.sh	no_log.bat	ALARMPROGRAM 设置为 no_log.sh 或 no_log.bat，禁用自动逻辑日志备份。
alarmprogram.sh	alarmprogram.bat	处理事件报警和控制逻辑日志备份。修改 alarmprogram.sh 或 alarmprogram.bat 并将 ALARMPROGRAM 设置为 alarmprogram.sh 或 alarmprogram.bat 的完整路径名。请参阅 定制 ALARMPROGRAM 脚本。

不使用提供的脚本，您可编写自己的 shell 脚本、批处理文件或二进制程序来执行事件。将 ALARMPROGRAM 设置为该文件的完整路径名。当值得通知的事件发生时，数据库服务器执行该脚本。这些事件包括数据库、表、索引或简单大对象故障；全部日志已满；内部子系统故障；初始化失败；以及长事务。您可通过电子邮件或传呼邮件消息通知这些事件。

将 ALARMPROGRAM 设置为 $GBASEDBTDIR/etc/alarmprogram.sh

或 %GBASEDBTDIR%\etc\alarmprogram.bat 并修改相应文件，来生成事件报警。

重要： 当您选择自动逻辑日志备份时，备份介质应随时用于备份进程。

如果您通过 ALARMPROGRAM 参数建立自动日志备份，则不要使用连续日志备份命令 gbackuprestore -b -l -C。

2.1.6 ALLOW_NEWLINE 配置参数

使用 ALLOW_NEWLINE 配置参数来允许或不允许在所有会话的加引号字符串中出现换行符。要在分布式查询中允许所有远程会话支持嵌入换行符，在 onconfig 文件中指定 ALLOW_NEWLINE。

onconfig.std 值

ALLOW_NEWLINE 0

值

0 = 不允许在所有会话的加引号字符串中出现换行符。

1 = 允许在所有会话的加引号字符串中出现换行符。

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

您可指定要数据库服务器允许在所有会话或一个特定会话的加引号字符串中出现换行符（\n）。会话是客户端连接到数据库服务器的持续时间。

当未设置 ALLOW_NEWLINE 时，要想允许或不允许当前会话的加引号字符串中出现换行符，您可以执行内建 ifx_allow_newline() 例程，其唯一的参数为 't' 或 'f'。

- 't' 启用支持加引号字符串内部的换行符。
- 'f' 效果相反。

调用 ifx_allow_newline() 只影响调用那个例程的用户会话。

2.1.7 ALRM_ALL_EVENTS 配置参数

使用 ALRM_ALL_EVENTS 配置参数来对于在 MSGPATH 配置参数中生成日志的所有事件或只对于值得通知的事件，指定 ALARMPROGRAM 配置参数是否运行。

onconfig.std 值

ALRM_ALL_EVENTS 0

值

0 = 只对于值得通知的事件。

1 = 该参数触发 ALARMPROGRAM 配置参数，且 ALRM_ALL_EVENTS 配置参数显示所有事件报警。

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.8 AUTO_AIOVPS 配置参数

当数据库服务器检测到现有 AIO VP 的性能跟不上 I/O 工作负载时，AUTO_AIOVPS 配置参数使得数据库服务器能自动地增加异步 I/O 虚拟处理器（AIO VP）和页清除程序线程的数目。

onconfig.std 值

未设置。如果 AUTO_TUNE 配置参数设置为 1，则自动地增加 AIO VP 和页清除程序线程。

值

0 = 关

1 = 开

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 重置内存中的该值时。

如果在当前 onconfig 文件中未设置一个 AUTO_AIOVPS 值，您编辑 AUTO_TUNE 配置参数并重启数据库服务器。

用法

VPCLASS aio 配置参数控制 AIO VP 的数目，如果在 onconfig 文件中未设置 VP aio 参数，则当启用 AUTO_AIOVPS 时，数据库服务器启动的 AIO VP 初始数目等于 AIO chunk 的数目。 如果未设置 VP aio，则数据库服务器最多可以启动 128 个 AIO VP。

2.1.9 AUTO_CKPTS 配置参数

AUTO_CKPTS 配置参数允许服务器更频繁地触发检查点来避免事务阻塞。

onconfig.std 值

未设置。如果 AUTO_TUNE 配置参数设置为 1，则启用自动检查点。

值

0 = 关

1 = 开

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 在 onconfig 文件里动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

如果在当前 onconfig 中未设置 AUTO_CKPTS 值，编辑 AUTO_TUNE 配置参数并重启数据库服务器。

2.1.10 AUTO_LLOG 配置参数

使用 AUTO_LLOG 配置参数来在特定的 dbspace 中自动地添加逻辑日志，以提高性能。

onconfig.std 值

不在 onconfig.std 文件中。

如果您在安装期间创建服务器的缺省值

```
AUTO_LLOG 1,llog,max_size
```

max_size 值依赖于 AUTO_TUNE_SERVER_SIZE 配置参数的值。

值

0 = 缺省。禁用。不自动地添加逻辑日志来提高性能。

1, *dbspace_name*, *max_size*

- 1 = 启用。在需要提高性能时自动地添加逻辑日志。

- *dbspace_name* = 将逻辑日志文件添加到的那个 dbspace 的名称。该 dbspace 必须有操作系统的缺省页大小。
- *max_size* = 可选。缺省是 2048000 KB（2 GB）。所有逻辑日志文件的最大 KB 数，包括存储在 dbspace *dbspace_name* 的任何逻辑日志文件。当大小达到最大时，数据库不再添加逻辑日志文件来提高性能。如果未指定 *max_size*，AUTO_TUNE_SERVER_SIZE 配置参数的设置影响大小的最大值。请参阅“用法”部分。

分隔符

以逗号分隔这些区域。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

如果您在安装期间创建了一个服务器，则自动地启用 AUTO_LLOG 配置参数。为逻辑日志创建一个名为 llog 的 dbspace。安装程序根据 AUTO_TUNE_SERVER_SIZE 配置参数值来设置初始大小和 dbspace 的 *max_size* 选项值。您可以通过重置 AUTO_LLOG 配置参数值来更改 *max_size* 选项。

如果您在安装期间未创建服务器，则当数据库服务器检测到添加逻辑日志文件可提高性能时，您可以启用 AUTO_LLOG 配置参数来自动地添加逻辑日志文件。为了获得理想的性能，从 root dbspace 和物理日志在单独一个磁盘上选择一个 dbspace。

当启用 AUTO_LLOG 配置参数时，一旦逻辑日志不足导致过高的检查点百分比、阻塞检查点或长检查点，数据库服务器会添加逻辑日志。

当逻辑日志文件大小达到最大时，不再添加逻辑日志文件来提高性能。然而，如果启用 DYNAMIC_LOGS 配置参数，则会添加逻辑日志以防事务阻塞。DYNAMIC_LOGS 和 AUTO_LLOG 配置参数的设置不相互作用。类似地，您可继续手工添加逻辑日志文件。

如果 *max_size* 域值大于指定 dbspace 的大小，则要确保您的存储池有可用空间。

示例

下列设置使得能自动添加逻辑日志文件，直到逻辑日志文件大小为 204800 KB 并将逻辑日志文件的 dbspace 设置为 llog：

```
AUTO_LLOG 1,llog,204800
```

2.1.11 AUTO_TUNE_SERVER_SIZE 配置参数

使用 AUTO_TUNE_SERVER_SIZE 配置参数来根据预期的并发用户数设置要分配的内存和存储空间大小。

onconfig.std 值

不在 onconfig.std 文件中。

缺省值

未设置。

在安装期间创建服务器的值

依赖于您在安装程序中指定的用户数。

值

SMALL = 1 - 100 用户

MEDIUM = 101 - 500 用户

LARGE = 501 - 1000 用户

XLARGE = 多于 1000 用户

生效

如果您在安装期间创建一个服务器。

编辑 onconfig 文件并首次重启数据库服务器之后。

用法

如果您在安装期间创建一个服务器，则指定数据库服务器的预期用户数。将 AUTO_TUNE_SERVER_SIZE 配置参数设置成相应的大小，会影响下列特性的大小：

- 缓冲池的大小。
- 在数据库停止为了提高性能而自动地添加逻辑日志之前，逻辑日志文件大小的最大值。
- 下列已创建的存储空间的初始大小，这些存储空间是在安装期间自动创建的：
  - 物理日志的可扩展 plogspace
  - 逻辑日志的 dbspace
  - 数据库和表的 dbspace
  - 临时 dbspace
  - sbspace
  - 临时 sbspace

下表说明 AUTO_TUNE_SERVER_SIZE 配置参数值如何影响大小。

表. 对内存和存储空间分配的影响.

值	缓冲池（BUFFERPOOL）大小的最大值	自动地创建的存储空间的初始大小	逻辑日志文件（AUTO_LLOG）大小的最大值
SMALL	可用共享内存的 10%	50 MB	200 MB
MEDIUM	20%	100 MB	500 MB
LARGE	33%	200 MB	1 GB
XLARGE	50%	500 MB	2 GB

如果您在安装期间未创建一个数据库，或在初始化服务器后首次更改 AUTO_TUNE_SERVER_SIZE 配置参数的值，则新的值仅会影响下列特性的大小：

- 缓冲池的大小，如果 BUFFERPOOL 配置参数的设置包括 memory='auto' 选项。
- 在服务器停止为了提高性能而自动地添加逻辑日志之前，所有逻辑日志大小的最大值。

2.1.12 AUTO_LRU_TUNING 配置参数

使用 AUTO_LRU_TUNING 配置参数来启用自动 LRU 调整，自动地为页替换维护足够的干净页。

onconfig.std 值

未设置。如果设置 AUTO_TUNE 配置参数为 1，则启动自动 LRU 调整。

值

0 = 关

1 = 开

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中自动地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

如果在当前的 onconfig 文件中未设置 AUTO_LRU_TUNING 值，编辑 AUTO_TUNE 配置参数并重启数据库服务器。

用法

自动 LRU 调整更改影响所有缓冲池并调整 BUFFERPOOL 配置参数中 lru_min_dirty 和 lru_max_dirty 值。

2.1.13 AUTO_READAHEAD 配置参数

使用 AUTO_READAHEAD 配置参数来更改自动预读模式或禁用一个查询的自动预读操作。

onconfig.std 值

未设置。如果 AUTO_TUNE 配置参数设置为 1，则在标准模式下自动地执行预读。

值

一个 0 - 2 的整数指定模式，可选地后跟一个逗号和一个整数，指定自动地请求预读的页数。例如，值 1,4096 在标准模式下启用自动预读，每次 4096 页。

0 = 禁用自动预读请求。

1 = 在标准模式下启用自动预读请求。只有当查询等待 I/O 时，数据库服务器才自动地处理预读请求。

2 = 在 GBase 8s 积极（aggressive）模式下启用自动预读请求。在查询开始时数据库服务器自动地处理预读请求，并在查询期间持续进行。

number_of_pages = 4-4096，指出自动地请求预读的页数。缺省是 128 页。

分隔符

用逗号分隔模式和页数。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 重置内存中的该值时。

如果在当前 onconfig 文件中未设置 AUTO_READAHEAD 值，编辑 AUTO_TUNE 配置参数并重启数据库服务器。

用法

当数据库服务器检测到查询遇到 I/O 时，自动预读操作通过发出异步页请求帮助提高查询性能。通过将查询处理与从磁盘检索数据并放入缓冲池所需的处理相叠加，异步页请求可提高查询性能。

通常，缺省值 1 适用于大多数生产环境。

在任何特定的环境中，执行 GBase 8s 积极（aggressive）预读取操作都不会明显优于标准预读操作。不过 GBase 8s 积极（aggressive）预读略微有效些：

- 对于一些读取少量数据的扫描
- 有些情况下，您会在关预读与开预读之间切换，对于小扫描关预读，对于较大扫描开预读。
- 对于只看少量行的扫描，因为服务器立即执行预读操作而不是等待扫描遇到 I/O。

因为扫描碰到高速缓存数据的 GBase 8s pocket，有些扫描可能会关闭预读操作再开启，GBase 8s 积极（aggressive）预读操作不关闭预读操作。

只有在测试了两种设置并知道 GBase 8s 积极（aggressive）预读操作更有效的情况下，才使用 GBase 8s 积极（aggressive）预读操作。如果不肯定积极预读操作更有效的话，则不要使用。

您可使用 SQL 的 SET ENVIRONMENT 语句的 AUTO_READAHEAD 环境选项来启用或禁用会话的 AUTO_READAHEAD 配置参数值。

预读设置的优先次序如下：

1. 会话的 SET ENVIRONMENT AUTO_READAHEAD 语句。
2. AUTO_READAHEAD 配置参数值为 1 或 2。
3. 如果在 onconfig 文件中未出现 AUTO_READAHEAD 配置参数值，当服务器完成查询时，服务器在 128 数据页上执行预读（等同于 AUTO_READAHEAD 模式设置为 1）。

2.1.14 AUTO_REPREPARE 配置参数

在 SPL 例程引用表模式之后，或引用表模式的已准备就绪的对象更改之后，AUTO_REPREPARE 配置参数控制数据库服务器是否自动地重新优化 SPL 例程和重新准备已准备就绪的对象。

onconfig.std 值

未设置。如果 AUTO_TUNE 配置参数设置为 1，则自动地重新优化 SPL 例程，并自动地重新准备已准备就绪的对象。

值

0 = 在修改一个直接引用表或间接引用表的模式之后，禁用已准备就绪对象的自动重新准备。还会在修改间接引用表的模式之后禁用 SPL 例程的自动重新优化。

1 = 启用自动重新准备和自动重新优化。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

如果在当前 onconfig 文件中未设置 AUTO_REPREPARE 值，编辑 AUTO_TUNE 配置参数并重启数据库服务器。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

通过启用 AUTO_REPREPARE 配置参数，您可以避免许多 -710 错误，并减少重新准备和重新优化操作的数目。修改 SPL 例程中动态 SQL 语句或 DML 语句引用的表模式之后，您必须明确地执行这些重新准备和重新优化操作。

例如，某些 DDL 语句修改表模式，诸如 CREATE INDEX、DROP INDEX、DROP COLUMN 和 RENAME COLUMN。如果在运行这些 DDL 语句时禁用了 AUTO_REPREPARE 配置参数，则用户会收到 -710 错误。 这些错误还会发生，如果您下次运行：

- 直接引用或间接引用被 DDL 语句修改的表的 SPL 例程。
- 引用被 DDL 语句修改的表的已准备就绪对象。

限制：

启用 AUTO_REPREPARE 可能不会影响已准备就绪的语句或引用表的 SPL 例程，DDL 操作改变这些表中的列数，或者一列的数据类型。这些模式更改之后，对于引用模式已经更改的表的 SPL 例程，您往往一定会重新发出 DESCRIBE 语句、PREPARE 语句（对于已准备就绪的对象）和 UPDATE STATISTICS 语句（对于例程相关的游标）。否则，不管 AUTO_REPREPARE 的设置，数据库服务器都会发出 SQL 错误 -710。

2.1.15 AUTO_STAT_MODE 配置参数

使用 AUTO_STAT_MODE 配置参数来启用或禁用有选择地更新模式，仅更新 UPDATE STATISTICS 操作中分布的过时的或丢失的数据，而不是更新分布的所有数据统计。

onconfig.std 值

未设置。如果 AUTO_TUNE 配置参数设置为1，则有选择地更新统计。

值

0 = 禁用有选择的 UPDATE STATISTICS 操作。

1 = 启用有选择的 UPDATE STATISTICS 操作。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

如果在当前 onconfig 文件中未设置 AUTO_STAT_MODE 值，且设置 AUTO_TUNE 配置参数。

用法

当 AUTO_STAT_MODE 配置参数或 AUTO_STAT_MODE 会话环境变量启用有选择地更新自动模式时，仅更新 UPDATE STATISTICS 操作中分布的过时的或丢失的数据， 数据库服务器使用 STATCHANGE 配置参数值来确定需要更新的表或分片分布统计。

2.1.16 AUTO_TUNE 配置参数

使用 AUTO_TUNE 配置参数来启用或禁用所有自动调整配置参数，这些参数的值不出现在 onconfig 文件中。

onconfig.std 值

AUTO_TUNE 1

值

0 = 禁用

1 = 启用

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

如果在当前 onconfig 文件中未设置单独的自动调整配置参数，则数据库服务器为那个配置参数使用在 AUTO_TUNE 配置参数中指定的值。

自动调整配置参数是：

- AUTO_AIOVPS
- AUTO_CKPTS
- AUTO_LRU_TUNING
- AUTO_READAHEAD
- AUTO_REPREPARE
- AUTO_STAT_MODE

如果在当前 onconfig 文件里设置自动调整配置参数，则数据库服务器使用 onconfig 文件中的值。AUTO_TUNE 配置参数不更改那个值。

onconfig 文件在 %GBASEDBTDIR%\conf 或 $GBASEDBTDIR%/conf 目录中。

示例

示例1：假定有些自动调整配置参数未设置，其他配置参数有值：

AUTO_LRU_TUNING （值未设置）

AUTO_STAT_MODE （值未设置）

AUTO_LRU_CKPTS （值未设置）

AUTO_AIOVPS 0

AUTO_REPREPARE 1

AUTO_READAHEAD 0

如果您设置 AUTO_TUNE 配置参数为 1，数据库服务器自动地更改未设置为 1 的那些值。以前设置的值保持不变。现在自动调整配置参数值如下：

AUTO_LRU_TUNING 1

AUTO_STAT_MODE 1

AUTO_CKPTS 1

AUTO_AIOVPS 0

AUTO_REPREPARE 1

AUTO_READAHEAD 0

示例2：假定所有自动调整配置参数都设置如下值：

AUTO_LRU_TUNING 1

AUTO_STAT_MODE 1

AUTO_LRU_CKPTS 1

AUTO_AIOVPS 0

AUTO_REPREPARE 1

AUTO_READAHEAD 0

在这种情况下，AUTO_TUNE 配置不更改任何值。

示例3：假定您从 onconfig 文件移除了自动调整配置参数，但现在想使用它们。您可设置 AUTO_TUNE 为 1 来重新启用所有自动调整配置参数。

2.1.17 AUTOLOCATE 配置参数

使用 AUTOLOCATE 配置参数来启用数据库、索引和表的自动定位，并启用自动表分片。

onconfig.std 和缺省值

AUTOLOCATE 0

值

0 = 禁用自动定位和分片。

1 - 32 = 启用自动定位和分片。数值表明初始分配给标的轮转法分片的多少。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置内存中的该值时。

当您通过运行 gadmin -wm 命令动态地重置内存中的该值时。

用法

使用 AUTOLOCATE 配置参数来控制数据库服务器是否控制新数据库、索引和表的定位，是否控制那些表的分片。如果您设定 AUTOLOCATE 配置参数为一个正整数，则数据库服务器执行下列任务：

- 将新数据库存储在您未在优化 dbspace 中而是在 root dbspace 中指定的位置。缺省情况下，除了那些专用于 tenant 数据库的 dbspace 之外，所有 dbspace 都可用。然而，您可控制可用 dbspace 列表。
- 通过轮转法分片新表，分片数目等于 AUTOLOCATE 配置参数的值。
- 随着表增长，添加更多表分片。

如果您设置 AUTOLOCATE 配置参数的值为 0，则缺省地在 root dbspace 中创建新数据库。新表和索引创建在数据库所在的同一个 dbspace 中，新表和索引不分片。

自动定位不适用于 tenant 数据库或 tenant 数据库内的表、分片和索引。

通过在 CREATE DATABASE 语句中以 IN 子句指定一个 dbspace，您可覆盖数据库的自动定位。类似地，通过在 CREATE TABLE 语句中以 IN 子句指定一个 dbspace 或以 FRAGMENT BY 子句指定一个分片策略，您可覆盖表的自动定位和分片。

当启用这个配置参数时，您可使用 admin() 或 task() 函数的 autolocate database 参数来：

- 管理自动定位和分片的 dbspace 列表。可用 dbspace 列表在 sysautolocate 系统目录表中。
- 对于指定的数据库禁用自动定位和分片。

您可使用 SQL 的 SET ENVIRONMENT 语句的 AUTOLOCATE 环境选项来启用或禁用会话的 AUTOLOCATE 配置参数值。

2.1.18 BATCHEDREAD_INDEX 配置参数

使用 BATCHEDREAD_INDEX 配置参数来启用优化器执行索引的轻扫描。这会减少读缓冲区的次数，从而提高性能。

onconfig.std 值

BATCHEDREAD_INDEX 1

值

0 = 禁用索引的轻扫描。

1 = 启用索引的轻扫描。

生效

在您编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

2.1.19 BATCHEDREAD_TABLE 配置参数

使用 BATCHEDREAD_TABLE 配置参数来启用或禁用在压缩表、行大于一页的表以及 VARCHAR、LVARCHAR 和 NVARCHAR 数据表上的轻扫描。

onconfig.std 值

BATCHEDREAD_TABLE 1

值

0 = 禁用可变记录长度表上的轻扫描。

1 = 启用可变记录长度表上的轻扫描。

此处将压缩表和行大于一页的表视同可变记录长度。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

除了压缩表、行大于一页的表和可变记录长度（比如 VARCHAR、LVARCHAR 和 NVARCHAR 列）的表之外，BATCHEDREAD_TABLE 的设置不影响查询优化器是否选择一个包括轻扫描的查询执行路径。

数据库服务器不在索引上、在系统表上执行轻扫描，对于行包括有下列这些存储属性中任何一种的大对象行的用户表，也不执行轻扫描：

- blobspace
- smartblob 空间
- 分区 blob。

您可使用 SET ENVIRONMENT 语句的 IFX_BATCHEDREAD_TABLE 环境选项来覆盖当前会话的 BATCHEDREAD_TABLE 配置参数的值。

2.1.20 BLOCKTIMEOUT 配置参数

使用 BLOCKTIMEOUT 配置参数来指定线程或数据库服务器挂起的秒数。超时之后，线程或数据库服务器或继续处理或失败。

onconfig.std 值

BLOCKTIMEOUT 3600

单位

秒

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

2.1.21 BTSCANNER 配置参数

使用 BTSCANNER 配置参数来设置 B-tree 扫描程序。当从一个带索引的表删除行时，B-tree 扫描程序改善事务处理。B-tree 扫描程序线程移除被删除的索引条目并重新平衡索引节点。 B-tree 扫描程序自动地确定删除那个索引项。

onconfig.std 值

BTSCANNER num=1,threshold=5000,rangesize=-1,alice=6,compression=default

取值范围

参阅“用法”部分。

分隔符

每个域之间使用逗号。

生效

编辑 onconfig 文件并重启数据库服务器之后。

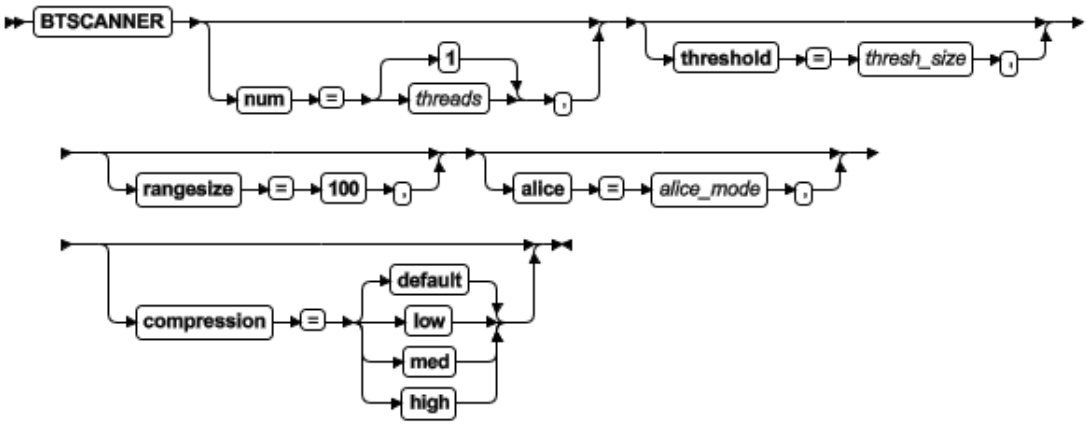
当您通过运行 gadmin -C 命令在 onconfig 文件中动态地重置该值时。

运行带有 gadmin 和 C 参数的 SQL 管理 API task() 或 admin() 函数之后。

用法

缺省情况下，BTSCANNER 配置参数启动一个索引清除程序线程，优先清除有超过 5000 个被删除项的索引，自动地调整索引清除模式，并在一个对适度增长和更改的索引适合的级别上合并索引页。

BTSCANNER 配置参数语法



BTSCANNER 配置参数值的选项表

域	值
num	threads 值是一个正整数，设置在系统启动时启动的 B-tree 扫描程序线程数。缺省为 1。
threshold	thresh_size 值是优先清除的索引必须达到的被删除项的最小数。缺省是 5000。
rangesize	指定是否允许小索引的叶扫描： ● -1 = 关。对所有索引清除使用 alice 模式。 ● 100 = 通过叶扫描模式扫描小索引。
alice	alice_mode 值控制索引清除： ● 0 = 关。 ● 1 = 精确地使用内存 8 字节。 ● 2 = 精确地使用内存 16 字节。 ● 3 - 12 = 缺省是 6。设置用于索引清除的内存初始量。接下去，B-tree 扫描程序根据以前清除操作的效率自动地调整模式。
compression	对两个部分地使用的索引页的合并级别： ● low = 如果您预期索引随着频繁的分裂会迅速地增长，则使用。 ● med 或 default = 缺省。如果索引适度增长或更改，则使用。 ● high = 如果索引的九成或更多是只读，或者索引没有许多更改，则使用。

清除高于阈值的所有索引之后，将低于阈值的索引添加到待清除索引的优先列表。频繁更新的系统应以 10 倍或 100 倍为系数增大这个值。

2.1.22 BUFFERPOOL 配置参数

使用 BUFFERPOOL 配置参数来配置在共享内存中高速缓存的数据页数，在检查点之间以何种频率将那些页清空到磁盘。对于许多系统而言，BUFFERPOOL 配置参数的缺省值足够。然而，您可更改这些值来调整系统性能。

onconfig.std 值

2 KB 缺省页大小的操作系统：

```
BUFFERPOOL default,buffers=10000,lrus=8,lru_min_dirty=50.00,
lru_max_dirty=60.50
BUFFERPOOL size=2k,buffers=50000,lrus=8,lru_min_dirty=50,
```

```
lru_max_dirty=60
```

4 KB 缺省页大小的操作系统：

```
BUFFERPOOL default,buffers=10000,lrus=8,lru_min_dirty=50.00,
lru_max_dirty=60.50
BUFFERPOOL size=4k,buffers=10000,lrus=8,lru_min_dirty=50,
lru_max_dirty=60
```

如果您在安装期间创建了服务器，则缺省值

```
BUFFERPOOL default,memory='auto'
BUFFERPOOL size=page_size,memory=memory_size
```

page_size 值是缺省页大小。缓冲池的最初大小是 32 MB。作为 auto 或 *memory_size* 值，指定 memory 域的值作为大小的最大值，这依赖于 AUTO_TUNE_SERVER_SIZE 配置参数的值。

值

请参阅“用法”部分。

分隔符

用逗号分隔域。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 glogadmin -b 命令在 onconfig 文件中动态地添加一个条目时。

当您通过运行 gspaces -c -d 命令通过添加一个页大小不同的 dbspace 来动态地添加一个条目时。

通过运行带有 add bufferpool 参数的 SQL 管理 API task() 或 admin() 函数，在 onconfig 文件中动态地添加一个条目之后。

用法

高速缓存数据页保存在缓冲区中。缓冲区包含在缓冲池中。用作存储空间的每个页大小需要一个缓冲池。当数据库服务器需要将新数据页移入共享内存时，最近最少使用的数据页移出内存。 BUFFERPOOL 配置参数控制缓冲池的大小和数据页清空到磁盘的频率。

如果您在安装期间创建了一个服务器，则在 onconfig.std 文件或 onconfig 文件中，BUFFERPOOL 配置参数有两个条目：

- 第一个条目为一个非缺省页大小的 dbspace 指定缓冲池的缺省值。
- 第二个条目根据系统的缺省页大小指定缓冲池的缺省值。

包括 size 域的 BUFFERPOOL 配置参数条目优先于包含 default 域的条目。

BUFFERPOOL 配置参数有两种格式：

- 如果您想用类似 MB 或 GB 这样的内存单位指定缓冲池的大小，则使用带有 memory 域的 BUFFERPOOL 配置参数。
- 如果你想以页单位指定缓冲池大小或保持先前版本的设置，则使用带有 buffers 域的 BUFFERPOOL 配置参数。

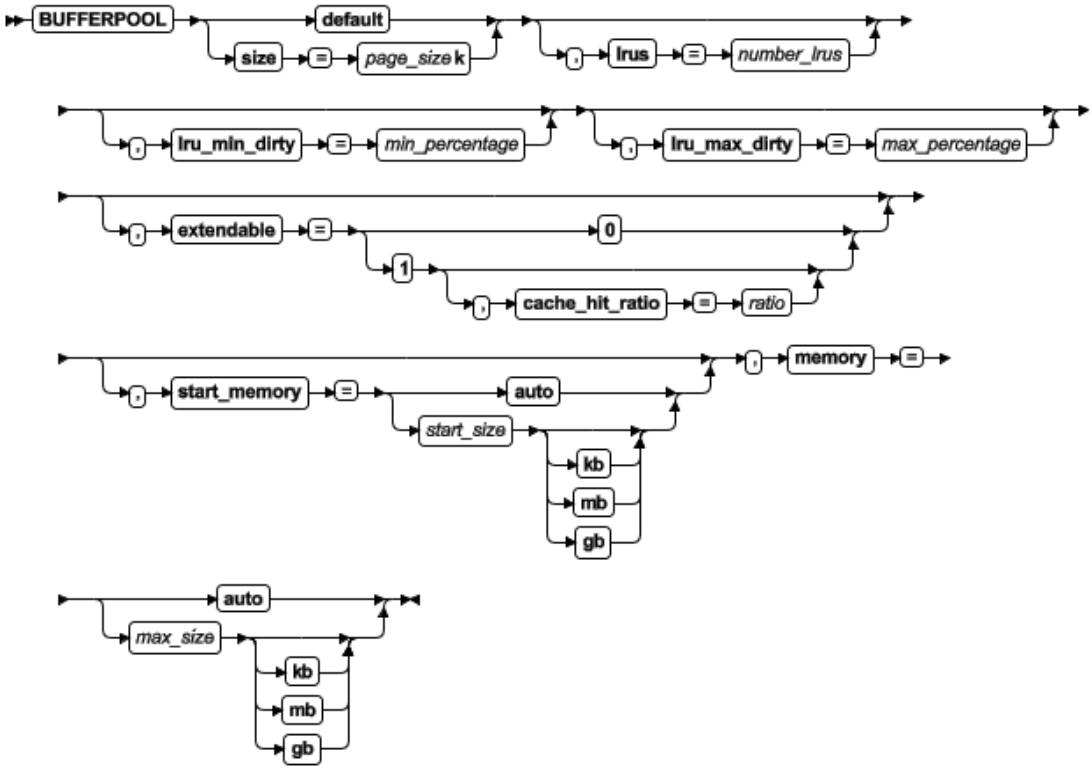
您可使用其中一种格式来在需要提高性能时使得数据库服务器能扩充缓冲池的大小。

限制： 您不可在 onconfig 文件中组合使用两种格式。onconfig 文件中的所有 BUFFERPOOL 配置参数条目都必须有相同的格式。否则数据库服务器不启动并显示下列错误：

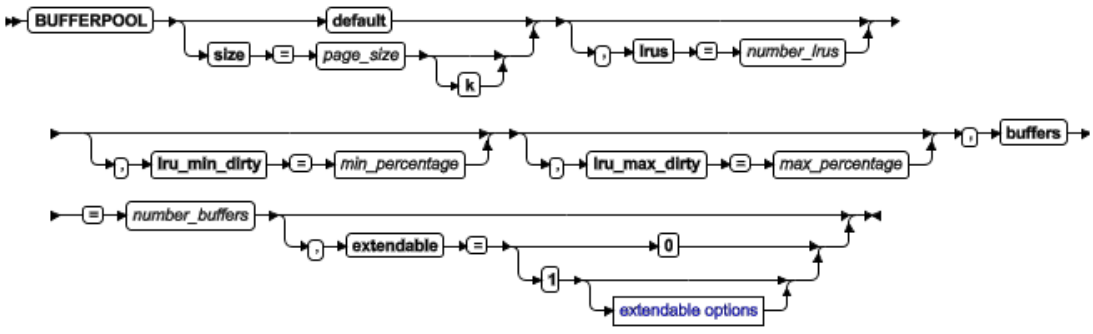
```
ERROR: Cannot mix buffer arguments with memory arguments.  (BUFFERPOOL)
```

BUFFERPOOL 条目中的域不是大小写敏感的，且这些域会按任何顺序罗列。

内存域的语法



缓冲区域的语法



可扩展的选项

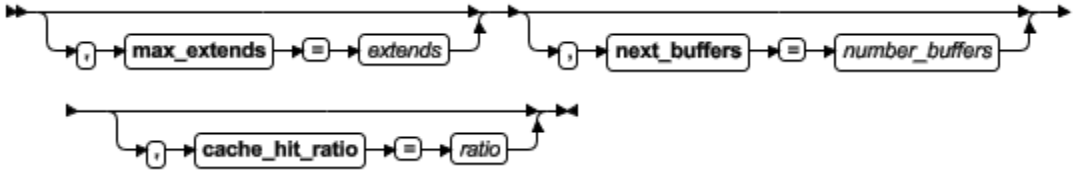


表 1. BUFFERPOOL 配置参数值的选项.

域	值
buffers	缺省是 1000。 *number_buffers* 值是一个大于等于 1000 的整数，指定共享内存缓冲区的最大数目。允许的缓冲区最大数目依赖于操作系统，位大小和页大小： ● UNIX™，32-bit，2 KB 页大小：1000 - 1843200 ● UNIX，32-bit，4 KB 页大小：1000 - 921600 ● Windows™ ，32-bit：100 - 524288 ● 64-bit：100 - (231-1)。请参阅 machine notes，了解您的64-bit 平台的实际值。例如，在 Solaris 平台上的缓冲区最大数目是 536,870,912。 每个用户 buffers 域的值设置为最少四个缓冲区。如果您的系统要处理超过 500 个并发用户，则至少指定 2000 个缓冲区。 每个缓冲区是操作系统页的大小。因此，数据库服务器需要的缓冲区数目依赖于物理内存的数量以及应用程序使用多少内存。例如，如果 90% 的时间数据库服务器访问 15% 的应用程序数据，则分配保持 15% 数据的充足缓冲区。提高缓冲区数目可提高系统性能。缓冲区数目可显著影响性能并使用大百分比的物理内存。 要了解更多信息，请参阅 BUFFERPOOL 配置参数与内存利用。
cache_hit_ratio	缺省是 90。 ratio 值是一个 0 - 100 的整数，表示缓冲池可以在之下扩展的阈值。当平均读高速缓存命中率保持在 ratio 之下大约五分钟时，数据库服务器扩展缓冲池。 只有设置 extendable=1时，cache_hit_ratio 域才有效。
extendable	如果设置 memory 域，则缺省是 1。 如果设置 buffers 域，则缺省是 0。

域	值
	数据库服务器能否扩展缓冲池的大小： ● 0 = 禁用。缓冲池不可增长。 ● 1 = 启用。缓冲池可增长。
lru_max_dirty	缺省是 60.00。 max_percentage 是一个 0 - 100.00 的十进制数，设置在 LRU 队列中当被修改页达到多少百分比时清除队列。 如果启用 AUTO_LRU_TUNING 配置参数，则这个值根据需要动态地更新。
lru_min_dirty	缺省是 50.00。 min_percentage 值是一个 0－100.00 的十进制数，设置在 LRU 队列中当被修改的页达到多少百分比时不再强制清除页。 在有些环境下，页清除程序可以在超过特定百分比时继续清除。 如果启用 AUTO_LRU_TUNING 环境变量，则这个值根据需要自动地更新。
lrus	缺省是 8。如果启用 MULTIPROCESSOR 配置参数，则缺省大于 8 或 CPU VP 的数目。 number_lrus 值是一个正整数，指定缓冲池中 LRU（最近最少使用）队列的数目。 取值范围依赖于操作系统的位大小： ● 32-bit 平台：8 - 128 ● 64-bit 平台：8 - 512 指定 LRU 队列越多，就有越多页清除程序并行工作。然而，设置 lrus 域的值过高，可能导致过度的页清除程序活动。 lrus 域的值与 lru_min_dirty 和 lru_max_dirty 域组合在一起，控制共享内存缓冲区清空到磁盘的频率。 要了解更多信息，请参阅 BUFFERPOOL 及其对页清除的影响。
max_extend	缺省是 8。 extends 表示数据库服务器可扩展缓冲池的最大次数。extends 的值时从 0 到段的最大数目，依赖于操作系统和位大小： ● 32 bit = 16

域	值
	● UNIX 64 bit = 24 ● Windows 64 bit = 8 只有设置 buffers 和 extendable=1，max_extend 域才有效。
memory	缺省是 auto。 max_size 表示缓冲池大小的最大值。max_size 取值范围是： ● 一个表示 32 MB - 4 TB 的整数。您可指定 KB、MB 或 GB 的大小单位。如果您不指定单位，则缺省单位是 KB。 ● auto = 数据库服务器决定分配给缓冲池的共享内存的最大数量。如果设置，AUTO_TUNE_SERVER_SIZE 配置参数的值控制缓冲池大小的最大值。
next_extend	缺省是 1000。 number_buffers 值是一个大于等于 1000 的整数，指定数据库服务器用于扩展缓冲池的共享内存缓冲区数目。number_buffers 的最大值受虚拟共享内存数量的限制。 每四次扩展，number_buffers 值翻倍。 只有设置 buffers 和 extendable=1，next_extend 域才有效。
size	page_size 值指定缓冲区的页大小，以 KB 为单位。页大小必须是 2 - 16 KB 且必须是缺省页大小的倍数。例如，如果缺省页大小是 2 KB，则页大小可是 2、4、6、8、10、12、14 或 16。如果缺省页大小是 4 KB，则页大小可是 4、8、12 或 16。缺省值依赖于系统缺省页大小： ● 2 KB 缺省页大小：size=2k ● 4 KB 缺省页大小：size=4k k 是可选的。
start_memory	缺省是 32 MB。 start_size 值表示当数据库服务器启动时缓冲池的初始大小： ● 一个表示 32 MB 直到可用共享内存的最大数量的整数。您可指定大小单位 KB、MB 或 GB。如果不指定单位，则缺省单位是 KB。缓冲池的初始大小可以大于 start_size 的值，因为该大小必须是共享内存段大小的倍数。

域	值
	● auto = 数据库服务器决定分配给缓冲池的共享内存的初始数量。 如果您不设置 start_memory 域，则缓冲池的初始大小等于 memory 域的值。 只有设置 memory 域，start_memory 域才有效。

默认情况下，GBase 8s 在安装时会自动根据当前内存大小设置 BUFFERPOOL 值。计算算法如下：

- 对于 2 KB 缺省页大小的操作系统：

  BUFFERPOOL = *memorybase* * 25000+10000

  其中 *memorybase* =总内存大小/950000。如果计算结果 < 1，则 *memorybase* = 1

  如果 BUFFERPOOL 值的计算结果大于 10000000 则，将其取值为 10000000。
- 对于 16 KB 缺省页大小的操作系统：

  BUFFERPOOL = *memorybase* * 25000

  如果 BUFFERPOOL 值的计算结果大于 3000000 则，将其取值为 3000000。

内存格式的缓冲池的大小

如果您使用内存格式，则缺省状态下，缓冲池大小根据需要增长。当平均高速缓存读命中率低于阈值时，共享内存段添加到缓冲池。您可设置缓冲池的初始大小和大小的最大值，或允许数据库服务器最优的大小。

如果 extendable 域设置为零，则缓冲池不增长。如果设置，则大小等于 start_memory 域的值，否则，等于 memory 域的值。

当您重启服务器时，缓冲池的大小重置为 start_memory 域的值。

缓冲区格式的缓冲池大小

如果您使用 buffers 格式，则缺省情况下，缓冲池大小不增长。大小等于 buffers 域的值。

如果您设置 extendable 域为 1，则当平均高速缓存读命中率低于阈值时，共享内存段添加到缓冲池。您必须在 buffers 域中设置缓冲区的初始数目。您可选择通过设置缓冲区的数目来扩展缓冲池，以及缓冲池可扩展的最大次数，以及高速缓存命中率。每四次扩展，添加到缓冲池的缓冲区数目翻倍。

示例：添加一个带 memory 域的 BUFFERPOOL 条目

下列条目创建一个有 10 KB 页大小的缓冲池：

```
BUFFERPOOL size=10k,start_memory=auto,memory=4gb
```

缓冲池可扩展到 4 GB。数据库服务器决定缓冲池的初始大小和扩展到缓冲池的大小。

示例：添加一个带有 buffers 域的 BUFFERPOOL 条目

下列条目创建一个有 2 KB 页大小的缓冲池：

```
BUFFERPOOL
size=2k,extendable=1,buffers=1000,next_buffers=2000,max_extends=8
```

缓冲池可扩展八次。缓冲池启动时有 1000 个缓冲区。最初三个缓冲池扩展添加 2000 个缓冲区。第四次到第七次扩展添加 4000 个缓冲区。第八次扩展添加 8000 个缓冲区。

示例：通过添加一个带有不同页大小的 dbspace 来添加一个 BUFFERPOOL 条目

当您用 gspaces 实用程序添加一个带有不同页大小的 dbspace 时，或者当您用 glogadmin 实用程序添加一个缓冲池时，在 onconfig 文件中添加一个 BUFFERPOOL 配置参数条目。下列示例显示一个第三条目：

```
BUFFERPOOL default,buffers=10000,lrus=8,lru_min_dirty=50.00,lru_max_dirty=60.50
        BUFFERPOOL
size=2k,buffers=10000,lrus=8,lru_min_dirty=50,lru_max_dirty=60
        BUFFERPOOL size=6k
```

当您创建一个非缺省页大小的 dbspace 时，如果现有的 BUFFERPOOL 条目存在，则数据库服务器使用那个条目的页大小。否则，数据库服务器使用来自 BUFFERPOOL 缺省行的值。

2.1.23 BULKBIND_THREAD_CNT 配置参数

使用 BULKBIND_THREAD_CNT 配置参数来设置批量更新的并发任务数。通过设置此参数来支持服务器端通过多任务的方式执行大量数据的批量更新功能。

onconfig.std 值

BULKBIND_THREAD_CNT 10

值

大于等于10；建议设置为略小于配置的 CPU VP 数。缺省值为 10。

单位

批量更新的并发任务数。

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.24 CHECKALLDOMAINSFORUSER 配置参数

使用 CHECKALLDOMAINSFORUSER 配置参数来为所有用户检查所有域。

onconfig.std 值

不在 onconfig.std 文件中

值

0 = 禁用

1 = 启用

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.25 CKPTINTVL 配置参数

使用 CKPTINTVL 配置参数来指定数据库服务器检查决定是否需要检查点的频率，用秒表示。当发生检查点时，所有共享内存缓冲池中的页都写到磁盘。

onconfig.std 值

CKPTINTVL 300

值

大于或等于 0 的任何值

单位

秒

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件内动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

RTO_SERVER_RESTART 和 CKPTINTVL 配置参数相互排斥。如果启用 RTO_SERVER_RESTART 配置参数，则会触发检查点并忽略 CKPTINTVL 值。否则，使用 CKPTINTVL 值来触发检查点。

如果您将 CKPTINTVL 配置参数设置为一个过短的间隔，则系统花费过多时间执行检查点，其他工作的性能受损。如果您将 CKPTINTVL 配置参数设置为一个过长的间隔，则快速恢复花费时间过长。

在实践中，30 秒是数据库服务器检查的最小间隔。如果您指定检查点间隔为 0，则数据库服务器不检查检查点间隔是否已过。然而，数据库服务器仍执行检查点。在其他情况下，比如物理日志满至 75%，也会导致数据库服务器执行检查点。

2.1.26 CLEANERS 配置参数

使用 CLEANERS 配置参数来指定数据库服务器运行期间可用的页清除程序线程数。缺省情况下，数据库服务器总是运行一个页清除程序线程。通用指导原则是每个磁盘驱动一个页清除程序。指定的值不影响共享内存的大小。

根据服务器工作负荷，服务器自动地尝试优化 AIO VP 和页清除程序线程，并在需要时向上调整 AIO VP 和页清除程序线程的数目。可使用环境变量 IFX_NO_AIOVP_TUNING 或 gadmin -wm 实用程序选项禁用 AIO VP 和页清除程序线程的自动调整。

onconfig.std 值

CLEANERS 8

值

1-128

单位

页清除程序线程数

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.27 CLUSTER_TXN_SCOPE 配置参数

设置 CLUSTER_TXN_SCOPE 配置参数来配置高可用性集群，以便在客户端会话发出提交时，服务器在辅助服务器上或跨集群地阻塞会话直到事务在那个会话中应用。

onconfig.std 值

CLUSTER_TXN_SCOPE SERVER

值

- SESSION = 当客户端会话发出提交时，数据库服务器阻塞会话，直到事务提交的影响返回到那个会话。控制返回到会话之后，同一个数据库服务器的其他会话或集群中其他数据库服务器上的其他会话可能觉察不到事务提交和事务的影响。
- SERVER （缺省行为） = 当客户端会话发出提交时，数据库服务器阻塞会话，直到事务应用在客户端发出提交的那个数据库服务器。那个数据库服务器的其他会话觉察到事务提交和事务的影响。集群中的其他数据库服务器会话可能觉察不到事务的提交及其影响。对于高可用性集群服务器，这是缺省行为。
- CLUSTER = 当客户端会话发出提交时，数据库服务器阻塞会话，直到事务应用在高可用性集群中所有数据库服务器，除了使用 DELAY_APPLY 或 STOP_APPLY 的 RS 辅助服务器之外，这些会话觉察到事务提交和事务的影响。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

您运行带有 -wf CLUSTER_TXN_SCOPE=value 或 -wm CLUSTER_TXN_SCOPE=value 参数的 SQL 管理 API task() 或 admin() 函数之后。

用法

设置 CLUSTER_TXN_SCOPE 配置参数来控制从高可用性集群返回事务提交到客户端应用程序。集群事务协调可延迟事务提交到客户端应用程序的返回，直到事务应用到辅助服务器或高可用性集群中的所有辅助服务器。 这个例程防止由于异步日志处理导致的操作失败，并确保多步骤处理中的那些步骤按串行顺序发生。

集群事务协调不应用到 RS 辅助服务器，该服务器有 DELAY_APPLY 或 STOP_APPLY 配置函数值而不是 0。在客户端应用程序可收到提交之前，事务不需要应用在 RS 辅助服务器上。CLUSTER_TXN_SCOPE 影响只读辅助服务器和可更新辅助服务器上的会话。

示例1：在高可用性集群服务器间的事务协调

在这个示例中，客户端应用程序启动一个两步骤进程。客户端应用程序在主数据库服务器上插入数据，然后在 HDR 辅助服务器上启动数据处理。

在来自主服务器的日志应用在 HDR 辅助服务器上之前，如果试图在 HDR 服务器上对插入的数据执行 SELECT，则操作失败。 为了防止此失败，设置主服务器的 CLUSTER_TXN_SCOPE 配置参数为 CLUSTER，以便客户端应用程序不收到提交，且不可启动数据处理，直到数据插入页应用在 HDR 辅助服务器上。

示例2：数据库服务器上的事务协调

在此示例中，您让一个客户端应用程序分作几个处理步骤。每个处理步骤使用一个不同的 SQL 会话来连接到数据库服务器。应用程序更新数据，然后应用程序的另一部分在不同的 SQL 会话中处理被更新的数据。

如果 CLUSTER_TXN_SCOPE 设置为 SESSION，则处理被更新的数据的那部分应用程序觉察不到更新的结果，可发生失败。为了防止此失败，设置数据库服务器的 CLUSTER_TXN_SCOPE 配置参数为 SERVER，以便客户端应用程序不接收提交，且不启动数据处理，直到在数据库服务器上更新完成。

2.1.28 CONSOLE 配置参数

使用 CONSOLE 配置参数来指定控制台消息文件的路径和名称。

onconfig.std 值

UNIX™ 上：$GBASEDBTDIR/tmp/online.con

Windows™ 上：online.con

值

pathname = online.con 文件的完整路径名。

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.29 CONVERSION_GUARD 配置参数

如果在升级过程中发生错误，使用 CONVERSION_GUARD 配置参数来指定 GBase 8s 是停止还是继续升级到服务器的新版本。

onconfig.std 值

CONVERSION_GUARD 2

值

0 = 禁用。

1 = 如果发生与捕获恢复点数据相关的错误，则启用一个恢复点作为升级进程的一部分，并停止升级。

2 = 即使发生与捕获恢复点数据相关的错误，也启用一个恢复点作为升级进程的一部分，并继续升级。

单位

整数

生效

当数据库服务器重启时。

用法

缺省情况下：

- CONVERSION_GUARD 配置参数为开（设置为 2）。如果升级到服务器新版本失败，则您可使用 onrestorept 实用程序来恢复数据。
- 服务器在 $GBASEDBTDIR/tmp 目录中存储恢复点数据。

如果 CONVERSION_GUARD 配置参数设置为 1 或 2 且升级到服务器新版本失败，则您可使用 onrestorept 实用程序来恢复数据。

如果 CONVERSION_GUARD 配置参数设置为 2 且转换防护操作失败（例如，由于服务器没有足够空间存储恢复点数据），升级失败，则您不可使用 onrestorept 实用程序来恢复数据。

在启动服务器初始化升级到服务器新版本前，您可更改 CONVERSION_GUARD 配置参数的值或更改在 RESTORE_POINT_DIR 配置参数中指定的目录。升级期间，您不可更改 CONVERSION_GUARD 或 RESTORE_POINT_DIR 值。

2.1.30 DATASKIP 配置参数

使用 DATASKIP 配置参数来控制在事务处理期间，数据库服务器是否跳过不可用的 dbspace。

onconfig.std 值

未设定。不跳过 dbspace。

值

请参阅“用法”部分。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gspaces -f 命令在 onconfig 文件中动态地重置该值时。

您运行带有 set dataskip 参数的 SQL 管理 API task() 或 admin() 函数之后。

用法

在查询期间无论何时只要数据库服务器跳过 dbspace，就会返回警告。

请慎重启用 DATASKIP 配置参数，因为结果总会有问题。仅在下列情况下启用该参数：

- 您可接受折中的事务完整性。
- 您可确定事务的完整性未作出让步，这样做困难又费时。

DATASKIP 配置参数的语法

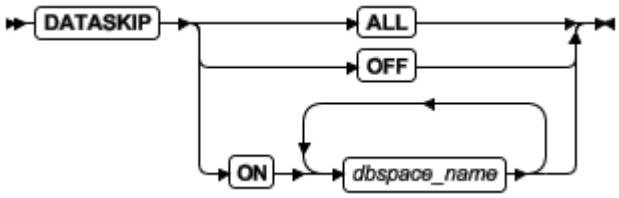


表 1. DATASKIP 配置参数值的选项.

域	描述
ALL	跳过所有不可用分片。
OFF	处理所有分片，包括不可用分片。
ON	*dbspace_name* 值指定一个或多个跳过的 dbspace，用逗号分隔。

应用程序可使用 SQL 语句 SET DATASKIP 来取代 DATASKIP 配置参数的值。

对于 GBase 8s ESQL/C，以前保留的 SQLCA 警告标志 **sqlwarn.sqlwarn7** 设置为 W。

2.1.31 DBCREATE_PERMISSION 配置参数

使用 DBCREATE_PERMISSION 配置参数来限制指定用户创建数据库的权限。

gbasedbt 用户总有创建数据库的权限。要限制 gbasedbt 用户创建数据库的能力，设定 DBCREATE_PERMISSION 配置参数为 gbasedbt。

onconfig.std 值

UNIX™ 上：未设置。任何用户都可创建数据库。

Windows™ 上：#DBCREATE_PERMISSION gbasedbt

缺省值

任何用户都可创建数据库。

单位

用户名

分隔符

逗号。您还可在 onconfig 文件中包括 DBCREATE_PERMISSION 配置参数的多个副本来授权更多用户创建数据库。

生效

编辑 onconfig 文件并重启数据库服务器之后。

DBCREATE_PERMISSION 配置参数不提供创建 tenant 数据库的许可。用户必须有 TENANT 权限来创建 tenant 数据库。通过运行带有 grant admin 参数的 admin() 或 task() SQL 管理 API 函数授予 TENANT 权限。

2.1.32 DB_LIBRARY_PATH 配置参数

使用 DB_LIBRARY_PATH 配置参数来指定一个逗号分隔的有效目录前缀位置的列表，数据库服务器可从此加载外部模块，比如 DataBlade Modules。您还可在该列表中包括服务器环境变量，比如 $GBASEDBTDIR。

您必须确切地指定外部模块的路径，与数据库服务器注册的路径一致。相对路径或包括双句号（..）的路径无效。不用这个参数指定的文件系统中的模块不可加载。在加载 C 语言模块前扫描这个列表。

如果您设置这个配置参数，您还必须包括字符串 $GBASEDBTDIR/extend 作为该值的一部分。如果 DB_LIBRARY_PATH 中未包括字符串 $GBASEDBTDIR/extend，则不加载内建扩展、DataBlade 模块和 BladeManager 实用程序。

onconfig.std 值

未设置

如未出现

数据库服务器可从任何位置加载外部模块

值

路径名列表（最多 512 字节）

分隔符

逗号

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.33 DBSPACETEMP 配置参数

使用 DBSPACETEMP 配置参数来指定 dbspace 列表，数据库服务器用来全面地管理临时表存储。

DBSPACETEMP 通过启用数据库服务器跨多磁盘有效地拓展临时表 I/O。备份期间，数据库服务器还使用临时 dbspace 来保存数据的前映像，这些数据在发生备份时会被重写。

onconfig.std 值

未设定。临时表存储在 root dbspace 中。

分隔符

逗号或冒号（无空白）

值

一个或多个 dbspace 名称。这些 dbspace 可是标准 dbspace、临时 dbspaces 或二者都是。用冒号号或逗号分隔 dbspace 名称。列表长度不可超过 254 字节。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

DBSPACETEMP 可包含非缺省页大小的 dbspace，但是 DBSPACETEMP 列表中的所有 dbspace 必须有相同的页大小。

如果客户端应用程序需要指定一个 dbspace 的替代列表来用于临时表定位，则客户端可使用 DBSPACETEMP 环境变量来罗列。仅当您使用 UPDATE STATISTICS 的 HIGH 选项时，数据库服务器才使用 DBSPACETEMP 环境变量指定的存储定位。

如果在 DBSPACETEMP 配置参数或环境变量中同时罗列了标准 dbspace 和临时 dbspace，则引用下列规则：

- 如果空间充足，则在临时 dbspace 中创建排序、备份、隐式和非日志显式临时表。
- 不用 WITH NO LOG 选项，在标准（而非临时）dbspace 中创建显式临时表。

当您用 gspaces 实用程序创建临时 dbspace 时，数据库服务器不适用新创建的临时 dbspace，直到您设置 DBSPACETEMP 配置参数或环境变量并重启服务器。

DBSPACETEMP 环境变量立即生效并取代 DBSPACETEMP 配置参数。

使用 GBase 8s 散列联接溢出（Hash Join Overflow）和 DBSPACETEMP

如果您不设置 DBSPACETEMP 环境变量或 DBSPACETEMP 配置参数，则 GBase 8s 使用操作系统目录或文件来指向一些数据库操作引起的任何溢出。

您可以下列方式指定操作系统目录或文件：

- 带有 GROUP BY 子句的 SELECT 语句
- 带有 ORDER BY 子句的 SELECT 语句
- 散列联接操作
- 嵌套循环联接操作

- 索引构建

排序溢出文件的位置

下表罗列您可使用来指定排序溢出文件的位置的环境变量和 ONCONFIG 配置参数。

表 1. 排序溢出文件的位置

变量或参数	排序溢出文件的位置
PSORT_DBTEMP 环境变量	在环境变量中指定的位置
DBSPACETEMP 环境变量	在环境变量中指定的位置
ONCONFIG 文件中指定的 DBSPACETEMP 配置参数	ONCONFIG 文件 DBSPACETEMP 配置参数中指定的 dbspace

如果指定多于一个变量或参数，则 GBase 8s 确定排序溢出文件位置优先次序是：

1. PSORT_DBTEMP 环境变量
2. DBSPACETEMP 环境变量
3. DBSPACETEMP ONCONFIG 变量
4. DUMPDIR
5. $GBASEDBTDIR/tmp

如果未设置环境变量或配置参数，则排序溢出文件置于 $GBASEDBTDIR /tmp 目录中，且临时表置于 rootdbspace 中。

2.1.34 DD_HASHMAX 配置参数

使用 DD_HASHMAX 配置参数来指定在数据字典高速缓冲中每个散列存储区中表的最大数目。

散列存储区是存储（通常是一页）的单位，通过散列函数计算其地址。一个散列存储区包含若干记录。

例如,如果 DD_HASHMAX 配置参数设置为 10 且 DD_HASHSIZE 配置参数设置为 59，则您可在数据字典高速缓存中存储大约 590 个表的信息，且每个散列存储区可有最多 10 个表。

使用文本编辑器来修改该配置文件。

onconfig.std 值

DD_HASHSIZE 10

值

正整数

单位

一个散列存储区中表的最大数目

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.35 DD_HASHSIZE 配置参数

使用 DD_HASHSIZE 配置参数来指定散列缓冲区的数目,或数据字典高速缓存中的列表数目。

使用文本编辑器来修改该配置文件。

onconfig.std 值

DD_HASHSIZE 31

值

任何正整数；推荐质数

单位

散列存储区或列表的数目

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.36 DEADLOCK_TIMEOUT 配置参数

使用 DEADLOCK_TIMEOUT 配置参数来指定数据库服务器可等待获取锁的最大秒数。仅为涉及远程数据库服务器的分布式查询使用此参数。对非分布式查询不使用此参数。

onconfig.std 值

DEADLOCK_TIMEOUT 60

值

正整数

单位

秒

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令设置内存中的该值时。

用法

如果分布式事务被强制等待的时间大于 DEADLOCK_TIMEOUT 配置参数指定的秒数时，则拥有该事务的线程假定存在多服务器死锁。

2.1.37 DEF_TABLE_LOCKMODE 配置参数

使用 DEF_TABLE_LOCKMODE 配置参数来指定新表在页级或行级的锁模式。

onconfig.std 值

PAGE

值

PAGE = 为新表设置页的锁模式

ROW = 为新表设置行的锁模式

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

优先规则

通过在 CREATE TABLE 或 ALTER TABLE 语句中包括 LOCK MODE 子句，您可取代对于特定表的所有其他锁模式。

在客户端设置的 IFX_DEF_TABLE_LOCKMODE 环境变量优先于服务器上的变量和 DEF_TABLE_LOCKMODE 配置参数。

在服务器上设置的 IFX_DEF_TABLE_LOCKMODE 环境变量优先于 DEF_TABLE_LOCKMODE 配置参数。

用法

如果 DEF_TABLE_LOCKMODE 配置参数设置为 ROW，对于连接到日志记录或非日志记录数据库的所有会话，则将每一个新创建表的锁模式都设置到行。该参数不影响现有表的锁模式。

如果 DEF_TABLE_LOCKMODE 配置参数设置为 PAGE，则 USELASTCOMMITTED 配置参数和 SET ISOLATION 语句的 COMMITTED READ LAST COMMITTED 选项不可启用访问未提交事务持有排他锁的表中的最近提交数据，除非明确地创建或更改这些表将 ROW 作为其锁定粒度。

2.1.38 DEFAULTESCCHAR 配置参数

DEFAULTESCCHAR 配置参数指定用于 LIKE 和 MATCHES 的缺省转义符。

onconfig.std 值

DEFAULTESCCHAR 反斜线字符（\）。

如未出现

如果在 onconfig 文件中未设置值，则使用反斜线字符（\）。

值

\ = 使用反斜线字符作为转义符。

NONE = 无缺省转义符。

character = 任何一字符值都可用作转义符。

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

通过使用带有您想使用的转义符的 SET ENVIRONMENT DEFAULTESCCHAR 语句，可以在会话中取代缺省值。例如：

```
SET ENVIRONMENT DEFAULTESCCHAR '\'
```

2.1.39 DELAY_APPLY 配置参数

使用 DELAY_APPLY 配置参数来配置 RS 辅助服务器应用日志之前等待指定的时间期间。

onconfig.std 值

DELAY_APPLY 0

缺省值

0

值

0 = 应用日志

后跟时间单位的整数：例如，1H 设置延迟一个小时。

number: 1-999 = 等待的天数、分钟数、小时数或秒数。

time_unit: D、H、M或S，此处 D = 天，H = 小时，M = 分钟，S = 秒。 这些值不区分大小写。

生效

编辑 onconfig 文件并重启数据库服务器后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

延迟日志文件的应用允许您通过从 RS 辅助服务器恢复数据来快速恢复错误的数据库修改。

当设置 DELAY_APPLY 的值时，您必须还要设置 LOG_STAGING_DIR。如果配置 DELAY_APPLY 而未将 LOG_STAGING_DIR 设置为一个有效且安全的目录，则服务器不可初始化。

您必须通过设置 LOG_STAGING_DIR 配置参数来指定一个日志文件的有效并安全位置。

在 RS 辅助服务器上处理完最后的检查点之后，清除 GBase 8s staging 目录中的日志。

要看到有关发送到为 RS 辅助服务器设置的 GBase 8s log-staging 目录的数据信息，请在 RS 辅助服务器上运行 gstat -g rss verbose 命令。

如果写 GBase 8s staging 文件失败，则 RS 辅助服务器发出事件报警 40007。

如果远程独立辅助（RSS）服务器的 DELAY_APPLY 配置参数设置为一个非 0 的值，则服务器不可用集群事务协调。

2.1.40 DIRECT_IO 配置参数（UNIX）

使用 DIRECT_IO 配置参数来控制用于 dbspace chunk 的熟文件直接 I/O。

该参数在 UNIX™ 平台上启用直接 I/O （绕过文件系统缓冲）或在 AIX® 操作系统上启用并发 IO（绕过文件系统缓冲和不必要的写串行化。）

onconfig.std 值

DIRECT_IO 0

值

0 = 不使用直接 I/O 或并发 I/O

1 = 如果可以使用，则直接 I/O，绕过文件系统缓冲

2 = 在 AIX 操作系统上启用并发 I/O（并发 I/O 选项包括直接 I/O 和并发 I/O。）

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

直接 I/O 仅可用于 dbspace chunk，其文件系统支持页大小的直接 I/O。

通过使用直接 I/O，您能减少 AIO 虚拟处理器的数目。

如果启用直接 I/O，则使用文件系统支持的 KAIO（内核异步 I/O）。然而，如果设置环境变量 KAIOOFF，则不使用 KAIO。当直接 IO 和 KAIO 都使用时，可减少 AIO 虚拟处理器的数目。如果使用直接 IO，但不使用 KAIO，则不应减少 AIO 虚拟处理器的数目。

对于用作临时 dbspace chunk 的熟文件，GBase 8s 不使用直接或并发 I/O。

在 AIX 上，如果 GBase 8s 使用 chunk 的并发 I/O，则另一个程序（比如一个联机外部备份程序）必须也使用并发 I/O。否则，文件打开操作将失败。

如果 GBase 8s 使用 chunk 的直接 I/O，且另一个程序试图不使用直接 I/O 来打开该 chunk，则打开操作正常成功，但可发生性能损失。之所以发生损失，是因为文件系统可能试图确保每个打开操作查看相同的文件数据，或者通过在冲突打开操作期间根本不使用直接 I/O，或者通过在每次直接 I/O 前清空文件系统高速缓存并在每次直接写之后使文件系统高速缓存无效。

在 Windows™ 平台上使用 dbspace chunk 的直接 I/O，与 DIRECT_IO 配置参数的值无关。

2.1.41 DIRECTIVES 配置参数

使用 DIRECTIVES 配置参数来启用或禁用优化器指令。这些指令指定在开发 SELECT、UPDATE 和 DELETE 语句的查询计划中查询优化器的行为。

onconfig.std 值

DIRECTIVES 1

值

0 = 禁用优化器指令

1 = 启用优化器指令

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

环境变量

IFX_DIRECTIVES

用法

设置 DIRECTIVES 为缺省值 1，来启用数据库服务器处理优化器指令。设置 DIRECTIVES 为 0 来禁用数据库服务器处理指令。

客户端程序还可设置 IFX_DIRECTIVES 环境变量为 ON 或 OFF 来启用或禁用数据库服务器处理指令。 IFX_DIRECTIVES 环境变量的设置取代 DIRECTIVES 配置参数的设置。如果您不设置 IFX_DIRECTIVES 环境变量，则客户端的所有会话继承数据库服务器配置处理指令。

2.1.42 DISABLE_B162428_XA_FIX 配置参数

使用 DISABLE_B162428_XA_FIX 配置参数来指定释放事务的时刻。

onconfig.std 值

不在 onconfig.std 文件中。

值

0 = （缺省）仅当调用 xa_rollback 时释放事务。

1 = 如果是非 xa_rollback 的事务回滚，则释放事务。

单位

整数

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

设置 DISABLE_B162428_XA_FIX 为 1，在事务回滚之后立即释放所有全局事务。

您可用 IFX_XASTDCOMPLIANCE_XAEND 环境变量取代客户端会话的 DISABLE_B162428_XA_FIX 配置参数。 设置 IFX_XASTDCOMPLIANCE_XAEND 为 1，

则仅当调用 xa_rollback 时释放事务。设置 IFX_XASTDCOMPLIANCE_XAEND 为 0，则在非 xa_rollback 的事务回滚时释放事务。

2.1.43 DRDA_COMMBUFFSIZE 配置参数

使用 DRDA_COMMBUFFSIZE 配置参数来指定 DRDA® 通信缓冲区的大小。

当建立一个 DRDA 会话时，为会话分配一个等于当前缓冲区大小的通信缓冲区。如果后来更改了缓冲区大小，则现有的连接不受影响，但新的 DRDA 连接使用新的大小。 GBase 8s 自动将大于 2 MB 的值重置到 2 MB，并将小于 4KB 的值重置到 32 KB 缺省值。

onconfig.std 值

不在 onconfig.std 文件中。

如果不出现

32K

值

Minimum = 4 KB

Maximum = 2 MB

生效

当初始化共享内存时

用法

通过为值添加“M”或“K”，用户可以 MB 或 KB 为单位指定 DRDA_COMMBUFFSIZE 值。字母不区分大小写，缺省是 KB。例如，可以下列方法之一指定一个 1 MB 缓冲区：

- DRDA_COMMBUFFSIZE 1M
- DRDA_COMMBUFFSIZE 1m
- DRDA_COMMBUFFSIZE 1024K
- DRDA_COMMBUFFSIZE 1024k
- DRDA_COMMBUFFSIZE 1024

2.1.44 DRAUTO 配置参数

设置 DRAUTO 配置参数来指定对 HDR 高可用性系统的 HDR 故障转移方式。

onconfig.std 值

DRAUTO 0

值域

值	描述
0	禁用自动故障转移。当主服务器故障或失去网络连接时，HDR 辅助服务器成为只读。

1	启用自动故障转移。当主服务器故障或失去网络连接时，将 HDR 辅助服务器转为标准服务器。HDR 辅助服务器适时地终止客户端连接并关闭，然后重启作为标准服务器。 当故障的主服务器重启或重新连接到网络时，将标准服务器转回为 HDR 辅助服务器。 如果您配置了连接管理器来执行故障转移，则不使用此设置。
2	启用自动故障转移。当主服务器故障或失去网络连接时，将 HDR 辅助服务器转为主服务器。HDR 辅助服务器保持客户端连接，且不关闭。 当故障的主服务器重启或重新连接到网络时，将其转为 HDR 辅助服务器。 如果您配置了连接管理器来执行故障转移，则不使用此设置。
3	由连接管理器控制故障转移。为了自动故障转移，连接管理器必须配置并是活动的。

生效

当初始化共享内存时。

用法

所有高可用性集群的服务器都必须有相同的 DRAUTO 配置参数设置。

DRAUTO 配置参数不控制 SDS 辅助服务器或 RS 辅助服务器的故障转移。要设置有 SD 辅助服务器或 RS 辅助服务器的高可用性集群的自动故障转移，请配置连接管理器。

重要： 如果您正在使用连接管理器来控制故障转移，则所有集群服务器上的 DRAUTO 配置参数都必须设置为 3。在连接管理器是活动的时候，您必须不执行手工故障转移。

2.1.45 DRIDXAUTO 配置参数

如果辅助 HDR 服务器检测到一个损坏的索引，则使用 DRIDXAUTO 配置参数来指定主高可用性数据复制（HDR）服务器是否自动地启动索引复制。

onconfig.std 值

DRIDXAUTO 0

值

0 = 关

1 = 开

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

要改变活动的服务器实例的 DRIDXAUTO 配置参数值，请使用 gadmin -d idxauto 命令。您不需要重启服务器实例。 然而，gadmin -d idxauto 命令将不修改 onconfig 文件中 DRIDXAUTO 配置参数值。

2.1.46 DRINTERVAL 配置参数

使用 DRINTERVAL 配置参数来指定两次清空数据复制缓冲区间隔的最大秒数，或是否使用 HDR_TXN_SCOPE 配置参数指定的异步模式。

onconfig.std 值

DRINTERVAL 0

值

-1 = 使用 HDR SYNC 模式。如果主服务器使用无缓冲的日志记录，则复制是同步的。

0 = HDR_TXN_SCOPE 配置参数的值决定 HDR 数据复制的同步模式。

正整数 = 使用 HDR ASYNC 模式。正整数是两次清空数据复制缓冲区之间间隔的最大秒数。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

DRINTERVAL 配置参数控制复制延迟，并用来设置复制同步。

如果使用无缓冲的日志记录，则 HDR SYNC 模式与同步模式几乎相同，通过 HDR_TXN_SCOPE 配置参数设置同步模式。

表 1. DRINTERVAL、HDR_TXN_SCOPE和日志记录设置的矩阵，及其 HDR 复制模式的结果。

DRINTERVAL	HDR_TXN_SCOPE	日志记录	结果
-1	无	有缓冲	异步复制
-1	无	无缓冲	将近同步复制
0	FULL_SYNC	有缓冲	完全同步复制
0	FULL_SYNC	无缓冲	完全同步复制
0	ASYNC	有缓冲	异步复制
0	ASYNC	无缓冲	异步复制
0	NEAR_SYNC	有缓冲	将近同步复制

0	NEAR_SYNC	无缓冲	将近同步复制
正整数	无	有缓冲	异步复制
正整数	无	无缓冲	异步复制

2.1.47 DRLOSTFOUND 配置参数

使用 DRLOSTFOUND 配置参数来指定到失而复得文件的路径名。此文件指出，当主数据库服务器发生故障时，有些事务在辅助数据库服务器上未提交之前，在 HDR 主数据库服务器上提交了。

onconfig.std 值

UNIX™ 上：$GBASEDBTDIR/etc/dr.lostfound

Windows™ 上：$GBASEDBTDIR\tmp

值

pathname = dr.lostfound 文件的路径名

生效

编辑 onconfig 文件并重启数据库服务器之后。

当 DRINTERVAL 配置参数设置为 -1时，如果在主和辅助数据库服务器之间同时发生更新，则 DRLOSTFOUND 不适用。

失而复得文件 dr.lostfound.timestamp 创建时，带着一个时间戳。该时间戳附在文件名后，以便当存在另一个文件时，数据库服务器不会重写另一个失而复得文件。您不能使用失而复得文件来重新应用丢失的事务。

2.1.48 DRTIMEOUT 配置参数

使用 DRTIMEOUT 配置参数来指定以秒计的时间长度，是在高可用性数据库服务器对中数据库服务器等待其他数据库服务器发来迁移确认的时长。此参数仅适用于高可用性数据复制对。

onconfig.std 值

DRTIMEOUT 30

值

正整数

单位

秒

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

使用下列公式来计算为 DRTIMEOUT 配置参数指定的值：

```
DRTIMEOUT = wait_time / 4
```

在此公式中，wait_time 时以秒计的时间长度，是在服务器认定高可用性数据复制发生故障之前，高可用性数据复制对中数据库服务器等待的时长。

例如，您确定系统的 wait_time 是 160 秒。使用先前的公式设置 DRTIMEOUT 如下：

```
DRTIMEOUT = 160 秒 / 4 = 40 秒
```

2.1.49 DS_HASHSIZE 配置参数

使用 DS_HASHSIZE 配置参数来指定在数据分布高速缓存和其他高速缓存中散列存储区的数目。数据库服务器存储和访问在数据分布高速缓存中 MEDIUM 或 HIGH 模式下 UPDATE STATISTICS 语句生成的列统计信息。

onconfig.std 值

DS_HASHSIZE 31

值

任何正整数；推荐质数

单位

散列存储区或列表的数目

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

更新 DS_HASHSIZE 和 DS_POOLSIZE 配置参数值来提高在多用户环境中频繁使用的查询性能。

DS_HASHSIZE 配置参数为下列高速缓存设置散列存储区的数目：

- 数据分布高速缓存
- 扩展类型名高速缓存
- 扩展类型 ID 高速缓存

- GBase 8s cast 高速缓存
- 运算符类实例高速缓存
- 例程解析高速缓存
- 合计高速缓存
- 辅助瞬态高速缓存

2.1.50 DS_MAX_QUERIES 配置参数

使用 DS_MAX_QUERIES 配置参数来指定可并发地运行的并行数据库查询(PDQ)的最大数目。DS_MAX_QUERIES 配置参数的值依赖于 DS_TOTAL_MEMORY 配置参数的设置：

- 如果设置 DS_TOTAL_MEMORY 配置参数，则 DS_MAX_QUERIES 的值是 DS_TOTAL_MEMORY / 128，向下取整到最接近的整数值。
- 如果未设置 DS_TOTAL_MEMORY 配置参数，则 DS_MAX_QUERIES 配置参数的值是 2 * num，此处 num 是在 VPCLASS 配置参数中指定的 CPU 数目。

onconfig.std 值

未设置。

如未出现

2* num * 128，此处 num 是在 VPCLASS 配置参数中指定的 CPU 数目。

值

最小值 = 1
最大值 = 8,388,608 （8 MB）

单位

查询的数目

生效

编辑 onconfig 文件并重启数据库服务器之后。
当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。
当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

“内存分配管理器”（MGM）根据下列公式为查询保留内存：

```
memory_reserved = DS_TOTAL_MEMORY *
```

```
        (PDQ-priority / 100) *
        (MAX_PDQPRIORITY / 100)
```

或者在 PDQPRIORITY 环境变量中，或者在 SQL 语句 SET PDQPRIORITY 中指定 PDQPRIORITY 的值。

2.1.51 DS_MAX_SCANS 配置参数

使用 DS_MAX_SCANS 配置参数来限定数据库服务器可并发执行的 PDQ 扫描线程的数目。

onconfig.std 值

DS_MAX_SCANS 1048576 或 (1024 * 1024)

值

10 - (1024 * 1024)

单位

PDQ 扫描线程数

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

当用户发出查询时，数据库服务器根据下列值分配一些扫描线程：

- PDQ 优先级的值（通过环境变量 PDQPRIORITY 或 SQL 语句 SET PDQPRIORITY 设置）
- 以 DS_MAX_SCANS 设置的上限
- 以 MAX_PDQPRIORITY 设置的因子
- 要扫描的表中的分片数（公式中的 nfrags）

内存分配器（MGM）试图根据下列公式为查询保留扫描线程：

```
reserved_threads = min (nfrags, (DS_MAX_SCANS * PDQPRIORITY / 100 *
MAX_PDQPRIORITY / 100) )
```

如果公式的 DS_MAX_SCANS 部分大于或等于要扫描的表中的分片数，则查询保持在就绪队列中，直到有表分片且可用的扫描线程一样多。一旦进行，查询快速地执行，因为线程在并行地扫描分片。

例如，如果 nfrags 等于 24，DS_MAX_SCANS 等于 90，PDQPRIORITY 等于 50， 且 MAX_PDQPRIORITY 等于 60，则查询不开始执行直到有 nfrags 个扫描线程可用。扫描并行发生。

如果 DS_MAX_SCANS 公式降到低于分片数，则查询会尽快开始执行，但由于线程串行地扫描分片，查询执行时间较长。

在前一个示例中，如果将 DS_MAX_SCANS 降低到 40，则查询需要较少的资源（12 个扫描线程）来开始执行，但每一个线程需要串行地扫描两个分片。执行耗时较长。

2.1.52 DS_NONPDQ_QUERY_MEM 配置参数

使用 DS_NONPDQ_QUERY_MEM 配置参数来增加查询可用的内存数量，该查询不是并行数据库查询（PDQ）。（如果 PDQ 优先级设为零，则您可仅使用该参数。）

onconfig.std 值

DS_NONPDQ_QUERY_MEM：

- UNIX™ 上：256
- Windows™ 上：128

值

从缺省值到 DS_TOTAL_MEMORY 值的 25%

单位

KB

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

如果您为 DS_NONPDQ_QUERY_MEM 参数指定一个值，则根据表行的数目和大小确定和调整该值。

提示： 通常设置该值不超过最大可用临时 dbspace 大小。

在数据库服务器初始化期间根据计算得到的 DS_TOTAL_MEMORY 值来计算 DS_NONPDQ_QUERY_MEM 值。如果在处理 DS_NONPDQ_QUERY_MEM 期间，数据库服务器更改您设置的该值，则服务器按此格式发送消息：

DS_NONPDQ_QUERY_MEM recalculated and changed from *old_value* Kb to *new_value* Kb.

在此消息中，*old_value* 表示您在用户配置文件中赋给 DS_NONPDQ_QUERY_MEM 的值，*new_value* 表示由数据库服务器确定的值。

2.1.53 DS_POOLSIZE 配置参数

使用 DS_POOLSIZE 参数来指定在数据分布高速缓存和其他高速缓存中条目的最大数目。数据库服务器存储和访问列统计信息，在数据分布高速缓存中由 UPDATE STATISTICS 语句在 MEDIUM 或 HIGH 模式下生成。

onconfig.std 值

DS_POOLSIZE 127

值

正值 127 或更大的表示高速缓存中条目初始的最大数目的一半。最大值依赖于共享内存配置和可用的服务器实例的共享内存。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wm 命令增加内存中的该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

使用 DS_HASHSIZE 和 DS_POOLSIZE 配置参数来提高在多用户环境中频繁地运行查询的性能。

高速缓存中的最初条目数是 DS_POOLSIZE 配置参数值的两倍。例如，如果 DS_POOLSIZE 配置参数设置为 127，则在高速缓存中允许 254 个条目。 如果高速缓存中所有条目都满，则高速缓存的大小自动地增长 10%。要减小高速缓存的大小，请降低 onconfig 文件中 DS_POOLSIZE 配置参数的值并重启服务器。

DS_POOLSIZE 配置参数设置在下列高速缓存中的条目数：

- 数据分布高速缓存
- 扩展类型名称高速缓存
- 扩展类型 ID 高速缓存
- GBase_85 cast 高速缓存
- 运算符类实例高速缓存
- 例程解析高速缓存
- 合计高速缓存
- 辅助瞬态高速缓存

2.1.54 DS_TOTAL_MEMORY 配置参数

使用 DS_TOTAL_MEMORY 配置参数来指定 PDQ 查询可用的内存数量。此数量应小于计算机物理内存，减去固定的开销，比如操作系统大小和缓冲池大小。

onconfig.std 值

未设置。

如未出现

如果设置 SHMTOTAL=0 和 DS_MAX_QUERIES，则 DS_TOTAL_MEMORY = DS_MAX_QUERIES * 128。

如果 SHMTOTAL=0 且未设置 DS_MAX_QUERIES，则 DS_TOTAL_MEMORY = num_cpu_vps * 2 * 128。

值

如果设置 DS_MAX_QUERIES，则最小值是 DS_MAX_QUERIES * 128。

如果未设置 DS_MAX_QUERIES，则最小值是 num_cpu_vps * 2 * 128。

没有最大值限制，除非在您机器上的软件有任何限制。

单位

KB

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 重置内存中的该值时。

用法

不要将 DS_TOTAL_MEMORY 跟配置参数 SHMTOTAL 和 SHMVIRTSIZE 弄混。SHMTOTAL 设置指定数据库服务器的所有内存（内存的常驻、虚拟和消息部分的总和）。SHMVIRTSIZE 设置指定虚拟部分的大小。DS_TOTAL_MEMORY 是 SHMVIRTSIZE 的一个逻辑子集。

对于 OLTP 应用程序，将 DS_TOTAL_MEMORY 设置在 20% 与 SHMTOTAL 值的 50% 之间的 KB 数。

对于涉及大的决策支持（DSS）查询的应用程序，将 DS_TOTAL_MEMORY 值增加到 50% 与 SHMTOTAL 的 80% 之间。如果您使用数据库服务器排他地进行 DSS 查询，则将此参数设置在 90% 与 SHMTOTAL 的 100% 之间。

设置 DS_TOTAL_MEMORY 配置参数为不大于 (SHMVIRTSIZE - 10 MB)的任意值。

有关您的平台上最大可用内存的信息，请参阅 GBase 8s machine notes。

DS_TOTAL_MEMORY 的算法

当您未设置 DS_TOTAL_MEMORY 时，或您设置一个不合适的值，数据库服务器得出一个 DS_TOTAL_MEMORY 的值。要了解关于算法的信息，请参阅 GBase 8s 性能指南。

2.1.55 DUMPCNT 配置参数（UNIX™）

使用 DUMPCNT 配置参数来指定会话中的断言失败数目，达到此数目时，数据库服务器线程转储共享内存或通过调用 gcore 实用程序生成核心文件。

onconfig.std 值

DUMPCNT 1

值

正整数

单位

转储的共享内存或可在会话中生成的核心文件数目

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

当数据库服务器不能继续正常处理时，发生断言失败。

断言失败可生成与 DUMPCNT 配置参数值相同数量的核心文件或共享内存转储。进一步的断言失败在消息日志中生成错误，或许还生成应用程序错误，但不保存进一步的诊断信息。

2.1.56 DUMPCORE 配置参数（UNIX™）

使用 DUMPCORE 配置参数来控制断言失败是否导致虚拟处理器转储一个核心映像。核心文件留在最后一次发起调用数据库服务器的那个目录中。（DUMPDIR 参数不影响核心文件的位置。）

onconfig.std 值

DUMPCORE 0

值

0 = 不转储核心映像。

1 = 转储核心映像。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

警告：当 DUMPCORE 设置为 1 时，断言失败导致虚拟处理器转储一个核心映像，反过来导致数据库服务器终止。仅在受控的环境里为了调试才设置 DUMPCORE。

2.1.57 DUMPDIR 配置参数

数据库服务器从失败的断言转储共享内存、gcore 文件或消息到一个目录中。DUMPDIR 指定这个目录。

因为共享内存可能很大，所以设置 DUMPDIR 到一个有大量空间的文件系统。为了服务器启动，DUMPDIR 设置的那个目录必须存在。

onconfig.std 值

UNIX™ 上：$GBASEDBTDIR/tmp

Windows™ 上：$ GBASEDBTDIR\tmp

值

用户 gbasedbt 有写访问权限的任何目录

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.58 DUMPGCORE 配置参数（UNIX™）

使用 DUMPGCORE 配置参数来指定是否转储 gcore 核心文件。在支持 gcore 的操作系统上使用此配置参数。

onconfig.std 值

DUMPGCORE 0

值

0 = 不转储 gcore。

1 = 转储 gcore。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

如果您设置 DUMPGCORE，但操作系统不支持 gcore，则数据库服务器消息日志中的消息表明试图转储核心映像，但数据库服务器未能找到预期的文件。（如果操作系统不支持 gcore，则转而设置 DUMPCORE。）

如果设置 DUMPGCORE，则每当虚拟处理器遇到断言失败时，数据库服务器就调用 gcore。gcore 实用程序指引虚拟处理器将核心映像转储到 DUMPDIR 指定的目录中的 core.pid.cnt 文件，并继续处理。

pid 值是虚拟处理器的进程标识号。此进程每次遇到断言失败，cnt 值都增加。cnt 值的范围从 1 到 DUMPCNT 的值。达到 DUMPCNT 的值之后，不再创建核心文件。如果虚拟处理

器继续遇到断言失败，则向消息日志报告错误（或许也向应用程序报告），但不保存进一步的诊断信息。

2.1.59 DUMPSHMEM 配置参数（UNIX™）

使用 DUMPSHMEM 配置参数来指出是否在断言失败时创建共享内存转储。此配置参数还指定将多少内存写到 shmem.pid.cnt 文件，该文件在 DUNPDIR 配置参数指定的目录中。

onconfig.std 值

DUMPSHMEM 1

值

0 = 不创建共享内存转储。

1 = 创建数据库使用的所有共享内存的共享内存转储。

2 = 创建不包括常驻内存中缓冲池的共享内存转储。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

如果 DUMPSHMEM 设置为 1，则转储数据库服务器使用的所有共享内存，这可导致生成一个大文件。当空间有限时，设置 DUMPSHMEM 为 2，因为此设置会生成一个较小的共享内存转储文件。

pid 值是虚拟处理器的进程标识号。虚拟处理器每次遇到断言失败 cnt 值就增加。cnt 值域可从 1 到 DUMPCNT 配置参数值。达到 DUMPCNT 值之后，不再创建文件。如果数据库服务器继续检测到不一致，则向消息日志报告错误（并或许向应用程序报告），但不再保存进一步的诊断信息。

2.1.60 DYNAMIC_LOGS 配置参数

使用 DYNAMIC_LOGS 配置参数来允许当有必要防止事务阻塞时动态地添加逻辑日志。

onconfig.std 值

DYNAMIC_LOGS 2

值

0 = 关闭动态日志分配。

1 = 关闭“log file required”报警并暂停来允许手工添加逻辑日志文件。您可在当前日志文件之后立即添加日志文件，或添加到日志文件列表尾部。

2 = 开启动态日志分配。当数据库服务器动态地添加日志文件时，会关闭“dynamically added log file”报警。

生效

对于 HDR：当数据库服务器关闭并重启时

对于 Enterprise Replication：当 Enterprise Replication 启动时

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

如果 DYNAMIC_LOGS 是 2，则当下一个活动的日志文件包含一个打开的事务时，数据库服务器动态地分配一个新的日志文件。动态日志分配防止长事务从阻塞的事务回滚。

如果您想要选择新逻辑日志文件的大小和位置，则设置 DYNAMIC_LOGS 为 1。使用 glogadmin -a 命令来在当前日志文件之后添加一个日志文件，该命令带有大小（-s）、位置（-d dbspace）和 -i 选项。

如果 DYNAMIC_LOGS 配置参数值是 0 且发生事务阻塞，则关闭数据库服务器，设置 DYNAMIC_LOGS 为 1 或 2，然后重启数据库服务器。

重要： 如果您正在使用带有动态日志分配的 Enterprise Replication，则请设置 LTXEHWM 不高于 70。

2.1.61 EILSEQ_COMPAT_MODE 配置参数

使用 EILSEQ_COMPAT_MODE 配置参数来控制 GBase 8s 是否检查客户端应用程序插入的字符数据，该数据是否包含指向队列的代码或当前数据库语言环境不识别。

onconfig.std 值

EILSEQ_COMPAT_MODE 0

值

0 = GBase 8s 以当前的语言环境验证来到的字符序列是否有效，如果有无效字符则返回错误 -202。

1 = GBase 8s 不验证来到的字符序列是否有效。

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

如果设置 EILSEQ_COMPAT_MODE 配置参数为 0，则仅可将有效的字节序列插入数据库。

EILSEQ_COMPAT_MODE 配置参数在这些情况下防止 202 错误：

- 当从数据库检索数据时。
- 当无效字符位于字符串尾部且该字符是部分字符时。

2.1.62 ENABLE_SNAPSHOT_COPY 配置参数

使用 ENABLE_SNAPSHOT_COPY 配置参数来启用或禁用使用 gclone 实用程序来克隆服务器的能力。

onconfig.std 值

0

值

0 = 禁止克隆

1 = 许可克隆

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

ENABLE_SNAPSHOT_COPY 配置参数确定您能否使用 gclone 实用程序创建服务器的克隆。设置 ENABLE_SNAPSHOT_COPY 配置参数为 1 来允许克隆。设置值为 0 来禁止使用 gclone 实用程序克隆服务器。

如果您在安装期间创建了服务器，则自动地启用 ENABLE_SNAPSHOT_COPY 配置参数。

2.1.63 ENABLE_NULL_STRING 配置参数

使用 ENABLE_NULL_STRING 配置参数来启用或禁用空字符串（’’）查询时是否等同于 NULL 的功能。

onconfig.std 值

ENABLE_NULL_STRING 1

值

1 = 启用（缺省值），空字符串（仅限于 ’’）等同于 NULL

0 = 禁用，空字符串（仅限于 ’’）不等同于 NULL

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

ENABLE_NULL_STRING 配置参数确定在对字符型字段执行 IS NULL 的条件查询时，空字符串（即，’’）是否等同于 NULL。缺省设置 ENABLE_NULL_STRING 配置参数为 1 来启用此功能。设置值为 0 来屏蔽此功能。

2.1.64 ENABLE_NULL_STRCAT 配置参数

使用 ENABLE_NULL_STRCAT 配置参数来启用或禁用值为 NULL的字段和字符串进行

拼接时等同空字符串的功能。

onconfig.std 值

ENABLE_NULL_STRCAT 1

值

1 = 启用（缺省值），NULL 等同于空字符串（’’）。

0 = 禁用，NULL 等同于字符串’NULL’。

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

ENABLE_NULL_STRCAT 配置参数确定在将值为 NULL的列与其它字符串进行拼接时，NULL 是否等同于空字符串（即，’’）。缺省设置 ENABLE_NULL_STRCAT 配置参数为 1 来启用此功能。设置值为 0 来屏蔽此功能。

例如，在以下示例中假定表 t1 的 name 字段为 NULL，启用此功能后，执行以下 SELECT 语句返回结果为‘abc’。

```
select ‘abc’|| name from t1;
```

若禁用此功能，则以上查询语句返回结果为‘abcNULL’

2.1.65 ENCRYPT_CIPHERS 配置参数

使用 ENCRYPT_CIPHERS 配置参数来定义当前数据库会话可使用的所有密码和方式。

ENCRYPT_CIPHERS 仅用于 Enterprise Replication 和高可用性数据复制。

onconfig.std 值

未设置。不使用加密密码。

值

请参阅“用法”部分。

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

在两个服务器之间在公共密码中随机地选择加密密码和方式。如果发现某个特定密码有弱点，您应使用 allbut 选项重置 ENCRYPT_CIPHERS 配置参数值来消除那个密码。

重要：包括所有密码比包括特定的密码更安全。

ENCRYPT_CIPHERS 配置参数的语法

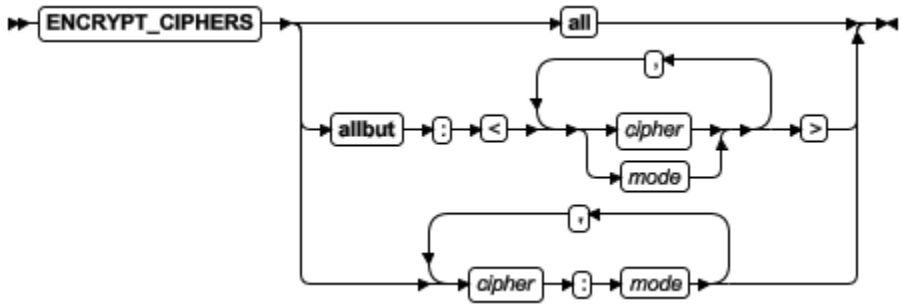


表 1. ENCRYPT_CIPHERS 配置参数值的选项

域	描述
all	包括所有可用的密码和方式，除了 ECB 模式之外，该模式被认为有弱点。 例如： ENCRYPT_CIPHERS all
allbut	包括所有密码和方式，除了 ECB 罗列的密码和方式之外。 例如：ENCRYPT_CIPHERS allbut:<cbc,bf> 密码列表可包括唯一的、缩写的条目。例如，bf 可表示 bf-1、bf-2 和 bf-3；然而，如果缩写是一个实际密码的名字，则仅消除那个密码。因此，des 仅 消除 des 密码，但 de 消除 des、des3 和 desx 密码。
cipher	支持下列密码： ● des = DES（64 位密钥） ● des3 = 三重 DES ● desx = 扩展 DES（128 位密钥）。只支持 cbc 方式。 ● aes = AES 128 位密钥 ● aes192 = AES 192 位密钥 ● bf-1 = Blow Fish（64 位密钥） ● bf-2 = Blow Fish（128 位密钥） ● bf-3 = Blow Fish（192 位密钥） ● aes128 = AES 128 位密钥 ● aes256 = AES 256 位密钥 所有方式支持所有密码，除了 desx 密码之外。 要了解最新的支持密码列表，请参阅 Release Notes®。
mode	支持下列模式：

域	描述
	● ecb = 电子密码本（ECB）。只包括指定的情况。 ● cbc = 密码分组链接 ● cfb = 密码反馈 ● ofb = 输出反馈

2.1.66 ENCRYPT_HDR 配置参数

onconfig.std 值

未设置。

值

0 = 禁用 HDR 加密

1 = 启用 HDR 加密

生效

当初始化服务器时

用法

ENCRYPT_HDR 启用或禁用 HDR 加密。启用 HDR 加密，在 HDR 对内从一台服务器向另一台传输数据时提供了一种安全的方法。HDR 加密连同 Enterprise Replication （ER）加密一同工作。 然而，不必为了 HDR 加密而启用 ER 加密。不论 ER 加密是否启用，HDR 加密都工作。HDR 和 ER 共享同样的加密配置参数：ENCRYPT_CIPHERS、ENCRYPT_MAC、ENCRYPT_MACFILE 和 ENCRYPT_SWITCH。

2.1.67 ENCRYPT_MAC 配置参数

使用 ENCRYPT_MAC 配置参数来控制消息认证代码（MAC）生成的级别。此配置参数仅用于 Enterprise Replication 和高可用性数据复制。

onconfig.std 值

未设置

值

off = 不使用 MAC 生成

low = 使用 XOR 折换所有消息

medium = 对所有大于 20 字节的消息使用 SHA1 MAC 生成，在较短的消息上使用 XOR 折换

high = 在所有消息上使用 SHA1 MAC 生成。

示例

ENCRYPT_MAC medium,high

生效

对于 HDR：当数据库服务器关闭并从重启时

对于 Enterprise Replication：当启动 Enterprise Replication 时

用法

此级别优先级高于最高值。例如，如果一个节点启用级别 high 和 medium，另一个节点仅启用 low，那么连接尝试失败。仅当确保网络连接安全时，才在服务器之间使用 off 条目。

2.1.68 ENCRYPT_MACFILE 配置参数

使用 ENCRYPT_MACFILE 配置参数来指定一个 MAC 密钥文件的全路径名列表。此配置参数仅用于 Enterprise Replication 和高可用性数据复制。

onconfig.std 值

未设置。

值

一个或以逗号分隔的多个全路径和文件名，以及可选的 builtin 关键词。例如：

```
ENCRYPT_MACFILE /usr/local/bin/mac1.dat, /usr/local/bin/mac2.dat,builtin
```

单位

路径名长度最大为 1536 字节

生效

对于 HDR：当数据库服务器关闭并重启时。

对于 Enterprise Replication：当启动 Enterprise Replication 时。

用法

在连接时刻，ENCRYPT_MACFILE 配置参数确定每一个条目的优先级并协商。由 GenMacKey 实用程序根据其创建时间确定 MAC 密钥文件的优先级。从 builtin 关键词创建的条目有最低优先级。因为协商 MAC 密钥文件，因此您应定期地更改密钥。

2.1.69 ENCRYPT_SMX 配置参数

使用 ENCRYPT_SMX 配置参数来设置在附属服务器上高可用性配置的加密级别。

onconfig.std 值

未设置。

值

0 = 关。不加密。

1 = 开。可能的情况下加密。当数据库服务器连接到也支持加密时加密 SMX 事务。

2 = 开。总是加密。仅允许连接到加密的数据库服务器。

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.70 ENCRYPT_SWITCH 配置参数

使用 ENCRYPT_SWITCH 配置参数来定义重新协商密码或秘密密钥的频率。此配置参数仅用于 Enterprise Replication 和高可用性数据复制。

秘密密钥和加密密码保持使用的时间越长，加密规则越可能被袭击者打破。为了避免发生这种情况，密码学家建议在长期连接上更改秘密密钥。发生重新协商的缺省时间是一小时一次。

onconfig.std 值

未设置。

值

以逗号分隔的两个正整数。第一个整数表示两次密码重新协商之间的分钟数。第二个整数表示两次秘密密钥重新协商之间的分钟数。例如：ENCRYPT_SWITCH 2,5。

单位

分钟

生效

对于 HDR：当数据库服务器关闭并重启时

对于 Enterprise Replication：当启动 Enterprise Replication 时

2.1.71 EXPLAIN_STAT 配置参数

使用 EXPLAIN_STAT 配置参数来启用或禁用在解释输出文件中包括“查询统计信息”部分。您可通过使用 SET EXPLAIN 语句或 gadmin -Y sessionid 命令生成输出文件。当您启用 EXPLAIN_STAT 配置参数时，在“查询计划”中“查询统计信息”部分显示预计的行数和返回的实际行数。

onconfig.std 值

EXPLAIN_STAT 1

值

0 = 禁用在解释输出文件中包含“查询统计信息”部分。

1 = 启用在解释输出文件中包含“查询统计信息”部分。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

2.1.72 EXT_DIRECTIVES 配置参数

onconfig.std 值

EXT_DIRECTIVES 0

值

0（缺省）= 关。即使 IFX_EXTDIRECTIVES 为开，也不可启用指令。

1 = 开。如果 IFX_EXTDIRECTIVES 为开，则为会话启用指令。

2 = 开。即使 IFX_EXTDIRECTIVES 未设置，也可使用指令。

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

EXT_DIRECTIVES 配置参数启用或禁用外部查询优化器指令的使用。通过与客户端侧 IFX_EXTDIRECTIVES 环境变量组合在一起，使用 EXT_DIRECTIVES 配置参数启用外部指令，如下：

IFX_EXTDIRECTIVES 环境变量的设置取代 EXT_DIRECTIVES 配置参数的设置。如果您不设置 IFX_EXTDIRECTIVES 环境变量，则客户端的所有会话继承数据库服务器处理外部指令的配置。

SQL 的 SET ENVIRONMENT EXTDIRECTIVES 语句指定的设置取代（仅对于当前用户会话）IFX_EXTDIRECTIVES 环境变量和 EXT_DIRECTIVES 配置参数二者的设置。

2.1.73 EXTSHMADD 配置参数

使用 EXTSHMADD 配置参数来指定虚拟扩展段的大小。当用户定义的例程或 DataBlade 例程运行在用户定义的虚拟处理器中时，会添加虚拟扩展段。

onconfig.std 值

EXTSHMADD 8192

值

32 位操作系统：1024 - 524288

64 位操作系统：1024 - 4294967296

单位

KB

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

当线程运行在用户定义的虚拟处理器中时，创建一个虚拟扩展段。在 gstat -g seg 命令的输出中，虚拟扩展段有一个 VX 类。如果在 onconfig 文件中未设置 EXTSHMADD 配置参数，则由 SHMADD 配置参数的值设置虚拟扩展段的大小。

2.1.74 FAILOVER_CALLBACK 配置参数

onconfig.std 值

未设置

值

pathname = FAILOVER_CALLBACK 配置参数指定的脚本全路径名。

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

当数据库服务器从辅助服务器向主服务器或标准服务器转换时，数据库服务器执行 FAILOVER_CALLBACK 指定的脚本。FAILOVER_CALLBACK 设置为脚本的全路径名。

2.1.75 FAILOVER_TX_TIMEOUT 配置参数

在高可用性集群环境中，在主服务器故障转移之后，使用 FAILOVER_TX_TIMEOUT 配置参数来启用事务完成。

使用 FAILOVER_TX_TIMEOUT 配置参数来指出在服务器回滚事务之前，服务器等待故障转移之后的最大秒数。在高可用性集群中， 所有服务器上的 FAILOVER_TX_TIMEOUT 配置参数都设置为同一个值。

onconfig.std 值

FAILOVER_TX_TIMEOUT 0

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

当在高可用性集群环境中发生故障转移时，辅助服务器之一取代主服务器的角色。成为新的主服务器的辅助服务器称为故障转移服务器。

通过设置 FAILOVER_TX_TIMEOUT 配置参数为一个大于零的值，启用事务存活。当启动事务存活时，故障转移服务器必须能够联系剩余的辅助服务器来同步并恢复任何打开的事务。类似地，存活的辅助服务器必须能建立到故障转移服务器的连接来重新发送任何挂起的事务。FAILOVER_TX_TIMEOUT 配置参数指定服务器开始回滚事务之前等待的时长。

在故障转移服务器上，如果超过了 FAILOVER_TX_TIMEOUT 指定的秒数，则任何不与存活服务器同步的打开的事务都终止并回滚。

在剩余的辅助服务器上，如果超过了 FAILOVER_TX_TIMEOUT 指定的秒数，则服务器上任何打开的事务都返回错误。

当发生故障转移时，设置 FAILOVER_TX_TIMEOUT 为 0 来立即回滚所有打开的事务。

如果主服务器故障，且辅助服务器未能取代主服务器的角色，那么回滚任何打开的事务，且客户端不能进行更新。例如，如果在辅助服务器上启动更新活动且主服务器故障，那么故障转移过程不完成且不建立新的主服务器，在预定的时间量之后，客户端要求超时，将 sqlexec 线程置于不确定状态。

在上述场景中，回滚活动的事务，但直到建立新主服务器才可发生物理回滚（因为主服务器管理日志）。在这些情况下，会话觉察不到在辅助服务器上执行的操作。会话觉察不到部分应用的事务回滚，因为直到建立新的主服务器才可发生部分事务的回滚。

2.1.76 FASTPOLL 配置参数

使用 FASTPOLL 配置参数来启用或禁用网络的快速轮询。FASTPOLL 是特定于平台的配置参数。

onconfig.std 值

FASTPOLL 1

值

0 = 禁用快速轮询。

1 = 启用快速轮询。

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.77 FILLFACTOR 配置参数

使用 FILLFACTOR 配置参数来指定索引页的充满程度。低值提供索引中的成长空间。高值压缩索引。

如果索引充满（100%），则任何新插入都会导致分裂节点。您还可设置 FILLFACTOR 作为 CREATE INDEX 语句的选项。在 CREATE INDEX 语句上的设置取代 ONCONFIG 文件值。

您不可将 FILLFACTOR 配置参数用于树状索引的森林。

onconfig.std 值

FILLFACTOR 90

值

1 - 100

单位

百分比

生效

当构建索引时。现有的索引不更改。要使用新值，就必须重新构建索引。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

2.1.78 FULL_DISK_INIT 配置参数

使用 FULL_DISK_INIT 配置参数来防止发生现有数据库服务器实例的意外磁盘重新初始化。当在 root 路径位置上存在页零时，该位置在第一个 chunk 位置的第一页，此配置参数指定在 GBase 8s 实例上可否运行磁盘初始化命令(oninit -i）。

onconfig.std 值

FULL_DISK_INIT 0

值

0 = 仅当在 root 路径位置上没有页零时，才运行 oninit -i 命令。

1 = 在所有情况下都运行 oninit -i 命令，但也在磁盘初始化之后重置 FULL_DISK_INIT 配置参数为 0。

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

当 FULL_DISK_INIT 配置参数设置为 1时，任何实例启动命令（例如，oninit 以及 oninit -i）都重置该配置参数为 0。

当 FULL_DISK_INIT 配置参数设置为 0 且数据库服务器发现页零时，如果您开始运行 oninit -i 命令，则 oninit -i 命令不运行且服务器在 online.log 中报告错误。

页零是 GBase 8s 系统页，包含关于服务器实例的通用信息。当服务器实例初始化时，创建此页。

2.1.79 GSKIT_VERSION 配置参数

使用 GSKIT_VERSION 配置参数来指定 GBase Global Security Kit (GSKit) 的主版本，数据库服务器使用该版本来加密和 SSL 通信。

onconfig.std 值

未设置。随着 GBase 8s 安装使用的 GSKit 版本。

值

GSKit 版本是运行在 7 与最新主发布号（当前为 15）之间的完整数目。

单位

正整数

生效

数据库初始化期间

如果在同一台计算机上数据库服务器与其他 GBase 产品一起使用，且随同其他 GBase 产品安装了不同的 GSKit 版本，则可使用不同 GSKit 版本配置数据库服务器。

2.1.80 HA_ALIAS 配置参数

使用可选的 HA_ALIAS 配置参数来指定高可用性集群节点的替代别名。

onconfig.std 值

未设置。

值

HA_ALIAS 配置参数的值必须是 sqlhosts 文件中 dbservername 条目指定的值之一。网络别名必须有 TCP 网络协议的连接类型。如果 dbservername 值包括监听线程的可选数据，则从 HA_ALIAS 忽略可选监听线程值。例如，如果 servername 列 设置为 reynolds-4，则 HA_ALIAS 设置为 reynolds。

生效

编辑 onconfig 文件并重启数据库服务器之后。

对于高可用性集群中的主服务器，通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值。此方法对高可用性集群中的辅助服务器不起作用。

对于高可用性集群中的主服务器，通过运行 gadmin -wm 命令重置内存中的该值。此方法对高可用性集群中的辅助服务器不起作用。

用法

HA_ALIAS 配置参数是一个定义辅助服务器网络别名的可选参数。当您在 gadmin -d 命令中指定辅助服务器时，使用通过 HA_ALIAS 配置参数指定的网络别名。

如果高可用性集群中的主服务器故障，则连接管理器找到一台辅助服务器并晋升为主服务器。如果设置了辅助服务器的 HA_ALIAS 配置参数，则使用 HA_ALIAS 网络别名来标识新的主服务器。

当设置了数据库服务器的 HA_ALIAS 配置参数时，所有与高可用性集群有关的服务器到服务器通信都通过该指定的网络别名发生。

2.1.81 HA_FOC_ORDER 配置参数

使用 HA_FOC_ORDER 配置参数来为服务器的高可用性集群定义单一的连接管理故障转移规则。

onconfig.std 值

HA_FOC_ORDER SDS,HDR,RSS

值

辅助服务器类型的列表，以逗号分隔并按优先级顺序罗列。例如，缺省值 SDS,HDR,RSS 表示主服务器故障转移到 SD 辅助服务器，然后是 HDR 辅助服务器，再然后是 RS 辅助服务器。

- HDR = 高可用性数据复制服务器
- RSS = 远程独立辅助服务器
- SDS = 共享磁盘辅助服务器

MANUAL = 对集群中所有连接管理器禁用自动故障转移。

分隔符

以逗号分隔值。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

运行带有 -wf HA_FOC_ORDER=value 或 -wm HA_FOC_ORDER=value 参数的 SQL 管理 API task() 或 admin() 函数之后。

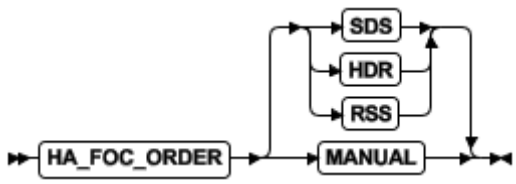
用法

如果在高可用性集群的主数据库服务器上设置 HA_FOC_ORDER 配置参数，则连接到该主服务器的每一个连接管理器都采用该设置。该值取代连接单元的 ORDER=rule 故障转移序列规则。然后，高可用性集群中的每一数据库服务器采用主服务器的 HA_FOC_ORDER 配置参数值作为自己的 HA_FOC_ORDER 配置参数。

如果主服务器上的 HA_FOC_ORDER 配置参数设置为 MANUAL。则在管理主服务器的集群的所有连接管理器上禁用自动故障转移。

如果连接管理器的配置文件中连接单元的 FOC ORDER 值设置为 DISABLED，则连接管理器不对那个连接单元执行故障转移。

HA_FOC_ORDER 配置参数的语法



示例

在下列示例中，您配置两个连接管理器来管理一个三服务器的集群。

三台服务器是：

- server_1（主服务器）
- server_2（SD 辅助服务器）

- server_3（HDR 辅助服务器）

第一个连接管理器有下列配置文件：

```
NAME connection_manger_1

CLUSTER cluster_1
{
    SERVERNUM servers_1
    SLA sla_1 DBSERVERS=ANY
    FOC ORDER=ENABLED \
        PRIORITY=1
}
```

第二个连接管理器有下列配置文件：

```
NAME connection_manger_2

CLUSTER cluster_1
{
    SERVERNUM servers_1
    SLA sla_2 DBSERVERS=ANY
    FOC ORDER=ENABLED \
        PRIORITY=2
}
```

server_1 的 onconfig 文件有下列值：

```
HA_FOC_ORDER SDS,HDR
```

当 connection_manger_1 和 connection_manger_2 与 server_1 连接时，其配置成为：

```
NAME connection_manger_1

CLUSTER cluster_1
{
   SERVERNUM servers_1
    SLA sla_1 DBSERVERS=ANY
    FOC ORDER=SDS,HDR \
        PRIORITY=1
}
NAME connection_manger_2

CLUSTER cluster_1
{
```

```
    SERVERNUM servers_1
    SLA sla_2 DBSERVERS=ANY
    FOC ORDER=SDS,HDR \
        PRIORITY=2
}
```

在 server_2 和 server_3 的 onconfig 文件中，HA_FOC_ORDER 条目的值更新为 SDS,HDR。

2.1.82 HDR_TXN_SCOPE 配置参数

HDR_TXN_SCOPE 配置参数与 DRINTERVAL 配置参数一起使用，来指定在高可用性集群中 HDR 复制的同步模式。

onconfig.std 值

HDR_TXN_SCOPE NEAR_SYNC

值

FULL_SYNC = 如果完全同步，则 HDR 复制。在事务可完成之前，事务需要 HDR 辅助服务器上完成确认。

NEAR_SYNC = 如果几乎同步，则 HDR 复制。在事务可完成之前，事务需要在 HDR 辅助服务器上收到确认。如果同无缓冲日志记录一起使用，SYNC 模式，则当 DRINTERVAL 设置为 -1 时打开，与几乎同步模式相同。

ASYNC = 如果完全异步，则 HDR 复制。在事务可完成之前，事务不需要在 HDR 辅助服务器上收到或完成确认。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

运行带有 "gadmin","-wf HDR_TXN_SCOPE=value" 或 "gadmin","-wm HDR_TXN_SCOPE=value" 参数的 SQL 管理 API task() 或 admin() 函数之后。

用法

当 DRINTERVAL 配置参数设置为 0 时，HDR_TXN_SCOPE 参数的值确定 HDR 复制的同步模式。

如果使用无缓冲日志记录,则 HDR SYNC 模式与几乎同步模式相同,通过 HDR_TXN_SCOPE 配置参数设置。

表 1. DRINTERVAL、HDR_TXN_SCOPE 和日志记录设置及其 HDR 复制模式结果的矩阵。

DRINTERVAL	HDR_TXN_SCOPE	日志记录	结果
-1	无	有缓冲的	异步复制

DRINTERVAL	HDR_TXN_SCOPE	日志记录	结果
-1	无	无缓冲的	几乎同步复制
0	FULL_SYNC	有缓冲的	完全同步复制
0	FULL_SYNC	无缓冲的	完全同步复制
0	ASYNC	有缓冲的	异步复制
0	ASYNC	无缓冲的	异步复制
0	NEAR_SYNC	有缓冲的	几乎同步复制
0	NEAR_SYNC	无缓冲的	几乎同步复制
正整数	无	有缓冲的	异步复制
正整数	无	无缓冲的	异步复制

2.1.83 HETERO_COMMIT 配置参数

onconfig.std 值

HETERO_COMMIT 0

值

1 = 启用异类提交。

0 = 禁用异类提交。

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

HETERO_COMMIT 配置参数指定在异类提交事务中数据库服务器是否准备与 GBase 8s Gateway 产品配合。设置 HETERO_COMMIT 为 1 允许单一事务来更新一个非 GBase 8s 数据库（与任何 Gateway 产品一起访问）和一个或多个 GBase 8s 数据库。

如果 HETERO_COMMIT 是 0，则单个事务可更新下列数据库：

- 一个或多个 GBase 8s 数据库，且没有非 GBase 8s 数据库
- 一个非 GBase 8s 数据库，且没有 GBase 8s 数据库

您可从任何数目的 GBase 8s 和非 GBase 8s 数据库读取数据，不管 HETERO_COMMIT的设置。

2.1.84 IFX_EXTEND_ROLE 配置参数

您的数据库系统管理员（DBSA），缺省为用户 gbasedbt，可使用 IFX_EXTEND_ROLE 参数来控制授权哪些用户注册 DataBlade 模块或外部用户定义的例程（UDR）。

onconfig.std 值

IFX_EXTEND_ROLE 1

值

1 或 On （缺省）= 启用 EXTEND 角色的要求，以便管理员可授予用户权限来创建或删除包括 EXTERNAL 子句的 UDR。

0 或 Off = 禁用 EXTEND 角色的要求，以便任何拥有适当外部语言（C 或 JAVA）USAGE ON LANGUAGE 权限的用户可注册或删除用那种语言编写的外部例程。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

2.1.85 IFX_FOLDVIEW 配置参数

使用 IFX_FOLDVIEW 配置参数来启用或禁用视图折换。对于某些查询中涉及视图的情况，视图折换可显著提高查询性能。在这些情况下，视图折换成父查询，而不是将查询结果放到临时表中。

onconfig.std 值

IFX_FOLDVIEW 1

值

0 或 Off = 禁用视图折换。

1 或 On = 缺省。启用视图折换。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

下列查询类型可利用视图折换：

- 包含 UNION ALL 且父查询有常规连接、GBase 8s 连接、ANSI 连接或 ORDER BY 子句的视图

对于下列执行涉及视图的 UNION ALL 操作的查询类型，创建临时表且不执行视图折换：

- 视图有下列子句之一：AGGREGATE、GROUP BY、ORDER BY、UNION、DISTINCT 或 OUTER JOIN（要么 GBase 8s 类型，要么 ANSI 类型）。

- 父查询有 UNION 或 UNION ALL 子句。

2.1.86 IFX_XA_UNIQUEXID_IN_DATABASE 配置参数

使用 IFX_XA_UNIQUEXID_IN_DATABASE 配置参数来启用事务管理器，在相同的数据库服务器实例中使用相同的 XID 来表示不同数据库上的全局事务。

onconfig.std 值

None

缺省值

0

值

0 = 禁用

1 = 启用

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

XID 是分布式 XA 事务的一个全局事务 ID。

如果设置 IFX_XA_UNIQUEXID_IN_DATABASE 配置参数为 1，则数据库服务器允许事务管理器在相同的数据库服务器实例中使用相同的 XID 来表示不同数据库上的全局事务。因此，数据库可是域而不是服务器。

2.1.87 GBASEDBTCONRETRY 配置参数

使用 GBASEDBTCONRETRY 配置参数来指定在初次连接尝试失败后，可对每一数据库服务器进行连接尝试的最大数目。进行这些尝试限定在 GBASEDBTCONTIME 配置参数指定的时间之内。

onconfig.std 值

GBASEDBTCONRETRY 1

值

正整数

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

GBASEDBTCONTIME 设置优先于 GBASEDBTCONRETRY 设置。超过 GBASEDBTCONTIME 值之后，但在达到 GBASEDBTCONRETRY 值之前，连接尝试可终止。

要取代当前会话的 GBASEDBTCONRETRY 配置参数的值，您可或者设置 SET ENVIRONMENT 语句的 GBASEDBTCONRETRY 环境选项，或者设置客户端的 GBASEDBTCONRETRY 环境变量。

2.1.88 GBASEDBTCONTIME 配置参数

使用 GBASEDBTCONTIME 配置参数来指定 CONNECT 语句尝试建立到数据库服务器连接的秒数。

onconfig.std 值

GBASEDBTCONTIME 60

值

正整数

单位

秒

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

要设置 GBASEDBTCONTIME 配置参数的最佳值，请考虑节点间的总距离、硬件速度、流量以及网络的并发级别。

GBASEDBTCONTIME 值除以 GBASEDBTCONRETRY 值的结果决定连接尝试之间的秒数。如果设置 GBASEDBTCONTIME 配置参数为零，则数据库服务器使用 60 秒的缺省值。

要取代当前会话的 GBASEDBTCONTIME 配置参数的值，您可设置 SET ENVIRONMENT 语句的 GBASEDBTCONTIME 环境选项，也可设置客户端的 GBASEDBTCONTIME 环境变量。

2.1.89 LIMITNUMSESSIONS 配置参数

使用 LIMITNUMSESSIONS 配置参数来定义要连接到 GBase 8s 的会话的最大数目。

如果指定一个最大数目，则你还可指定当会话数达到最大数目时，是否要 GBase 8s 打印消息到 online.log 文件。

如果启用 LIMITNUMSESSIONS 配置参数，且由于此限制导致会话受限，连接到任何数据库的常规用户线程和 DBSA 用户线程都要计入此限制。然而，即使已经达到此限制，仍然允许 DBSA 用户连接到数据库。

针对服务器的分布式查询也计入此限制。

不要打算使用 LIMITNUMSESSIONS 配置参数作为遵守许可协议的一种方法。

onconfig.std 值

未在 onconfig.std 文件中设置

值

maximum_number_of_sessions = 0 到 2,097,152（2*1024*1024）。缺省是 0。

print_warning = 0（关）或 1（开）。此可选的缺省值是 0。

分隔符

逗号

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

如果 print_warning 设置为 1，则当会话数目大于或等于 maximum_number_of_sessions 值的 95% 时，触发警告。如 果 print_warning 设置为零，或未设置，则不发出警告。达到 maximum_number_of_sessions 限制之后，不可打开新的用户会话。

如果 LIMITNUMSESSIONS 配置参数的 maximum_number_of_sessions 值设置为 0，或未设置，则不限制可连接到数据库的会话数。

下列示例指定您想要最多 100 个会话连接到服务器，且当连接的会话数接近 100 时，打印警告消息。

```
LIMITNUMSESSIONS 100,1
```

此示例中的设置导致当并发的连接多于 94 个时，打印警告。仅 DBSA 组成员可在已有 100 个会话连接时启动新的会话。

使用 gadmin -wf 或 gadmin -wm，或者对等的 SQL 管理 API ONMODE 命令，来动态地增加或临时地禁用 LIMITNUMSESSIONS 设置。 如果数据库服务器达到 maximum_number_of_sessions ，则使用此配置参数来允许运行管理性实用程序。

2.1.90 LISTEN_TIMEOUT 配置参数

使用 LISTEN_TIMEOUT 配置参数来指定服务器等待连接的秒数。

onconfig.std 值

LISTEN_TIMEOUT 60

单位

秒

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

您可设置 LISTEN_TIMEOUT 为一个较低数目来防止那些可能指出“服务拒绝”袭击的错误连接要求。

依赖于机器保持线程（数目）的能力，您可配置 MAX_INCOMPLETE_CONNECTIONS 为一个较高值，依赖于网络流量，您可设置 LISTEN_TIMEOUT 为一个较低值来减少袭击达到上限的可能性。

2.1.91 LOCKS 配置参数

LOCKS 配置参数指定锁表的初始大小。

锁表为每个锁保留一个条目。如果分配的锁数目超过 LOCKS 配置参数的值，则数据库服务器增加锁表的大小。锁表最多可增加 99 次。

onconfig.std 值

LOCKS 20000

值

对于 32 位数据库服务器，2,000 至 8,000,000；对于 64 位数据库服务器，2,000 至 500,000,000

单位

内部锁表中锁的数目

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

通过尝试每次增加锁表一倍，数据库服务器增加锁表的大小。然而，每次增加期间添加的数量受最大值限制。对于 32 位平台，每次增加可最多添加 100,000 个锁。因此，对于 32 位平台，最多锁数合计是 8,000,000 （启动锁的最大数目）+ （99（动态锁表扩展的最大数目）x 100,000（每个锁表扩展添加的锁的最大数目））。对于 64 位平台，每次增加可最多添加 1,000,000 个锁。因此，允许的最多锁数合计是 500,000,000（启动锁的最大数目）+(99(动态锁表扩展的最大数目)x 1,000,000(每个锁表扩展添加的锁的最大数目))。随着初始锁表存储在常驻内存中，而每个附加锁存储在虚拟内存中，如果你有数量有限的共享内存，则锁资源可变得枯竭。单个锁占据的存储数量依赖于字大小和操作系统，且易更改。当前，存储的数量范围从大约 100 字节至 200 字节。通过以 LOCKS 配置参数的一个不同值（未做其他更改）重启服务器，您可看到为支持附加锁而需要的存储数量。通过常驻池的“gstat -g mem”，观察使用的内存的增加情况。

默认情况下，GBase 8s 在安装时会自动根据当前内存大小设置 LOCKS 值。计算算法如下：

LOCKS = *memorybase* *100000+100000

其中 *memorybase* =总内存大小/950000。如果计算结果 < 1，则 *memorybase* = 1

如果 LOCKS 值的计算结果大于 20000000 则，将其取值为 20000000。

提示： 当您删除数据库时，需要一个锁并在数据库中的每个表上保持，直到删除了数据库。

2.1.92 LOGBUFF 配置参数

使用 LOGBUFF 配置参数来指定共享内存中三个逻辑日志缓冲区的 KB 大小。

onconfig.std 值

LOGBUFF 64

单位

KB

值

从 32 至（32767 * pagesize / 1024）的整数，此处 pagesize 是缺省系统页大小。该值必须被缺省系统页大小均分。 如果该值未被页大小均分，则数据库服务器将大小向下取整为最接近于被页大小均分的那个值。

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

在其他缓冲区之一清空到磁盘时，三个逻辑日志缓冲区允许用户线程写到活动的缓冲区。

如果到活动的缓冲区充满的时候清空仍未完成，则用户线程开始写到第三个缓冲区。

如果启用 RTO_SERVER_RESTART 配置参数，则设置 LOGBUFF 配置参数的值为 256 KB。如果 LOGBUFF 配置参数的值小于 256 KB，则当您重启服务器时显示警告消息。

否则，对于标准工作负载，设置 LOGBUFF 配置参数值为 32 KB；对于重工作负载，设置为 64 KB。数据库服务器使用 LOGBUFF 参数来设置恢复期间使用的内部缓冲区的大小。如果您设置 LOGBUFF 过高，则在恢复期间数据库服务器可用光内存并关闭。

如果您将用户数据日志记录在智能大对象中，则增加日志缓冲区大小以使系统更为高效。数据库服务器仅对更改的智能大对象页的部分做日志记录。

通过运行 gstat -l 命令，您可查看关于逻辑日志缓冲区的信息。

2.1.93 LOGFILES 配置参数

使用 LOGFILES 配置参数来指定在磁盘初始化期间数据库服务器创建的逻辑日志文件的数目。

onconfig.std 值

LOGFILES 6

值

3 - 32,767（仅整数）

单位

逻辑日志文件的数目

生效

磁盘初始化期间且当您添加新的日志文件时。您使用 onparms 实用程序之一添加一个新日志。

用法

要更改逻辑日志文件的数目，请添加或删除逻辑日志文件。

如果您使用 glogadmin 来添加或删除日志文件，则数据库服务器自动地更新 LOGFILES。

2.1.94 LOG_INDEX_BUILDS 配置参数

使用 LOG_INDEX_BUILDS 配置参数来启用或禁用索引页日志记录。

onconfig.std 值

未设置

值

0 = 禁用

1 = 启用

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

如果启用 LOG_INDEX_BUILDS，则会依赖于索引的大小，增加逻辑日志文件空间的消耗。这会导致更频繁地要求备份逻辑日志文件。当索引页日志记录状态更改时，消息写到 online.log 文件。

RS 辅助服务器的提示：使用 gadmin -wm 仅启用或禁用当前会话的索引页日志记录，不影响 onconfig 文件中的设置。如果服务器停止并重启， 则 onconfig 文件中的设置决定是否启用索引页日志记录。因此当使用 RS 辅助服务器时，不建议使用 gadmin -wm 启用索引页日志记录；相反， 使用 gadmin -wf 来更新 onconfig 文件，以便在重启服务器之后启用索引页日志记录。当使用 RS 辅助服务器时，需要索引页日志记录。

2.1.95 LOG_STAGING_DIR 配置参数

使用 LOG_STAGING_DIR 配置参数来指定当在 RS 辅助服务器上配置延迟的日志文件应用时，从主服务器收到的日志文件的位置。

onconfig.std 值

未设置。

值（第一个参数）

任何有效的安全目录。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

LOG_STAGING_DIR 配置参数指定目录，在下列情况下，从主服务器发送的日志文件存储在此：

- 在一台 RS 辅助服务器上设置 DELAY_APPLY 配置参数来延迟日志的应用
- 在一台 RS 辅助服务器上设置 STOP_APPLY 配置参数来停止日志的应用
- 一台 RS 辅助服务器必须临时地缓冲日志
- 在 HDR 服务器上设置 LOG_INDEX_BUILDS 参数，且 HDR 辅助服务器正在处理检查点

通过存储来自 RS 辅助服务器的数据，延迟日志文件的应用允许您快速地从错误数据库修改恢复。

LOG_STAGING_DIR 配置参数指定的目录必须是安全的。此目录必须为用户 gbasedbt 所拥有，必须属于组 gbasedbt，且必须没有公共读、写或执行权限。

此目录应有足够的空间来保持 GBase 8s staged 所有逻辑日志。请在主服务器上选择一个至少能存储逻辑日志总计大小两倍的目录。要估算存储大小，请将 LOGBUFF 配置参数的值与 LOGFILES 配置参数的值相乘，然后再加倍。

要看到关于送到为 RS 辅助服务器设置的日志 staging 目录的数据信息，请在 RS 辅助服务器上运行 gstat -g rss verbose 命令。

如果写到 staging 文件失败，则 RS 辅助服务器发出事件报警 40007。

2.1.96 LOGSIZE 配置参数

当创建逻辑日志文件时，使用 LOGSIZE 配置参数来指定使用的文件大小。

onconfig.std 值

LOGSIZE 10000

单位

KB

值

整数值。

最小值 = 200

当数据库服务器首次初始化时，最大值 = (ROOTSIZE - PHYSFILE - 512 - (63 * pagesize/1024)) / LOGFILES

pagesize 值是操作系统的缺省系统页大小。

如果您扩展 root dbspace 或将逻辑日志移到一个不同的 dbspace，则逻辑日志文件大小的最大值不能超过下列与页大小相依的值：

- 1 GB，当页大小 = 2 KB
- 2 GB，当页大小 = 4 KB

此限制是日志位置可以为那些页大小描述的页的最大数目。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

当您更改 LOGSIZE 配置参数的值时，仅影响新的日志文件。现有的日志文件的大小不更改。逻辑日志大小的合计是 LOGSIZE 配置参数设置与 LOGFILES 配置参数值的乘积。然而，如果您更改 LOGSIZE 配置参数的值，则所有逻辑日志文件的大小合计依赖于每个大小的日志文件的数目。

如果启用 AUTO_LLOG 配置参数，则根据需要自动地添加逻辑日志文件来提高性能，直到逻辑日志大小合计达到可配置最大值。

要验证您的平台上数据库服务器的页大小，请运行 gstat -b 命令。

如果您声明一个智能大对象列的日志记录，则您必须确保此逻辑日志比插入或更新期间日志记录的数据量大许多。数据库服务器不可备份打开的事务。如果许多事务是活动的，则合计日志记录活动必须不强制打开的事务到日志备份文件。例如，如果您的日志大小是 1000 KB 且高水印是 60%，则不为智能大对象更新使用多于 600 KB 逻辑日志。当达到高水印 600 KB 时，数据库服务器启动回滚事务。

2.1.97 LOW_MEMORY_MGR 配置参数

使用 LOW_MEMORY_MGR 配置参数来启用自动低内存管理，当达到内存限制时，您可用来更改主服务器或标准服务器的缺省行为。

onconfig.std 值

LOW_MEMORY_MGR 0

值

1 = 当数据库服务器启动时，启用自动低内存管理。

0 = 禁用自动低内存管理。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

用法

如果您配置主服务器或标准服务器来使用 SHMTOTAL 配置参数值的一个百分率，该值用做自动低内存管理启动和停止阈值，则 SHMTOTAL 配置参数必须设置为一个正整数值。

注意： 更改 SHMTOTAL 配置参数的值可导致自动低内存管理的配置变得无效，强制数据库服务器使用缺省设置。

要启用自动低内存管理，请指定：

```
LOW_MEMORY_MGR 1
```

2.1.98 LOW_MEMORY_RESERVE 配置参数

当关键活动需要而服务器的空闲内存有限时，使用 LOW_MEMORY_RESERVE 配置参数来保留使用特定的内存量。

如果您通过将其设定为一个特定的 KB 值来启用新的 LOW_MEMORY_RESERVE 配置参数，则即使您收到内存用尽的警告，诸如回滚活动这样的关键活动仍可完成。

onconfig.std 值

LOW_MEMORY_RESERVE 0

值

0 或 128 - 2147483648，虽然最大值不可高于 SHMVIRTSIZE 配置参数值的 20%

单位

KB

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

不管如何设置 LOW_MEMORY_RESERVE 配置参数，保留的内存大小最大值是 SHMVIRTSIZE 配置参数值的 20%。

例如，要保留 512 KB 内存，请指定：

```
LOW_MEMORY_RESERVE 512
```

您可使用 gstat -g seg 命令来查看低内存保留信息。输出包括若干行，显示保留的内存大小、服务器已经使用的保留内存次数和需要的最大内存。

2.1.99 LTAPEBLK 配置参数

用于逻辑日志备份的磁带的块大小（以千字节为单位）。

2.1.100 LTAPEDEV 配置参数

逻辑日志磁带设备或文件系统目录。

在云环境中备份或恢复时，将以下语法用于 LTAPEDEV 配置参数：

```
LTAPEDEV 'local_path, keep=option, cloud=cloud_vendor, url=url'
```

- local_path 是将日志备份对象临时存储在的目录的完整路径名。
- option 可以设置为 yes 或 no。如果 keep 设置为 yes，那么 gtape 实用程序将备份对象保留在本地目录中。如果 keep 设置为 no，那么在云存储位置中传输备份对象后，会将这些对象删除。
- cloud_vendor 是云存储供应商的名称。
- url 是将日志备份数据永久存储在的云存储位置。

2.1.101 LTAPESIZE 配置参数

用于逻辑日志备份的磁带的大小（以千字节为单位）。值可以为 0 - 2,097,151。

2.1.102 LTXEHWM 配置参数

使用 LTXEHWM 配置参数来指定长事务、排他访问、高水印。当逻辑日志空间达到 LTXEHWM 阈值时，当前正在回滚的长事务获准排他访问逻辑日志。

onconfig.std 值

LTXEHWM 80

如未出现

90（如果 DYNAMIC_LOGS 设置为 1 或 2）60（如果 DYNAMIC_LOGS 设置为 0）

值域

LTXHWM 至 100

单位

百分比

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

如果当事务达到长事务高水印时，事务不提交或回滚，则这个事务是长事务。

如果在回滚完成前耗尽您的系统长空间，则请降低 LTXEHWM 值。

如果您不想添加过多的逻辑日志，则应将 LTXEHWM 设置为一个较小值（大约 60）。如果关闭动态日志记录（DYNAMIC_LOGS = 0），则应将 LTXEHWM 设置更低（大约 50）来避免耗尽逻辑空间。

提示： 要允许用户继续访问逻辑日志，即使在长事务回滚期间，请设置 LTXEHWM 为 100。设置 DYNAMIC_LOGS 为 1 或 2 以便数据库服务器可添加足够数目的日志文件，来防止长事务挂起并允许长事务回滚。

2.1.103 LTXHWM 配置参数

使用 LTXHWM 配置参数来指定长事务高水印。长事务高水印是可用的日志空间百分率，当填满时，触发数据库服务器来检查长事务。

onconfig.std 值

LTXHWM 70

如未出现

80（如果 DYNAMIC_LOGS 设置为 1 或 2）50（如果 DYNAMIC_LOGS 设置为 0）

值

1 - 100

单位

百分比

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

当逻辑日志空间达到 LTXHWM 阈值时，数据库服务器启动回滚事务。如果您降低 LTXHWM 值，则增加日志文件的大小或数目来减少回滚的可能性。

如果 DYNAMIC_LOGS 设置为 1 或 2，则当有长事务时，数据库服务器可添加足够数目的日志文件来完成事务或防止回滚挂起。

如果您不想添加过多的逻辑日志，则应将 LTXHWM 设置为一个较小值（大约 60）。如果关闭动态日志记录（DYNAMIC_LOGS = 0），则应将 LTXHWM 设置更低（大约 50）来避免逻辑空间耗尽。

警告： 如果您将 LTXHWM 和 LTXEHWM 都设置为 100，则从不终止长事务。虽然您可为了对自己有利而使用此配置，但是对于正常的数据库服务器操作，您应将 LTXHWM 设置低于 100。

如果您设置 LTXHWM 为 100，则数据库服务器发出警告消息：

LTXHWM is set to 100%. This long transaction high water mark will never be reached. Transactions will not be aborted automatically by the server, regardless of their length.

如果事务挂起，则请遵循 GBase 8s 管理员指南中关于管理逻辑日志文件的章节指导，从长事务挂起恢复。

2.1.104 MAX_FILL_DATA_PAGES 配置参数

onconfig.std 值

MAX_FILL_DATA_PAGES 0

值

0 或 1

单位

整数

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

设置 MAX_FILL_DATA_PAGES 值为 1 来允许在有变长行的表中每页插入更多的行。该设置可减少磁盘空间，更有效地利用缓冲池，减少表扫描次数。

如果启用 MAX_FILL_DATA_PAGES，则服务器将添加一个新行到最近更改的带着现有行的页，如果添加该行留下了至少 10% 的空闲页用于在此页中所有行的未来扩展。如果未设置 MAX_FILL_DATA_PAGES，则服务器仅当页上有足够空间时才添加行，来允许新行增长到其最大长度。

启用 MAX_FILL_DATA_PAGES 并允许每页更多的变长行可能带来问题，就是服务器会以不同的物理顺序存储这些行。而且，当页填满时，对一行中可变长列的更新可能导致行扩展，以致于在页上不再完全适合。这导致服务器将该行分裂到两个页上，增加该行的访问时间。要利用这个设置，随着进一步插入，必须重新加载带可变长行的现有的表或必须更改现有的页。

2.1.105 MAX_INCOMPLETE_CONNECTIONS 配置参数

使用 MAX_INCOMPLETE_CONNECTIONS 配置参数来指定一个会话中未完成连接的最大数目。

onconfig.std 值

MAX_INCOMPLETE_CONNECTIONS 1024

单位

未完成连接的数目

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

达到 MAX_INCOMPLETE_CONNECTIONS 配置参数中指定的数目之后，在 online 消息中写入错误消息，说明服务器可能受到“拒绝服务”袭击。另请参阅关于 LISTEN_TIMEOUT 配置参数的信息，那个配置参数指定服务器等待连接的秒数。

依赖于机器保持线程（数目）的能力，您可配置 MAX_INCOMPLETE_CONNECTIONS 为一个较高值。依赖于网络流量，您还可设置 LISTEN_TIMEOUT 配置参数为一个较低值来降低袭击可达到最大限制的可能性，该参数指定服务器等待连接的秒数。

2.1.106 MAX_PDQPRIORITY 配置参数

使用 MAX_PDQPRIORITY 配置参数来限制 PDQ 资源，数据库服务器可将这些资源分配给任一 DSS 查询。

onconfig.std 值

MAX_PDQPRIORITY 100

值

0 = 关闭 PDQ。DSS 查询不使用并行性。

1 = 从分片表并行取回数据（并行扫描），但不使用其他形式的并行性。

2 - 100 = 设置实际地分配给该查询的用户要求的 PDQ 资源百分率。100 使用所有可用资源来并行处理查询。

生效

在所有用户会话上，编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

MAX_PDQPRIORITY 是用来衡量用户设置的 PDQ 优先权的一个因素。例如，假定数据库管理员设置 MAX_PDQPRIORITY 为 80。如果一个用户设置 PDQPRIORITY 环境变量 为 50 然后发出一个查询，则数据库服务器自动地以 PDQ 优先权 40 处理该查询。

在数据库服务器 online 时，您可使用 gadmin 实用程序来更改 MAX_PDQPRIORITY 的值。在 GBase 8s 中，PDQ 资源包括内存、CPU、磁盘 I/O 和扫描线程。MAX_PDQPRIORITY 让数据库管理员与 OLTP 并发地运行决策支持，未影响 OLTP 性能。 然而，如果 MAX_PDQPRIORITY 过低，可降低决策支持查询的性能。

2.1.107 MIRROR 配置参数

onconfig.std 值

MIRROR 0

值

0 = 禁用镜像

1 = 启用镜像

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

MIRROR 参数指明是否启用数据库服务器的镜像。推荐您镜像 root dbspace 和作为初始化部分的关键数据。否则，禁用镜像。如果您后来决定添加镜像，则可编辑配置文件来更改该参数值。

在高可用性数据复制对中的两台数据库服务器上，您不必设置 MIRROR 配置参数为同样的值。在主数据库服务器或辅助数据库服务器上，您可任意地启用或禁用镜像。请不要设置 MIRROR 配置参数为 1，除非您正在使用镜像。

2.1.108 MIRROROFFSET 配置参数

在 GBase 8s 中，为达到作为 root dbspace 的初始 chunk 镜像的 chunk，MIRROROFFSET 指定磁盘分区中的偏移量或设备中的偏移量。

onconfig.std 值

MIRROROFFSET 0

值

大于或等于 0 的任意值

单位

KB

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.109 MIRRORPATH 配置参数

使用 MIRRORPATH 配置参数来为 root dbspace 的初始 chunk 指定镜像 chunk 的全路径名。

onconfig.std 值

UNIX™ 上：$GBASEDBTDIR/tmp/demo_on.root_mirror

Windows™ 上：None

值

65 个或更少字符

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

MIRRORPATH 应为一个指向实际镜像 chunk 的 chunk 路径名的链接，由于相同的原因，指定 ROOTPATH 为一个链接。类似地，选取镜像 chunk 的一个短路径名。

您必须设置 MIRRORPATH 指定的那个文件的权限为 660。所有者和组都必须是 gbasedbt。

如果您在 UNIX 平台上使用镜像 chunk 的原始磁盘空间，则推荐您定义 MIRRORPATH 为链接到镜像 dbspace 的初始 chunk 的一个链接，而不要输入初始 chunk 的实际设备名称。

要在数据库服务器上启动镜像数据，而该服务器没有运行启用的镜像功能，请：

1. 将数据库服务器 offline。
2. 更改 MIRROR 配置参数为 1 并将 MIRRORPATH 配置参数置空。
3. 将数据库服务器 online。
4. 为镜像 chunk 分配磁盘空间。您可在任何时间分配此磁盘空间，然而，当您在下一步骤中指定镜像 chunk 时，该磁盘空间必须可用。镜像 chunk 必须与对应的主 chunk 位于不同的磁盘上。
5. 指定 gspaces -m 选项来启动 dbspace、blobspace 或 sbspace 的镜像。您必须从 root dbspace 开始。成功地运行 root dbspace 命令之后，服务器自动地设置 MIRRORPATH 值。

2.1.110 MSG_DATE 配置参数

使用 MSG_DATE 配置参数来启用在打印到 online 日志的每一消息的开头插入 MM/DD/YY 格式的日期。

onconfig.std 值

不在 onconfig.std 文件中。

值

0 = OFF（缺省）

1 = ON

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

在下列示例中， MSG_DATE 设置为 1（ON）。

```
04/10/11 10:26:06  MSG_DATE 值已修改为 1
04/10/11 10:27:35  MSG_DATE 值已修改为 1
```

2.1.111 MSGPATH 配置参数

使用 MSGPATH 配置参数来指定消息日志文件的全路径名。在操作期间，数据库服务器将状态消息和诊断消息写到这个文件。

onconfig.std 值

UNIX™ 上：$GBASEDBTDIR/tmp/online.log

Windows™ 上：%GBASEDBTDIR%\online.log

在 Windows 上，如果您在安装期间创建服务器实例：%GBASEDBTDIR%\server_name.log。server_name 是程序组中的服务器名称。

值

online.log 文件的路径名。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

如果 MSGPATH 指定的文件不存在，则数据库服务器在特定目录中创建该文件。如果 MSGPATH 指定的目录不存在，则数据库向系统控制台发送消息。

如果 MSGPATH 指定的文件确实存在，则数据库服务器打开它并当消息发生时追加消息到这个文件。

2.1.112 MULTIPROCESSOR 配置参数

使用 MULTIPROCESSOR 配置参数来指定数据库服务器是否以一种适合于单处理器计算机或多处理器计算机的方式执行锁定。

如果 MULTIPROCESSOR 设置为 0，则忽略设置处理器 GBase 8s affinity 的参数。

onconfig.std 值

MULTIPROCESSOR 0

值

0 = 无多处理器

1 = 多处理器可用

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.113 NET_IO_TIMEOUT_ALARM 配置参数

如果网络写操作被阻塞达 30 分钟或更多，则使用 NET_IO_TIMEOUT_ALARM 配置参数来控制是否要通告。

阻塞网络写操作通常表明操作系统问题。对特定的网络流量类型，使用 NET_IO_TIMEOUT_ALARM 配置参数来启用事件报警 82。

onconfig.std 值

不在 onconfig.std 中

值

下列值之一或者一个或多个下列值的总计：

- 0 = 禁用
- 1 = 对 Enterprise Replication 操作启用
- 2 = 对分布式查询启用
- 4 = 对 HDR 操作启用
- 8 = 对 SMX 操作启用
- 16 = 对其他组件操作启用

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

2.1.114 NETTYPE 配置参数

使用 NETTYPE 参数来调整您在 sqlhosts 信息中定义的网络协议。

onconfig.std 值

UNIX™: ipcshm,1,50,CPU

Windows™: 未设置。

缺省值

connection_type,1,50,vp_class

缺省连接类型依赖于操作系统：

- UNIX：来自 sqlhosts 文件的 protocol 域值。

分隔符

以逗号分隔域。不包括空格。如果您可省略域值，但您必须为每个域包括一个逗号。然而，您可省略尾部的逗号。

值

请参阅“用法”部分。

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

NETTYPE 提供对协议与接口组合的调整选项，与 sqlhosts 信息中的 dbservername 条目相关联。

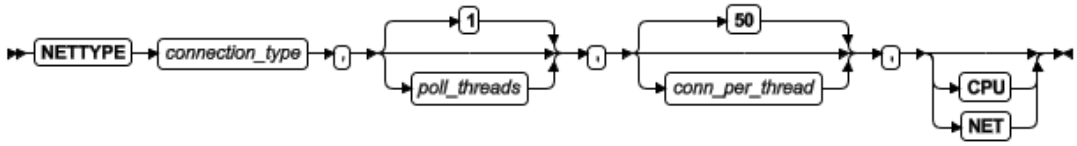


表 1. NETTYPE 配置参数值的选项.

域	值
connection_type	有效协议与接口的组合，带有或没有数据库服务器前缀 on、ol 或 dr。
poll_threads	指定给该连接类型的轮询线程数。缺省是 1。值域依赖于操作系统和虚拟处理器类： ● UNIX：如果虚拟处理器类是 NET，则是一个大于或等于 1 的整数。每一轮询线程需要一个单独的虚拟处理器， 因此当您指定一个接口、协议组合的轮询线程数并指定由 NET 类运行它们时，请直接指定网络虚拟处理器的数目。 ● UNIX：如果虚拟处理器类是 CPU，则是一个从 1 至 CPU VP 数目的一个整数。 ● Windows：一个大于或等于 1的整数。 如果数据库服务器有许多连接，则您可能通过增加轮询线程数来提高性能。通常，每一轮询线程可控制大约 200 - 250 个连接。 Windows：如果指定 soctcp 协议，则只创建一个轮询线程，反之，一个套接字 I/O 线程（soctcpio）被创建在自己的每一轮询线程的 SOC VP 中，由 NETTYPE 参数指定轮询线程。套接字 IO 线程使用 I/O 完成端口来接收完成通告，控制所有连接的接收操作。在 Windows 平台上，这些线程执行大量服务网络连接工作。
conn_per_thread	1 - 32767 的整数设置每一轮询线程的最大连接数。缺省是 50。 对于共享内存连接，conn_per_thread 的值是每个线程的最大连接数。通常，指定预期连接数的两倍。 对于网络连接，可超出 conn_per_thread 的值。根据需要，轮询线程动态地重新分配资源来支持更多连接。请避免设置并发连接数目

域	值
	的值远高于您的预期。否则，您可能浪费系统资源。 如果仅有几个连接在并发地使用一个协议，则您可通过明确地设置连接的预计数目来节省内存。
CPU	指定一个 CPU 虚拟处理器。对于共享内存连接，使用 CPU 虚拟处理器，共享内存连接应运行在每个 CPU 虚拟处理器中。
NET	指定一个 NET 虚拟处理器。对于网络连接，使用 NET 虚拟处理器。

您可为想让数据库服务器使用的每一协议指定一个 NETTYPE 参数。下列示例表明到数据库服务器的两类连接的 NETTYPE ：本地客户端的一个共享内存连接，以及使用套接字的一个网络连接：

```
NETTYPE ipcshm,3,,CPU
NETTYPE soctcp,8,300,NET
```

共享内存连接的 NETTYPE 参数（ipcshm）指定在 CPU 虚拟处理器中三个轮询线程来运行。连接数未指定，于是设置为 50。对于 ipcshm，轮询线程数对应于内存段的数目。

套接字连接的 NETTYPE 参数（soctcp）为这个协议指定每一线程预期 300 个同时发生的连接，且在 NET 虚拟处理器中将运行 8 个轮询线程。

UNIX：在 NETTYPE 和 NUMFDSERVERS 配置参数设置之间可有依赖。当有多个 CPU 虚拟处理器和轮询线程，且 gstat -g ath 命令的线程状态输出表明网络共享文件（NSF）锁定时，您可增加轮询线程的 NUMFDSERVERS 值来减少 NSF 锁争用。

GBase 8s MaxConnect

如果您正在使用 GBase 8s MaxConnect，要了解如何指定 NETTYPE 参数中的域，请参阅 GBase 8s MaxConnect 用户指南。ontliimc 和 onsocimc 协议使用 TCP/IP 来与 GBase 8s MaxConnect 通信。您可使用这些协议来连接 GBase 8s MaxConnect 或者连接应用程序客户端到数据库服务器。

2.1.115 NS_CACHE 配置参数

使用 NS_CACHE 配置参数来定义 GBase 8s 名称服务高速缓存中条目的最大保留时间：主机名 /IP 地址高速缓存、地址高速缓存、服务高速缓存、用户高速缓存和组高速缓存。

onconfig.std 值

```
NS_CACHE host=900,service=900,user=900,group=900
```

值

每一域都是一个等于或大于 0 的整数值。

host = 设置在主机名或 IP 地址高速缓存中高速缓存信息的秒数。

service = 设置在服务高速缓存中高速缓存信息的秒数。

user = 设置在用户高速缓存中高速缓存信息的秒数。

group = 设置在组高速缓存中高速缓存信息的秒数。

0 = 禁用高速缓存。服务器总是从操作系统取得信息。您可设置一个别高速缓存为 0 或设置所有服务高速缓存为 0：NS_CACHE 0。

单位

秒

分隔符

用逗号分隔值。不包括空格。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

为了查找和解析主机名（或 IP 地址）、服务名、用户（及口令）或组，数据库服务器使用适当的系统调用查询操作系统（OS）。通过使用 GBase 8s 名称服务高速缓存机制，您可避免这些 OS 查找中的许多，在可配置的时间量中，该机制可保持和重用每一检索的信息条。如果操作系统不提供其自己的高速缓存，则您应设置 NS_CACHE 配置参数。

服务器从高速缓存获得信息比在查询操作系统时更快。然而，如果您通过设置保留时间为 0 禁用这些高速缓存中的一个或多个，则数据库服务器为主机、服务、用户或组信息查询操作系统。

在操作系统级别对名称服务进行的更改不立即反映在 GBase 8s 名称服务高速缓存中：例如，IP 地址的更改、用户添加到组或从组中移走、或者新的口令。然而， 您可使用 gadmin -wf 或 gadmin -wm 命令来立即更改 NS_CACHE 信息。当您用 gadmin -wf 或 gadmin -wm 命令更改一个特别的高速缓存值时 ，服务器立即终止那个高速缓存中所有现存的条目。

2.1.116 NUMFDSERVERS 配置参数

对于 UNIX™ 上的网络连接，使用 NUMFDSERVERS 配置参数来指定轮询线程的最大数目，以控制网络连接在 GBase 8s 虚拟处理器（VP）之间迁移。

如果 GBase 8s 有高频率的新建连接和断开连接要求，或者如果您发现在网络共享文件(NSF)锁之间存在大量争用，则指定 NUMFDSERVERS 信息是有用的。 您可使用 gstat -g ath 命令来显示关于所有线程的信息。这个信息包括诸如 mutex wait nsf.lock 之类的状态，表明您有大量的 NSF 锁争用。

onconfig.std 值

NUMFDSERVERS 4 （仅每一 nettype 的前 4 个轮询线程，涉及管理连接迁移。）

值

1 - 50

实际数目依赖于在 NETTYPE 配置参数中指定的轮询线程的数目。

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

指定的 NUMFDSERVERS 值对共享内存（SHM）连接没有影响。

如果您使用 NUMFDSERVERS 配置参数，如有必要还会审核它，则在 NETTYPE 配置参数中更改轮询线程的数目。例如，如果您有多个 CPU VP 和轮询线程，且导致 NSF 锁定，则可增加 NUMFDSERVERS 和轮询线程来减少 NSF 锁争用。

2.1.117 OFF_RECVRY_THREADS 配置参数

当数据库服务器 offline（冷恢复期间）时，使用 OFF_RECVRY_THREADS 配置参数来指定用在逻辑恢复中的恢复线程数。该线程的数目还用于在快速恢复中的前滚逻辑日志记录。

onconfig.std 值

OFF_RECVRY_THREADS 10

值

正整数

单位

并行运行的恢复线程的数目

生效

编辑 onconfig 文件并重启数据库服务器之后。

参考

- GBase 8s 备份与恢复指南
- GBase 8s 性能指南

用法

在执行冷恢复之前，您可设置该参数的值约等于逻辑日志中有大量事务的表的数目。对于单处理器计算机或节点，多于 30 至 40 个线程可能就太多了，因为线程管理的开销抵消了并行处理中的增长。

2.1.118 ON_RECVRY_THREADS 配置参数

当数据库服务器 online（热恢复期间）时，ON_RECVRY_THREADS 配置参数是数据库服务器用户用于逻辑恢复的恢复线程的最大数。

onconfig.std 值

ON_RECVRY_THREADS 1

值

正整数

单位

并行运行的恢复线程数

生效

编辑 onconfig 文件并重启数据库服务器之后。

参考

- GBase 8s 备份与恢复指南
- GBase 8s 性能指南

用法

您可调整 ON_RECVRY_THREADS 为可能被恢复的表数，因为在恢复期间，根据表数为处理的逻辑日志指定线程。当恢复线程数与正在恢复的表数相配时，会发生最大程度的并行处理。要提高热恢复的性能，请用 ON_RECVRY_THREADS 参数增加快速恢复线程的数目。

2.1.119 ONDBSPACEDOWN 配置参数

当非关键 dbspace 内的主 chunk 上发生任何禁用事件时，使用 ONDBSPACEDOWN 配置参数来定义数据库服务器采取的行动。

onconfig.std 值

ONDBSPACEDOWN 2

值

0 = 数据库服务器将 dbspace 标记为 offline 并继续。

1 = 数据库服务器终止。

2 = 数据库服务器将 chunk 的状态写到日志并等待用户输入。如果您设置这个选项，但您不想数据库服务器将一个禁用的 dbspace 标记为 down 并继续处理，则使用 gadmin -O 来取代这个 ONDBSPACEDOWN 设置。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

当不应用 ONDBSPACEDOWN 时的数据库服务器行为

如果任何关键 dbspace（例如，rootdbs 或 logsdbs）内的 chunk 丢失，则数据库服务器将不 online。

ONDBSPACEDOWN 的值不影响临时 dbspace。对于临时 dbspace，数据库不管 ONDBSPACEDOWN 设置，继续处理。如果临时 dbspace 需要修理，则您应停止并重新创建它。

对于非关键 dbspace 内的非主 chunk，当禁用事件发生时，数据库服务器的行为依赖于 chunk 的事务状态：

- **无事务：**如果未检测到那个 chunk 的事务，则单独地将该 chunk 标记为关闭。在这种情况下，随后对那个 chunk 的写尝试失败，回滚相关的事务。您可安全地放回该 chunk，然后使用 gspaces -s 实用程序来将该 chunk 标记为 back online。
- **检测到事务：**如果有事务前滚或回滚，那么数据库终止并伴以适当的快速恢复错误。在这种情况下，您应放回该 chunk 并重启数据库服务器。

2.1.120 ONLIDX_MAXMEM 配置参数

使用 ONLIDX_MAXMEM 配置参数来限定分配给单个 preimage 池和单个 updator 日志池的内存数量。

onconfig.std 值

ONLIDX_MAXMEM 5120

值

16 - 4294967295

单位

KB

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

preimage 和 updator 日志池,pimage_partnum 和 ulog_partnum,是当执行 CREATE INDEX ONLINE 语句时创建的共享内存池。当语句执行完毕时，释放这些池。

如果您为这个参数指定一个值，然后创建一个表，添加行到该表并在一列上启动执行 CREATE INDEX ONLINE 语句,则您还可在此列上执行其他操作,诸如运行 UPDATE STATISTICS HIGH，而不会引起内存问题。

2.1.121 OPTCOMPIND 配置参数

使用 OPTCOMPIND 来指定信息，帮助优化器选择一个对您的应用适当的查询计划。

提示： 您可认为此变量名出自“OPTimizer COMPare (the cost of using) INDexes (with other methods)。”

onconfig.std 值

OPTCOMPIND 2

值

0 = 当表的每一排序的对存在适当的索引时，用户器选择索引扫描（嵌套循环联接），不计成本，不选择表扫描（散列联接）。

1 = 如果隔离级别不是 Repeatable Read，则优化器根据成本来确定执行路径。否则，优化器选择索引扫描（效果与值 0 相同）。此设置是优化性能的推荐设置。

2 = 优化器根据成本来确定任何隔离级别的执行路径。索引扫描不优先于表扫描；优化器纯粹地根据成本做决定。如果未设置变量，则该值是缺省值。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

由于散列联接的本质，一个隔离模式设置为 Repeatable Read 的应用可能临时地锁定表中的所有记录，这些表参与表的每一排序集的联接（即使那些记录不符合联接条件）。这种情况导致联接中发生较高的争用。相反地，嵌套循环联接锁定较少记录，但当数据库服务器检索大量行时性能较差。因此，两种联接方式都各有利弊。客户端应用也可影响优化器对联接方式的选择。

2.1.122 OPT_GOAL 配置参数

onconfig.std 值

OPT_GOAL -1

值

0 或 -1

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

OPT_GOAL 参数使您能指定下列查询的优化目标之一：

优化 FIRST ROWS

优化 ALL ROWS

值 0 设置优化目标为 FIRST_ROWS。值 -1 设置优化目标为 ALL_ROWS，这是缺省值。

当您设置优化目标为优化 FIRST ROWS 时，请指定您想要数据库服务器优化感知响应时间的查询。换句话说，交互应用的用户感知的响应时间，就是花费在屏幕上显示数据的时间。设置优化目标为 FIRST ROWS，配置数据库服务器来返回满足查询的前几行数据。

当您设置优化目标为优化 ALL ROWS 时，请指定您想要数据库服务器优化的查询执行时间的总计。使 ALL ROWS 优化目标指导数据库服务器来尽快处理总计查询，不管将前几行数据返回到应用需要花费多长时间。

您可用四种方法之一指定优化目标：

- 通过查询（SELECT 语句）

  使用 ALL_ROWS 和 FIRST_ROWS 指令。

- 通过会话

  使用 SET OPTIMIZATION 语句。

- 通过环境

  设置 OPT_GOAL 环境变量。

- 通过数据库服务器

  设置 OPT_GOAL 配置参数。

上述列表按优先的降序罗列设置这个目标的机制。要确定优化目标，数据库服务器按上述顺序检查设置。以遇到的第一个设置来确定优化目标。例如，如果查询包括 ALL_ROWS 指令，但 OPT_GOAL 配置参数设置为 FIRST_ROWS，则 数据库服务器按照查询指定的，优化 ALL_ROWS。

2.1.123 PC_HASHSIZE 配置参数

使用 PC_HASHSIZE 来指定在数据库服务器使用的高速缓存中散列存储区的数目。

PC_HASHSIZE 仅适用于 UDR 高速缓存。

onconfig.std 值

PC_HASHSIZE 31

值

任何正整数，推荐质数。

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.124 PC_POOLSIZE 配置参数

使用 PC_POOLSIZE 配置参数来指定存储在 UDR 高速缓存中的用户定义的例程的最大数目。

onconfig.std 值

PC_POOLSIZE 127

值

一个正值 127 或更大，表示高速缓存中条目的初始最大数目的一半。最大值依赖于共享内存配置和服务器实例的可用共享内存。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wm 命令增加内存中的该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

高速缓存中条目的初始数目是 PC_POOLSIZE 配置参数值的两倍。例如，如果 PC_POOLSIZE 配置参数设置为 127，则高速缓存中允许 254 个条目。 如果高速缓存中所有条目填满，则高速缓存大小自动地增长 10%。要降低高速缓存的大小，请减小 onconfig 文件中 PC_POOLSIZE 配置参数的值并重启服务器。

2.1.125 PHYSBUFF 配置参数

使用 PHYSBUFF 配置参数来指定以 KB 为单位的共享内存中两个物理日志缓冲区的大小。

onconfig.std 值

PHYSBUFF 128

单位

KB

值

范围从 4 -（32767 * pagesize / 1024）的整数，此处 pagesize 是缺省系统页大小。该值必须能被缺省系统页大小均分。如果该值不被页大小均分，则数据库服务器向下取整这个大小为最接近可被页大小均分的值。

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

当其他缓冲区正被清空到磁盘上的物理日志时，加倍缓冲准许用户线程写到活动的物理日志缓冲区。向物理日志缓冲区的写长度恰好为一页。PHYSBUFF 参数的值确定数据库服务器需要将物理日志缓冲区清空到物理日志文件的频繁程度。

如果启用 RTO_SERVER_RESTART 配置参数，则使用 512 KB 为 PHYSBUFF 的缺省值。当启用 RTO_SERVER_RESTART 配置参数时，如果 PHYSBUFF 配置参数的值小于 512 KB，则在重启服务器时显示警告消息。

智能大对象的用户数据部分不通过物理日志缓冲区。

2.1.126 PHYSFILE 配置参数

当您初始化磁盘空间并使数据库服务器 online 时，使用 PHYSFILE 配置参数来指定物理日志文件的大小。

onconfig.std 值

PHYSFILE 50000

如未出现

200

值

整数 200 或更大

单位

KB

生效

编辑 onconfig 文件并通过运行 oninit -i 命令初始化磁盘空间之后。

运行 glogadmin -p -s 命令之后。

用法

初次启动服务器之后，您不可通过编辑 onconfig 更改 PHYSFILE 配置参数的值。

在下列情况下，数据库服务器更新 onconfig 文件中的 PHYSFILE 配置参数值：

- 通过运行 glogadmin -p -s 命令更改物理日志文件的大小。
- 自动地扩展 plogspace。如果物理日志存储在 plogspace 中，则数据库服务器根据需要扩展物理日志的大小来提高性能。

当启用 RTO_SERVER_RESTART 配置参数时，请确保物理日志的大小至少等于缓冲池大小的 110%。在下列时刻，会向消息日志打印警告消息：

- PHYSFILE 配置参数的值更改为小于所有缓冲池的 110%
- 重启服务器
- 添加新的缓冲池

2.1.127 PLOG_OVERFLOW_PATH 配置参数

如果物理日志文件溢出，则 PLOG_OVERFLOW_PATH 参数指定在快速恢复期间使用的文件的位置。

该文件是 plog_extend.servernum，缺省位置是 $GBASEDBTDIR/tmp。使用全路径名以 PLOG_OVERFLOW_PATH 参数来指定文件的不同位置。

onconfig.std 值

UNIX™ 上：$GBASEDBTDIR/tmp

Windows™ 上：None

生效

当数据库服务器启动时（初始化共享内存）

2.1.128 PLCY_HASHSIZE 配置参数

PLCY_HASHSIZE 配置参数指定安全策略信息高速缓存中散列存储区的数目。

onconfig.std 值

PLCY_HASHSIZE 31

值

任意正整数

单位

KB

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.129 PLCY_POOLSIZE 配置参数

使用 PLCY_POOLSIZE 配置参数来指定安全策略信息高速缓存的每一散列存储区中的最大条目数。

onconfig.std 值

PLCY_POOLSIZE 127

值

正值 127 或更大，表示高速缓存中条目的初始最大数目的一半。最大值依赖于共享内存配置和服务器实例的可用共享内存。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wm 命令增加内存中的该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

共享内存中的初始条目数是 PLCY_POOLSIZE 配置参数值的两倍。例如，如果 PLCY_POOLSIZE 配置参数设置为 127，则高速缓存中允许 254 个条目。如果高速缓存中的所有条目填满，则高速缓存大小自动地增长 10%。 要减小高速缓存的大小，请减小 onconfig 文件中 PLCY_POOLSIZE 配置参数的值并重启服务器。

2.1.130 PN_STAGEBLOB_THRESHOLD 配置参数

使用 PN_STAGEBLOB_THRESHOLD 配置参数来为轮转法分片中的 BYTE 和 TEXT 数据保留空间。

onconfig.std 值

未设置。

如未出现

0

值

0 - 1000000

单位

KB

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

设置这个配置参数为存储在表中的 BYTE 或 TEXT 数据的典型或平均大小。

限制： 如果 extent 的数目已经达到允许的最大 extent，或如果 dbspace 填满，则 PN_STAGEBLOB_THRESHOLD 配置参数没有影响。

当表达到一个分片的页的最大数目时，可通过添加新的分片来向表添加更多页。然而，如果表中包含 BYTE 或 TEXT 列，且按轮转法分发方案分片表，则添加新分片不会自动地使新行能够插入到新分片内。

例如，如果表中的分片之一达到了最大页数，则添加新分片不会扩展表来存储更多行。因为 BYTE 和 TEXT 数据属于大型数据，所以在所有分片中平均地分发之前，该数据在其中一个分片中 staged。 此 staging 分片必须有充足的空间来存储 BYTE 或 TEXT 数据。使用 PN_STAGEBLOB_THRESHOLD 配置参数以便数据库服务器可在一个 staging 分片中临时地 stage BYTE 或 TEXT 数据，直到 INSERT 操作完成且永久地将数据存储在表中。

在 UPDATE 操作期间，如果分片没有 PN_STAGEBLOB_THRESHOLD 配置参数中指定的空间，则受更新处理影响的表行会移到另一分片内。

2.1.131 PRELOAD_DLL_FILE 配置参数

PRELOAD_DLL_FILE 配置参数指定共享库文件的路径名，当数据库服务器启动时预先加载该文件。

onconfig.std 值

未设置。无预先加载的共享库文件。

值

pathname = 共享库文件的全路径名。下列是一些示例：

- /finance/jeffzhang/mylib.udr
- home/gbasedbt/extend/blade.so
- \work4\lauragupta\userfuncdir\cudrs.ddl

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

使用这个参数来预先加载用户定义的例程的共享库文件，用 C 编程语言创建这些例程（C UDR）。对于您想要预先加载的每一库文件，请添加这个参数的一个单独条目。 预先加载的 C UDR 共享库保持活动状态直到服务器停止。

限制： 您不可使用 gadmin -wm 或 gadmin -wf 命令来设置 PRELOAD_DLL_FILE 配置参数。

2.1.132 QSTATS 配置参数

QSTATS 配置参数指定 gstat -g qst 来打印队列统计信息的能力。

onconfig.std 值

QSTATS 0

值

0 = 禁用队列统计信息

1 = 启用队列统计信息

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.133 RA_PAGES 配置参数

RA_PAGES 配置参数指示数据库服务器在顺序扫描数据或索引页期间在单个 I/O 操作中引入内存的页数。

2.1.134 RA_THRESHOLD 配置参数

RA_THRESHOLD 配置参数指示数据库服务器发出 I/O 请求以从磁盘引入下一组页的点。

因为 I/O 等待时间的较大一部分与寻找磁盘上的正确起始点有关，所以可通过增加随每次传输带入的连续页数来提高顺序扫描的效率。

但是，就 BUFFERPOOL 配置参数中 buffers 的值而言，将 RA_PAGES 设置得太大或将 RA_THRESHOLD 设置得太高会触发不必要的页面清除从而为没有迫切需要的页腾出空间。

使用以下公式计算 RA_PAGES 和 RA_THRESHOLD 的值：

```
RA_PAGES = ((BUFFERS * bp_fract) / (2 * large_queries)) + 2
RA_THRESHOLD = ((BUFFERS * bp_fract) / (2 * large_queries)) - 2
```

bp_fract

是用于要求预先读取的大型扫描的数据缓冲区部分。如果希望大型扫描占用缓冲区的 75%，那么 bp_fract 设置为 0.75。

large_queries

是要求您所要支持的预先读取的并发查询数。

2.1.135 REMOTE_SERVER_CFG 配置参数

使用 REMOTE_SERVER_CFG 配置参数来指定罗列可信的远程主机的文件。

onconfig.std 值

未设置。使用系统 hosts.equiv 文件。

值

文件名。路径假定为 $GBASEDBTDIR/etc。请考虑使用下列命名惯例：

authfile.server_name

REMOTE_SERVER_CFG 配置参数指定的文件必须位于 $GBASEDBTDIR/etc 中。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

1. REMOTE_SERVER_CFG 配置参数指定的文件格式与系统 hosts.equiv 文件的格式相同。
2. 如果未设置 REMOTE_SERVER_CFG 配置参数，且您运行带有 cdr add trustedhost 参数的 SQL 管理 API task() 或 admin() 函数,则数据库服务器采取下列行动：
3. REMOTE_SERVER_CFG 配置参数设置为 authfile.DBSERVER。
4. 在 $GBASEDBTDIR/etc 中创建 authfile.DBSERVER 文件。
5. 将指定的可信的主机信息添加到 $GBASEDBTDIR/etc/authfile.DBSERVER。
6. 如果数据库服务器是高可用性集群的一部分，则将可信的主机信息传播到其他集群服务器的可信的主机文件。

注： 如果数据库服务器的 sqlhosts 文件使用 s=6 选项，则您必须还设置 S6_USE_REMOTE_SERVER_CFG 配置参数为 1 来使用 REMOTE_SERVER_CFG 配置参数指定的文件。否则，数据库服务器使用系统 hosts.equiv 文件，而不是 REMOTE_SERVER_CFG 配置参数指定的文件。

2.1.136 REMOTE_USERS_CFG 配置参数

使用 REMOTE_USERS_CFG 配置参数来指定罗列在远程主机上存在的可信的用户名的文件。

onconfig.std 值

未设置。

值

文件名。路径假定为 $GBASEDBT/etc。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

REMOTE_USERS_CFG 配置参数指定的文件必须位于 $GBASEDBTDIR/etc 中。如果设置配置参数，则使用指定的文件而不是 ~/.rhosts 文件。如果在 $GBASEDBTDIR/etc 中不存在指定的文件，则认证失败。

REMOTE_USERS_CFG 配置参数指定的文件格式与 ~/.rhosts 文件的格式相同。

对于 REMOTE_USERS_CFG 配置参数指定的文件，请考虑使用下列命名惯例：

users.server_name

2.1.137 RESIDENT 配置参数

使用 RESIDENT 配置参数来指定共享内存的常驻段和虚拟段是否保持驻留在操作系统物理内存中。

onconfig.std 值

RESIDENT 0

值

-1 - 99

0 = 关

1 = 仅锁定常驻段

-1 = 锁定所有常驻段和虚拟段

n = 锁定常驻段和接下来的 n -1 个虚拟段。例如，如果您指定值为 99，则锁定常驻段并锁定接下来的 98 个虚拟段。

有些平台有不同的值。要了解信息，请参阅您的 machine notes。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

一些系统允许您指定共享内存的常驻部分必须在所有时间都常驻内存。如果您的操作系统支持强制的常驻，则您可指定共享内存的常驻段和虚拟段不交换到磁盘。

警告： 在您决定强制驻留之前，请确保可用的物理内存量足以执行所有需要的操作系统和应用进程。如果可用的内存不足，则系统挂起可能导致需要 reboot。

在支持大型内存页的 AIX®、Solaris 或 Linux™ 系统上，DBSA 可使用操作系统命令来配置一个大型页的池。

如果您执行下列步骤，GBase 8s 可在这些大型页上存储非消息虚拟内存段：

- 通过设置 IFX_LARGE_PAGES 环境变量启用大型页大小。
- 对于您打算存储在大型页上的虚拟内存段，请设置 RESIDENT 参数来锁定物理内存中的那些段，以便不可将它们交换到磁盘。

在大型页上存储虚拟内存段可在大型内存配置中提供非常出色的性能帮助。

2.1.138 RESTARTABLE_RESTORE 配置参数

onconfig.std 值

RESTARTABLE_RESTORE ON

值

ON = 启用可重启恢复

OFF = 禁用可重启恢复

生效

编辑 onconfig 文件并重启数据库服务器之后。

如果您设置 RESTARTABLE_RESTORE 为 ON，则您使数据库服务器能够在发生故障的点上重启一个失败的物理恢复或冷逻辑恢复。要以 ON-Bar 执行可重启的恢复，请使用 gbackuprestore -RESTART 命令。

如果您打算使用可重启恢复，则请增加物理日志的大小。如果需要恢复许多日志，虽然可重启恢复降低逻辑恢复的速度，但您会节约许多时间，不必重复进行完整恢复。

重要： 如果在热逻辑恢复期间数据库服务器发生故障，则您必须重复进行完整恢复。如果数据库服务器仍在运行，则请使用 gbackuprestore -r -l 来完成恢复。

如果您在不完全相同的系统上进行冷恢复，则可指定到 chunk 的新路径名，且您可在恢复期间重命名关键 chunk 的设备。重命名和恢复操作完成之后，您必须执行 0 级归档。

数据库服务器使用下列物理恢复和逻辑恢复来恢复数据：

- **物理恢复**。数据库服务器从备份介质将数据页写到磁盘。这个行动使得存储空间与其最初被备份的点保持一致。然而，每一存储空间的备份时间通常不相同。可重启恢复可重启到存储空间的级别。当恢复失败时，如果仅恢复了存储空间的部分 chunk，则当您重启恢复时，需要再次恢复整个存储空间。
- **逻辑恢复**。数据库服务器重放介质上的逻辑日志记录来使所有存储空间为最新。在逻辑恢复结束时，所有存储空间都与同一点保持一致。

2.1.139 RESTORE_POINT_DIR 配置参数

使用 RESTORE_POINT_DIR 配置参数来更改目录的路径名，在升级到新版本服务器失败时，恢复点文件将置于这个目录。仅当启用 CONVERSION_GUARD 配置参数时，GBase 8s 将恢复点文件存储在指定目录的一个子目录中，以服务器号为子目录名称

onconfig.std 值

$GBASEDBTDIR/tmp

值

目录的完整路径名

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

您可更改目录，例如，如果您认为 $GBASEDBTDIR/tmp 没有恢复点数据所需的充足空间。如果您想要更改目录，则必须在初始化升级到一个新版本服务器之前更改。您不可在升级期间更改目录。

当升级开始时，RESTORE_POINT_DIR 配置参数中指定的目录必须为空。如果该目录包含任何以前升级的恢复点文件，则必须在新升级开始新的恢复点之前移除这些文件。

重要：

空目录是进行升级的先决条件，从失败的升级恢复时则不是。失败的升级之后，在您尝试运行 onrestorept 实用程序之前不清空 RESTORE_POINT_DIR 目录。

2.1.140 ROOTNAME 配置参数

ROOTNAME 为这个数据库服务器配置指定一个 root dbspace 的名称。

在数据库服务器管理的所有 dbspace 中，此名称必须是唯一的。推荐您选择一个易于识别为 root dbspace 的名称。

onconfig.std 值

ROOTNAME rootdbs

值

最多 128 字节。ROOTNAME 必须以一个字母或下划线开始，且必须仅包含字母、数字、下划线或 $ 字符。

单位

一个 dbspace

生效

当初始化磁盘时（毁掉所有数据）

2.1.141 ROOTOFFSET 配置参数

ROOTOFFSET 指定对磁盘空间（文件、磁盘分区或设备）分配的偏移量，root dbspace 的初始 chunk 始于这里。

仅限于 UNIX：

在一些 UNIX™ 平台上，设置 ROOTOFFSET 为 0 是无效的。当这个参数设置不正确时，您必须重新初始化磁盘空间并重新加载数据来恢复数据库服务器的正当操作。在您配置数据库服务器之前，请经常检查 machine notes 文件，了解正当的设置信息。

onconfig.std 值

ROOTOFFSET 0

值

大于或等于 0 的任意值

单位

KB

生效

当初始化磁盘时（毁掉所有数据）

2.1.142 ROOTPATH 配置参数

使用 ROOTPATH 配置参数来指定 root dbspace 的初始 chunk 全路径名，包括设备或文件名。ROOTPATH 配置参数作为 chunk 名存储在保留页中。

onconfig.std 值

UNIX™ 上：$GBASEDBTDIR/tmp/demo_on.rootdbs

Windows™ 上：None

值

pathname

生效

当初始化磁盘时（毁掉所有数据）

参考

GBase 8s 管理员指南 中关于管理磁盘空间的章节中的下列资料：

- 分配磁盘空间
- 创建裸设备的链接

用法

在 UNIX 上，您必须设置用 ROOTPATH 配置参数指定的文件的许可为 660，所有者和组都必须是 gbasedbt。在 Windows 上，Gbasedbt-Admin 组的成员必须拥有您以 ROOTPATH 配置参数指定的文件。

仅限于 UNIX：

对于 UNIX 上的初始 chunk，如果您使用无缓冲的磁盘空间，则应定义 ROOTPATH 配置参数为 pathname，那是一个到 root dbspace 的初始 chunk 的链接，而不是输入初始 chunk 的实际设备名。

2.1.143 ROOTSIZE 配置参数

使用 ROOTSIZE 配置参数来指定以 KB 为单位的 root dbspace 的初始 chunk 的大小。您选择的大小依赖于数据库服务器的即时计划。

仅在完全的磁盘初始化期间,数据库服务器使用 ROOTSIZE 配置参数的值。在 root dbspace 的初始 chunk 已经创建之后，更改 ROOTSIZE 值将没有效果。

onconfig.std 值

ROOTSIZE 300000

如未出现

0

值

50,000 至存储设备的最大容量

单位

KB

生效

当初始化磁盘时（毁坏所有数据）

2.1.144 RSS_FLOW_CONTROL 配置参数

在一个包含至少一台远程独立辅助服务器的高可用性集群中，指定发生流量控制的时刻。

onconfig.std 值

RSS_FLOW_CONTROL 0

值

0 = 当前日志位置与最近响应的日志的差异超过日志缓冲区大小的 12 倍时，激活流量控制。

-1 = 禁用流量控制。禁用流量控制可能导致日志文件交换和数据丢失。

start_value,end_value = start_value 和 end_value 确定在当前日志位置与最后响应的日志页之间滞后的量。start_value 大于 end_value。这些值必须包括下列单位之一：

- K（KB）
- M（MB）
- G（GB）

例如，设置 RSS_FLOW_CONTROL 128M,100M，当日志之间的滞后超过 128 MB 时启动流量控制，当滞后降到 100 MB 时停止流量控制。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

流量控制提供一种在主服务器上限制日志活动的方法，以便集群中的 RS 辅助服务器处理事务不落后太远。如果服务器在忙碌的或间歇的网络上，则启用流量控制确保 RS 辅助服务器上的日志保持现状。当启用流量控制，且当前日志位置与最后响应的日志页之间日志大小的差异超过 start_value 时，则主服务器上的日志活动受到限制。当启用流量控制时，连接到主服务器的用户会感到响应时间变慢。当日志之间的滞后大于 start_value 时启动流量控制，当日志滞后已降低到 stop_value 时停止流量控制。

请仅在主服务器上设置 RSS_FLOW_CONTROL 配置参数。集群中的所有 RS 辅助服务器都受 RSS_FLOW_CONTROL 配置参数影响。日志通常按照接收的顺序发送到 RS 辅助服务器。

要检查 RS 辅助服务器的流量控制是否是活动的，请使用 gstat -g rss verbose 命令，并比较 RSS flow control 值与 Approximate Log Page Backlog 值。如果 Approximate Log Page Backlog 高于 RSS flow control 的第一个值，则流量控制是活动的。如果 Approximate Log Page Backlog 低于 RSS flow control 的第二个值，则流量控制被禁用。

2.1.145 RTO_SERVER_RESTART 配置参数

使用 RTO_SERVER_RESTART 配置参数来指定以秒计时间量的恢复时间目标（RTO）标准，您重启服务器并使其进入 online 或 quiescent 模式之后，GBase 8s 必须从问题中恢复。

onconfig.std 值

RTO_SERVER_RESTART 0（禁用）

值域

0 = 禁用

60 - 1800

单位

秒

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

2.1.146 S6_USE_REMOTE_SERVER_CFG 配置参数

使用 S6_USE_REMOTE_SERVER_CFG 配置参数来控制由 REMOTE_SERVER_CFG 配置参数指定的文件是否用于认证服务器集群和 Enterprise Replication 的安全连接。

onconfig.std 值

S6_USE_REMOTE_SERVER_CFG 0

缺省值

0

值

0 = 使用系统 hosts.equiv 文件来认证通过一个安全端口的服务器连接。

1 = 使用由 REMOTE_SERVER_CFG 配置参数指定的文件来认证通过一个安全端口的服务器连接。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

使用 REMOTE_SERVER_CFG 配置参数来指定一个文件，这个文件罗列被装载数据库服务器的那台计算机信任的远程服务器主机。如果使用 sqlhosts 文件连接安全选项 s=6 配置罗列的服务器中的一台或多台，则您必须设置 S6_USE_REMOTE_SERVER_CFG 配置参数为 1。

如果未设置 S6_USE_REMOTE_SERVER_CFG 或设置为 0， 则使用系统 hosts.equiv 文件，而不是 REMOTE_SERVER_CFG 配置参数指定的文件，来认证通过一个安全端口的服务器连接。

2.1.147 SB_CHECK_FOR_TEMP 配置参数

使用 SB_CHECK_FOR_TEMP 配置参数来防止将临时智能大对象复制到永久表内。

onconfig.std 值

未设置。

如未出现

准许将临时智能大对象复制到永久表内。

值

0 = 准许将临时智能大对象复制到永久表内。等同于在 onconfig 文件中未设置该配置参数。

1 = 防止将临时智能大对象复制到永久表内。数据库服务器返回下列错误消息，而不是复制临时智能大对象的句柄：

- -9810: Smart-large-object error.
- -12246: Smart large objects: You cannot put a temporary smart large object into a permanent table

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

缺省情况下，您可将临时智能大对象复制到永久表内。智能大对象数据类型，BLOB 和 CLOB，包含两部分：数据，存储在 sbspace 中，以及句柄，存储在表中。当您将临时智能大对象复制到永久表内时，仅将 BLOB 或 CLOB 句柄复制到永久表内。如果您接下来删除该临时智能大对象，则永久表包含一个不再有效的句柄。

要防止将临时智能大对象复制到永久表内，请在 onconfig 文件中设置 SB_CHECK_FOR_TEMP 配置参数为 1。例如，如果 SB_CHECK_FOR_TEMP 配置参数设置为 1，则将临时智能大对象复制到永久表内的 INSERT INTO . . . SELECT FROM . . . 语句失败。

2.1.148 SBSPACENAME 配置参数

使用 SBSPACENAME 配置参数指定缺省 sbspace 的名称。

onconfig.std 值

未设置。

如未出现

0

值

最多 128 字节。

SBSPACENAME 必须是唯一的，以一个字母或下划线开头，且仅包含字母、数字、下划线或 $ 字符。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

如果数据库表包含智能大对象列，未明确地指定其存储空间，则那个数据保存在 SBSPACENAME 指定的 sbspace 中。

内建的加密和解密函数也使用缺省 sbspace 来存储 BLOB 或 CLOB 值。如果 DECRYPT_BINARY 或加密函数找不到存储 BLOB 或 CLOB 参数或者返回值的 sbspace，则函数失败并带有下列错误消息：

Fatal error in server row processing - SQL error -9810 ISAM error -12053

如果您在调用一个有 BLOB 或 CLOB 参数的加密或解密函数之后，看到这个错误消息，则请使用 SBSPACENAME 配置参数配置缺省 sbspace，然后重复函数调用。

在使用之前，您必须用 gspaces -c -S 实用程序创建缺省 sbspace。当发生下列情况之一时，数据库服务器验证缺省 sbspace 的名称：

- 在 CREATE TABLE 或 ALTER TABLE 语句的 PUT 子句中为 CLOB 或 BLOB 列指定缺省 sbspace 作为存储选项。
- 当未为该列指定 sbspace 时，数据库服务器尝试将智能大对象写到缺省 sbspace。
- 在缺省 sbspace 中存储多重表示数据。

JAVA 语言支持：

如果您正在使用 J/Foundation，则必须在数据库服务器可存储 Java™ 归档（JAR）文件的地方提供智能大对象。这些 JAR 文件包含 Java 用户定义的例程（UDR）。当您使用 Java UDR 时，建议为了保存智能大对象创建单独的 sbspace。

警告： 当您使用 Enterprise Replication 时，在定义复制服务器之前，必须设置 CDR_QDATA_SBSPACE 配置参数并创建 sbspace。

缺省 sbspace 的自动创建

如果您创建一个 bts 索引且未明确地指定 sbspace 名称，则即使未设置 SBSPACENAME 参数，也会创建缺省 sbspace。

为数据库服务器在 root dbspace 中创建大小 10 000 KB 的缺省 sbspace。当缺省 sbspace 填满时，您必须手工地增加其大小。

2.1.149 SBSPACETEMP 配置参数

使用 SBSPACETEMP 配置参数来指定缺省临时 sbspace 的列表，用于存储没有元数据或用户数据日志记录的临时智能大对象。如果您在标准 sbspace 中存储临时智能大对象，则日志记录元数据。

onconfig.std 值

未设置。临时智能大对象存储在缺省 sbspace 中，用 SBSPACENAME 配置参数指定该 sbspace。

分隔符

逗号

值

一个或多个 sbspace 名称。以逗号分隔名称。列表长度不可超过 128 字节。

每一 sbspace 名称必须是唯一的，以一个字母或下划线开头，并仅包含字母、数字、下划线或 $ 字符。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

2.1.150 SDS_ALTERNATE 配置参数

在高可用性集群中，使用 SDS_ALTERNATE 配置参数来定义在主服务器与 SD 辅助服务器之间通信的替代方式。

onconfig.std 值

NONE（未配置 SD 辅助服务器替代通信路径。）

值

在主服务器与 SD 辅助服务器之间，用作替代通信路径的 blobspace 名称。

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

如果在主服务器与 SD 辅助服务器之间的网络不可用，则您设置 SDS_ALTERNATE 配置参数并创建共享 blobspace 来允许高可用性集群中的主服务器和所有 SD 辅助服务器使用替代通信路径。 当 SD 服务器即将故障转移并成为主服务器，但 TCP/IP 通信不可用时，使用 SDS_ALTERNATE 配置参数设置的共享 blobspace 将关闭程序传递到原先的主服务器。

在主服务器和所有 SD 辅助服务器上，设置 SDS_ALTERNATE 配置参数为相同的值。

设置 SDS_ALTERNATE 配置参数之前，您必须在主服务器上创建共享 blobspace。例如，在主服务器上输入下列命令来创建名为 sds_alt_comm 的 blobspace：

gspaces -c -b sds_alt_comm -g <pagesize> -p <path> -o <offset> -s <size>

请运行下列命令来切换到下一个逻辑日志文件，以便启用新创建的 blobspace：

gadmin -l

在高可用性集群中的每一 SD 辅助服务器上，设置 SDS_ALTERNATE 配置参数来指向主服务器上的 blobspace。

SDS_ALTERNATE sds_alt_comm

2.1.151 SDS_ENABLE 配置参数

使用 SDS_ENABLE 配置参数来启用 SD 辅助服务器功能。

onconfig.std 值

未设置。

如未出现

0

值

0 = 禁用

1 = 启用

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

您必须在 SD 辅助服务器上设置 SDS_ENABLE 为 1（启动）来启用 SD 辅助服务器功能。

当您运行下列命令时，自动地设置 SDS_ENABLE 为 1（启用）：

gadmin -d set SDS primary

当您运行下列命令时，SDS_ENABLE 设置为 0（禁用）：

gadmin -d clear SDS primary

要防止数据终端，如果 SDS_ENABLE 设置为 1（启用），则您不可使用 oninit -i 或 oninit -iy 命令来初始化服务器上的磁盘空间。 要初始化 SD 辅助服务器，请使用不带参数的 oninit 仅初始化共享内存。要初始化关联一台或多台 SD 辅助服务器的主服务器，且其磁盘从未初始化，则设置 SDS_ENABLE 为 0 并 使用 oninit -i 初始化服务器内存和磁盘。要初始化关联 SD 辅助服务器的主服务器，且其磁盘已初始化，则设置 SDS_ENABLE 为 1 并使用不带参数的 oninit 初始化共享内存。

2.1.152 SDS_FLOW_CONTROL 配置参数

在包含至少一台共享磁盘（SD）辅助服务器的高可用性集群中，当发生流量控制时指定。

onconfig.std 值

SDS_FLOW_CONTROL 0

值

0 = 当当前日志位置与最近响应日志之间的差异超过日志缓冲区大小的 12 倍时，激活流量控制。

-1 = 禁用流量控制。禁用流量控制可能导致日志文件的交换和数据丢失。

start_value,end_value = start_value 和 end_value 确定当前日志位置与最后响应日志页之间的滞后量。start_value 必须大于 end_value。这些值必须包括下列单位之一：

- K（KB）
- M（MB）
- G（GB）

例如，设置 SDS_FLOW_CONTROL 128M,100M，当日志之间的滞后为 128 MB 时启动流量控制，当滞后降至 100 MB 时停止流量控制。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

用法

流量控制提供一种限制主服务器上日志活动的方法，以便集群中的 SD 辅助服务器不在处理事务上落后太远。当启用流量控制时，且当当前日志位置与最后响应日志页之间日志大小的差异超过 start_value 时，主服务器上的日志活动受到限制。当流量控制是活动的时，连接到主服务器的用户可感觉到响应时间变慢。当日志之间的滞后大于 start_value 时启动流量控制，当日志滞后降至 stop_value 时停止流量控制。

仅在主服务器上设置 SDS_FLOW_CONTROL 配置参数。集群中的所有 SD 辅助服务器都受到 SDS_FLOW_CONTROL 配置参数的影响。日志通常按照被接收的顺序发送到 SD 辅助服务器。

2.1.153 SDS_LOGCHECK 配置参数

使用 SDS_LOGCHECK 配置参数来确定主服务器是否正在生成日志活动，并允许或防止主服务器的故障转移。

例如，如果 SDS_LOGCHECK 配置参数设置为 10，且主服务器故障，则 SD 辅助服务器等待最多 10 秒，或者检测主服务器正在生成日志记录（若是防止故障转移），或者 SD 辅助服务器检测主服务器没在生成日志记录并发生故障转移。

onconfig.std 值

SDS_LOGCHECK

UNIX™ 上：10

Windows™ 上：0

值

0 = 不检测日志活动；允许故障转移。

n = 等待最多 n 秒。如果从主服务器检测日志活动，则防止故障转移；否则，允许故障转移。

单位

秒

生效

当在主服务器上启用共享磁盘功能时

用法

重要： 您必须为主服务器和为所有辅助服务器指定相同的值。如果您指定的值不相同，则数据库服务器自动地更改辅助服务器上不同的值，更改为主服务器上设置的值。

SDS_LOGCHECK 配置参数指定 SD 辅助服务器等待检测主服务器是否在生成日志活动的秒数。如果在指定的时间量中主服务器未创建日志活动，则允许启动故障转移处理。如果从主服务器检测日志活动，则不启动故障转移处理。

如果您没有配置 I/O GBase 8s fencing，且您的系统包含一台主服务器和一台或多台 SD 辅助服务器，则设置 SDS_LOGCHECK 配置参数为一个大于零的值。

如果您的系统配置了 I/O GBase 8s fencing，且如果一台 SD 辅助服务器成为主服务器，则 I/O GBase 8s fencing 脚本必须防止故障的主服务器更新任何共享磁盘。 如果系统未配置 I/O GBase 8s fencing，如果原先的主服务器正在生成日志记录，则通过不故障转移到 SD 辅助服务器，SDS_LOGCHECK 配置参数防止多台主服务器的并发。

2.1.154 SDS_PAGING 配置参数

SDS_PAGING 配置参数指定作为缓冲区 GBase_8s paging 文件的两个文件的位置。

onconfig.std 值

未设置。

生效

当启动 SD 辅助服务器时

用法

设置 SDS_PAGING 配置参数为两个缓冲区 GBase_8s paging 文件的路径，以逗号分隔。SDS_PAGING 配置参数必须设置为有效值以确保 SD 辅助服务器启动。由于 GBase_8s paging 文件根据需要动态地增长，您应分配足够的磁盘空间来存储 PHYSFILE 配置参数指定值的两倍大小。

2.1.155 SDS_TEMPDBS 配置参数

使用 SDS_TEMPDBS 配置参数来指定共享磁盘（SD）辅助服务器用于动态地创建临时 dbspace 的信息。只可在 SD 辅助服务器上指定这个配置参数。

onconfig.std 值

未设置。不创建共享磁盘辅助服务器的临时 dbspace。

值

以逗号分隔，包含按照下列顺序的下列值的字符串：

dbspace = 要创建的 dbspace 的名称。在所有现有的 dbspace、blobspace 和 sbspace 中必须是唯一的，包括从主服务器继承的那些任何临时空间。名称不可超过 128 字节，必须以一个字母或下划线开头，且必须仅包含字母、数字、下划线或 $ 字符。

dbpath = dbspace 的路径，或者是一个完全路径名，或者是一个相对路径名。如果您使用相对路径名，必须是相对于您初始化数据库服务器时是当前目录的那个目录。

pagesize = 以 KB 为单位的表示页大小的整数。页大小必须在 2 KB 和 16 KB 之间，且必须是缺省页大小的倍数。

offset = 一个等于或大于 0 的整数，指定在磁盘分区内或设备内达到 dbspace 的初始 chunk 的偏移量。起始偏移量加上 chunk 大小不可超过 chunk 大小的最大值。偏移量必须是页大小的倍数。最大偏移量是 2 TB 或 4 TB，这依赖于平台。缺省情况下，该值以 KB

计。您可通过给该值附加单个字符修饰符来标明不同的单位：M 或 m 表示 MB，G 或 g 表示 GB，T 或 t 表示 TB。

size = 等于或大于 1000 KB 的正整数和页大小的倍数指定 dbspace 的初始 chunk 的大小。offset 的值加上 size 的值不可超过 chunk 大小的最大值。chunk 大小的最大值等于 2 147 483 647 页乘以页大小。缺省情况下，以 KB 为单位这个值。您可通过给该值附加单个字符修饰符来标明不同的单位:M 或 m 表示 MB,G 或 g 表示 GB,T 或 t 表示 TB。

分隔符

以逗号分隔每一值。不使用空格。

生效

编辑 onconfig 文件并重启 SD 辅助服务器之后。

用法

当 SD 辅助服务器启动时，如果以前存在 dbsapce，则创建或初始化临时 dbspace。临时 dbspace 用于创建临时表。为了 SD 辅助服务器启动，在 SD 辅助服务器的 onconfig 文件中必须有至少一个 SDS_TEMPDBS 配置参数。通过使用多个 SDS_TEMPDBS 配置参数，您可在 onconfig 文件中指定最多 16 个 SD 辅助临时 dbspace。

对于 onconfig 文件中的每一 SDS_TEMPDBS 配置参数：

- dbsname 值必须是每一服务器唯一的，且不与其他 SD 辅助服务器或主服务器分享。
- dbspath、pagesize、offset 与 size 的组合必须不导致与现有的 chunk 重叠，或与 SDS_TEMPDBS 配置参数指定的临时 dbspace 空间之间重叠。
- 对每一 SDS_TEMPDBS 配置参数值，pagesize 值必须是相同的。

下列示例展示 SDS_TEMPDBS 配置参数的两个条目：

```
SDS_TEMPDBS sds_space1,/dev/raw_dev1,2,0,60M
SDS_TEMPDBS sds_space2,/dev/raw_dev2,2,0,80M
```

如果高可用性集群中的主服务器故障，且一台 SD 辅助服务器接替作为主服务器，则在 SD 辅助服务器上设置的值用于临时 dbspace，直到服务器重启。您应确保在 SD 辅助服务器上指定的 SDS_TEMPDBS 配置参数值不同于在主服务器上指定的值。SD 辅助服务器重启之后，使用 DBSPACETEMP 配置参数。

2.1.156 SDS_TIMEOUT 配置参数

使用 SDS_TIMEOUT 配置参数来指定以秒计的时间量，在这段时间里，高可用性集群中主服务器将等待从共享磁盘（SD）辅助服务器发送日志位置响应。

onconfig.std 值

SDS_TIMEOUT 20

如未出现

10

值

2 - 2147483647

单位

秒

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置 SDS_TIMEOUT 值。

当您通过运行 gadmin -wm 命令重置内存中的 SDS_TIMEOUT 值。

用法

在指定的时间量内，如果从 SD 辅助服务器未收到日志位置响应，则主服务器将从 SD 辅助服务器断开连接并继续。等待 SDS_TIMEOUT 配置参数设置中指定的秒数之后，如果在等待 SD 辅助服务器时页清空已超时，则主服务器将启动移除 SD 辅助服务器。

2.1.157 SECURITY_LOCALCONNECTION 配置参数

使用 SECURITY_LOCALCONNECTION 配置参数，通过验证正在运行程序的本地用户的 ID 与正在试图访问数据库的用户的 ID 是否相同，来验证本地连接的安全性。

onconfig.std 值

未设置。

值

0 = 不发生安全性检查。

1 = GBase 8s 检查正在运行程序的用户的 ID 与正在试图连接到数据库的用户的 ID 是否相配。

2 = 与 1 相同，加之 GBase 8s 从网络 API 检索同级端口号并验证该连接是否正来自客户端程序。 如果您的系统有 SOCTCP 或 IPCSTR 网络协议，则可仅指定 2。

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.158 SEQ_CACHE_SIZE 配置参数

使用 SEQ_CACHE_SIZE 配置参数来指定在序列高速缓存中可有预先分配值的序列对象的最大数目。

onconfig.std 值

SEQ_CACHE_SIZE 10

值

1 - 2147483647

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

如果未设置 SEQ_CACHE_SIZE 配置参数，则在缺省情况下，您不可用 CREATE SEQUENCE 语句的 CACHE 选项定义 10 个以上不同的序列对象。

例如，如果您想增加可有提前分配值的序列对象的最大数目为 15，则请指定：

SEQ_CACHE_SIZE 15

2.1.159 SERVERNUM 配置参数

SERVERNUM 配置参数指定共享内存中的相对位置。

onconfig.std 值

SERVERNUM 0

值

0 - 255

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

在本地计算机上，您为每一数据库服务器选择的值必须是唯一的。在您的网络上，该值不需要是唯一的。因为在 onconfig.std 文件中包括值 0，所以建议您选择一个非 0 的值来避免无意中重复 SERVERNUM 配置参数。

2.1.160 SESSION_LIMIT_LOCKS 配置参数

SESSION_LIMIT_LOCKS 配置参数指定非管理员用户在一个会话中可用锁的最大数目。

onconfig.std 值

无

如未出现

2147483647

值

500 至 2147483647

单位

在内部锁表中锁的数目

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

对于非管理员用户，在一个会话中锁的最大数目可以是从 500 至 2147483647 范围内任何指定的值。

如果指定的值在 1 至 500 的范围内，则对于非管理员用户有效的最大数目是 500。

如果指定的值为 0 或负数，则缺省值 2147483647 对所有用户都有效。

如果使用 gadmin -wm 或 gadmin -wf 来设置这个参数，则对于无效值显示一个适当的消息。例如，

```
% gadmin -wm SESSION_LIMIT_LOCKS=200

New value is invalid. Suggested value: (500).
```

SESSION_LIMIT_LOCKS 设置不可限制拥有管理权限的用户在会话中允许的锁数目，诸如用户 gbasedbt 或 DBSA 用户，缺省情况下，允许其每会话的最大值总是 2147483647 个锁。对于管理员的大的最大锁限制，通常不必设置这个参数，或设置相应的 SQL 的 SET ENVIRONMENT 语句的 IFX_SESSION_LIMIT_LOCKS 会话环境选项。

然而，在要求非常大量锁的数据处理上下文中，管理员可能设置 SESSION_LIMIT_LOCKS 为一个值，以图降低普通用户在并发会话中用尽数据库服务器锁资源的风险，由此妨碍大量密集锁操作。

重要：

在可重复的读隔离级别中，因为要求锁的活动集中的每一行，请小心设置服务器上锁的限制过低。类似地，设置锁限制过小可妨碍 Enterprise Replication 任务，或妨碍非 DBSA 用户发出的 cdr 命令。

2.1.161 SHMADD 配置参数

使用 SHMADD 配置参数来指定那些动态地添加到共享内存虚拟部分的段的大小。

onconfig.std 值

与平台有关

值

32 位平台：1024 - 524288

64 位平台：1024 - 4294967296

单位

KB

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

当需要添加内存时，SHMADD 配置参数的值表示数据库服务器添加到共享内存的虚拟部分的第一批段的大小。通过 SHMVIRTSIZE 配置参数设置第一个虚拟共享内存段。请设置 SHMVIRTSIZE 和 SHMADD 配置参数的值，以便在数据库服务器正常操作期间添加最少量的段。通常，较多的段会降低性能。

GBase 8s 共享内存段的最大数目是 1024。如果 SHMADD 值低，或者数据库服务器意外地有大量活动或内存使用，则可能要求许多共享内存段。要防止数据库服务器达到共享内存段的最大数目，数据库动态地添加的虚拟段的大小应两倍于每个 16 虚拟段。在大型段中添加内存更为有效，但如果添加的内存未被使用，则造成浪费。而且，操作系统可能要求您在少数大型段中，而不是许多小段中添加内存。

下列表包含设置 SHMADD 初始值的建议。

表. SHMADD 推荐值

物理内存数量	SHMADD 推荐值
小于 256 MB	8192
256 - 512 MB	16,384
大于 512 MB	32,768

默认情况下,GBase 8s 在安装时会自动根据当前内存大小设置 SHMADD 值。计算算法如下：

SHMADD = *memorybase* * 51200+25600

其中 *memorybase* =总内存大小/950000。如果计算结果 < 1，则 *memorybase* = 1

如果 SHMADD 值的计算结果大于 1024000 则，将其取值为 1024000。

通过运行 gstat -g seg 命令，您可查看关于虚拟内存段的信息。

2.1.162 SHMBASE 配置参数

使用 SHMBASE 配置参数来指定共享内存附加到虚拟处理器的内存空间的基本地址。

onconfig.std 值

与平台有关

值

正整数

单位

地址

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

共享内存段的地址起始于 SHMBASE 值并增长直到达到上限，这与平台相关。

请不更改 SHMBASE 的值。SHMBASE 的 onconfig.std 值依赖于平台，以及处理器是 32 位还是 64 位。要了解关于使用 SHMBASE 值的信息，请参阅 machine notes。

2.1.163 SHMNOACCESS 配置参数

SHMNOACCESS 配置参数指定一个不用于附加共享内存的虚拟内存地址范围。

onconfig.std 值

On UNIX™: None

On Windows™: #SHMNOACCESS 0x70000000-0x7FFFFFFF, and this value is commented out in the onconfig.std template file.

UNIX 上：无

Windows 上：#SHMNOACCESS 0x70000000-0x7FFFFFFF，且在 onconfig.std 模板文件中注释掉这个值。

值

1 - 10 地址范围

分隔符

逗号

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

SHMNOACCESS 配置参数用于避开特定的范围进程地址，反过来避免与操作系统库冲突。

每一范围中的每一地址必须以十六机制格式起始。范围中的每一地址必须以连字符分隔，且每一范围必须以逗号分隔，如下例所示：

```
SHMNOACCESS 0x70000000-0x75000000,0x7A000000-0x80000000
```

2.1.164 SHMTOTAL 配置参数

使用 SHMTOTAL 配置参数来指定数据库服务器用于所有内存分配的共享内存的总量（常驻部分、虚拟部分、通信部分和虚拟扩展部分）。onconfig.std 值 0 暗指约定对内存分配不设限。

onconfig.std 值

SHMTOTAL 0

值

0 = （无特定限制）或者大于或等于 1 的任何整数

单位

KB

生效

编辑 onconfig 文件并重启数据库服务器。

用法

您可使用 SHMTOTAL 配置参数来限制对数据库服务器可在系统上放置的内存的需求。然而，如果数据库服务器需要的内存多于 SHMTOTAL 强制的限度，则应用可能失败。 当发生这种情况时，数据库服务器将下列消息写到消息日志中：

```
size of resident + virtual segments xx + yy > zz total allowed by configuration parameter SHMTOTAL
```

这个消息包括下列值。

值

描述

xx

常驻段的当前大小

yy

虚拟段的当前大小

zz

需要的共享内存合计

如果您启用 LOW_MEMORY_MGR 配置参数，且正在配置服务器来使用 SHMTOTAL 配置参数值的百分率作为自动低内存管理起始和终止阈值，则 SHMTOTAL 配置参数必须不设置为 0（不限制）。

注意： 更改 SHMTOTAL 配置参数值可导致自动低内存管理的配置成为无效，强制数据库服务器使用缺省设置。

仅限于 UNIX：

对于最大共享内存段大小，通常是 SHMMAX、SHMSIZE 或 SHMALL，请设置操作系统参数为数据库服务器配置需要大小的合计。要了解关于操作系统允许的共享内存量的信息，请参阅 machine notes。

如果您有比 machine notes 中指定的值更多的物理内存，且将由 GBase 8s 使用该内存，则您可增加 SHMALL 参数的值，直到计算机指定的物理内存的 90%。建议您不要达到或超过可用的 RAM。

2.1.165 SHMVIRT_ALLOCSEG 配置参数

使用 SHMVIRT_ALLOCSEG 配置参数来指定 GBase 8s 应分配新的共享内存段的阈值，以及如果服务器不能分配新内存段时激活的事件报警级别。

onconfig.std 值

SHMVIRT_ALLOCSEG 0,3

值

一个数值，可选地后跟一个逗号和另一个数值。

threshold = 当数据库服务器应添加一个共享内存段时指出的数值。：

- 0 = 缺省。当需要时数据库服务器分配的共享内存段。
- .40 - .99 = 添加一个段之前使用的内存百分比。
- 256 - 10000000 = 添加一个段之前保留的 KB 数。

alarm_level：可选的。一个从 1 至 5 的整数值，指定发出的事件报警级别：1 = 不值得注意，2 = 信息，3 = 关注（缺省），4 = 紧急，5 = 致命。事件报警的类 ID 是 24，事件 ID 是 24003。

分隔符

以逗号分隔这些值。

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

设置 SHMVIRT_ALLOGSEG 配置参数来主动地添加共享内存段，而不是等到数据库服务器自动地添加共享内存段。

如果不能分配新的内存段，则事件报警每三十分钟重复一次。

2.1.166 SHMVIRTSIZE 配置参数

使用 SHMVIRTSIZE 配置参数来指定虚拟共享内存段的初始大小。

onconfig.std 值

与平台有关

如未出现

如果 SHMADD 出现：SHMADD 配置参数的值。

如果 SHMADD 未出现：8192。

值

32 位平台：正整数，最大值 2 GB

64 位平台：正整数，最大值 4 TB

由于操作系统限制，在一些平台上最大值可能较小。 要查看您的 UNIX™ 平台的实际最大值，请参阅 machine notes。

单位

KB

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

要确定 SHMVIRTSIZE 配置参数适当的值，应使用下列算法来确定共享内存的虚拟部分大小：

```
shmvirtsize = fixed overhead + ((stack size + heap) * number of users)
```

变量	使用的值
fixed overhead	这包括 AIO 向量的大小、排序内存、dbspace 备份缓冲区、字典大小、存储过程高速缓存的大小、直方图池、其他池和其他开销。 要获得固定开销的估计，请启动数据库服务器，如果有的话，还要看分配了多少附加的内存段。当您启动服务器时，您在系统上拥有的用户数影响内存段的分配。当您启动服务器时： ● 如果用户数是您环境的典型用户数，那么将内存段的大小添加到 SHMVIRTSIZE 配置参数的当前值并重启服务器。 ● 如果用户数远少于您环境的典型用户数，则必须计算要使用的内存段的适当开销值。当您启动服务器时，将分配的附加内存段数除以那时服务器上的用户数，您可计算每个用户消耗多少内存段。将每一用户内存段的值乘以系统上典型的用户数。将这个计算值的内存段添加到 SHMVIRTSIZE 配置参数的当前值并重启服务器。
stack size	在 32 位系统上，使用 32 KB 堆栈大小。在 64 位系统上，您典型地使用 64 KB 堆栈大小。然而，一些 64 位系统使用不同的值。
heap	每用户使用 30 KB。
用户数	使用服务器上您预计的并发用户会话的最大数目。

如果可能，创建一个共享内存的虚拟部分，其大小大于您每日处理的大小。

默认情况下，GBase 8s 在安装时会自动根据当前内存大小设置 SHMVIRTSIZE 值。计算算法如下：

SHMVIRTSIZE = *memorybase* * 51200+25600

其中 *memorybase* =总内存大小/950000。如果计算结果 < 1，则 *memorybase* = 1

如果 SHMVIRTSIZE 值的计算结果大于 4096000 则，将其取值为 4096000。

使用 gstat -g seg 命令来确定峰值用量并相应地降低 SHMVIRTSIZE 配置参数的值。

2.1.167 SINGLE_CPU_VP 配置参数

SINGLE_CPU_VP 配置参数指定数据库服务器是否正在运行仅一个 CPU 虚拟处理器。

onconfig.std 值

SINGLE_CPU_VP 0

值

0 = 运行多个 CPU VP

1 = 运行一个 CPU VP

生效

当数据库服务器关闭并重启时

用法

如果您想在数据库服务器启动时自动地增加 CPU VP 的数目，请通过将 SINGLE_CPU_VP 配置参数设置为 0 来禁用它。

设置 SINGLE_CPU_VP 为非零，允许数据库服务器根据仅一个 CPU 虚拟处理器在运行的情况，使用优化的代码。这使得数据库服务器能够绕过许多必须在运行多个 CPU 虚拟处理器时才使用的互斥调用。

当数据库服务器仅运行一个 CPU 虚拟处理器时，强烈地建议您设置这个参数。依赖于应用和工作负载，设置这个参数最大可提高性能 10%。

如果设置 SINGLE_CPU_VP 为非零，并试图添加一个 CPU 虚拟处理器，则您收到下列消息之一：

```
gadmin: failed when trying to change the number of classname VPs by n.
gadmin: failed when trying to change the number of cpu virtual processors by n.
```

如果您设置 SINGLE_CPU_VP 为非零，然后尝试在 VPCLASS **cpu,** *num* 设置为一个大于 1 的值的情况下启动数据库服务器，则会收到下列错误消息，且数据库服务器初始化失败：

```
Cannot have SINGLE_CPU_VP non-zero and CPU VPs greater than 1.
```

VPCLASS 值和 SINGLE_CPU_VP 配置参数

GBase 8s 将用户定义的虚拟处理器类视作 CPU 虚拟处理器。如果您设置 SINGLE_CPU_VP 配置参数为非零值，则不可创建任何用户定义的虚拟处理器类。

使用用户定义的 VPCLASS

如果您设置这个配置参数为非零值，然后尝试以一个用户定义的 VPCLASS 启动数据库服务器，则会收到下列错误消息，且数据库服务器初始化失败：

```
oninit: Cannot have SINGLE_CPU_VP non-zero and user-defined VP classes
```

使用 cpu VPCLASS

如果您设置这个配置参数为非零值，然后尝试以 num 设置为大于 1 的值的 VPCLASS cpu 值来启动数据库服务器，则会收到下列错误消息，且数据库服务器初始化失败：

```
Cannot have SINGLE_CPU_VP non-zero and CPU VPs greater than 1.
```

2.1.168 SMX_COMPRESS 配置参数

在从源数据库服务器向目标数据库服务器发送数据之前，使用 SMX_COMPRESS 配置参数来指定数据库服务器使用的压缩级别。

在缓慢的连接之上，网络压缩节约网络带宽，但使用更多的 CPU 来压缩和解压缩数据。比较两服务器的 SMX_COMPRESS 配置参数值并更改到较高的压缩值。

onconfig.std 值

SMX_COMPRESS 0

值

-1 = 源数据库服务器从不压缩数据，不管目标站点是否使用压缩。

0 = 仅当目标数据库服务器期待压缩的数据时，源数据库服务器才压缩数据。

1 = 数据库服务器执行最小量的压缩。

9 = 数据库服务器执行最大可能的压缩。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

2.1.169 SMX_NUMPIPES 配置参数

此参数定义由远程独立辅助服务器（RSS）使用的网络 pipe 数。使用此值来为服务器多路复用器组（SMX）设置网络线程处理的级别，可提升较慢网络连接上的传输速率。

onconfig.std 值

SMX_NUMPIPES 1

值

大于等于 1 的网络连接数。缺省值为 1。

单位

在一给定的 RSS 服务器与其主服务器之间的网络连接数目

生效

需要在主节点和 RSS 节点的 onconfig 文件里都设置该参数，才能保证生效。

2.1.170 SMX_PING_INTERVAL 配置参数

使用 SMX_PING_INTERVAL 配置参数来指定超时间隔中的秒数，在服务器多路复用器组（SMX）连接中，辅助服务器在间隔内等待来自主服务器的活动。

onconfig.std 值

SMX_PING_INTERVAL 10

值

0 = 不确定地等待。

1 与 60 之间且包括 1 和 60 的正整数。= 超时间隔中的秒数。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

运行带有 "gadmin","-wf SMX_PING_INTERVAL=value" 或 "gadmin","-wm SMX_PING_INTERVAL=value" 参数的 SQL 管理 API task() 或 admin() 函数之后。

用法

如果在 SMX_PING_INTERVAL 配置参数指定的时间长度期间，以及 SMX_PING_RETRY 配置参数指定的间隔数之后，辅助服务器未收到任何消息，则辅助服务器打印错误消息到 online.log 并关闭 SMX 连接。如果 SMX 超时消息在 online.log 中，则您可增加 SMX_PING_INTERVAL 值、SMX_PING_RETRY 值，或者增加二者的值。

这个配置参数仅应用于辅助服务器。如果您在主服务器上设置 SMX_PING_INTERVAL，则在主服务器成为辅助服务器时该参数生效。

如果高可用性集群中的辅助服务器的 onconfig 文件有下列条目，则辅助服务器总共等待来自主服务器的活动 180 秒。如果在那 180 秒期间没有来自主服务器的活动，则辅助服务器关闭 SMX 连接并将错误消息写到 online 日志。

SMX_PING_INTERVAL 30 SMX_PING_RETRY 6

2.1.171 SMX_PING_RETRY 配置参数

如果没有收到来自主服务器的响应，则使用 SMX_PING_RETRY 配置参数来指定辅助服务器重复 SMX_PING_INTERVAL 配置参数指定的超时间隔的最大次数。如果达到最大数目而没有响应，则辅助服务器在 online.log 中打印错误消息并关闭服务器多路复用器组（SMX）连接。

onconfig.std 值

SMX_PING_RETRY 6

值

任何正整数 = 重复超时间隔的最大次数。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

运行带有 "gadmin","-wf SMX_PING_RETRY=value" 或 "gadmin","-wm SMX_PING_RETRY=value" 参数的 SQL 管理 API task() 或 admin() 函数之后。

用法

如果在 SMX_PING_INTERVAL 配置参数指定的时间长度期间，以及 SMX_PING_RETRY 配置参数指定的间隔数之后，辅助服务器未收到任何消息，则辅助服务器打印错误消息到 online.log 并关闭 SMX 连接。如果 SMX 超时消息在 online.log 中，则您可增加 SMX_PING_INTERVAL 值、SMX_PING_RETRY 值，或者增加二者的值。

这个配置参数仅应用于辅助服务器。如果您在主服务器上设置 SMX_PING_RETRY，则在主服务器成为辅助服务器时该参数生效。

如果高可用性集群中的辅助服务器的 onconfig 文件有下列条目，则辅助服务器总共等待来自主服务器的活动 60 秒。如果在那 60 秒期间没有来自主服务器的活动，则辅助服务器关闭 SMX 连接并将错误消息写到 online 日志。

```
SMX_PING_INTERVAL 12
SMX_PING_RETRY 5
```

2.1.172 SP_AUTOEXPAND 配置参数

使用 SP_AUTOEXPAND 配置参数来启用或禁用 chunk 的自动创建或扩展。

onconfig.std 值

SP_AUTOEXPAND 1

值

0 = 禁用 chunk 的自动创建或扩展。

1 = 启用 chunk 的自动创建或扩展。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

当启用 SP_AUTOEXPAND 配置参数，且一个诸如 dbspace 这样的存储容器有一个定义的非零的建大小或扩展大小时，容器是可自动扩展的。

2.1.173 SP_THRESHOLD 配置参数

GBase 8s 自动地运行任务来扩展空间之前，使用 SP_THRESHOLD 配置参数来定义可在存储空间中存在的空闲最小 KB 数量。或者通过扩展空间中现有的 chunk 或者通过添加新 chunk。

onconfig.std 值

SP_THRESHOLD 0

值

0 = 无阈值。当空间低于禁用的阈值时，运行存储空间监控（mon_low_storage）任务的触发器添加空间。

1 - 50 = 存储空间中空闲 KB 的百分率阈值。

如果该值是 50 或以下，则 GBase 8s 将该值解释为一个百分率（例如，10 = 10%，2.84 = 2.84%）。

1000 至 chunk 大小的最大值 = 阈值或者是 1000 KB，或者是当前平台上 chunk 大小的最大值。

如果该值是 1000 或更高，则 GBase 8s 将该值解释为一个特定的 KB 数。

值 50 - 1000 不是有效的。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

当您设置 SP_THRESHOLD 配置参数为一个大于 0 的有效值时，当 dbspace、临时 dbspace、sbspace、临时 sbspace 或 blobspace中的空闲空间低于该阈值时，内建的 Scheduler 任务，mon_low_storage，自动地运行。

假定 SP_THRESHOLD 配置参数的值是 5.5，服务器解释为 5.5%。如果空间在低空闲页上运行，且空闲空间百分率低于 5.5%，且保持低于那个级别直到 mon_low_storage 任务下一次运行，则那个任务将尝试扩展该空间。如果 SP_THRESHOLD 配置参数设置为 50000 且空间小于 50000 空闲 KB，则下一次 mon_low_storage 任务运行时将扩展那个空间。

值 0 关闭 mon_low_storage 任务，并防止服务器扩展任何空间。然而，当所有空闲页用尽并需要更多时，值 0 不影响服务器扩展空间的能力。

SP_THRESHOLD 配置参数中指定的值适用于属于服务器的所有空间。

2.1.174 SP_WAITTIME 配置参数

在返回一个空间用尽错误之前，使用 SP_WAITTIME 配置参数来指定线程等待 dbspace、临时 dbspace、plogspace、sbspace、临时 sbspace 或 blobspace 空间扩展的最大秒数。

onconfig.std 值

SP_WAITTIME 30

值

0 - 2147483647

单位

秒

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

服务器用来自动地添加或扩展 chunk 的时间可差异很大，这依赖于不同的因素，诸如 chunk 的大小、相关磁盘驱动的速度以及系统上的负载。当 GBase 8s 自动地添加或扩展 chunk 以防止空闲空间降至 SP_THRESHOLD 配置参数指定的阈值之下时， GBase 8s 强制那些需要该空间的线程等待直到空间可用。如果您想更改线程将等待更多空间的最大秒数，则可更改 SP_WAITTIME 配置参数的值。

仅当存储池包含条目时，线程才会等待存储空间扩展。如果存储池为空，则线程将不等待。

2.1.175 SQL_LOGICAL_CHAR 配置参数

使用 SQL_LOGICAL_CHAR 配置参数来启用或禁用在内建的字符数据类型声明中大小规格的扩展。

onconfig.std 值

SQL_LOGICAL_CHAR OFF（ = 以字节为单位解释大小规格）

值

OFF = 声明的大小无扩展。

1 = 声明的大小无扩展。

2 = 使用 2 作为声明的大小的扩展因子。

3 = 使用 3 作为声明的大小的扩展因子。

4 = 使用 4 作为声明的大小的扩展因子。

ON = 使用 M 作为扩展因子，此处 M 是在当前数据库代码集中任何逻辑字符需要的以字节记的最大长度。依赖于 DB_LOCALE 设置， M 有一个从 1（单字节语言环境中）直至 4 的整数范围。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

对于在单语言环境中开发但是在多语言环境中部署的应用，这个特性可降低在数据条目操作期间截断多字节逻辑字符的风险。

在诸如 UTF-8 这样的单字节代码集中，或一些东亚语言的多字节代码集中，单个逻辑字符可要求多于一个字节的存储。设置这个参数可以指导 SQL 分析器将逻辑字符语义应用到这些内建的字符数据类型的声明：

- CHAR
- CHARACTER
- CHARACTER VARYING
- LVARCHAR
- NCHAR
- NVARCHAR
- VARCHAR
- DISTINCT 类型，声明任意这些数据类型作为基本类型
- ROW 类型（命名的和未命名的），包括这些数据类型的域
- 集合类型（LIST、MULTISET 或 SET），包括这些类型作为元素。

您为这个参数指定的设置必须是下列值之一：

SQL_LOGICAL_CHAR 配置参数是否设置为启用或禁用声明的存储大小的扩展，对于 GBase 8s 实例的所有会话，其设置指定数据类型声明如何解释。

扩展因子的自动调整大小

当 SQL_LOGICAL_CHAR 设置为有效的数字，且当前会话创建数据库时， GBase 8s 比较 SQL_LOGICAL_CHAR 值与任何逻辑字符将用于数据库的代码集的最大字节数。

如果 SQL_LOGICAL_CHAR 设置大于最大字节数，则数据库使用语言环境的最大值作为新的扩展因子，取代配置文件指定的因子。在配置文件中的 SQL_LOGICAL_CHAR 设置保持不变，继续作为其他用户数据库的缺省扩展因子发挥作用。

类似地，如果会话的 SQL_LOGICAL_CHAR 值自动地重置为一个数字，如上所述，但同一会话随后连接到另一个数据库，其语言环境使用一个代码集，在该代码集中逻辑字符要求比当前扩展因子更大的存储大小， 则在用户会话连接到那个数据库时， GBase 8s 使用新代码集的最大字节数作为新的扩展因子，而不使用 SQL_LOGICAL_CHAR 的当前设置。

扩展因子的自动重置以匹配代码集中最大的逻辑字符大小，在连接时 DB_LOCALE 指定该代码集，当 SQL_LOGICAL_CHAR 设置为 ON 时也会发生，但是，当 SQL_LOGICAL_CHAR 以两种方式设置为数字（1、2、3 或 4）时，ON 的影响与对数据库服务器行为的影响不相同：

- 如果 SQL_LOGICAL_CHAR 设置为 ON，则扩展因子可自动地重置为较小的值。
- SQL_LOGICAL_CHAR = 4 与 SQL_LOGICAL_CHAR = ON 之间没有不同。

如果当前会话连接到一个数据库时，该数据库在 DB_LOCALE 代码集中的最大逻辑字符要求一个比当前 SQL_LOGICAL_CHAR 设置更小的字节数，则必须设置 SQL_LOGICAL_CHAR 为 ON，而不是一个数字。有效的扩展因子将总是小于或等于语言环境的最大字符大小。

2.1.176 SQLTRACE 配置参数

使用 SQLTRACE 参数来控制 SQL 跟踪的启动环境。

onconfig.std 值

UNIX™ 上：未设置。禁用 SQL 跟踪。

Windows™ 上：#SQLTRACE level=low,ntraces=1000,size=2,mode=global

值

请参阅“用法”部分。

生效

编辑 onconfig 文件并重启数据库服务器之后。

运行带有 set sql tracing 参数的 SQL 管理 API task() 或 admin() 函数之后。

用法

从 onconfig 移除 # 符号来保持关于任何用户运行的最后 1000 个 SQL 语句的基本信息，大小最多 2 KB。通过调整 SQLTRACE 配置参数的域值，您可定制 SQL 跟踪信息的范围。

SQLTRACE 配置参数语法图

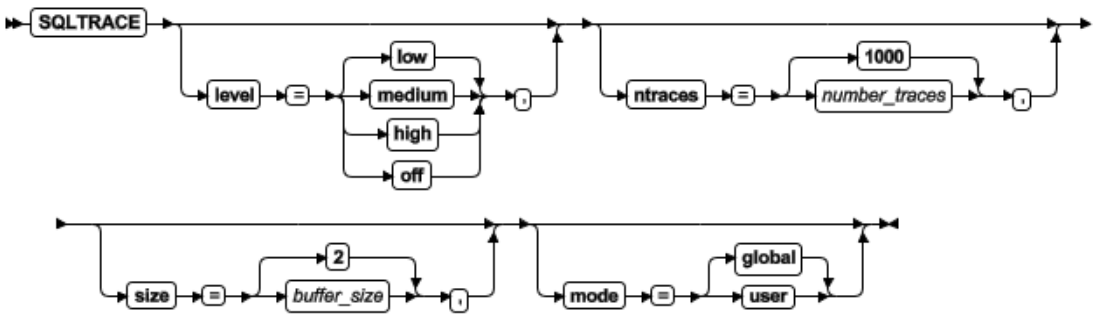


表 1. SQLTRACE 配置参数值的选项.

域	值
level	跟踪的信息量： ● Low = 缺省。捕获语句统计信息、语句文本和语句迭代器。 ● Medium = 捕获包括在 low 级别跟踪中的所有信息，外加表名、数据库名和存储过程堆栈。

	● High = 捕获包括在 medium 级别跟踪的所有信息，外加主机变量。 ● Off = 指定无 SQL 跟踪。
ntraces	*number_traces* 值是重新使用资源之前 SQL 语句跟踪的数目。缺省是 1000。范围是 500 - 2147483647。
size	*buffer_size* 值是要存储的可变长度数据大小的最大值，以 KB 为单位。缺省是 2。范围是 1 -100。如果超过这个缓冲区大小，则数据库服务器丢弃保存的数据。
mode	执行跟踪的范围： ● Global = 缺省。所有用户。 ● User = 通过 SQL 管理 API task() 或 admin() 函数启用跟踪的用户。如果您想要得到一小部分用户正在运行的 SQL 样例，请指定这个模式。

gstat -g his 命令显示 SQL 跟踪信息。

2.1.177 SSL_KEYSTORE_LABEL 配置参数

使用 SSL_KEYSTORE_LABEL 配置参数来指定在 GBase_8s keystore 数据库中使用的服务器数字证书的标签。GBase_8s keystore 数据库是一个存储 SSL 密钥和数字证书的受保护数据库。

onconfig.std 值

未设置。

值

最多 512 个字符的在安全套接字层（SSL）协议通信中使用的 GBase 8s 证书标签

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

缺省值与缺省 SSL 认证的标签名称相同。SSL 证书存储在 GBASEDBTDIR/ssl/servername.kdb 目录中的 GBase 8s keystore 中。

要了解关于您需要在客户端设置的配置参数的信息，请参阅 GBase 8s 安全指南 。

2.1.178 STACKSIZE 配置参数

使用 STACKSIZE 配置参数来指定数据库服务器用户线程的堆栈大小。

onconfig.std 值

STACKSIZE 32，32 位数据库服务器

STACKSIZE 64，64 位数据库服务器

值

32 至由数据库服务器配置和可用内存量确定的限度

单位

KB

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

STACKSIZE 的值没有上限，但设置过大的值会浪费虚拟内存空间并可导致交换空间问题。对于 32 位平台，对非递归的数据库活动而言，32 KB 的缺省 STACKSIZE 值足够。对于 64 位平台，推荐的 STACKSIZE 值是 64 KB。当数据库服务器执行递归的数据库任务时，例如在一些 SPL 例程中，数据库服务器检查堆栈大小溢出的可能性并自动地扩展堆栈。

用户线程执行用户定义的例程。要增加一个特定例程的堆栈大小，请在 CREATE FUNCTION 语句上使用 stack 修饰符。

警告： 设置 STACKSIZE 的值过低可导致堆栈溢出，虽未定义其结果，但其结果通常是不好的。

2.1.179 STARTWITH_OPTIMIZE_MODE 配置参数

指定是否开启START WITH的HASH优化，0：不开启；1：开启。此优化为实例级优化，默认为不开启。

在START WITH语句中增加HINT /*+STARTWITH_BIGHASH*/，对应的语句执行时会进行HASH优化，例如：

```
SELECT /*+STARTWITH_BIGHASH*/ a FROM t3
  START WTIH a=1 CONNECTED BY b=PRIOR a;
```

2.1.180 STATCHANGE 配置参数

当启用 UPDATE STATISTICS 操作的自动模式时，如果分布统计有资格更新，则使用 STATCHANGE 配置参数来指定数据库用来确定更改阈值的全局百分率的一个正整数。

onconfig.std 值

STATCHANGE 10

值

0 - 100

单位

更改阈值的百分率

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

当 AUTO_STAT_MODE 配置参数、AUTO_STAT_MODE 会话环境变量或 UPDATE STATISTICS 语句的 AUTO 关键词启用 UPDATE STATISTICS 操作的自动模式时，数据库服务器使用 STATCHANGE 配置参数的值。

当启用 UPDATE STATISTICS 操作的自动模式时，STATCHANGE 设置为数据库服务器指定一个更改阈值，用来确定分布统计是否有资格更新。当启用这个模式时，UPDATE STATISTICS 语句比较该 STATCHANGE 设置与自动计算当前数据分布以来每一表或分片更改行的百分率，并有选择地仅更新在 UPDATE STATISTICS 语句范围内每一表或分片丢失的或陈旧的分布统计。

2.1.181 STD_TO_CHAR 配置参数

使用 STD_TO_CHAR 配置参数来控制 TO_CHAR() 函数的行为。

onconfig.std 值

STMT_TO_CHAR 1

值

1= 缺省值，体现 TO_CHAR() 函数本身的功能。

0 = TO_CHAR() 函数等同于 GBASE_TO_CHAR() 函数，体现 GBASE_TO_CHAR() 函数的功能。

有关 GBASE_TO_CHAR() 函数和 TO_CHAR() 函数的用法，请参阅《GBase 8s SQL 指南：语法》。

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.182 STD_TO_DATE 配置参数

使用 STD_TO_DATE 配置参数来控制 TO_DATE() 函数的行为。

onconfig.std 值

STMT_TO_DATE 1

值

1= 缺省值，体现 TO_DATE() 函数本身的功能。

0 = TO_DATE() 函数等同于 GBASE_TO_DATE() 函数，体现GBASE_TO_DATE() 函数的功能。

有关 GBASE_TO_DATE() 函数和 TO_DATE() 函数的用法，请参阅《GBase 8s SQL 指南：语法》。

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.183 STMT_CACHE 配置参数

使用 STMT_CACHE 配置参数来确定数据库服务器是否使用 SQL 语句高速缓存。

onconfig.std 值

STMT_CACHE 0

值

0 = 不使用 SQL 语句高速缓存（等同于 gadmin -e OFF）。

1 = 启用 SQL 语句高速缓存，但用户会话不使用高速缓存。仅当设置环境变量 STMT_CACHE 为 1 或执行 SQL 语句 SET STATEMENT CACHE ON 时，用户使用该高速缓存。

2 = 开启 SQL 语句高速缓存。高速缓存所有语句。要关闭语句高速缓存，请设置环境变量 STMT_CACHE 为 0 或执行 SQL 语句 SET STATEMENT CACHE OFF。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

您可以两种模式之一启用 SQL 语句高速缓存：

- 总是使用 SQL 语句高速缓存除非用户明确地指定不使用它。设置 STMT_CACHE 配置参数为 2 或 gadmin -e ON。
- 仅当用户明确地指定使用 SQL 语句高速缓存时才使用它。设置 STMT_CACHE 配置参数为 1 或 gadmin -e ENABLE。

2.1.184 STMT_CACHE_HITS 配置参数

使用 STMT_CACHE_HITS 配置参数来指定在语句完全地插入到 SQL 语句高速缓存中之前命中（引用）语句的数目。

onconfig.std 值

STMT_CACHE_HITS 0

值

0 = 在 SQL 语句高速缓存中完全地插入所有限定的语句。

>0 = 用户首次发出唯一的语句，数据库服务器在该语句标识的高速缓存中插入一个 key-only 条目。随后的同一语句增加 key-only 高速缓存条目的命中数。当 key-only 高速缓存条目的命中数达到指定的命中数时，数据库服务器在高速缓存中完全地插入语句。设置 hits 为 1 或更多来排除 GBase_8s ad hoc 查询进入高速缓存。

单位

整数

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

2.1.185 STMT_CACHE_NOLIMIT 配置参数

使用 STMT_CACHE_NOLIMIT 配置参数来控制是否将限定的语句插入到 SQL 语句高速缓存内。

onconfig.std 值

STMT_CACHE_NOLIMIT 0

如未出现

1

值

0 = 防止将语句插入到高速缓存中。如果当前高速缓存中的大多数语句当前在用，则高速缓存可增长超过大小限制，因为高速缓存清除赶不上插入的速度。如果您关注内存使用情况，则请关闭 STMT_CACHE_NOLIMIT 以防数据库服务器为高速缓存分配大量内存。

1 = 总是在 SQL 语句高速缓存中插入语句，不管高速缓存的大小。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

2.1.186 STMT_CACHE_NUMPOOL 配置参数

使用 STMT_CACHE_NUMPOOL 配置参数来为 SQL 语句高速缓存指定内存池的数目。要获得关于这些内存池的信息，请使用 gstat -g ssc pool。

因为数据库服务器不插入所有从高速缓存中的内存池分配内存的语句，高速缓存大小可能比内存池的大小合计小一些。

onconfig.std 值

STMT_CACHE_NUMPOOL 1

值

1 - 256

单位

正整数

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.187 STMT_CACHE_SIZE 配置参数

使用 STMT_CACHE_SIZE 配置参数来指定以 KB 为单位的 SQL 语句高速缓存的大小。在下一次添加语句到高速缓存时，新的高速缓存大小生效。

onconfig.std 值

STMT_CACHE_SIZE 512

值

正整数

单位

KB

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.188 STOP_APPLY 配置参数

使用 STOP_APPLY 配置参数来停止 RS 辅助服务器应用从主服务器收到的日志文件。

onconfig.std 值

STOP_APPLY 0

缺省值

0

值

0 = 应用日志

1 = 立即停止应用日志

"YYYY:MM:DD-hh:mm:ss" = 在指定的时间停止日志应用，此处：

- YYYY = 年
- MM = 月
- DD = 日
- hh = 时

- mm = 分
- ss = 秒

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

参考

GBase 8s 管理员指南 中的 RS 辅助服务器灾难恢复的延迟

用法

停止日志文件的应用允许您通过从 RS 辅助服务器恢复数据快速地从错误的数据库修改恢复。您可配置服务器或立即或在指定的时间点停止日志的应用。当设置 STOP_APPLY 值时，您还必须设置 LOG_STAGING_DIR。如果配置 STOP_APPLY 而未将 LOG_STAGING_DIR 设置为一个有效且安全的目录，则不可初始化服务器。

日志文件以二进制格式存储在 LOG_STAGING_DIR 配置参数指定的目录中。您必须为日志文件指定一个有效和安全的位置。

要查看关于发送到为 RS 辅助服务器设置的日志 GBase_8s staging 目录的数据信息，请在 RS 辅助服务器上运行 gstat -g rss verbose 命令。

如果向 staging 文件写失败，则 RS 辅助服务器发起事件报警 40007。

假定为 STOP_APPLY 配置参数指定的时间值与 RS 辅助服务器在同一个时区中。

dbexport 实用程序不可支持在可更新的辅助服务器上的写操作，除非设置 STOP_APPLY 参数。（除 STOP_APPLY之外，UPDATABLE_SECONDARY 和 USELASTCOMMITTED 配置参数还必须通过在辅助数据复制服务器上的 dbexport 设置为启用写操作。）

如果远程独立辅助（RSS）服务器将其 STOP_APPLY 配置参数设置为一个非零的值，则那台服务器不可使用群组事务协调。

2.1.189 STORAGE_FULL_ALARM 配置参数

使用 STORAGE_FULL_ALARM 配置参数来配置当存储空间填满时消息和报警的频率和严重程度。

onconfig.std 值

STORAGE_FULL_ALARM 600,3

值

seconds = 0（关）或指明通告之间秒数的正整数。

severity_level = 0（无报警）或 1 - 5

单位

seconds,severity_level

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

当诸如 dbspace、sbspace、blobspace 或 tblspace 这样的存储空间或分区填满时，发起报警并发送消息到 online 消息日志。您可以这个参数的第一个值指定通告之间的秒数。您可指定返回的事件报警的最低严重程度。设置特定的严重程度防止发出严重程度更低的事件。但会发出与指定的严重程度相同或更严重的事件。您可通过设置这个参数为 0 防止当存储空间填满时报警。

不管 STORAGE_FULL_ALARM 的值，当存储空间或分区填满时，都将消息发送到 online 消息日志。

2.1.190 SYSALARMPROGRAM 配置参数

使用 SYSALARMPROGRAM 配置参数来指定 evidence.sh 脚本的全路径名。当数据库服务器发生故障时，数据库服务器执行 evidence.sh。 您可使用从 evidence.sh 脚本的输出来诊断数据库故障的原因。

onconfig.std 值

UNIX™ 上：$GBASEDBTDIR/etc/evidence.sh

Windows™ 上：未设置。

值

pathname = evidence.sh 脚本的全路径名。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

在 Windows 上，您必须启用 evidence.bat 的命令扩展来成功地完成。通过发出下列命令，您可启用或禁用您正在工作的命令提示的扩展：

- 启用：cmd /x
- 禁用：cmd /y

您还可从 Windows XP 注册表启用或禁用命令扩展：

表 1. 从 Windows 注册表启用命令扩展

属性	值

Hive	HKEY_CURRENT_USER
Key	Software\Microsoft\Command Processor
Name	EnableExtensions
Type	REG_DWORD
Values	0（禁用），1（启用）

2.1.191 SYSSBSPACENAME 配置参数

使用 SYSSBSPACENAME 配置参数来指定 sbspace 的名称，数据库服务器在其中存储分片级别数据分布统计信息，syfragsdist 系统目录表在其 encsdist 列中存储为 BLOB 对象。还使用 SYSSBSPACENAME 来指定 sbspace 的名称，数据库服务器在其中存储 UPDATE STATISTICS 语句为某些用户定义的数据类型收集的统计信息。

onconfig.std 值

未设置。

如未出现

0

值

最多 128 字节。SYSSBSPACENAME 必须是唯一的，以一个字母或下划线开头，且仅包含数字、字母、下划线或 $ 字符。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

参考

- 更新统计信息，在 GBase 8s 性能指南 中关于个别查询性能的章节中
- sbspace 特征，在 GBase 8s 性能指南 中关于配置对 I/O 影响的章节中
- 写用户定义的统计信息，在 GBase 8s 用户定义的例程和数据类型开发者指南 中的性能章节中
- 提供一列的统计信息数据，在 GBase 8s DataBlade API 程序员指南 中

用法

要支持分片级别统计信息，您必须指定一个 sbspace 的名称作为 SYSSBSPACENAME 设置，且您必须分配那个 sbspace（通过使用 gspaces 实用程序，如下所示。对于任何其

STATLEVEL 属性设置为 FRAGMENT 的表，如果未设置 SYSSBSPACENAME，或如果未正当地分配 SYSSBSPACENAME 设置到的那个 sbspace，则数据库服务器返回错误）。

对于分片表中一列的分布统计信息，您可根据这个公式估计该 sbspace 要求的存储容量为多少字节：

nfrags * 1.25 * ((10000 / resolution) * ((2 * column_width) + 6))

此处 1.25 接近溢出 GBase_8s bin 的数目。公式还包括这些变量：

- column_width 是 UPDATE STATISTICS 语句指定列的以字节计的宽度。
- nfrags 是表的分片数。
- resolution 是计算分布的 UPDATE STATISTICS 语句的 resolution 子句中 percent 值。

resolution 还是 dbschema -hd table 命令为该列分布统计信息显示的内容。

SYSSBSPACENAME 还指定 sbspace 的名称，数据库服务器在其中存储 UPDATE STATISTICS 语句为某些用户定义的数据类型收集的统计信息。通常情况下，数据库服务器在 sysdistrib 系统目录表中存储统计信息。

请不要混淆 SYSSBSPACENAME 配置参数与 SBSPACENAME 配置参数。

由于用户定义的数据类型的数据分布可很大，您有在 sbspace 中存储它们，而不存储在 sysdistrib 系统目录表中的选项。如果您在 sbspace 中存储数据分布，则请使用 DataBlade API 或 GBase 8s ESQL/C 功能来检查统计信息。

即使您以 SYSSBSPACENAME 参数指定一个 sbspace，在可使用它之前，您也必须以 gspaces 实用程序的 -c -S 选项创建该 sbspace。当发生下列情况之一时，数据库服务器验证这个 sbspace 的名称：

- 当数据库服务器执行带有 MEDIUM 或 HIGH 关键词的 UPDATE STATISTICS 语句时，数据库服务器尝试将多重表示类型的数据分布写到 SYSSBSPACENAME。
- 当数据库服务器执行带有 DROP DISTRIBUTIONS 关键词的 UPDATE STATISTICS 语句时，数据库服务器尝试将多重表示类型的数据分布删除到 SYSSBSPACENAME。

如果未设置 SBSSPACENAME，或如果未分配存储到那个 sbspace，则数据库服务器可能存储分布统计信息，以便 UPDATE STATISTICS 操作以错误 -9814失败。

虽然您可在 SYSSBSPACENAME 中指定的 sbspace 中存储智能大对象，但推荐将分布统计信息与智能大对象保持在分开的 sbspace 中，因为：

- 您要避免当查询正在访问智能大对象，且查询优化器正在使用分布来确定查询计划时发生磁盘争用。
- 当每一 sbspace 用于不同的目的，磁盘空间花费较长时间才填满。

2.1.192 TAPEBLK 配置参数

用于存储空间备份的磁带的块大小（以千字节为单位）。

2.1.193 TAPEDEV 配置参数

用于存储空间备份的目录文件系统或磁带设备的绝对路径名称。指定 gtape 在归档过程中写入存储空间数据的目标位置，以及 gtape在恢复过程中读取数据的源位置。

要配置 gtape 以使用 stdio，将 TAPEDEV 设置为 STDIO。

在云环境中备份或恢复时，将以下语法用于 TAPEDEV 配置参数：

```
TAPEDEV 'local_path, keep=option, cloud=cloud_vendor, url=url'
```

- local_path 是将存储空间备份对象临时存储在的目录的完整路径名。
- option 可以设置为 yes 或 no。如果 keep 设置为 yes，那么 gtape 实用程序将备份对象保留在本地目录中。如果 keep 设置为 no，那么在云存储位置中传输备份对象后，会将这些对象删除。
- cloud_vendor 是云存储供应商的名称。
- url 是将存储空间备份数据永久存储在的云存储位置。

2.1.194 TAPESIZE 配置参数

用于存储空间备份的磁带的大小（以千字节为单位）。 值可以为 0 - 2,097,151。

2.1.195 TBLSPACE_STATS 配置参数

使用 TBLSPACE_STATS 配置参数来开启或关闭 tblspace 统计信息的收集。使用 gstat -g ppf 命令来罗列 tblspace 统计信息。

onconfig.std 值

TBLSPACE_STATS 1

值

0 = 关闭 tblspace 统计信息的收集。gstat -g ppf 命令显示 partition profiles disabled。

1 = 开启 tblspace 统计信息的收集。

单位

整数

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.196 TBLTBLFIRST 配置参数

如果您想要指定 root dbspace 中 tblspace tblspace 的第一个 extent 大小，则请使用 TBLTBLFIRST 配置参数。如果您不想要数据库服务器自动地管理 extent 大小，则请设置这个参数。

onconfig.std 值

TBLTBLFIRST 0

值

从以 KB 指定的 250 页等同的大小，至第一个 chunk 的大小减去任何系统对象需要的空间。

单位

以 KB 为单位的页大小的倍数

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

您可能想要指定第一个和下一个 extent 大小来减少 tblspace tblspace extent 的数目并减少将 tblspace tblspace extent 置于非主 chunk 中的频率（主 chunk 是一个 dbspace 中的初始 chunk。）

您可使用 gcheck -pt 和 gcheck -pT 来显示 tblspace tblspace 第一个和下一个 extent 大小。

如果您想要为非 root dbspace 配置第一个 extent，则请使用 gspaces 实用程序。

2.1.197 TBLTBLNEXT 配置参数

TBLTBLNEXT 配置参数指定 root dbspace 中 tblspace tblspace 的下一个 extent 大小。如果您不想要数据库自动地管理 extent，则请设置这个参数。

onconfig.std 值

TBLTBLNEXT 0

值

从以 KB 指定的 4 页的等同大小，至最大 chunk 大小减去三页

单位

KB

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

如果在主 chunk 中没有下一个 extent 的足够空间，则从另一个 chunk 分配该 extent。如果指定的空间不可用，则分配最近的可用空间。

2.1.198 TEMPTAB_NOLOG 配置参数

使用 TEMPTAB_NOLOG 配置参数来在临时表上禁用日志记录。

onconfig.std 值

TEMPTAB_NOLOG 0

值

0 = 在临时表操作上启用逻辑日志记录

1 = 在临时表操作上禁用逻辑日志记录

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

因为防止 GBase 8s 在网络上转移临时表，所以这个参数可提高应用程序中的性能。可用 gadmin -wf 实用程序动态地更新该设置。

如果您启用这个设置，则请注意，当使用临时表时因为没有日志记录数据，在临时表上回滚事务将不再在临时表中恢复工作的原状。

对于高可用性集群中的HDR、RSS 和 SDS 辅助服务器，应总是通过设置 TEMPTAB_NOLOG 配置参数为 1 来禁用临时表上的逻辑日志记录。

2.1.199 TLS_VERSION 配置参数

使用 TLS_VERSION 配置参数来指定数据库服务器用于网络连接的“传输层安全”（TLS）版本。缺省情况下，启用 TLS 版本 1.0、1.1 和 1.2。

onconfig.std 值

未设置。启用所有版本。

缺省值

1.0,1.1,1.2

值

一个或多个 TLS 版本。以逗号分隔多个版本。

- 1.0 = TLS 版本 1.0。
- 1.1 = TLS 版本 1.1。
- 1.2 = TLS 版本 1.2。

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

TLS 是“安全套接字层”（SSL）的后继者，提供客户端/服务器连接的密码协议。为了计算机的通信，它们必须有一个共同的 TLS 版本，与那个 TLS 级别的一个有效数字证书在一起。 例如，如果两台计算机启用 TLS 1.0、1.1 和 1.2，但其中一台计算机的数字证书仅支持 TLS 1.0，则该连接将在 TLS 1.0 一级。

2.1.200 TXTIMEOUT 配置参数

使用 TXTIMEOUT 配置参数来指定在开始参与者恢复之前，在两阶段提交中一个参与者等待的时间量。这个参数仅用于涉及远程数据库服务器的分布查询。非分布查询不使用这个参数。

onconfig.std 值

TXTIMEOUT 300

值

正整数

单位

秒

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

2.1.201 UNSECURE_ONSTAT 配置参数

使用 UNSECURE_ONSTAT 配置参数来移除数据库系统管理员（DBSA）用户 gstat 命令的访问限制。

onconfig.std 值

未设置。

值

1 = 所有用户可运行 gstat 命令来查看正在运行的 SQL 语句

生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

缺省情况下，限制 DBSA 用户从活动的会话使用显示 SQL 语句文本的 gstat 命令。要移除这个限制，请设置 UNSECURE_ONSTAT 配置参数为 1。显示 SQL 语句的 gstat 命令包括 gstat -g his、gstat -g ses、gstat -g stm、gstat -g ssc 和 gstat -g sql。

2.1.202 UPDATABLE_SECONDARY 配置参数

使用 UPDATABLE_SECONDARY 配置参数来设置在主服务器与辅助服务器之间要建立的连接数。设置这个配置参数使得客户端应用能够在一台高可用性辅助服务器上执行更新、插入和删除操作。

onconfig.std 值

UPDATABLE_SECONDARY 0

值

从零（缺省值）直到两倍于 CPU VP 数目的任何数。设置该值为 0 配置辅助服务器为只读。设置该值为从 1 值 CPU VP 数目的两倍，使得辅助服务器可更新并配置连接线程。

单位

在一给定的辅助服务器与其主服务器之间的网络连接数目

生效

编辑 onconfig 文件并重启数据库服务器之后。

“辅助数据复制服务器”的隔离级别

如果 UPDATABLE_SECONDARY 配置参数未设置或设置为零，则辅助数据复制服务器为只读。在这种情况下，在辅助服务器上仅 DIRTY READ 或 READ UNCOMMITTED 事务隔离级别可用。如果 UPDATABLE_SECONDARY 参数设置为一个大于零的有效连接数，则辅助数据复制服务器可支持 COMMITTED READ、COMMITTED READ LAST COMMITTED 或 COMMITTED READ 事务隔离级别，或者 USELASTCOMMITTED 会话环境变量。仅 SQL DML 语句，诸如 INSERT、UPDATE、MERGE 和 DELETE 和 dbexport 实用程序，可支持在可更新的辅助服务器上写操作。（除了 UPDATABLE_SECONDARY 之外，还必须设置 STOP_APPLY 和 USELASTCOMMITTED 配置参数通过在辅助数据复制服务器上的 dbexport 来启用写操作。）

2.1.203 USELASTCOMMITTED 配置参数

使用 USELASTCOMMITTED 配置参数来指定隔离级别，COMMITTED READ 隔离级别的 LAST COMMITTED 特征是隐含有效的。

onconfig.std 值

USELASTCOMMITTED "NONE"

缺省值

"NONE"

值

"NONE" = 未标识隔离级别。当尝试在 Committed Read、Dirty Read、Read Committed 或 Read UncommittedNo 隔离级别中读一行时，如果您的会话遇到排他锁，则您的事务不可读那一行，直到持有该排他锁的并发事务提交或回滚。

"COMMITTED READ" = 所有来自 Committed Read 隔离级别的事务都视为最后提交事务。当数据库服务器在试图读一个 Committed Read 或 Read Committed 隔离级别中的行时遇到一个排他锁，数据库服务器读该数据的最近提交版本。

"DIRTY READ" = 所有来自 Dirty Read 隔离级别的事务都视为最后提交事务。如果数据库服务器在试图读一个 Dirty Read 或 Read Uncommitted 隔离级别中的行时遇到一个排他锁，则数据库服务器读该数据的最近提交版本。

"ALL" = 所有来自 Committed Read 和 Dirty Read 两个隔离级别的事务都视为最后提交事务。如果数据库服务器在试图读一个 Committed Read、Dirty Read、Read Committed 或 Read Uncommitted 隔离级别中的行时遇到一个排他锁，则数据库服务器读该数据的最近提交版本。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

LAST COMMITTED 特征可减小在有排他行锁的表上并发事务之间发生锁定冲突的风险。

USELASTCOMMITTED 配置参数还可为 SET TRANSACTION 语句的 READ COMMITTED 和 READ UNCOMMITTED 隔离级别启用 LAST COMMITTED 语义。

USELASTCOMMITTED 配置参数仅与那些已经创建或更改为锁定颗粒度的 ROW 级别的表一起工作。没有以任何锁模式设置创建的表使用 DEF_TABLE_LOCKMODE 中的缺省设置。如果 DEF_TABLE_LOCKMODE 设置为 PAGE，则 USELASTCOMMITTED 配置参数不可访问表中的最近提交数据，未提交的事务在这些表上持有排他锁，除非这些表明确地变更为锁定颗粒度的 ROW 级别。

与共享磁盘辅助数据库服务器一起使用

在“共享磁盘”（SD）辅助数据库服务器上，USELASTCOMMITTED 配置参数也有效。下列表显示 SD 辅助服务器上 USELASTCOMMITTED 配置参数的有效值及其描述。

表 1. 有效的辅助服务器 USELASTCOMMITTED 值

USELASTCOMMITTED 值	描述
NONE	COMMITTED READ LAST COMMITTED 不是会话的缺省隔离级别
COMMITTED READ	COMMITTED READ LAST COMMITTED 是所有带有 Committed Read 隔离的会话的缺省隔离级别
DIRTY READ	COMMITTED READ LAST COMMITTED 是所有带有 Dirty Read 隔离

USELASTCOMMITTED 值	描述
	的会话的缺省隔离级别
ALL	COMMITTED READ LAST COMMITTED 是所有带有 Committed Read 或 Dirty Read 隔离的会话的缺省隔离级别

2.1.204 USEOSTIME 配置参数

onconfig.std 值

USEOSTIME 0

值

0 = 关

1 = 开

生效

初始化期间

参考

- GBase 8s 性能指南
- 使用 CURRENT 函数来返回 datetime 值，在 GBase 8s SQL 指南：语法 中

用法

设置 USEOSTIME 为 1 指定当数据库服务器从操作系统为 SQL 语句获取当前时间时，数据库服务器要使用亚秒精度。下列示例显示 datetime 值中的亚秒：

2001-09-29 12:50:04.612

如果不需要亚秒精度，则数据库服务器每秒从操作系统检索当前时间一次，使得客户端应用的时间精度为一秒。如果设置 USEOSTIME 为 0，则当前函数为该年向分数域返回零（.000）。

当数据库服务器的主机计算机有亚秒精度时钟时，其 SQL 语句依赖于亚秒精确度的应用应设置 USEOSTIME 为 1。

与关闭 USEOSTIME 运行的系统相比，USEOSTIME 设置为非零的系统注意到性能下降达 4% 至 5%。

从应用程序到 GBase 8s 内嵌语言库函数，这个设置不影响任何关于时间的调用。

2.1.205 USERMAPPING 配置参数（UNIX™，Linux™）

使用 USERMAPPING 配置参数来设置数据库服务器是否接受来自映射用户的连接。

缺省值

OFF

值

OFF = 仅在有登录服务的 GBase 8s 主机计算机 OS 中注册的那些用户可连接到数据库服务器。在 GBase 8s 主机计算机上没有 OS 账号的外部认证的用户不可连接到数据库服务器资源。

BASIC = 没有 OS 账户的用户可连接到 GBase 8s 。即使没有 OS 账户的用户映射到服务器管理员用户或组 ID，该用户也不可在数据库服务器上执行有权限的用户操作。

ADMIN = 没有 OS 账户的用户可连接到 GBase 8s 。如果用户以一个有权限的用户身份得到认证，且映射到正确的服务器管理员组 ID，则该用户可在数据库服务器上执行 DBSA、DBSSO 或 AAO 的工作。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

当通过设置带有 BASIC 或 ADMIN 值的参数开启 USERMAPPING 时，在 GBase 8s 主机计算机上没有操作系统（OS）账户的外部认证的用户可访问数据库服务器资源。BASIC 或 ADMIN 的设置还确定映射用户能否获得管理权限。

重要：在为映射的用户建立 GBase 8s 中，将 USERMAPPING 配置参数从 OFF 更改为 ADMIN 或 BASIC 不是唯一的步骤。 要以适当的用户属性映射用户，您还必须使用诸如 CREATE USER 和 ALTER USER 这样的 DLL 语句，在 SYSUSER 数据库的适当系统表中来注册这些值。依赖于使用的 DDL 语句和定义的表映射，将更新或填入下列表：

- SYSINTAUTHUSERS
- SYSUSERMAP
- SYSSURORGATES
- SYSSURROGATEGROUPS

2.1.206 USRC_HASHSIZE 配置参数

USRC_HASHSIZE 配置参数指定 LBAC 凭证内存高速缓存中散列存储区的数目。这个内存高速缓存持有关于用户 LBAC 凭证的信息。

onconfig.std 值

USRC_HASHSIZE 31

值

任何正整数

单位

KB

生效

编辑 onconfig 文件并重启数据库服务器之后。

2.1.207 USRC_POOLSIZE 配置参数

USRC_POOLSIZE 配置参数指定在 LBAC 凭证内存高速缓存的每一散列存储区中条目的最大数。这个内存高速缓存持有关于用户 LBAC 凭证的信息。

onconfig.std 值

USRC_POOLSIZE 127

值

正值 127 或更大，表示高速缓存中条目的初始最大数目的一半。该最大值依赖于共享内存配置和服务器实例的可用共享内存。

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wm 命令增加内存中的该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

高速缓存中条目的初始数目是 USRC_POOLSIZE 配置参数值的两倍。例如，如果 USRC_POOLSIZE 配置参数设置为 127，则该高速缓存中允许 254 个条目。 如果高速缓存中的所有条目填满，则高速缓存大小自动地增长 10%。要减小高速缓存的大小，请降低 onconfig 文件中 USRC_POOLSIZE 配置参数的值并重启服务器。

2.1.208 USTLOW_SAMPLE 配置参数

当您在 LOW 模式中运行 UPDATE STATISTICS 语句时，使用 USTLOW_SAMPLE 配置参数来根据样例启用索引统计信息的生成。

对于有多于 100 K 叶子页的索引，使用样例收集统计信息可提高 UPDATE STATISTICS 操作的速度。

onconfig.std 值

USTLOW_SAMPLE 1

值

0 = 禁用样例

1 = 启用样例

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

2.1.209 VP_MEMORY_CACHE_KB 配置参数

使用 VP_MEMORY_CACHE_KB 参数来为每一 CPU 虚拟处理器创建私有内存高速缓存。

onconfig.std 值

VP_MEMORY_CACHE_KB 0

值

0 = 关

所有私有内存高速缓存的合计大小，可选地后跟逗号和高速缓存的模式。

大小，以 KB 为单位：

- SHMTOTAL 配置参数指定从 800 到等于内存限制的 40% 的一个数目。

模式：

- STATIC = 缺省。指定的大小是结合了所有私有内存高速缓存大小的最大值。
- DYNAMIC = 指定的大小是所有私有内存高速缓存的初始大小。高速缓存大小动态地变更，但不能超过 SHMTOTAL 配置参数的值。

生效

编辑 onconfig 文件并重启数据库服务器。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

用法

在 GBase 8s 服务器中，私有内存高速缓存可提高线程分配的内存性能。私有内存高速缓存不影响分配给缓冲池或共享内存通信使用的内存。

当您将 VP_MEMORY_CACHE_KB 配置参数的值设置为非零数目时，为每一 CPU 虚拟处理器创建一个私有内存高速缓存。缺省情况下，所有私有内存高速缓存相结合的大小限定在指定的 KB 数。

如果您想要根据需要自动地增加或减少每一私有内存高速缓存的大小，则请在大小之后包括一个逗号和词 DYNAMIC，例如，VP_MEMORY_CACHE_KB 1000,DYNAMIC。虽然所有私有内存高速缓存相结合的最大初始大小不可超过 SHMTOTAL 配置参数值的 40%，带有 DYNAMIC 模式设置，但高速缓存的大小可扩展超出初始的限制。高速缓存的合计大小不可超过 SHMTOTAL 配置参数的值。

注意： 在繁忙系统上的动态内存高速缓存可快速地增长，并使用大量可用内存。为此，如果您设置模式为 DYNAMIC，则请设置 SHMTOTAL 配置参数为一特定限制，而不是缺省值 0，设置为 0 不限制内存的数量。

如果您重置 VP_MEMORY_CACHE_KB 配置参数为 0，则内存高速缓存被清空并禁用。

gstat -g vpcache 命令返回关于私有内存高速缓存的统计信息。

2.1.210 VPCLASS 配置参数

使用 VPCLASS 配置参数来创建和配置虚拟处理器。

onconfig.std 值

UNIX™ : VPCLASS cpu,num=1,noage

Windows™ :

VPCLASS cpu,num=1,noage

#VPCLASS aio,num=1

#VPCLASS jvp,num=1

值

最多 128 字节字符。每一 VPCLASS 配置参数值必须是唯一的，以一个字母或下划线开头，且仅包含数字、字母、下划线或 $ 字符。不包括空格。请参阅“用法”部分。

分隔符

以逗号分隔每一域。

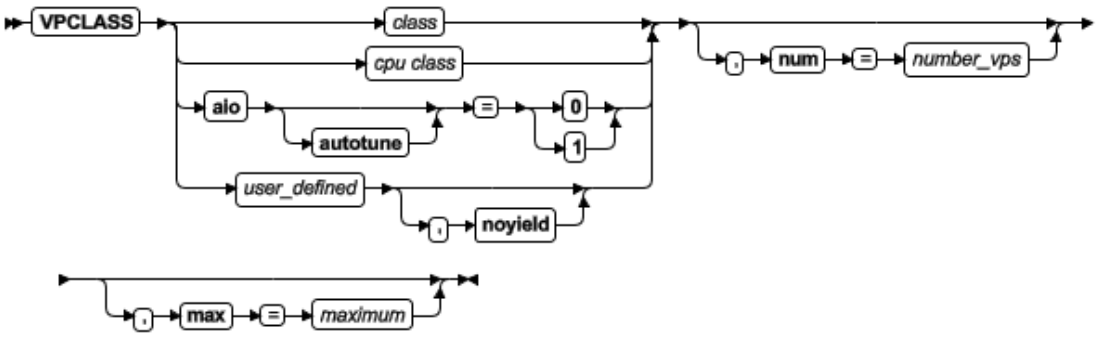
生效

编辑 onconfig 文件并重启数据库服务器之后。

用法

您可在 onconfig 文件中添加多个 VPCLASS 配置参数条目。每一 VPCLASS 配置参数必须描述一个不同的虚拟处理器类。每一定义置于单独的行上。

VPCLASS 配置参数的语法



CPU CLASS

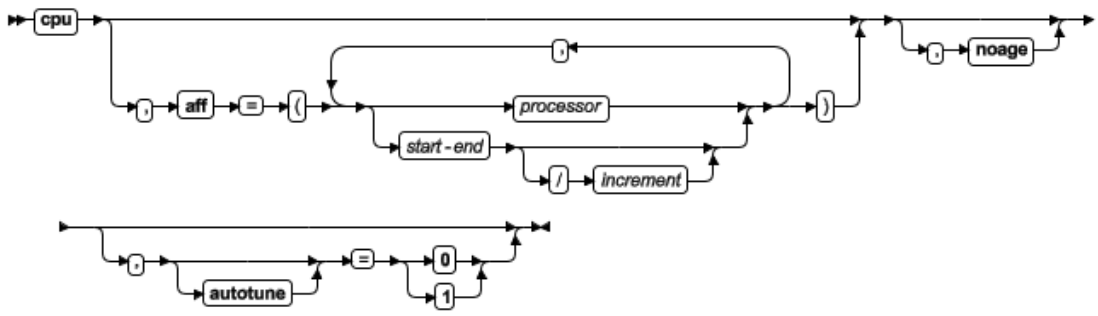


表 1. VPCLASS 配置参数值的选项

域	值
class	class 值是虚拟处理器类的名称。数据库服务器根据需要启动大多数虚拟处理器。典型地，您可能为 CPU、AIO、JVP 和用户定义的虚拟处理器类设置 VPCLASS 配置参数。 虚拟处理器类名不区分大小写。 要查看类名称的列表，请参阅虚拟处理器的类型。
user_defined	user_defined 值是您为用户定义的例程创建的虚拟处理器类名称。请确保 SINGLE_CPU_VP 配置参数设置为 0。
autotune	指定数据库服务器是否根据需要为指定的类添加虚拟处理器来提高性能，如果包括 max 选项，则最多达 max 选项的值。 ● autotune=0 防止自动添加虚拟处理器 ● autotune=1 启用虚拟处理器的自动添加 如果该类是 cpu，则自动地添加的任何 CPU 虚拟处理器都没有 affinity。忽略 aff 选项。
cpu	指定 CPU 虚拟处理器类。
num	number_vps 值设置指定类的虚拟处理器数目，当数据库服务器启动时数据库服务器启动这个类。缺省值是 1。cpu 和 aio 虚拟处理器类的值域是 1 - 10000。所有其他虚拟处理器类的值域是 0 - 10000。您可使用 gadmin -p 命令来为当前会话的类添加虚拟处理器。
max	maximum 值指定数据库服务器可为该类启动的虚拟处理器的最大数目。该值可为大于 0 的任何整数。缺省情况下，该数目是无限制的。
aff	在支持处理器 affinity 的多处理器计算机上，aff 选项指定数据库服务器绑定 CPU 虚拟处理器的那些 CPU。操作系统将这些 CPU 编号，从 0 至小于 CPU 数目的一个数。缺省情况下，以轮转法方式将CPU 虚拟处理器分派到可用处理器。aff 选项为一个或多个整数： ● processor = 将 CPU 虚拟处理器绑定到得那个 CPU 编号。CPU 编号可以任何顺序罗列。 ● start = CPU 编号范围的起始。

域	值
	● end = CPU 编号范围的终止。 ● increment = 指定在一个使用的范围中 CPU 编号的因子。例如，aff=(1-5/2) 指定使用 CPU 编号 1、3 和 5。
noage	如果操作系统实现优先级 aging，则对于 CPU 虚拟处理器禁用优先级 aging。缺省情况下，优先级 aging 有效。
noyield	指定用户定义的虚拟处理器类不 yield，允许 C UDR yield 给其他需要访问用户定义的虚拟处理器类的线程。缺省情况下，用户定义的虚拟处理器的线程 yield。 非 yielding 用户定义的虚拟处理器类以一种方式运行用户定义的线程，即给线程排他使用虚拟处理器类。使用非 yield 虚拟处理器类的用户定义的线程串行地运行，且从不将该虚拟处理器 yield 给另一个线程。 因为 UDR 运行在单个虚拟处理器上直到完成，且任何附加的虚拟处理器都会空闲，所以在非 yielding 用户定义的虚拟处理器类中仅指定一个虚拟处理器。

这些选项可以任何顺序出现，以逗号分隔。

使用 gadmin -p 命令来动态地添加或移除当前数据库会话的虚拟处理器。gadmin -p 命令不更新 onconfig 文件。

CPU 虚拟处理器

在单处理器计算机上，仅分配一个 CPU 虚拟处理器。在多处理器计算机上，分配的 CPU 虚拟处理器加上用户定义的虚拟处理器的总数目最多达到计算机上的 CPU 数。

当数据库服务器启动时，CPU 虚拟处理器的数目自动地增加到数据库服务器计算机上 CPU 处理器数目的一半，除非启用 SINGLE_CPU_VP 配置参数。

如果包括 autotune 选项，则数据库服务器根据需要添加 CPU 虚拟处理器来提高性能，最多达到计算机上的 CPU 数。

当数据库服务器自动地添加 CPU 虚拟处理器时，不更新 CPU 类 VPCLASS 配置参数的 num 选项。

您可配置处理器 affinity 和是否允许 aging。例如，下列条目创建绑定到 CPU 编号 7、8、9 和 10 的四个 CPU 虚拟处理器，且不受优先级 aging 的影响：

VPCLASS CPU,num=4,aff=(7-10),noage

AIO 虚拟处理器

使用 AIO 虚拟处理器类的 VPCLASS 配置参数条目来指定 AIO 虚拟处理器的确切数目，或启用数据库服务器来根据需要添加 AIO 虚拟处理器。

当没有为 AIO 虚拟处理器类设置 VPCLASS 配置参数条目时，由 AUTO_AIOVPS 配置参数的设置确定 AIO 虚拟处理器的数目，且限定到 128：

- 如果 AUTO_AIOVPS 设置为 1（开），则初始启动的 AIO 虚拟处理器数目等于 AIO chunk 的数目。
- 如果 AUTO_AIOVPS 设置为 0（关），则启动的 AIO 虚拟处理器数目等于 6 或 AIO chunk 的数目之中较大的一个。

Java™ 虚拟处理器

如果您使用 Java 用户定义的例程或 Java 应用，则通过为 JVP 虚拟处理器类添加 VPCLASS 配置参数条目，创建至少一个 Java 虚拟处理器。如果您设置 JVP 的数目为零，或如果没有 JVP 类的 VPCLASS 参数，则您不可运行 Java UDR。

2.1.211 WSTATS 配置参数

使用 WSTATS 配置参数来指定 gstat -g wst 命令是否显示系统内线程的等待统计信息。

注意： 由于收集统计信息的成本，您应预期产生小的性能影响。不建议为生产系统启用 WSTATS 配置参数。

onconfig.std 值

WSTATS 0

值域

0 = 禁用等待统计信息

1 = 启用等待统计信息

生效

编辑 onconfig 文件并重启数据库服务器之后。

当您通过运行 gadmin -wf 命令在 onconfig 文件中动态地重置该值时。

当您通过运行 gadmin -wm 命令重置内存中的该值时。

2.2 sysmaster 数据库

本章描述 sysmaster 数据库并提供关于系统监视接口（ SMI ）的参考信息。

它提供了有关以下主题的信息:

- 什么是 sysmaster 数据库
- 如何使用 SMI 表
- SMI 表的描述

- 已记载的 SMI 表的映射

有关 ON-Bar 表的信息，请参阅 GBase 8s 备份与恢复指南。

2.2.1 sysmaster 数据库

数据库服务器创建并维护 sysmaster 数据库。它类似于数据库的系统目录，数据库的系统目录在 GBase 8s SQL 参考指南 中描述。正如数据库服务器管理的每个数据库的系统目录对数据库中的对象和特权进行跟踪那样，每个数据库服务器的 sysmaster 数据库对有关数据库服务器的信息进行跟踪。

sysmaster 数据库包含了系统监视接口（ SMI ）表。SMI 表提供了有关数据库服务器的状态信息。可以查询这些表以识别处理瓶颈、确定资源的使用、跟踪会话或数据库服务器的活动等等。本章对这些表进行描述，它们与普通表稍有不同。

警告： 数据库服务器依赖于 sysmaster 数据库。请不要更改 sysmaster 中任何表和表中的任何信息。这种更改可能会导致不可预料和削弱性能的结果。

数据库服务器在其初始化磁盘空间时创建 sysmaster 数据库。数据库服务器创建具有未缓冲的日志记录的数据库。您不能删除数据库或其中的任何表，并且不能关闭日志记录。

作为在 UNIX™ 上的用户 Gbasedbt 或者 Windows™ 上的 Gbasedbt-Admin 组的成员，您可以在 sysmaster 数据库创建 SPL 例程。（也可以在 sysmaster 中的表上创建触发器，但数据库服务器从不执行那些触发器。）

联接 sysmaster 中的多个表可能返回不一致的结果，因为数据库服务器在联接过程中并不锁定这些表。可以将 sysmaster 表与其他数据库中的表进行联接。然而，要将 sysmaster 表与非日志记录数据库中的表进行联接，首先要使非日志记录数据库成为当前数据库。

buildsmi 脚本

当第一次启动数据库服务器时，它运行名为 buildsmi 的脚本，该脚本位于 etc 目录。该脚本构建支持 SMI 的数据库和表。数据库服务器需要大约 1750 页逻辑日志空间可用页以构建 sysmaster 数据库。

如果接收到指示您运行 buildsmi 脚本的错误信息，那么可能在数据库构建 SMI 数据库、表和视图时发生了问题。使用 buildsmi 时，将删除然后重新创建现有的 sysmaster 数据库。

在确保创建数据库过程中 sysmaster 数据库没有相关联接后，该脚本才能被执行。而且在 UNIX™ 上该脚本必须以 gbasedbt 用户的身份运行，在 Windows™上该脚本必须以 Gbasedbt-Admin 群组成员的身份运行。例如:若有一个调度程序的任务在 buildsmi 脚本开始时运行，且该程序试图去联接 sysmaster 中的任意一表，那么脚本就会运行失败。

buildsmi 脚本运行中产生的错误信息（在 UNIX ）会被写入 /tmp/buildsmi.out 文件中，或者写入 %GBASEDBTDIR%\etc\buildsmi_out 文件中（在 Windows 上）。

bldutil.sh 脚本

在第一次初始化数据库服务器时，在 UNIX™ 上运行名为 bldutil.sh 的脚本，或者在 Windows™ 上运行名为 bldutil.bat 的脚本。该脚本构建了sysutils 数据库。如果它失败，数据库服务器会在 tmp 目录中创建一个输出文件。该输出文件在 UNIX 为 bldutil.process_id ，在 Windows 上为 bldutil.out 。该输出文件中的消息反映脚本执行过程中发生的错误。

2.2.2 系统监视接口

本节描述 SMI 表及如何访问它们以监视数据库服务器操作。

了解 SMI 表

SMI （系统监视接口）由数据库服务器自动维护的表和伪表组成。当 SMI 表作为表向用户显示时，它们不像普通的表那样记录在磁盘上。相反，数据库服务器视需要在内存中根据该瞬时共享内存中的信息构造表。当查询SMI 表时，数据库服务器从这些共享内存中读取信息。由于数据库服务器不断更新共享内存中的数据，所以 SMI 所提供的信息允许您检查数据库服务器的当前状态。

SMI 表提供关于以下主题的信息：

- 审计
- 检查点
- Chunk I/O
- Chunks
- 数据库日志记录状态
- Dbspaces
- 磁盘使用
- 环境变量
- Extents
- 锁
- 网络
- SQL 语句高速缓存统计信息
- SQL 跟踪
- 构建系统概要文件
- 表
- 构建用户概要文件
- 虚拟处理器 CPU 使用

当用户访问并修改数据库服务器所管理的数据库时，SMI 表中的数据动态更改。

访问 SMI 表

任何用户都可以使用 SQL SELECT 语句查询 SMI 表，但是标准用户不能执行 SELECT 以外的语句。如果试图这样做，那么会导致权限错误。管理员可以执行 SQL 以外的语句，但这类语句的结果是不可预料的。

提示： 对于可预料的结果，应查询与每个表相关联的的视图而不是直接查询表。

如果直接查询 systabpaghdrs 表，必须为 pg_partnum 参数指定一个合适的值。pg_partnum 的值 > 1048576 。然而，如果查询与 systabpaghdrs 表关联的视图，就不用指定 pg_partnum 参数的值。

GBase 8s 包括 sysadtinfo 和 sysaudit 表。只有用户 gbasedbt（在 UNIX™ 上）或 Gbasedbt-Admin 组的成员 （在 Windows™ 上）可以查询 sysadtinfo 和 sysaudit 表。

在 sysmaster 数据库的任何表上都无法使用 dbschema 或 dbexport 。如果这样做，数据库服务器将生成以下错误信息：

Database has pseudo tables - can't build schema

SELECT 语句

您可以在对普通表使用 SELECT 的任何场合对 SMI 表使用 SELECT 语句。

例如，从 DB-Access ，在 SPL 例程中，以 GBase 8s ESQL/C 等，您可以使用 SELECT 语句。

限制： 当查询 SMI 表时，无法有意义地引用 rowid 。使用 rowid 的 SELECT 语句不返回任何错误，但是结果是不可预料的。

包括表之间的连接、输出顺序排序等等在内所有标准 SQL 语法，都可使用于 SMI 表。如果要连接 SMI 表和非 SMI 表，那么用以下标准语法给出 SMI 表的名称：

sysmaster[@dbservername]:[owner.]tablename

触发器和事件报警

基于对 SMI 表的更改的触发器从不运行。尽管可以在 SMI 表上定义触发器，但触发器只在表上出现 INSERT、UPDATE 或 DELETE 语句时才被激活。对 SMI 数据的更新发生在数据库服务器中，这不需要使用 SQL ，因此尽管由 SELECT 语句返回的数据提示它应被激活，但 SMI 表上的触发器不会被激活。

要创建事件警报，请以预定时间间隔查询特定条件，并在警报的必需条件满足时执行 SPL 例程。

SPL 和 SMI 表

可以在 SPL 例程中访问 SMI 表。当引用 SMI 表时，请使用与引用标准表相同的语法。

锁定和 SMI 表

SMI 表中信息根据数据库服务器活动而变更。然而，数据库服务器不使用 SQL 语句更新信息。当以锁定对象的隔离级别使用 SMI 表时，它阻止其他用户访问该对象但不阻止更改数据。在这种意义上，所有 SMI 表都具有永久的 Dirty Read 隔离级别。

2.2.3 系统监视接口表

sysmaster 数据库包含了许多表，您可以用这些表来维护系统。

提示： 对于每一个系统监视接口（SMI）表都有一个命名相同的视图与之对应。为了获得更好的结果，应查询与表相关联的视图，而不是直接查询基础表。

sysmaster 数据库中的许多其他表是系统监视接口的一部分但未加记载。其模式和列内容可以随版本而更改。flags_text 表包含更多的行。要查看新行，必须首先断开，然后重新创建 sysmaster 数据库。

下表列出了 SMI 表。

表	描述	参考
sysadtinfo	审计配置信息	sysadtinfo
sysaudit	审计事件掩码	sysaudit
syscheckpoint	Checkpoint 信息	syscheckpoint
syschkio	Chunk I/O 统计信息	syschkio
syschunks	Chunk 信息	syschunks
syscluster	高可用性集群信息	syscluster
syscmsmsla	连接管理器信息	syscmsmsla
syscmsmtab	连接管理器信息	syscmsmtab
syscmsmunit	每个连接管理器单元在连接管理器配置文件中的信息	syscmsmunit
syscompdicts_full	压缩字典信息	syscompdicts_full
sysconfig	配置信息	sysconfig
sysdatabases	Database 信息	sysdatabases
sysdbslocale	本地语言环境信息	sysdbslocale
sysdbspaces	Dbspace 信息	sysdbspaces
sysdri	数据复制信息	sysdri
sysdual	是单行表	sysdual
sysenv	联机服务器的启动环境	sysenv
sysenvses	会话级别的环境变量	sysenvses
sysextents	Extent 分配信息	sysextents

表	描述	参考
sysextspaces	外部空间信息	sysextspaces
sysha_lagtime	辅助服务器等待时间的统计信息	sysha_lagtime
sysha_type	关于已连接服务器的信息	sysha_type
sysha_workload	辅助服务器工作负载的统计信息	sysha_workload
sysipl	索引页日志记录信息	sysipl
syslocks	活动锁的信息	syslocks
syslogs	逻辑日志文件信息	syslogs
syslogfil	系统日志文件信息	syslogfil
sysmgminfo	内存分配管理器/并行数据查询信息	sysmgminfo
sysnetclienttype	客户机类型网络活动	sysnetclienttype
sysnetglobal	全局网络信息	sysnetglobal
sysnetworkio	网络 I/O	sysnetworkio
sysonlinelog	联机日志信息	sysonlinelog
sysprofile	系统概要文件信息	sysprofile
sysproxyagents	关于代理线程的信息	sysproxyagents
sysproxydistributors	代理分发器信息	sysproxydistributors
sysproxysessions	关于用于可更改辅助服务器的会话信息	sysproxysessions
sysproxytxnops	通过代理分发服务器运行事务的信息	sysproxytxnops
sysproxytxns	现在所有的通过代理分发服务器运行的事务的信息	sysproxytxns
sysptprof	表信息	sysptprof
sysrepevtreg	发布到连接管理器和 GBase OpenAdmin Tool (OAT) for GBase 8s 的事件	sysrepevtreg
sysrepstats	发布到连接管理器和 OAT 的事件	sysrepstats
sysrsslog	RS 辅助服务器的信息	sysrsslog
sysscblst	用户的内存	sysscblst
syssesprof	各种用户操作计数	syssesprof
syssesappinfo	分布式关系数据库体系结构™ (DRDA®) 客户端会话信息	syssesappinfo
syssessions	对每个已连接用户的描述	syssessions
syssmx	SMX （服务器多路复用器组） 的连接信息	syssmx

表	描述	参考
syssmxses	SMX （服务器多路复用器组）的会话信息	syssmxses
syssqexplain	SET EXPLAIN 声明的 SQL 语句信息	syssqexplain
syssqltrace	SQL 语句信息	syssqltrace
syssqltrace_info	SQL 概要文件跟踪系统信息	syssqltrace_info
syssqltrace_iter	SQL 语句迭代器	syssqltrace_iter
syssrcrss	RS 辅助服务器的统计信息	syssrcrss
syssrcsds	SD 辅助服务器的统计信息	syssrcsds
systabnames	表空间 tblspace 的数据库、所有者和表名称	systabnames
systabpaghdrs	页面页眉	None
systhreads	等待统计信息	systhreads
systrgrss	RS 辅助服务器的统计信息	systrgrss
systrgsds	SD 辅助服务器的统计信息	systrgsds
sysvpprof	每个虚拟处理器所使用的用户和系统	sysvpprof

sysutils

ON-Bar 使用 sysutils 数据库中的下表。有关更多信息，请参阅 GBase 8s 备份与恢复指南.

表 描述

bar_action

列出除去在冷备份过程中的之外，对对象尝试过的所有备份和恢复操作。使用该表中的信息跟踪备份和恢复历史的记录。

bar_instance

为每个成功备份向此表写入一条记录。ON-Bar 以后可能会将该信息用于恢复操作。

bar_object

描述每个备份对象。该表提供至少作过一次备份尝试的每个数据库服务器的所有存储空间和逻辑日志的列表。

bar_server

列出处于安装状态的数据库服务器。该表用于确保备份对象在恢复过程中返回到正确的位置。

sysadtinfo

sysadtinfo 表包含有关数据库服务器审计配置的信息。有关更多信息，请参阅 GBase 8s 安全指南 。要从 sysadtinfo 表中检索信息，必须是用户 gbasedbt 或用户 root（在 UNIX™ 上）或 Gbasedbt-Admin 组的成员（在 Windows™ 上）。

列	类型	描述
adtmode	integer	控制审计的级别
adterr	integer	当数据库服务器在写入审计记录时发生错误时，指定对错误采取的操作
adtsize	integer	审计文件的最大大小
adtpath	char(256)	审计文件所写入的目录
adtfile	integer	审计文件的数量

sysaudit

对于每个已定义的审计掩码（即，对于每个用户名）， sysaudit 表包含代表生成审计记录的数据库事件的标志。 success 和 failure 列代表组合审计掩码的位掩码。如果一个位在 success 和 failure 列都进行了设置，那么相应的事件生成一条有关该事件是否成功的审计记录。

要从 sysaudit 表中检索信息，必须是用户 gbasedbt 或用户 root （在 UNIX™ 上）或 Gbasedbt-Admin 组的成员（在 Windows™ 上）。

使用 gaudit 实用程序列出或修改审计掩码。有关 gaudit 和审计的信息，请参阅 GBase 8s 安全指南 。

列	类型	描述
username	char(32)	掩码的名称
succ1	integer	审计成功掩码的位掩码
succ2	integer	审计成功掩码的位掩码
succ3	integer	审计成功掩码的位掩码
succ4	integer	审计成功掩码的位掩码
succ5	integer	审计成功掩码的位掩码
fail1	integer	审计失败掩码的位掩码

列	类型	描述
fail2	integer	审计失败掩码的位掩码
fail3	integer	审计失败掩码的位掩码
fail4	integer	审计失败掩码的位掩码
fail5	integer	审计失败掩码的位掩码

syschkio

syschkio 系统监视接口表提供数据库服务器所管理的个别 chunk 的 I/O 统计信息。

列	类型	描述
chunknum	smallint	Chunk 编号
reads	integer	物理读取数
pagesread	integer	读取的页数
writes	integer	物理写入数
pageswritten	integer	写入的页数
mreads	integer	物理读取（镜像）数
mpagesread	integer	读取（镜像）的页数
mwrites	integer	物理写入（镜像）数
mpageswritten	integer	写入（镜像）的页数

syscheckpoint

syscheckpoint 表提供有关检查点的信息和其统计信息。

列	类型	描述
interval	integer	服务器启动以来的检查点数
type	char(12)	Hard 或 Interval
caller	char(10)	检查点的调用者
clock_time	integer	检查点的发生时间

列	类型	描述
crit_time	float	等待释放临界区所用时间
flush_time	float	将页清仓到磁盘所用的时间
cp_time	float	检查点暂挂到检查点完成的持续时间
n_dirty_buffs	integer	脏缓冲区的数量
plogs_per_sec	integer	1秒钟处理的物理日志页数
llogs_per_sec	integer	1秒钟处理的逻辑日志页数
dskflush_per_sec	integer	1秒钟清空的缓冲池页面数
ckpt_logid	integer	检查点的逻辑日志的唯一标识
ckpt_logpos	integer	检查点
physused	integer	物理日志中所用的页数
logused	integer	逻辑日志中所用的页数
n_crit_waits	integer	必须等待进入临界区的用户数
tot_crit_wait	float	为检查点临界区块中等待的所有用户所用的等待持续时间
longest_crit_wait	float	最久的临界区等待
block_time	float	阻塞系统的检查点的持续时间

syschunks

syschunks 表描述数据库服务器所管理的每个 chunk 。

在 flags 和 mflags 列中，每个位置代表一个单独的位置。这样，如果这些值是使用 HEX 函数返回的，那么读取 flags 和 mflags 列中的 值会更容易。

列	类型	描述
chknum	smallint	Chunk 编号
dbsnum	smallint	Dbspace 编号
nxchknum	smallint	dbspace 中下一个 chunk 的编号

列	类型	描述
chksize	integer	chunk 中页的数量（在系统默认的页大小的单位中）
offset	integer	在设备或路径中 chunk 的页偏移量
pagesize	integer	页大小（in bytes）
nfree	integer	chunk 中的空闲页数 可用空间的容量取决于空间的类型 ● dbspace = 可用页面的数量乘以系统缺省页面的大小 2KB 或 4KB 。 ● blobspace = 可用页面的数量乘以 blobpages 的大小。 ● sbspace = 可用页面的数量乘以 sbpage 的大小 （和系统缺省页面一样大）。
is_offline	integer	1 = chunk 处于脱机模式 0 = chunk 处于联机模式
is_recovering	integer	1 = chunk 正在恢复 0 = chunk 没在恢复
is_blobchunk	integer	1 = chunk 在 blobspace 中 0 = chunk 不在 blobspace 中
is_sbchunk	integer	1 = chunk 在 sbspace中 0 = chunk 不在 sbspace 中
is_inconsistent	integer	1 = chunk 正在进行逻辑恢复 0 = chunk 不在逻辑恢复

列	类型	描述
is_extendable	integer	1 = chunk 是扩展的 0 = chunk 是不可扩展的
flags	smallint	flag 有以下数值和十六进制值和含义： ● 16 (0x0010) = Chunk 是已镜像的 chunk ● 32 (0x0020) = Chunk 处于脱机模式 ● 64 (0x0040) = Chunk 处于联机模式 ● 128 (0x0080) = Chunk 处于恢复模式 ● 256 (0x0100) = Chunk 刚好镜像过 ● 512 (0x0200) = Chunk 是 blobspace 的一部分 ● 1024 (0x0400) = 正在删除 Chunk ● 4096 (0x1000) = Chunk 是不一致的 ● 8192 (0x2000) = Chunk 是可扩展的 ● 16384 (0x4000) = Chunk 在前滚过程中被添加 ● 32768 (0x8000) = Chunk 被重命名 ● 65536 (0x10000) = Chunk 使用大 chunk 页面页眉 ● 131072 (0x20000) = Chunk 有一 tblspace tblspace长度 ● 262144 (0x40000) = 自 chunk 被初始化（主要供内部使用），没有检查点被完成

列	类型	描述
fname	char(256)	该 chunk 的文件或设备的路径名
mdsize	integer	在页中元数据领域的 chunk 的大小属于一 sbspace 。 如果 chunk 不是 sbspace 的一部分，其值为 -1。
mfname	char(256)	已镜像的 chunk （如果有）的文件或设备的路径名
moffset	integer	已镜像的 chunk 的页偏移量
mis_offline	integer	1 = 镜像是脱机的 0 = 镜像是联机的
mis_recovering	integer	1 = 镜像正被恢复 0 = 镜像没有在被恢复
mflags	smallint	已镜像 chunk 标志；值和含义与 flags 列相同
udfree	integer	chunk 在用户数据区域中的页可用空间属于一个 sbspace 。如果不属于，其值为 -1 。
udsize	integer	chunk 在用户数据区域中的页的大小属于 sbspace 。 如果不属于，其值为 -1 。

sysckptinfo

sysckptinfo 系统监视接口表提供有关前 20 个检查点的历史记录信息。

列	类型	描述
ckpt_status	int	0x0011 = 检查点被阻止，由于物理日志耗尽了资源

列	类型	描述
		0x0021 = 检查点被阻止，由于逻辑日志耗尽了资源 0x0041 = 检查点被阻止，由于事务运行时间过长 0x1000 = 物理日志太小 0x2000 = 逻辑日志太小 0x4000 = 为 RTO 准备的物理日志太小
plogs_per_S	int	物理日志活动的平均率
llogs_per_S	int	逻辑日志活动的平均率
dskF_per_S	int	页面缓冲到磁盘的平均率
longest_dskF	int	在 checkpoint 处理期间，其刷新到磁盘缓冲池的最长持续时间
dirty_pgs_S	int	页被修改的平均率
sug_plog_sz	int	建议的物理日志大小
sug_llog_sz	int	建议的逻辑日志空间大小
ras_plog_sp	int	恢复物理日志的最快速率
ras_llog_sp	int	恢复重新加载逻辑日志的最快速率
boottime	int	服务器启动共享内存和打开 chunk 所用的时间
auto-ckpts	int	1 = on , 0 = off
auto_lru	int	1 = on , 0 = off
cur_intvl	int	现有的 checkpoint 时间间隔
auto_aiovp	int	1 = on , 0 = off

syscluster

syscluster 系统目录表存储了在高可用集群下服务器的信息。syscluster表有以下列。

列	类型	描述

列	类型	描述
name	CHAR(128)	主服务器的名称
role	CHAR(1)	用于区分服务器是主服务器还是辅助服务器的标志
syncmode	CHAR(8)	主服务器和辅助服务器之间的同步方式：sync 或 async
nodetype	CHAR(8)	服务器的类型： HD 、RSS 或 SDS
supports_updates	CHAR(1)	标示客户端是否能在辅助服务器上进行修改、插入、删除操作(由 UPDATABLE_SECONDARY 配置参数所指定).
server_status	CHAR(32)	标示辅助服务器的状态
connection_status	CHAR(32)	标示辅助服务器的连接状态
delayed_apply	INTEGER	标示辅助服务器在应用日志时是否需要等待一定的时间 （由 DELAY_APPLY 配置参数指定）
stop_apply	CHAR(24)	标示辅助服务器是否阻止其应用从主服务器上接收的日志（由 STOP_APPLY 配置参数指定）
logid_sent	INTEGER	标示最新的日志页由主服务器发送到辅助服务器的日志 ID
logpage_sent	INTEGER	标示最新的日志页由主服务器发送到辅助服务器的页数
logid_acked	INTEGER	标示辅助服务器确认的最新日志页的日志 ID
logpage_acked	INTEGER	标示辅助服务器确认的最新日志页的页数
ack_time	DATETIME YEAR TO SECOND	标示最新确认日志的时间
sdscycle	INTEGER	标示主服务器先行循环次数。内部由 GBase 支持监视主服务器和辅助服务器的协调

列	类型	描述
sdscycle_acked	INTEGER	标示辅助服务器确认共享磁盘的循环次数。内部由 GBase 支持监视主服务器和辅助服务器的协调

syscmsm

syscmsm 表是 syscmsmtab 和 syscmsmsla 表的视图。它包含连接管理器服务等级协议（ SLA ）的信息。该表每五秒更新一次。

列	类型	描述
sid	integer	连接管理器会话 ID
name	char(128)	连接管理器名称
host	char(256)	主机名
unit	char(128)	单位名
type	char(128)	单位类型
servers	char(1024)	单位服务器
foc	char(128)	故障切换配置 (FOC)
flag	integer	仲裁标志 1 = 连接管理器仲裁是激活的 0 = 连接管理器仲裁未被激活
sla_name	char(128)	SLA 名称
sla_define	char(128)	SLA 定义
connections	integer	通过连接管理器的连接数

syscmsmsla

syscmsmsla 表包含连接管理器服务等级协议（ SLA ）的信息。该表每五秒更新一次。

列	类型	描述
address	int8	CMSLA 内部地址

列	类型	描述
sid	integer	连接管理器会话 ID
sla_name	char(128)	SLA 名称
sla_define	char(128)	SLA 定义
connections	integer	通过连接管理器的连接数

syscmsmtab

syscmsmtab 包含了连接管理器的信息。

列	类型	描述
address	int8	连接管理器的内部地址
sid	integer	连接管理器的会话 ID
name	char(128)	连接管理器名称
host	char(256)	主机名称
flag	integer	仲裁标志 1 = 连接管理器仲裁是激活的 0 = 连接管理器仲裁未被激活

syscmsmunit

syscmsmunit 表包含了每个连接管理器单元在连接管理器配置文件中的信息。

列	类型	描述
address	int8	连接管理器内部地址
sid	integer	连接管理器会话 ID
unit	char(128)	单位名称
type	char(128)	单位类型
servers	char(1024)	单位服务器
foc	char(128)	故障切换配置 （ FOC ）

列	类型	描述
flag	integer	仲裁标志 1 = 连接管理器仲裁是激活的 0 = 连接管理器仲裁未被激活

syscompdicts_full

syscompdicts_full 表和 syscompdicts 视图提供了所有压缩字典的信息。该表和视图的唯一区别是，出于安全性的目的，视图不包含 dict_dictionary 列。

只有用户 gbasedbt 可以检索 syscompdicts_full 表的信息。syscompdicts 视图没有限制用户 gbasedbt 。

下表显示了 syscompdicts_full 表和 syscompdicts 视图为每个压缩字典提供的信息。

列	类型	描述
dict_partnum	integer	压缩字典所适用的分区号
dict_code_version	integer	创建压缩字典代码的版本 1 是一个版本
dict_dbsnum	integer	字典中驻留的 dbspace 的数目
dict_create_timestamp	integer	时间戳显示字典被创建的时间
dict_create_loguniqid	integer	字典被创建时已激活的逻辑日志的唯一 ID
dict_create_logpos	integer	字典被创建时在逻辑日志中的位置
dict_drop_timestamp	integer	时间戳显示字典被删除的时间
dict_drop_loguniqid	integer	字典被删除时已激活的逻辑日志的唯一 ID
dict_drop_logpos	integer	字典被删除时在逻辑日志中的位置
dict_dictionary	byte	压缩字典的二进制对象 syscompdicts 视图中没有该列

syscompdicts 信息样本

syscompdicts 视图的一列包含的信息如下所示：

```
dict_partnum         1048939
dict_code_version    1
dict_dbsnum          1
dict_create_times+   1231357656
dict_create_logun+   11
dict_create_logpos   1695768
dict_drop_timesta+   0
dict_drop_loguniq+   0
dict_drop_logpos     0
```

使用 UNLOAD 语句可将该压缩字典导出到一个压缩字典文件中，该语句示例如下：

```
UNLOAD TO 'compression_dictionary_file'
        SELECT * FROM sysmaster:syscompdicts_full;
```

sysconfig

sysconfig 表描述了参数的有效、原始和缺省值。有关 ONCONFIG 文件以及配置参数的更多信息，请参阅 数据库配置参数。

列	类型	描述
cf_id	integer	唯一的数字标识
cf_name	char(128)	配置参数名称
cf_flags	integer	保留，供将来使用
cf_original	char(256)	启动时 ONCONFIG 文件中的值
cf_effective	char(256)	当前正在使用的值
cf_default	char(256)	如果 ONCONFIG 文件中没有指定任何值，那么为数据库服务器所提供的值

sysdatabases

sysdatabases 视图描述数据库服务器所管理的每个数据库。

列	类型	描述
name	char(128)	数据库名称

列	类型	描述	
partnum	integer	数据库的 systables 表的分区编号（tblspace ID）	
owner	char(32)	数据库创建者的用户 ID	
created	date	创建日期	
is_logging	integer	如果日志记录是活动的，那么为 1 ；如果不是，那么为 0	
is_buff_log	integer	如果日志记录已缓冲，那么为 1 ；如果不是，那么为 0	
is_ansi	integer	如果 ANSI/ISO 兼容，那么为 1 ；如果不，那么为 0	
is_nls	integer	如果是启用GLS的，那么为 1 ；如果不，那么为 0	
is_case_insens	integer	如果 NCHAR 和 NVARCHAR 列区分大小写，那么为 1 ；如果不，那么为 0	
flags	smallint	日志记录标志 （十六进制值）	
		0	没有日志记录
		1	未缓冲的日志记录
		2	已缓冲的日志记录
		4	兼容 ANSI/ISO 的数据库
		8	只读数据库
		10	GLS 数据库
		20	忽略对 syscdr 数据库日志记录方式的检查
		100	已将状态更改为已缓冲的日志记录
		200	已将状态更改为未缓冲昂的日志记录
		400	已将状态更改为兼容 ANSI/ISO 的日志记录
		800	数据库日志记录已关闭
		1000	启用了长 ID 支持

sysdbslocale

sysdbslocale 表列出了数据库服务器所管理的每个数据库的语言环境。

列	类型	描述
dbs_dbsname	char(128)	数据库名称
dbs_collate	char(32)	数据库的语言环境

sysdbspaces

sysdbspaces 表描述了数据库服务器所管理的每个 dbspaces 。

在 flags 列中，每个位位置代表一个单独的标志。这样，如果值是使用 HEX 函数返回的，那么读取 flags 列中的值会更容易。

列	类型	描述		
dbsnum	smallint	Dbspace 编号		
name	char(128)	Dbspace 名称		
owner	char(32)	dbspace 所有者的用户 ID		
fchunk	smallint	dbspace 中第一个 chunk 的编号		
nchunks	smallint	dbspace 中 chunk 的数量		
create_size	decimal	可以为此空间使用存储池所创建的 chunk 的最小容量		
extend_size	decimal	存储空间中可扩展的 chunk 的最小容量，可以是手动或自动方式		
pagesize	integer	页大小		
is_mirrored	integer	如果 dbspace 已镜像，那么为 1 ；如果不是，那么为 0		
is_blobspace	integer	如果 dbspace 是 blobspace 那么为 1 ； 如果不是,那么为 0		
is_sbspace	integer	如果 dbspace 是 sbspace，那么为 1 ； 如果不是,那么为 0		
is_temp	integer	如果 dbspace 是临时 dbspace ，那么为1；若果不是，那么为 0		
flags	smallint	标志	十六进制值	含义
		1	0x0001	Dbspace 没有镜像
		2	0x0002	Dbspace 使用镜像

列	类型	描述		
		4	0x0004	Dbspace 镜像已被禁用
		8	0x0008	Dbspace 最近镜像过
		16	0x0010	Space 是 blobspace
		32	0x0020	Blobspace 在可移动介质上
		128	0x0080	Blobspace 已被删除
		512	0x0200	Space 正在恢复
		1024	0x0400	Space 已被物理移除
		2048	0x0800	Space 正在逻辑恢复
		32768	0x8000	Space 是 sbspace

sysdri

sysdri 表提供关于数据库服务器的高可用性数据复制状态的信息。

列	类型	描述
type	char(50)	高可用性数据复制类型值: ● 主 ● 辅助 ● 标准 ● 未初始化
state	char(50)	高可用性数据复制的状态值： ● 关闭 ● 开启 ● 正在连接 ● 失败 ● 只读
name	char(128)	高可用性数据复制对中的其他数据库服务器的名称
intvl	integer	高可用性数据复制时间间隔

列	类型	描述
timeout	integer	该数据服务器的高可用性数据复制超时值
lostfound	char(256)	丢失和找到文件的路径名

sysdual

sysdual 表仅返回 1 列 1 行。

列	类型	描述
dummy	char(1)	返回 "X" 的 Dummy 列

sysenv

sysenv 表显示数据库服务器的启动环境设置。

列	类型	描述
env_id	integer	标识变量编号
env_name	char(128)	环境变量名称
env_value	char(512)	环境变量值

sysenvses

sysenvses 表显示会话级别的环境变量。

列	类型	描述
envses_sid	integer	会话 ID
envses_id	integer	标识变量号
envses_name	char(128)	会话环境变量名称
envses_value	char(512)	会话环境变量值

sysextents

sysextents 表提供有关扩展数据块分配的信息。

列	类型	描述
dbsname	char(128)	数据库名

列	类型	描述
tabname	char(128)	表名
chunk	integer	Chunk 编号
offset	integer	Chunk 中扩展数据块开始处的页数
size	integer	扩展数据块的大小，在页中

sysextspaces

sysextspaces 表提供关于外部空间的信息。 id 列和 name 列的索引只允许唯一的值。

列	类型	描述
id	integer	外部空间 ID
name	char(128)	外部空间名称
owner	char(32)	外部空间所有者
flags	integer	外部空间标识 （保留供将来使用）
refcnt	integer	外部空间引用计数
locsize	integer	外部空间位置大小（字节）
location	char (256)	外部空间的位置

sysfeatures

sysfeatures 视图提供 GBase 8s 数据库服务器实例的各种功能信息。sysfeatures 视图是从一个名为 syslicenseinfo 的外部表创建而来的，该表永久地存储在磁盘上。当数据库服务器实例初始化时，该表会预分配固定的可以追踪 260 周数据大小的容量。这些数据每五年更新一次。

每隔 15 分钟会抽样度量标准，并且只存储某一周的最高值。表中每一行包含的数据仅代表特定周。

列	类型	描述
week	smallint	记录信息的周数
year	smallint	记录信息的年份

列	类型	描述
version	char(12)	GBase 8s 服务器版本
max_cpu_vps	smallint	CPU 虚拟处理器的最大数量
max_vps	smallint	虚拟处理的最大数量
max_conns	integer	在独立或高可用性的群集主服务器实例上的并发物理连接的最大数目
max_sec_conns	integer	在 HDR 辅助服务器或 RS 辅助服务器实例上的并发物理连接的最大数目
max_sds_clones	smallint	SD 辅助服务器实例连接主服务器的最大数目
max_rss_clones	smallint	RS 辅助服务器实例连接主服务器的最大数目
total_size	integer	在所有 chunk 中分配的最大磁盘空间 （以兆字节）
total_size_used	integer	在所有 chunk 中使用的最大的磁盘空间 （以兆字节）
max_memory	integer	在所有段中分配的最大内存（以兆字节）
max_memory _used	integer	在所有段中使用的最大内存（以兆字节）
is_primary	integer	标示在特定的周该服务器是否为主服务器：是则为 1 ，不是则为 0
is_secondary	integer	标示在特定的周该服务器是否为 HDR 辅助服务器：是则为 1 ，不是则为 0
is_sds	integer	标示在特定的周该服务器是否为 SD 辅助服务器：是则为 1 ，不是则为 0（未实施的；经常为 0 ）
is_rss	integer	标示在特定的周该服务器是否为 RS 辅助服务器：是则为 1 ，不是则为 0
is_er	integer	标示在特定的周该服务器是否为 Enterprise Replication 服务器：是则为 1 ，不是则为 0
is_pdq	integer	标示在特定的周服务器实例是否使用了 PDQ 功能：若使用则为 1 ，没有则为 0

sysha_lagtime

sysha_lagtime 表提供之前将日志记录应用到任何辅助节点上所需的时间。

sysha_lagtime 表包含最近为特定的辅助服务器执行的 20 个取样。

列	类型	描述
lt_secondary	CHAR(128)	辅助服务器的名称
lt_time_last_update	INTEGER	上次更新日志记录的时间
lt_lagtime_1	FLOAT	最近 5 秒间隔内应用日志记录所需的时间
lt_lagtime_2	FLOAT	上个 5 秒间隔内应用日志记录所需的时间
lt_lagtime_3	FLOAT	前 3 个 5 秒间隔内应用日志记录所需的时间
lt_lagtime_4	FLOAT	前 4 个 5 秒间隔内应用日志记录所需的时间
lt_lagtime_5	FLOAT	前 5 个 5 秒间隔内应用日志记录所需的时间
lt_lagtime_6	FLOAT	前 6 个 5 秒间隔内应用日志记录所需的时间
lt_lagtime_7	FLOAT	前 7 个 5 秒间隔内应用日志记录所需的时间
lt_lagtime_8	FLOAT	前 8 个 5 秒间隔内应用日志记录所需的时间
lt_lagtime_9	FLOAT	前 9 个 5 秒间隔内应用日志记录所需的时间
lt_lagtime_10	FLOAT	前 10 个 5 秒间隔内应用日志记录所需的时间
lt_lagtime_11	FLOAT	前 11 个 5 秒间隔内应用日志记录所需的时间
lt_lagtime_12	FLOAT	前 12 个 5 秒间隔内应用日志记录所需的时间
lt_lagtime_13	FLOAT	前 13 个 5 秒间隔内应用日志记录所需的时间
lt_lagtime_14	FLOAT	前 14 个 5 秒间隔内应用日志记录所需的时间
lt_lagtime_15	FLOAT	前 15 个 5 秒间隔内应用日志记录所需的时间
lt_lagtime_16	FLOAT	前 16 个 5 秒间隔内应用日志记录所需的时间
lt_lagtime_17	FLOAT	前 17 个 5 秒间隔内应用日志记录所需的时间
lt_lagtime_18	FLOAT	前 18 个 5 秒间隔内应用日志记录所需的时间

列	类型	描述
lt_lagtime_19	FLOAT	前 19 个 5 秒间隔内应用日志记录所需的时间
lt_lagtime_20	FLOAT	前 20 个 5 秒间隔内应用日志记录所需的时间

sysha_type

sysha_type 表是一个单行表，用来描述连接的服务器的类型。

列	类型	描述
ha_type	integer	服务器类型 （参见下表）
ha_primary	char(128)	服务器名称 （参见下表）

ha_type 的值	*ha_primary* 的值	描述
0	NULL	不是高可用性环境的组成部分
1	*<primary server name>*	主服务器
2	*<primary server name>*	HDR 辅助服务器
3	*<primary server name>*	SD 辅助服务器
4	*<primary server name>*	RS 辅助服务器

sysha_workload

sysha_workload 表包含了每台服务器上工作负载的统计信息。

列	类型	描述
wl_secondary	char(128)	辅助服务器的名称
wl_time_last_update	integer	上次更新工作负载的时间
wl_type	char(12)	该行包含了就绪队列大小、用户 CPU 时间以及系统 CPU 时间
wl_workload_1	float	最近的工作负载活动
wl_workload_2	float	前两次的工作负载活动

列	类型	描述
wl_workload_3	float	前三次的工作负载活动
wl_workload_4	float	前四次的工作负载活动
wl_workload_5	float	前五次的工作负载活动
wl_workload_6	float	前六次的工作负载活动
wl_workload_7	float	前七次的工作负载活动
wl_workload_8	float	前八次的工作负载活动
wl_workload_9	float	前九次的工作负载活动
wl_workload_10	float	前十次的工作负载活动
wl_workload_11	float	前十一次的工作负载活动
wl_workload_12	float	前十二次的工作负载活动
wl_workload_13	float	前十三次的工作负载活动
wl_workload_14	float	前十四次的工作负载活动
wl_workload_15	float	前十五次的工作负载活动
wl_workload_16	float	前十六次的工作负载活动
wl_workload_17	float	前十七次的工作负载活动
wl_workload_18	float	前十八次的工作负载活动
wl_workload_19	float	前十九次的工作负载活动
wl_workload_20	float	前二十次的工作负载活动

sysipl

sysipl 表提供了关于主服务器上索引页日志记录的状态信息。

列	类型	描述
ipl_status	integer	索引页日志记录状态
ipl_time	integer	启用索引页日志记录的时间

syslocks

syslocks 表提供了有关数据库服务器中所有当前活动锁的信息 。

列	类型	描述	
dbsname	char(128)	数据库名称	
tabname	char(128)	表名	
rowidlk	integer	实际 rowid （如果它是索引键锁）	
keynum	smallint	索引键锁的键编号	
type	char(4)	锁的类型	
		B	字节锁
		IS	意向共享锁
		S	共享锁
		XS	由可重复阅读器持有的共享锁值
		U	更新锁
		IX	意向互斥锁
		SIX	共享的意向互斥锁
		X	互斥锁
		XR	由可重复阅读器持有的互斥键值
owner	integer	锁所有者的会话 ID	
waiter	integer	等待锁的用户的会话 ID 。如果一个以上用户正在等待，那么只有第一个会话 ID 出现	

syslogs

syslogs 表提供有关逻辑日志文件中使用的空间的信息。在 flags 列中，每个位位置代表一个单独的标志。例如：对于日志文件， flags 列可能具有当前日志文件和临时日志文件所设置的标志。这样，如果值是使用 HEX 函数返回的，那么读取 flags 列中的值可能会更容易。

列	类型	描述		
number	smallint	逻辑日志文件编号		
uniqid	integer	日志文件 ID		
size	integer	日志文件中的页数		
used	integer	日志文件中已用的页数		
is_used	integer	如果文件被使用，其值为 1 ，否则为 0		
is_current	integer	如果文是当前文件，其值为 1 ，否则为 0		
is_backed_up	integer	如果文件已备份过，其值为 1 ，否则为 0		
is_new	integer	如果自上一次 0 级 dbspace 备份以来添加过该日志，其值为 1 ，否则为 0		
is_archived	integer	如果文件已置于备份磁带上，其值为 1 ，否则为 0		
is_temp	integer	如果文件已标识为临时日志文件，其值为 1 ，否则为 0		
flags	smallint	标志	十六进制	含义
		1	0x01	日志文件在使用中
		2	0x02	文件是当前的日志文件
		4	0x04	日志文件已进行了备份
		8	0x08	文件是新添加的日志文件
		16	0x10	日志文件已写入 dbspace 备份介质
		32	0x20	日志是临时日志文件

syslogfil

syslogfil 表提供有关逻辑日志文件的信息。

列	类型	描述
address	int8	逻辑文件结构的内存地址
number	small	逻辑文件编号

列	类型	描述
	integer	
flags	integer	要了解该值及值含义的信息，请参阅下面 **标志值** 部分
fillstamp	integer	填充日志文件的内部时间戳
filltime	integer	填充日志文件的 UNIX™ 时间
uniqid	integer	日志文件的唯一 ID
chunk	integer	包含日志文件的 chunk 的编号
offset	integer	chunk 中日志文件开始位置的页偏移量
size	integer	日志文件的总页数
used	integer	日志文件中已用的页数

标志值

标志值对应于 gstat -l 命令的标志值。

十六进制	Gstat -l 标志值	含义
0x1	U	文件在使用中
0x2	C	文件是当前的日志文件
0x4	B	日志文件已进行了备份
0x8	A	文件是新添加的日志文件
0x20	None	临时日志文件
0x40	D	文件存档后将要被删除的日志文件
0x4000	L	包含最近写入检查点的日志文件

sysmgminfo

sysmgminfo 表提供内存分配管理器（ MGM ）和并行数据查询信息（ PDQ ）的概述。

列	类型	描述
max_query	integer	允许的活动查询的最大数量

列	类型	描述
total_mem	integer	MGM 内存总量
avail_mem	integer	MGM 空闲内存量
total_seq	integer	顺序扫描总数
avail_seq	integer	未使用的顺序扫描数
active	integer	活动 MGM 查询的数量
ready	integer	就绪 MGM 查询的数量
min_free_mem	integer	最低 MGM 空闲内存量
avg_free_mem	float	平均 MGM 空闲内存量
std_free_mem	float	标准 MGM 空闲内存量
min_free_seq	integer	最低空闲 MGM 顺序扫描数
avg_free_seq	float	平均空闲 MGM 顺序扫描数
std_free seq	float	标准空闲 MGM 顺序扫描数
max_active	integer	活动 MGM SQL 操作最大数量
cnt_active	integer	活动 MGM SQL 操作数量
avg_active	float	活动 MGM SQL 操作平均数量
std_active	float	活动 MGM SQL 操作标准数量
max_ready	integer	就绪 MGM SQL 操作最大数量
cnt_ready	integer	就绪 MGM SQL 操作的数量
avg_ready	float	就绪 MGM SQL 操作平均数量
std_ready	float	就绪 MGM SQL 操作标准数量

sysnetclienttype

sysnetclienttype 表提供各客户机类型的网络活动概述。

列	类型	描述

列	类型	描述
nc_cons_allowed	integer	是否允许连接
nc_accepted	integer	接收的连接数
nc_rejected	integer	拒绝的网络连接数
nc_reads	int8	针对该客户机类型的网络读取数
nc_writes	int8	针对该客户机类型的网络写入数
nc_name	char(18)	客户机类型的名称

sysnetglobal

sysnetglobal 表提供有关系统网络概述。

列	类型	描述
ng_reads	int8	网络读取数
ng_writes	int8	网络写入数
ng_connects	int8	网络连接数
ng_his_read_count	int8	已断开 ng_his_read_bytes 的用户进行的网络读取数
ng_his_read_bytes	int8	已断开连接的用户转发给服务器的数据
ng_his_write_count	int8	已断开连接的用户写入的网络数
ng_his_write_bytes	int8	已断开连接的用户转发给客户机的数据
ng_num_netscbs	integer	网络订户数
ng_max_netscbs	integer	网络订户的最大数量
ng_free_thres	integer	缓冲表中已释放缓冲区最大数量的阈值
ng_free_cnt	integer	达到 ng_free_thres 限制的次数
ng_wait_thres	integer	可以与缓冲表建立缓冲区的最大数量的阈值
ng_wait_cnt	integer	达到 ng_wait_thres 限制的次数

列	类型	描述
ng_pvt_thres	integer	私有缓冲队列中的最大数量的已释放缓冲区的阈值
ng_netbuf_size	integer	传输网络缓冲区的大小
ng_buf_alloc	integer	已分配的网络缓冲区数量
ng_buf_alloc_max	integer	已分配的网络缓冲区的最大值
ng_netscb_id	integer	下个 netscb ID

sysnetworkio

sysnetworkio 表提供系统网络概述。

列	类型	描述
net_id	integer	Netscb ID
sid	integer	会话 ID
net_netscb	int8	Netscb 端口
net_client_type	integer	客户机类型 Int
net_client_name	char(12)	客户机协议名称
net_read_cnt	int8	网络读取数
net_write_cnt	int8	网络写入数
net_open_time	integer	此会话的连接时间
net_last_read	integer	上次从该网络读取的时间
net_last_write	integer	上次从该网络写入的时间
net_stage	integer	连接 / 断开 / 接收
net_options	integer	来自 SQL 主机的选项
net_protocol	integer	协议
net_type	char(10)	网络协议类型
net_server_fd	integer	服务器 fd

列	类型	描述
net_poll_thread	integer	轮询线程

sysonlinelog

sysonlinelog 表提供 online.log 文件中存储的信息的视图。

列	类型	描述
offset	int8	文件偏移量
next_offset	int8	到下一个消息的偏移量
line	char(4096)	来自文件的一行文本

sysprofile

sysprofile 表包含有关数据库服务器的概要文件信息。

列	类型	描述
name	char(13)	概要文件事件的名称（参阅下表已获得可能事件的列表）
value	integer	概要文件事件的值 （参阅下表已获得可能事件的列表）

下表列出了事件以及相应的值，它们一起组成了 sysprofile 表。

sysprofile 中的概要文件事件	描述
dskreads	从磁盘的实际读取数
bufreads	从共享内存的实际读取数
dskwrites	向磁盘的实际写入数
bufwrites	向共享内存的写入数
isamtot	调用总数
isopens	isopen 调用
isstarts	isstart 调用
isreads	isread 调用

sysprofile 中的概要文件事件	描述
iswrites	iswrite 调用
isrewrites	isrewrite 调用
isdeletes	isdelete 调用
iscommits	iscommit 调用
isrollbacks	isrollback 调用
ovlock	溢出锁表
ovuser	溢出用户表
ovtrans	溢出事务表
latchwts	锁存器请求等待数
bufwts	锁缓冲数
lockreqs	锁请求数
lockwts	锁等待数
ckptwts	检查点等待数
deadlks	死锁数
lktouts	死锁超时数
numckpts	检查点数
plgpagewrites	已写入的物理日志页数
plgwrites	物理日志写入数
llgrecs	逻辑日志记录数
llgpagewrites	逻辑日志写入数
llgwrites	已写入的逻辑日志页数
pagreads	页读取数

sysprofile 中的概要文件事件	描述
pagwrites	页写入数
flushes	缓冲池清仓数
compress	页压缩数
fgwrites	前台写入数
lruwrites	最近最少使用（ LRU ）写入数
chunkwrites	检查点过程中的写入数
btradata	通过索引页节点读取的预先读取数据页数
btraidx	通过索引分支或根节点读取的预先读取数据页数
dpra	以预先读取功能读入内存的数据页数
rapgs_used	用户已用的预先读取数据页数
seqscans	顺序扫描数
totalsorts	排序总数
memsorts	适合内存的排序数
disksorts	不适合内存的排序数
maxsortspace	排序所使用的最大磁盘空间

sysproxyagents

sysproxyagents 表包含关于所有代理服务器线程的信息。代理服务器线程在主服务器上运行并且接受来自辅助服务器处理的 DML 操作。主服务器也有一个用于处理辅助服务器更新的代理服务器。辅助服务器决定代理服务器创建在辅助服务器的 ONCONFIG 文件上，基于 UPDATABLE_SECONDARY 设置的实例的数量。

列	类型	描述
tid	integer	运行在主服务器上的代理线程事务的 ID 。该 ID 是代理服务器在处理来自辅助服务器会话工作时被创建的

列	类型	描述
flags	integer	代理服务器线程的标识
proxy_id	integer	代理服务器代表当前正在执行的代理线程 ID（ TID ）
source_session_id	integer	用户在辅助服务器上的会话 ID
proxy_txn_id	integer	当前事物的编号。这些编号对于代理服务器是唯一的
current_seq	integer	在当前事务上的当前操作的顺序编号
sqlerrno	integer	任何 SQL 错误（或者 0 成功）的错误编号
iserrno	integer	任何 ISAM/RSAM 错误（或者 0 成功）的错误编号

sysproxydistributors

sysproxydistributors 表包含关于代理分发器的信息。

在主服务器上，该表包含所有高可用性集群下的代理分发器的信息。在辅助服务器上，该表仅包含那些被分配来处理更新到辅助服务器的代理分发器的信息。

列	类型	描述
node_name	char	能被主服务器识别的辅助服务器的名称（例如：HA_ALIAS 等等）
proxy_id	integer	代理分发器的 ID 。这些 ID 在高可用性集群上是唯一的。
transaction_count	integer	代理分发器当前处理的事务的数量
hot_row_total	integer	由代理服务器处理过的 hot row 总数。Hot row是在辅助分发器上被客户机修改过多次的行。如果最近来自不同会话的更新操作不在辅助服务器上重演，当行更新多次时，辅助服务器会读取之前从主服务器在该行上放置一个更新锁的视图。

sysproxysessions

sysproxysessions 表包含了使用重定向写功能的每个会话的信息。该表只在辅助服务器上是可用的。

列	类型	描述
session_id	integer	在辅助服务器上用户会话 ID
proxy_id	integer	代表正在运行代理服务器线程 (TID)的代理服务器 ID
proxy_tid	integer	在辅助服务器上正运行的代理线程的事务的 ID 。该 ID 是由代理服务器在处理来自辅助服务器的会话工作时创建的。
proxy_txn_id	integer	当前事务的编号。这些编号对代理服务器是唯一的
current_seq	integer	在当前事务中当前操作的顺序编号
pending_ops	integer	缓冲在辅助服务器上还没有被传送到主服务器的操作的数目
reference_count	integer	表示使用该事务的信息的线程(例如:sqlexec、sync 响应、恢复等等）的数目。当返回的总数为 0 时，该事务处理完成，不管它是否成功。

sysproxytxnops

sysproxytxnops 表包含了在每个代理服务器上运行的每个事务的信息。

在主服务器上，此表包含有关于高可用性集群上的所有的代理服务器的信息。而在辅助服务器上，此表仅包含用于处理更新到辅助服务器的代理服务器的信息。

列	类型	描述
proxy_id	integer	代理服务器的 ID 。这些 ID 在高可用集群里是唯一的
proxy_txn_id	integer	事务的 ID 。这些编号对于代理服务器是唯一的
sequence_number	integer	操作数目
operation_type	char(10)	执行的操作的类型：修改、删除或其它。
rowidn	integer	应用该操作的行的 ID
table	char	完整的表名，修整以适应合理的长度。其格式为：*database*:*owner*.*tablename*

列	类型	描述
sqlerrno	integer	任一 SQL 错误（或者是 0 成功）的错误代码

sysproxytxns

sysproxytxns 表包含有关在每个代理器上运行的所有的当前的事务的信息。

在主服务器上，此表包含有关于高可用性集群上的所有的代理服务器的信息。而在辅助服务器上，此表仅包含用于处理更新到辅助服务器的代理服务器的信息。

列	类型	描述
proxy_id	integer	代理服务器的 ID 。这些 ID 在高可用集群里是唯一的
proxy_txn_id	integer	事务的 ID 。这些编号对于代理服务器是唯一的
reference_count	integer	表示使用该事务信息的线程（例如：sqlexec、sync reply、recovery 等等）的数量。当数量为 0 时，表示事务处理已经完成。（不论它是否成功或不成功）
pending_ops	integer	在主服务器上，表示从辅助服务器上接收的还未被处理的操作的的数量。在辅助服务器上，表示缓冲在辅助服务器上还未被发送到主服务器上的操作的数量
proxy_sid	integer	代理会话 ID

sysptprof

sysptprof 表列出了有关 tblspace 的信息。tblspace 对应于表。

表的概要信息只在表打开时可用。当最后一个使用打开的表的用户关闭此表时，共享内存中的 tblspace 释放，而且所有概要文件统计信息都将丢失。

列	类型	描述
dbsname	char(128)	数据库名称
tabname	char(128)	表名
partnum	integer	分区 (tblspace) 编号
lockreqs	integer	锁请求数
lockwts	integer	锁等待数

列	类型	描述
deadlks	integer	死锁数
lktouts	integer	锁超时数
isreads	integer	isreads 的数量
iswrites	integer	iswrites 的数量
isrewrites	integer	isrewrites 的数量
isdeletes	integer	isdeletes 的数量
bufreads	integer	缓冲区读取数
bufwrites	integer	缓冲区写入数
seqscans	integer	顺序扫描数
pagreads	integer	页读取数
pagwrites	integer	页写入数

sysrepevtreg

可以使用 sysrepevtreg 伪表为一组预定义的连接管理器、GBase OpenAdmin Tool (OAT) for GBase 8s 或任一客户端中的事件注册。完成注册后，连接管理器、OAT 或者任一客户端都可以通过查询此表来接收事件数据。

列	类型	描述
evt_bitmap	integer	事件 ID 位图
evt_timeout	integer	客户端可以等待事件数据的最大时间（以秒为单位）。可用的 timeout 值有： ● 0；不用等待 (缺省) ● -1；一直等待 ● *n*（此处 *n* > 0）等待 *n* 秒
evt_hwm	integer	待定事件列表高水位标记
evt_info	char(256)	事件信息（尚未实现）

sysrepstats

使用 sysrepstats 表向连接管理器、GBase OpenAdmin Tool (OAT) for GBase 8s 连接管理器、OAT 发布事件，并且客户端应用程序之间可以通过发布事件到伪表来交换信息。

列	类型	描述
repstats_type	integer	事件 ID
repstats_subtype	integer	子事件 ID
repstats_time	integer	事件初始化的时间
repstats_ver	integer	事件数据的版本号
repstats_desc	lvarchar	事件数据

sysrepstats 和 sysrepevtreg 表的用户接口

客户端应用程序可以通过往 sysrepstats 伪表中插入事件信息来发布事件到连接管理器或者其他客户端。可以用伪sysrepevtreg 表来注册事件并且可在 sysrepstats 伪表中执行 select 或 fetch 语句来接收事件。

发布事件到 sysrepstats 伪表这一行为为像 GBase OpenAdmin Tool (OAT) for GBase 8s 类似的程序提供了与连接管理器交流的能力。通过发布事件到 sysrepstats 您可以发出控制消息给连接管理器，而无需直接连接到管理器本身。

当连接管理器注册它希望接收事件，它将传递一个位图，它想要接收的事件类型。接收到的事件被发布到请求的线程中。

事件类型

下表列出了每个事件类型的信息，它的位值和描述。

事件类型名称	位值	描述
REPEVT_CLUST_CHG	0x1	高可用性集群的事件类型
REPEVT_CLUST_PERFSTAT	0x2	高可用性集群中的服务器节点的工作负载的事件类型
REPEVT_CLUST_LATSTAT	0x4	高可用集群中服务器节点的复制延迟信息的事件类型
REPEVT_CM_ADM	0x8	连接管理器管理命令
REPEVT_SRV_ADM	0x10	用于服务器节点更新的事件类型

事件类型名称	位值	描述
REPEVT_ER_ADM	0x20	与 Enterprise Replication (ER) 相关的事件的事件类型
REPEVT_CLIENT	0x40	用户自定义事件

REPEVT_CLUST_CHG 事件类型的子事件

下表列出了REPEVT_CLUST_CHG事件类型的子事件：

子事件名称	值	描述	可用条件
REPEVT_SUB_CLUST_ADD	1	向高可用集群添加新节点	只在高可用集群的主服务器上可用
REPEVT_SUB_CLUST_DROP	2	删除高可用集群下的一个节点	只在高可用集群的主服务器上可用
REPEVT_SUB_CLUST_CON	3	高可用性备节点连接到主服务器	只在高可用集群的主服务器上可用
REPEVT_SUB_CLUST_DIS	4	高可用性备节点从主服务器断开连接	只在高可用集群的主服务器上可用
REPEVT_SUB_CLUST_NEWPRIM	5	高可用性主节点更改	只在高可用集群的辅助服务器上可用
REPEVT_SUB_CLUST_DROFF	6	HDR 备节点从主服务器断开连接	HDR 主、备服务器
REPEVT_SUB_CLUST_DRON	7	HDR 节点从主服务器断开连接	HDR 主、备服务器

REPEVT_CLUST_PERFSTAT 事件类型的子事件

下表列出了 REPEVT_CLUST_PERFSTAT 事件类型的子事件：

子事件名称	值	描述	可用条件
REPEVT_ SUB_LOCAL_PERFSTAT	1	本地服务器的工作负载统计	在高可用集群下的所有服务器
REPEVT_ SUB_REMOTE_PERFSTAT	2	辅助服务器的工作负载统计	只在高可用集群下的主服务器上可用

REPEVT_CLUST_LATSTAT 事件类型的子事件

下表列出了 REPEVT_CLUST_LATSTAT 事件类型的子事件：

子事件名称	值	描述	可用条件
REPEVT_SUB_LOCAL_LATSTAT	1	高可用集群中辅助服务器的复制延迟统计	只在高可用集群下的主服务器上可用

REPEVT_CM_ADM 事件类型的子事件

下表列出了 REPEVT_CM_ADM 事件类型的子事件：

子事件名称	值	描述	可用条件
REPEVT_SUB_CM_ADM_REQ	1	命令请求	所有 GBase 8s 服务器实例
REPEVT_SUB_CM_ADM_ACK	2	命令响应	所有 GBase 8s 服务器实例
REPEVT_SUB_CM_REG	3	连接管理器注册服务器	所有 GBase 8s 服务器实例
REPEVT_SUB_CM_DEREG	4	连接管理器注销服务器	所有 GBase 8s 服务器实例
REPEVT_SUB_CM_FATAL	5	连接管理器终止而没有与服务器注销	所有 GBase 8s 服务器实例

REPEVT_SRV_ADM 事件类型的子事件

下表列出了 REPEVT_SRV_ADM 事件类型的子事件：

子事件名称	值	描述	可用条件
REPEVT_ SUB_SRV_BLK	1	由于 DDRBLOCK 造成的服务器阻塞	所有 GBase 8s 服务器实例
REPEVT_ SUB_SRV_UBLK	2	DDRBLOCK 移除后，服务器畅通	所有 GBase 8s 服务器实例

REPEVT_ER_ADM 事件类型的子事件

下表列出了 REPEVT_ER_ADM 事件类型的子事件：

子事件名称	值	描述	可用条件
REPEVT_SUB_ER_SPOOL_FULL	1	ER 在 sbspace 或 dbspace 数据队列中，或在 paging sbspace 组等待空间被添加时阻塞	Enterprise Replication 服务器节点

sysrsslog

sysrsslog 表捕获在主服务器上 RS 辅助服务器的信息。

列	类型	描述
server_name	char(128)	服务器名称
from_cache	integer	从高速缓存读取的总页数
from_disk	integer	从磁盘读取的总页数
logpages_tossed	integer	没有写入日志缓冲区高速缓存的日志总页数

sysscblst

sysscblst 表的这些列提供有关会话内存量的信息。

列	类型	描述
memtotal	integer	可用内存量
memused	integer	已用内存量

syssesappinfo

syssesappinfo 表在 sysmaster 中显示了有关分布式关系数据库体系结构™ (DRDA®) 的客户会话信息。 syssesappinfo 表有以下列。

列	类型	描述
sesapp_sid	INTEGER	客户会话 ID
sesapp_name	CHAR(128)	客户会话名称
sesapp_value	CHAR(512)	会话值

syssesprof

syssesprof 表列出用户操作（例如：写入、删除或提交）发生数的累计计数。

列	类型	描述
sid	integer	会话 ID
lockreqs	integer	所请求锁的数量
locksheld	integer	当前持有的锁的数量
lockwts	integer	等待锁的次数
deadlks	integer	所检测到的死锁数量
lktouts	smallint	死锁超时数
logrecs	integer	已写入的逻辑日志记录数
isreads	integer	读取数
iswrites	integer	写入数
isrewrites	integer	重写数
isdeletes	integer	删除数
iscommits	integer	提交数
isrollbacks	integer	回滚数
longtxs	integer	长事务数
bufreads	integer	缓冲区读取数
bufwrites	integer	缓冲区写入数
seqscans	integer	顺序扫描数
pagreads	integer	页读取数
pagwrites	integer	页写入数
total_sorts	integer	排序总数
dsksorts	integer	不适合内存的排序数
max_sortdiskspace	integer	排序所使用的最大空间

列	类型	描述
logspused	integer	当前会话事务所使用的逻辑日志空间字节数
maxlogsp	integer	会话曾经使用的逻辑日志空间最大字节数

syssessions

syssessions 表提供有关连接到数据库服务器上的每个用户的一般信息。在 state 列中，每个位位置代表一个单独的标志。这样，如果值是使用 HEX 函数返回的，那么读取 state 列中的值可能会更容易。

列	类型	描述
sid	integer	会话 ID
username	char(32)	用户 ID
uid	smallint	用户 ID 号
pid	integer	客户端进程 ID
hostname	char(256)	客户端的主机名
tty	char(16)	用户 stderr 文件的名称
connected	integer	用户连接到数据库服务器上的时间
feprogram	char(255)	保留、供将来使用
pooladdr	integer	会话池地址
is_wlatch	integer	如果会话主线程正在等待锁存器，那么为 1
is_wlock	integer	如果会话主线程正在等待锁，那么为 1
is_wbuff	integer	如果会话主线程正在等待缓冲区，那么为 1
is_wckpt	integer	如果会话主线程正在等待检查点，那么为 1
is_wlogbuf	integer	如果会话主线程正在等待日志缓冲区，那么为 1
is_wtrans	integer	如果会话主线程正在等待事务，那么为 1
is_monitor	integer	如果会话是特殊的监视进程，那么为 1

列	类型	描述		
is_incrit	integer	如果会话主线程处于临界段中，那么为 1		
state	integer	标识	十六进制	含义
		1	0x00000001	用户结构在使用中
		2	0x00000002	正在等待锁存器
		4	0x00000004	正在等待锁
		8	0x00000008	正在等待缓冲区
		16	0x00000010	正在等待检查点
		32	0x00000020	在读调用中
		64	0x00000040	正在向备份磁带写如逻辑日志
		256	0x00000100	在临界段中
		512	0x00000200	特殊守护程序
		1024	0x00000400	正在归档
		2048	0x00000800	清除死进程
		4096	0x00001000	正在等待写入日志缓冲区
		8192	0x00002000	特殊的缓冲区清仓线程
		16384	0x00004000	远程数据库服务器
		32768	0x00008000	用于设置 RS_timeout 死锁超时
		65536	0x00010000	常规死锁超时
		262144	0x00040000	正在等待事务
		524288	0x00080000	会话主线程
		1048576	0x00100000	用于构建索引的线程
		2097152	0x00200000	B-tree 清除线程

syssmx

syssmx 表提供了 SMX （服务器多路复用器组）的连接信息。

列	类型	描述
address	int8	SMX 管道地址
name	char(128)	目标服务器名称
encryption_status	char(20)	启用或禁用保留供将来使用
buffers_sent	integer	已发送的缓冲区数
buffers_recv	integer	接收到的缓冲区数
bytes_sent	int8	已发送的字节数
bytes_recv	int8	接收到的字节数
reads	integer	读取调用数
writes	integer	写入调用数
retries	integer	重试的写入调用数

syssmxses

syssmxses 表提供了 SMX （服务器多路复用器组）的会话信息。

列	类型	描述
name	char(128)	目标服务器名称
address	int8	SMX 会话地址
client_type	char(20)	SMX 客户端类型
reads	integer	读取调用数
writes	integer	写入调用数

syssqexplain

syssqexplain 伪表存储了有关 SQL 查询的信息。

存储的信息包括查询优化器的计划及估计的返回行数、查询的相对成本。

列	类型	描述

列	类型	描述
sqx_sessionid	INTEGER	与 SQL 语句相关联的会话 ID
sqx_sdbno	INTEGER	查询会话 ID 在数组中的位置
sqx_iscurrent	CHAR	该查询是否为当前的 SQL 语句
sqx_executions	INTEGER	执行查询的总次数
sqx_cumtime	FLOAT	运行查询的累计时间 **重要：** 如果禁用 SQL 跟踪，那么显示零。
sqx_bufreads	INTEGER	运行查询时执行的缓冲区读取数 **重要：** 如果禁用 SQL 跟踪，那么显示零。
sqx_pagereads	INTEGER	运行查询时执行的页读取数 **重要：** 如果禁用 SQL 跟踪，那么显示零。
sqx_bufwrites	INTEGER	运行查询时执行的缓冲区写入数 **重要：** 如果禁用 SQL 跟踪，那么显示零。
sqx_pagewrites	INTEGER	运行查询时执行的页写入数 **重要：** 如果禁用 SQL 跟踪，那么显示零。
sqx_totsorts	INTEGER	运行查询时执行的排序数 **重要：** 如果禁用 SQL 跟踪，那么显示零。
sqx_dsksorts	INTEGER	运行查询时磁盘执行的排序数 **重要：** 如果禁用 SQL 跟踪，那么显示零。
sqx_sortspmax	INTEGER	排序所需的最大磁盘空间
sqx_conbno	SMALLINT	在 conblock 列表中位置

列	类型	描述
sqx_ismain	CHAR	查询是否在主语句块中
sqx_selflag	VARCHAR(200,0)	SQL语句类型；例如：SELECT 、UPDATE 、 DELETE
sqx_estcost	INTEGER	查询的估计成本
sqx_estrows	INTEGER	估计查询返回的行数
sqx_seqscan	SMALLINT	此查询所用的顺序扫描数
sqx_srtscan	SMALLINT	此查询所用的排序扫描数
sqx_autoindex	SMALLINT	此查询所用的自动索引扫描数
sqx_index	SMALLINT	此查询所用的索引路径数
sqx_remsql	SMALLINT	此查询所用的远程路径数
sqx_mrgjoin	SMALLINT	此查询所用的排序联接合并数
sqx_dynhashjoin	SMALLINT	此查询所用的动态哈希连接数
sqx_keyonly	SMALLINT	此查询所用的key-only 扫描数
sqx_tempfile	SMALLINT	此查询所用的临时文件数
sqx_tempview	SMALLINT	临时表查询所创建的视图数量
sqx_secthreads	SMALLINT	查询所使用的辅助线程数
sqx_sqlstatement	CHAR	已运行的 SQL 查询

syssqltrace

syssqltrace 表提供有关单个 SQL 语句的详细信息。

列	类型	描述
sql_id	int8	唯一 SQL 执行 ID
sql_address	int8	代码块中语句的地址
sql_sid	int	运行 SQL 语句的用户的数据库的会话 ID
sql_uid	int	运行 SQL 的语句的用户 ID
sql_stmttype	int	语句类型
sql_stmtname	char(40)	显示为单词的语句类型

列	类型	描述
sql_finishtime	int	次语句的完成时间 (UNIX™)
sql_begintxtime	int	此事务的启动时间
sql_runtime	float	语句执行时间
sql_pgreads	int	此 SQL 语句的磁盘读取数
sql_bfreads	int	此 SQL 语句的缓冲区读取数
sql_rdcache	float	从缓冲池读取页的时间百分比
sql_bfidxreads	int	索引页缓冲区读取数
sql_pgwrites	int	写入磁盘的页数
sql_bfwrites	int	已修改并返回到缓冲池的页数
sql_wrcache	float	页已写入缓冲池，但未写入磁盘的时间百分比
sql_lockreq	int	此 SQL 语句所需锁总数
sql_lockwaits	int	SQL 语句等待锁的次数
sql_lockwttime	float	SQL 语句期间系统等待锁定的时间
sql_logspace	int	逻辑日志中 SQL 语句所用空间量
sql_sorttotal	int	为语句运行的排序数
sql_sortdisk	int	磁盘上运行的排序数
sql_sortmem	int	内存中运行的排序数
sql_executions	int	SQL 语句运行的次数
sql_totaltime	float	运行语句所用的时间总量
sql_avgtime	float	运行语句所用的平均时间
sql_maxtime	float	执行 SQL 语句所使用最大时间量
sql_numiowaits	int	I/O 操作必须等待的次数
sql_avgiowaits	float	SQL语句必须等待的平均时间量
sql_totaliowaits	float	SQL 语句必须等待 I/O 的时间总量。这不包含任何异步 I/O
sql_rowspersec	float	产生的平均行数（每妙）
sql_estcost	int	与SQL 语句关联的成本
sql_estrows	int	按照优化器的预测为 SQL 语句返回的预估计行数
sql_actualrows	int	为 SQL 语句返回的行数
sql_sqlerror	int	SQL 错误号
sql_isamerror	int	RSAM/ISAM 错误编号
sql_isollevel	int	SQL 语句的隔离级别

列	类型	描述
sql_sqlmemory	int	执行 SQL 语句所需字节数
sql_numiterators	int	语句所用迭代器数
sql_database	char(128)	数据库名称
sql_numtables	int	执行 SQL 语句中所用表数
sql_tablelist	char(4096)	SQL 语句中直接引用的表名列表。如果 SQL 语句激发对其他表执行语句的触发器，将不列出这些表
sql_statement	char(1600)	已运行的 SQL 语句

syssqltrace_info

syssqltrace_info 表描述有关 SQL 概要文件跟踪系统的信息。

列	类型	描述
flags	integer	SQL 跟踪标志
ntraces	integer	要跟踪的项数
tracesize	integer	为各 SQL 跟踪项存储的文本的大小
duration	integer	跟踪缓冲区（以秒为单位）
sqlseen	int8	启动或调整大小来跟踪的 SQL 项数
starttime	integer	跟踪的启动时间
memoryused	int8	SQL 跟踪所用内存的字节数

syssqltrace_iter

syssqltrace_iter 表列举 SQL 语句迭代器。

列	类型	描述
sql_id	int8	SQL 执行 ID
sql_address	int8	SQL 语句块的地址
sql_itr_address	int8	迭代器的地址
sql_itr_id	int	迭代器 ID
sql_itr_left	int	向左的迭代器 ID

列	类型	描述
sql_itr_right	int	向右的迭代器 ID
sql_itr_cost	int	迭代器成本
sql_itr_estrows	int	迭代器预估行数
sql_itr_numrows	int	迭代器实际处理的行数
sql_itr_type	int	迭代器类型
sql_itr_misc	int	迭代器杂项标志
sql_it_info	char(256)	显示为文本的迭代器杂项标志

syssrcrss

syssrcrss 表提供了主服务器上 RS 辅助服务器的相关统计信息。

列	类型	描述
address	int8	RS 辅助服务器控制块地址
server_name	char(128)	数据库服务器名称
server_status	char(20)	静止、活动或不活动
connection_status	char(20)	已连接或已断开连接
log_transmission_status	char(20)	活动或阻塞
next_page_tosend_log_uniq	integer	要发送的下一个页面的唯一日志 ID
next_page_tosend_log_page	integer	要发送的下一页的页号
seq_tosend	integer	上次发送的缓冲区的顺序 ID
last_seq_acked	integer	上次应答的缓冲区的顺序 ID

syssrcsds

syssrcsds 表提供了主服务器上 SD 辅助服务器的相关统计信息。

列	类型	描述
address	int8	SD 辅助服务器控制块地址

列	类型	描述
source_server	char(128)	主数据库服务器的名称
connection_status	char(20)	已连接或已断开连接
last_received_log_uniq	integer	上次接收到的日志页的唯一日志 ID
last_received_log_page	integer	上次接收到的日志页的页号
next_lpgtoread_log_uniq	integer	下次要读取的日志页的唯一日志 ID
next_lpgtoread_log_page	integer	下次要读取的日志页的页号
last_acked_lsn_uniq	integer	上次应答的 LSN 的唯一日志 ID
last_acked_lsn_pos	integer	上次应答的 LSN 的日志位置
last_seq_received	integer	上次接收到的缓冲区的顺序 ID
last_seq_acked	integer	上次应答的缓冲区的顺序 ID
cur_pagingfile	char(640)	当前的 GBase_8s paging 文件名称
cur_pagingfile_size	int8	当前的 GBase_8s paging 文件大小
old_pagingfile	char(640)	以前的 GBase_8s paging 文件名称
old_pagingfile_size	int8	以前的 GBase_8s paging 文件大小

systabnames

systabnames 表描述数据库所管理的每个表。

列	类型	描述
partnum	integer	tblspace 标识
dbsname	char(128)	数据库名
owner	char(32)	所有者的用户 ID
tabname	char(128)	表名
collate	char(32)	与支持 GLS 的数据库相关联的对照

systhreads

systhreads 表描述有关线程的信息。

列	类型	描述
th_id	INTEGER	线程的 ID
th_addr	INTEGER	线程控制块的内存地址
th_joinlist	INTEGER	如果有等待线程， th_joinlist 列显示等待线程列表中第一个线程的地址
th_joinnext	INTEGER	如果有等待线程，那么 th_joinnext 列显示等待线程列表中第二个线程的地址
th_joinee	INTEGER	该线程等待的要退出线程的地址
th_name	CHAR(12)	线程的名称
th_state	INTEGER	线程状态代码
th_priority	INTEGER	线程的优先级
th_class	INTEGER	线程所在的虚拟处理器类代码
th_vpid	INTEGER	线程上次计划运行的虚拟处理器的 ID
th_mtxwait	INTEGER	此线程正在等待的互斥对象的地址
th_conwait	INTEGER	此线程正在等待条件的地址
th_waketime	INTEGER	最近休眠到期的时间。时间由内部时钟计算。值为 -1 的表示时间值不定。
th_startwait	INTEGER	最近开始等待的时间。时间由内部时钟计算
th_startrun	INTEGER	最近开始执行的时间。时间由内部时钟计算

systrgrss

systrgrss 表提供了 RS 辅助服务器上 RS 辅助服务器的相关统计信息。

列	类型	描述
address	int8	RS 辅助服务器控制块地址
source_server	char(128)	服务 RS 辅助服务器的源服务器

列	类型	描述
connection_status	char(20)	已连接或已断开连接
last_received_log_uniq	integer	上次接收到的日志页的唯一日志 ID
last_received_log_page	integer	上次接收到的日志页的页号
last_seq_received	integer	上次接收到的缓冲区的顺序 ID
last_seq_acked	integer	上次应答的缓冲区的顺序 ID

systrgsds

systrgsds 表提供了 SD 辅助服务器上 SD 辅助服务器的相关统计信息。

systrgsds 表包含以下列：

列	类型	描述
address	int8	SD 辅助服务器控制块地址
source_server	char(128)	服务 SD 辅助服务器的源服务器
connection_status	char(20)	已连接或已断开连接
last_received_log_uniq	integer	上次接收到的日志页的唯一日志 ID
last_received_log_page	integer	上次接收到的日志页的页号
next_lptoread_log_uniq	integer	下次要读取的日志页的唯一日志 ID
next_lptoread_log_page	integer	下次要读取的日志页的页号
last_acked_lsn_uniq	integer	上次应答的 LSN 的唯一日志 ID
last_acked_lsn_pos	integer	上次应答的 LSN 的日志位置
last_seq_received	integer	上次接收到的缓冲区的顺序 ID
last_seq_acked	integer	上次应答的缓冲区的顺序 ID
cur_pagingfile	char(640)	当前的 GBase_8s paging 文件名称
cur_pagingfile_size	int8	当前的 GBase_8s paging 文件大小
old_pagingfile	char(640)	以前的 GBase_8s paging 文件名称

列	类型	描述
old_pagingfile_size	int8	以前的 GBase_8s paging 文件大小

sysvpprof

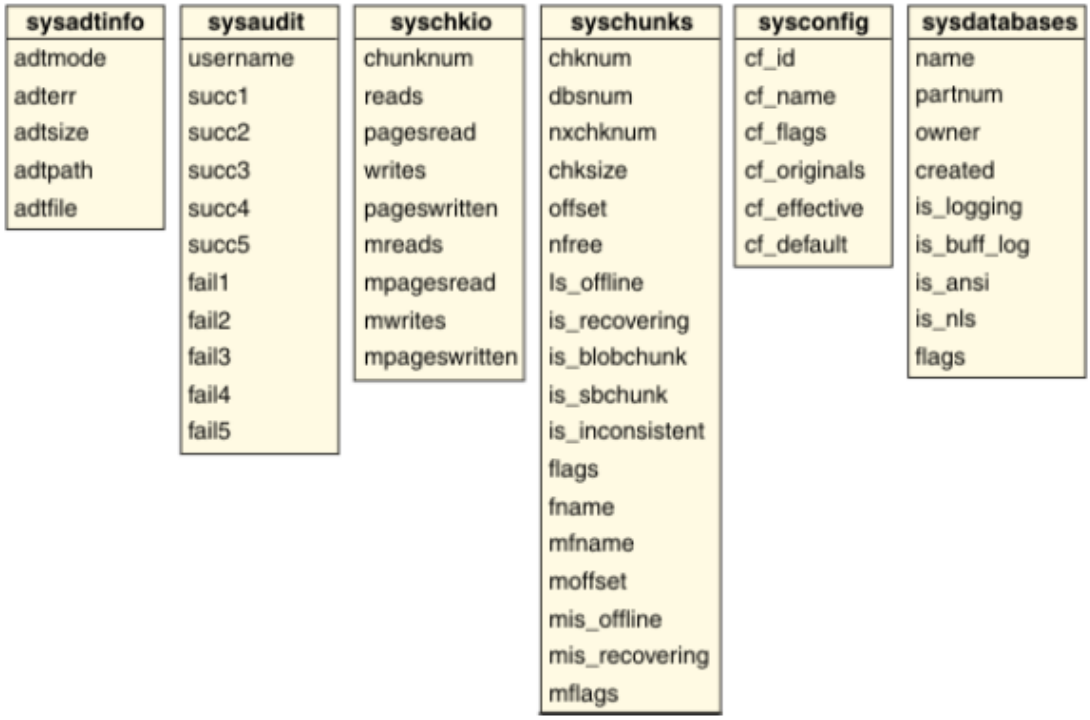
sysvpprof 表列出每个虚拟处理器的用户和系统 CPU 的时间。

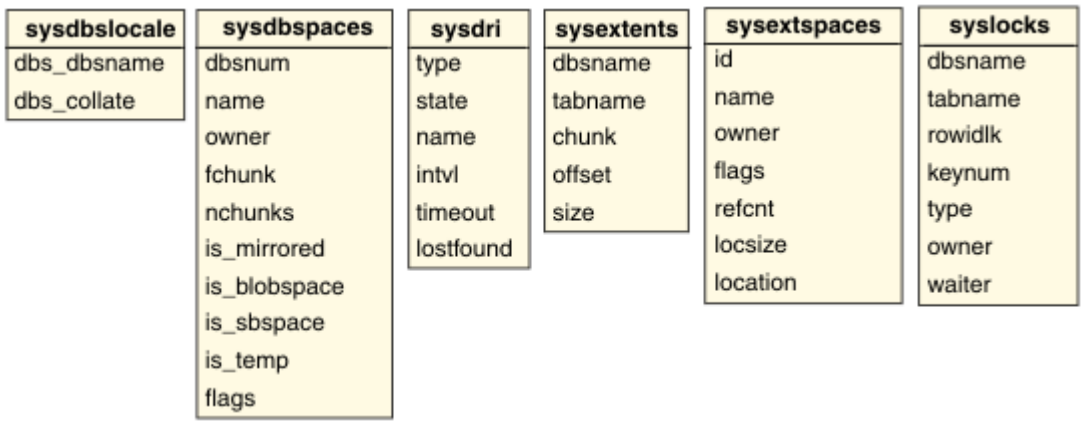
列	类型	描述
vpid	integer	虚拟处理器 ID
	char(50)	虚拟处理器的类型： ● cpu ● adm ● lio ● pio ● aio ● tli ● soc ● str ● shm ● opt ● msc ● adt
usercpu	float	用户时间（微妙数）
syscpu	float	系统时间（微妙数）

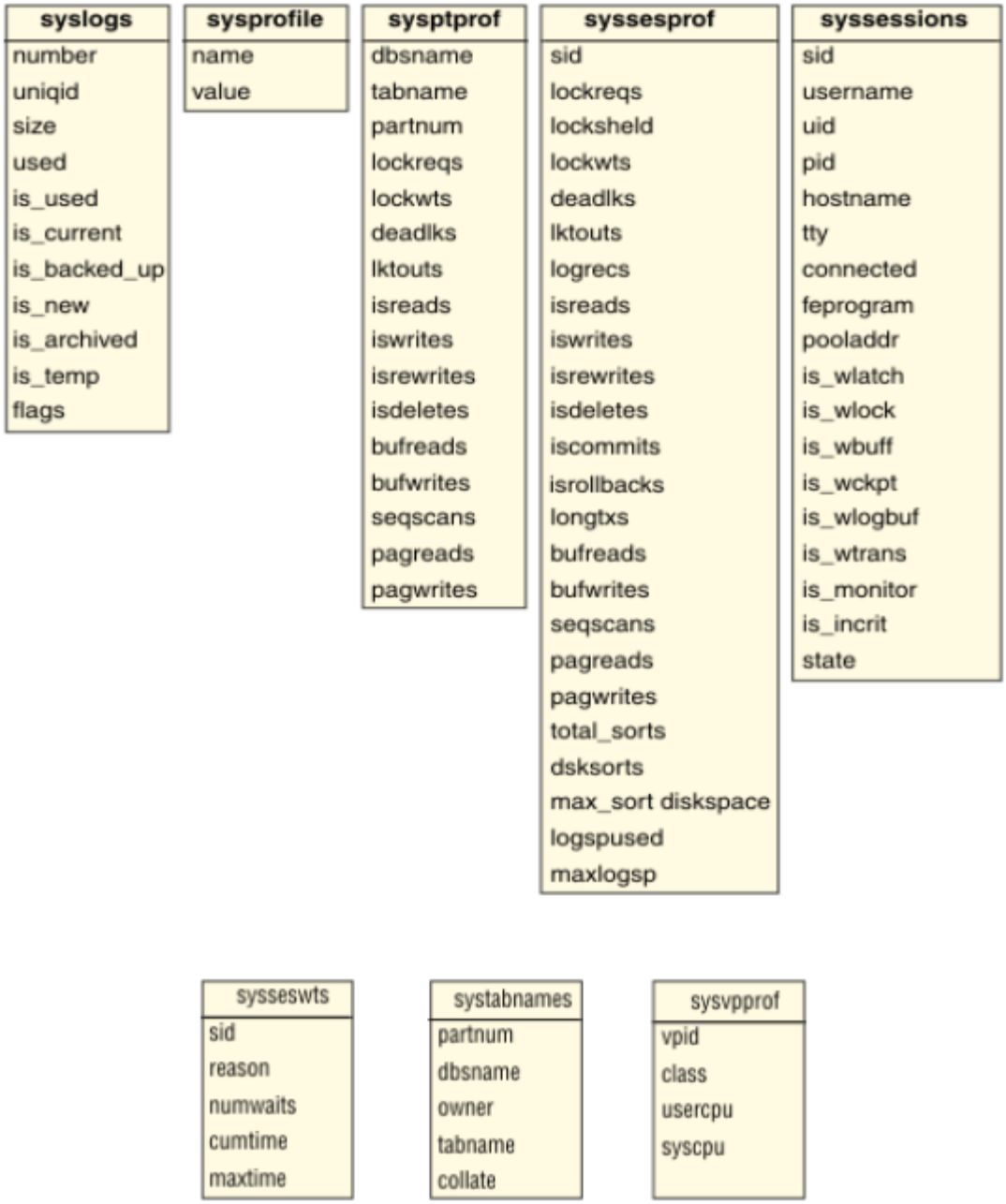
SMI 表映射

图 1 显示了 SMI 表中的某些列。

图: SMI 表中的列


sysadtinfo
adtmode
adterr
adtsize
adtpath
adtfile
sysaudit
username
succ1
succ2
succ3
succ4
succ5
fail1
fail2
fail3
fail4
fail5
syschkio
chunknum
reads
pagesread
writes
pageswritten
mreads
mpagesread
mwrites
mpageswritten
syschunks
chknum
dbsnum
nxchknum
chksize
offset
nfree
ls_offline
is_recovering
is_blobchunk
is_sbchunk
is_inconsistent
flags
fname
mfname
moffset
mis_offline
mis_recovering
mflags
sysconfig
cf_id
cf_name
cf_flags
cf_originals
cf_effective
cf_default
sysdatabases
name
partnum
owner
created
is_logging
is_buff_log
is_ansi
is_nls
flags


sysdbslocale
dbs_dbsname
dbs_collate
sysdbspaces
dbsnum
name
owner
fchunk
nchunks
is_mirrored
is_blobspace
is_sbspace
is_temp
flags
sysdri
type
state
name
intvl
timeout
lostfound
sysextents
dbsname
tabname
chunk
offset
size
sysextspaces
id
name
owner
flags
refcnt
locsize
location
syslocks
dbsname
tabname
rowidlk
keynum
type
owner
waiter



SMI 表中 gstat 的信息

要了解 gstat 实用程序所提供的信息，可以使用 SQL 查询相应的 SMI 表。下表指示了要查询哪些 SMI 表以获取由给定 gstat 选项提供的信息。关于 gstat 选项的描述，请参阅 监视数据库服务器状态。

gstat 选项	要查询的 SMI 表	不在 SMI 表中的 gstat 字段
-d	sysdbspaces syschunks	address bpages
-D	sysdbspaces	

gstat 选项	要查询的 SMI 表	不在 SMI 表中的 gstat 字段
	syschkio	
-F	sysprofile	address flusher snoozer state data
-g ath	systhreads	
-g dri	sysdri	Last DR CKPT (id/pg)
-g glo	sysvpprof	按类排列的虚拟处理器列表
-g ipl	sysipl	
-g rss	sysrsslog systrgrss syssrcrss	
-g his	syssqltracing	
-g sds	syssrcsds systrgsds	
-g smx	syssmx	
-g smx ses	syssmxses	
-k	syslocks	address lklist tblsnum
-l	syslogs sysprofile	所有的物理日志字段（numpages 和 numwrits 除外）所有的逻辑日志缓冲区字段（ numrecs、numpages 和 numwrits 除外）address begin % used
-p	sysprofile	
-u	syssessions syssesprof	address wait nreads nwrites

2.3 sysadmin 数据库

sysadmin 数据库包含具有以下内容的表。这些表中包含并组织调度任务和传感器，存储传感器收集的数据，还记录了调度工作和 SQL 管理 API 命令的结果。

缺省情况下，只授予用户 gbasedbt 对数据库 sysadmin 的访问权；可以授予其他用户对 sysadmin 数据库的访问权。

因为有些重要的数据库服务器组件使用它，您不能删除或更改 sysadmin 数据库。然而，如果 root dbspace 没有足够的空间存储任务属性和命令历史信息，您可以将 sysadmin 数据库从其缺省的 root dbspace 位置移除。移除 sysadmin 数据库请使用 reset sysadmin SQL 管理 API 命令：admin() 或 task() 。

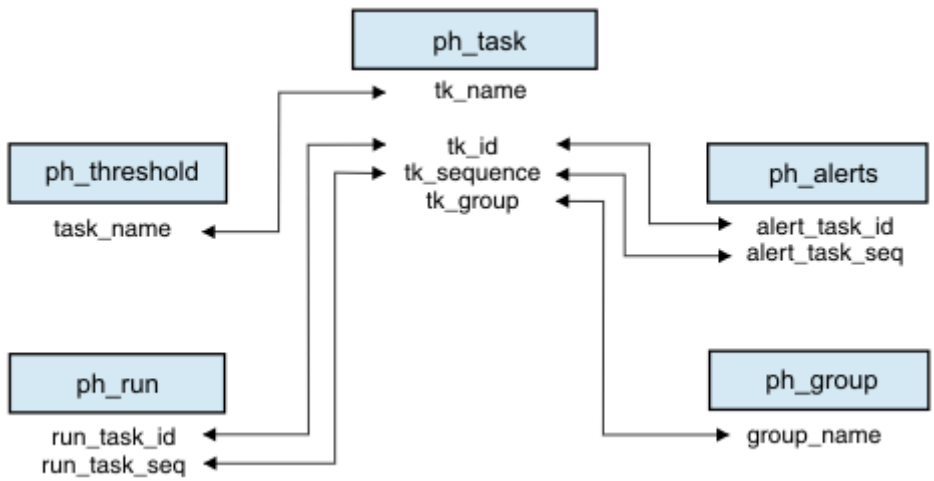
重要： 在 sysadmin 移除数据库的过程中会将该数据库恢复到初次创建视的状态；所有的数据、操作历史和结果集表的信息都会丢失。只有内置任务、传感器和阈值保留在 sysadmin 表中。

2.3.1 Scheduler 表

Scheduler 表存储了 sysadmin 数据库中有关任务和传感器的五个表。它们分别是：ph_task 、ph_run 、ph_group 、ph_alert 和ph_threshold 。

Scheduler 是一个管理工具，它是数据库服务器可以在预定义的时间或有服务器内部驱动的时间执行功能和过程。它所使用的这五个表包含了自动运行的内置任务和传感器。您也可以通过往这些表中插入数据来添加您自己的任务和传感器。这些表有它们在下图中描述的列之间的关系。

图：Scheduler 表之间的关系



更多有关如何使用 Scheduler 的详细信息，请参阅 GBase 8s 管理员指南 。

ph_task 表

ph_task 表包含关于调度任务和传感器的信息。ph_task 表所包含内置任务和传感器是按计划自动运行的。

列	类型	描述
tk_id	serial	顺序任务 ID

列	类型	描述
		由系统更新；不要修改 引用于 ph_alert 表中的 alert_task_id 列和表 ph_run 中的 run_task_id 列
tk_name	char(36)	任务名称。在该列上有一个唯一索引确保没有两个名称是相同的。 引用于 ph_threshold 表中的 task_name 列
tk_description	lvarchar	关于该任务的描述
tk_type	char(18)	任务的类型： ● TASK：以特定的时间和频率调用操作。 ● SENSOR：（缺省）一个从结果表中收集、存储和清除数据的任务 ● STARTUP TASK：仅在服务器启动时运行的任务 ● STARTUP SENSOR：仅在服务器启动时运行的传感器
tk_sequence	integer	当前的数据收集号 由系统更新；不要修改 引用于 ph_alert 表中的 alert_task_id 列和 ph_run 表中的 run_task_seq 列
tk_result_table	varchar	传感器用来收集数据的结果表名称。该表由 tk_create 列中的 CREATE TABLE 语句创建
tk_create	lvarchar	用于创建传感器收集数据的结果表的 CREATE TABLE 语句。 表中必须有一名为 ID 的列，该列用来存储 tk_sequence 值。此值用来表示该行的年龄和清除行
tk_dbs	varchar(250)	运行任务的数据库 缺省值 sysadmin 数据库
tk_execute	lvarchar	要执行的 SQL 语句 命令长度限制在 2048 字节内
tk_delete	interval day(2) to second	删除结果表中早于该时间间隔的数据

列	类型	描述
		缺省值 1:00:00 （一天）
tk_start_time	datetime hour to second	任务或传感器启动的时间 缺省值 08:00:00
tk_stop_time	datetime hour to second	该任务或传感器应停止运行的时间 数据库服务器在下一个有效日调度下一个执行 缺省值 19:00:00 。可以为 NULL ，表示没有停止时间
tk_frequency	interval day(2) to second	该任务或传感器运行的频率 缺省值 1 （一天一次）
tk_next_execution	datetime year to second	该任务或传感器下次执行时间. 如果已启动的任务或传感器已经运行，该值为 NULL 。当一个任务或传感器启动时，数据库服务器用 tk_start_time 、tk_stop_time 和 tk_frequency 列的值来计算此次时间。并根据 tk_monday 、tk_tuesday 、tk_wednesday 、tk_thursday 、tk_friday 、tk_saturday 和 tk_sunday 列来计算启动当天的星期数。例如：在 *new_next_execution_time* 比 *current_next_execution_time* 长的情况下，*new_next_execution_time* = *current_next_execution_time* + tk_frequency 。如果 tk_frequency 没有指示，任务只运行一次
tk_total_executions	integer	该任务或传感器运行的总次数 由系统更新；不要修改 缺省值为 0
tk_total_time	float	执行该任务或传感器所使用的时间 由系统更新；不要修改 缺省值为 0.0 秒
tk_monday	boolean	该任务或传感器是否在星期一执行 缺省值为 T (true)
tk_tuesday	boolean	该任务或传感器是否在星期二执行

列	类型	描述
		缺省值为 T (true)
tk_wednesday	boolean	该任务或传感器是否在星期三执行 缺省值为 T (true)
tk_thursday	boolean	该任务或传感器是否在星期四执行 缺省值为 T (true)
tk_friday	boolean	该任务或传感器是否在星期五执行 缺省值为 T (true)
tk_saturday	boolean	该任务或传感器是否在星期六执行 缺省值为 T (true)
tk_sunday	boolean	该任务或传感器是否在星期日执行 缺省值为 T (true)
tk_attributes	integer	标志 由系统更新；不要修改
tk_group	varchar(128)	组名 必须与 ph_group 表中的列 group_name 的值相同。 缺省值 MISC
tk_enable	boolean	该任务或传感器是否已启用 缺省值为 T（该任务已启用）
tk_priority	integer	作业优先级，数值范围为 0-5 ，数字越大级别越高。如果有几个作业要同时执行，则具有最高优先级的作业先执行。 缺省值为：0 （低优先级）

ph_run 表

ph_run 表包含关于每个调度程序任务或传感器的运行方式和运行时间的信息。

列	类型	描述
run_id	serial	在执行期间生成的顺序 ID

列	类型	描述
		由系统更新；不要修改
run_task_id	integer	任务 ID 引用于 ph_task 表的 tk_id 列
run_task_seq	integer	该任务或传感器的唯一顺序编号 引用于表 ph_task 的 tk_sequence 列
run_retcode	integer	用户定义例程或存储过程的返回码或 SQL 命令的 SQLCode
run_time	datetime year to second	该任务或传感器的执行时间
run_duration	float	执行改作业所使用的是案件（以秒为单位）
run_ztime	integer	服务器统计（gstat -z 命令）上次运行的时间
run_btime	integer	服务器启动的时间
run_mttime	integer	服务器的内部计数器

ph_group 表

ph_group 表包含有关调度程序组名的信息。它包含内置任务和传感器的分类组以及 MISC 中缺省的组。

列	类型	描述
group_id	serial	组 ID 由系统更新；不要修改
group_name	varchar(128)	组的唯一名称 引用于 ph_task 表中的 tk_group 列
group_description	lvarchar	组的描述

ph_alert 表

ph_alert 表包含调度程序的关于错误、警告或参考消息的信息。与内置任务和传感器有关的警报将自动添加到 ph_alert 表内。

列	类型	描述
id	serial	警报 ID 由系统更新；不要修改
alert_task_id	serial	任务或传感器 ID 必须与 ph_task 表中 tk_id 列的值相同 事件报警的任务 ID 为 15
alert_task_seq	integer	识别出哪些调度程序任务创建警报 由系统更新；不要修改 引用于 ph_task 表的the tk_sequence 列
alert_type	char(8)	警报的类型： ● INFO (缺省的) ● WARNING ● ERROR 警报和事件报警的严重程度是由警报的类型和颜色综合确定的。
alert_color	char(15)	警报的颜色： ● 绿色 （默认） ● 黄色 ● 红色 警报和事件报警的严重程度是由警报的类型和颜色综合确定的。
alert_time	datetime year to	产生警报的时间

列	类型	描述
	second	由系统更新；不要修改
alert_state	char(15)	指示对象当前所处的状态： NEW （默认）警报是新添加的，对该警报尚未发生任何其他操作 IGNORED 警报已由 DBA 确认，但未执行任何操作 ACKNOWLEDGED 警报已由 DBA 确认 ADDRESSED 警报已由 DBA 处理
alert_state_changed	datetime year to second	最近一次状态被更新的时间 由系统更新；不要修改
alert_object_type	char(15)	发生警报对象的类型： ● ALARM ● CHUNK ● DATABASE ● DBSPACE ● INDEX ● MISC (Default) ● SERVER ● SQL_STATEMENT ● TABLE ● USER
alert_object_name	varchar(255)	警报对象或事件报警类 ID 的名称
alert_message	lvarchar	警报或事件报警的详细消息
alert_action_dbs	lvarchar(256)	用于修正操作的数据库名称

列	类型	描述
		缺省值为 sysadmin
alert_action	lvarchar	修正操作 可调用 SQL 脚本来修正该问题。该脚本必须符合所有多语句准备规则。 如果没有可用的操作，那么为 NULL
alert_object_info	bigint	事件报警的事件ID

下表为 3 种不同类型的消息定义的警报颜色。

消息类型	绿色	黄色	红色
Informative	指示组件操作状态的状态消息 事件报警的严重程度为 1（不值得注意）	重要的状态消息 事件报警的严重程度为 2（可供参考）	需要执行操作的状态信息
Warning	自动处理的来自数据库的警告	需要处理的未来事件 事件报警的严重程度为 3（需要注意）	即将发生预测的故障。需要立即执行操作
Error	组件中的故障，由组件自身修正	组件中的故障，由组件自身修正，但可能需要	组件中故障，需要 DBA 操作 事件报警的严重程度为 4（紧急）或

消息类型	绿色	黄色	红色
		DBA 操作	5（致命）

ph_alerts 表显示了警报的信息和与之相关的任务和传感器的信息。

ph_threshold 表

ph_threshold 表包含关于调度程序任务阈值的信息。

ph_threshold 表包含了与内置任务和传感器相关联的内置阈值。例如：名为 COMMAND HISTORY RETENTION 的阈值决定了在 command_history 表中应保留的时间行长度。

列	类型	描述
id	integer	阈值 ID
name	char	阈值的名称
task_name	varchar	与阈值相关联的调度程序任务名称 必须与 ph_task 表中 tk_name 列的值相同
value	lvarchar	阈值的值
value_type	char	值列的数据类型： ● STRING ● NUMERIC ● NUMERIC(p,s)
description	lvarchar	阈值的描述

2.3.2 结果表

结果表包含调度程序任务执行情况的历史数据。

大多数传感器创建一个新的表来存储它们的结果。表的名称在 ph_task 表的 tk_result_table 列中列出。表的结构是由 ph_task 表中 tk_create 列里的 CREATE TABLE 语句定义的。

当内置传感器以前缀 mon_ 启动时，它会自动创建结果表。

列	类型	描述
ID	integer	传感器的迭代顺序编号。必须在 $DATA_SEQ_ID

列	类型	描述
		中设置 引用于 ph_run 表中的 run_task_seq 列
user columns	any	可指定任意数据类型的列来保存由传感器返回的信息

2.3.3 command_history 表

command_history 表包含近 30 天内管理 API 函数已运行的所有 SQL 命令的列表。该表还会显示命令的结果。

command_history 表显示管理 API 的每一条 SQL 命令，并显示执行命令的用户的相关信息、命令执行的时间、命令、以及数据库服务器完成命令运行时返回的消息。

列	数据类型	描述
cmd_number	serial	每行的唯一 ID
cmd_exec_time	datetime year-to-second	命令的启动时间
cmd_user	varchar	执行命令的用户
cmd_hostname	varchar	执行命令的主机的名称
cmd_executed	varchar	所执行的命令
cmd_ret_status	integer	返回码
cmd_ret_msg	lvarchar	返回消息

下表显示了示例命令和 command_history 表中关联的结果。

所执行的命令 (cmd_executed)	返回的消息 (cmd_ret_msg)
set sql tracing on	SQL tracing on with 1000 buffers of 2024 bytes.
create dbspace	Space 'space12' added.
checkpoint	Checkpoint completed.

所执行的命令 (cmd_executed)	返回的消息 (cmd_ret_msg)
add log	Added 3 logical logs to dbspace logdbs.

要显示命令历史记录，请在 sysadmin 数据库中运行以下 SQL 语句：

```
SELECT * FROM command_history;
```

command_history 表的大小

依赖于所运行的管理 API 的 SQL 命令的数目。command_history 表可以扩展到很大。

command_history 表中数据都有一个保留的时间周期。您可以通过更改 ph_threshold 表中的 COMMAND HISTORY RETENTION 行的 value 字段的来修改该时间周期。

您可以使用诸如 DELETE 或 TRUNCATE TABLE 之类的 SQL 命令从表中手动移除数据。

2.3.4 storagepool 表

storagepool 表在 sysadmin 数据库中，它包含了在 GBase 8s 实例存储池中的所有条目的信息。每个条目代表当自动扩展存储空间时，服务器可以使用的自由空间。

列	类型	描述
entry_id	SERIAL	存储池条目 ID
path	VARCHAR (255)	为服务器可以使用的文件、目录或设备需要额外的存储空间的路径
beg_offset	BIGINT	初始的偏移量，以千字节为单位到设备的服务器可以开始分配空间 如果是一个目录的存储池信息，最终的偏移量值是 0
end_offset	BIGINT	初始的偏移量，以千字节为单位到设备的服务器必须停止分配空间 如果是一个目录的存储池信息，最终的偏移量值是 0
chunk_size	BIGINT	条目所分配的 chunk 的初始大小

列	类型	描述
status	VARCHAR (255)	储存池条目的状态。状态值可以为： • Active = 一个实用的存储池条目。服务器可以从此条目分配 chunk 。 • Full = 存储池条目上没有可用空间。服务器不能从该条目分配 chunk 。 • Error = 当服务器尝试从此条目分配 chunk ，存储池报错。
priority	INT	当服务器通过存储池搜寻空间时，目录、文件或设备的优先级。服务器会先从高级别的条目分配空间，之后才是低级别的条目。 ● 1 = 高优先级 ● 2 = 中优先级 ● 3 = 低优先级
last_alloc	DATETIME (year to second)	该条目最近一次被分配的时间
logid	INT	当时最后使用此条目的当前日志的 ID 。服务器使用该标志和 logused 值在相同优先级别条目之间做出选择
logused	INT	当时最后使用此条目的当前日志的位置。服务器使用该标志和 logid 值在相同优先级别条目之间做出选择

2.3.5 tenant 表

tenant 表在 sysadmin 数据库中，它包含了有关 tenant 数据库的信息。

列	类型	描述
tenant_id	int	tenant 数据库的唯一 ID
tenant_dbsname	varchar(128)	tenant 数据库的名称
tenant_resources	bson	tenant 数据库的属性 将这列切换成 JSON 去查看该属性
tenant_last_updated	datetime year to second	tenant 数据库最近一次更新的时间点
tenant_comment	lvarchar(2048)	tenant 数据库的详细信息

2.4 磁盘结构和存储

本章内容

数据库服务器通过管理自己的 I/O 获得高性能。数据库服务器管理存储、搜索和检索。当数据库服务器存储数据时，它创建在以后搜索和检索数据时需要的结构。数据库服务器磁盘结构还存储和跟踪管理日志记录和备份所需的控制信息。数据库服务器包含确保数据一致性（物理和逻辑）所需的所有信息。

在阅读本章之前，请先熟悉 GBase 8s 管理员指南 中的 数据存储在哪里 一章中磁盘空间术语和定义。

本章讨论与磁盘数据结构相关的以下主题：

- 数据库空间的结构和存储
- 简单大对象的存储
- Sbspace 结构
- 时间戳记
- 数据库和表的创建：磁盘上发生什么

2.4.1 数据库空间结构和存储

本节研究数据库服务器用于在数据库空间中存储数据的磁盘结构和存储技术。

root dbspace 结构

作为磁盘空间初始化的一部分，数据库服务器初始化 root dbspace 初始 chunk 中的特定结构。

这些结构是：

- 12 个保留页

- 第一个 chunk 可用列表页
- tblspace tblspace
- 物理日志
- 逻辑日志文件
- 数据库 tblspace

ROOTNAME 、ROOTOFFSET 、ROOTPATH 和 ROOTSIZE 配置参数指定 root dbspace 初始 chunk 的大小和位置。如果 root dbspace 已镜像，那么 MIRROROFFSET 和 MIRRORPATH 配置参数指定镜像块的位置。有关这些参数的更多信息，请参阅 数据库配置参数。

可使用 gcheck -pe 命令来查看 root chunk 的结构。有关更多信息,请参阅 gcheck -ce 、-pe: 检查可用 chunk 列表。

保留页

Root Dbspace 的初始 chunk 的前 12 页是保留页。每个保留页包含数据库服务器所使用的特定控制和跟踪信息。

要获取保留页内容的列表，请执行 gcheck -pr 命令。要同时列出有关物理日志和逻辑日志页（包括物理日 志页）的信息，请执行 gcheck -pR 命令。

以下示例显示了时间间隔检查点的 gcheck -pr 输出：

```
        Time of checkpoint          10/25/2005 17:05:20
      Checkpoint Interval                 1234
```

数据库服务器也存储了当前配置参数的信息，这些信息存储在一个名为 PAGE_CONFIG 的保留页中。如果你用命令行更改参数配置信息，并在没有关闭和重启数据库服务器的情况下运行 gcheck -pr 命令，该命令输出的配置值将与保留页中当前的值不匹配。gcheck 程序用例会返回一个警告消息。

以下示例显示了 PAGE_CONFIG 保留页的输出。

```
...
Validating GBase 8s database server reserved pages - PAGE_CONFIG
    ROOTNAME                     rootdbs
    ROOTPATH                     /home/dyn_srv/root_chunk
    ROOTOFFSET                    4
    ROOTSIZE                     8000
    MIRROR                       0
    MIRRORPATH
    MIRROROFFSET                  0
    PHYSFILE                     1000
    LOGFILES                      5
    LOGSIZE                      500
```

MSGPATH	/home/dyn_srv/online.log
CONSOLE	/dev/ttyp5
...	...

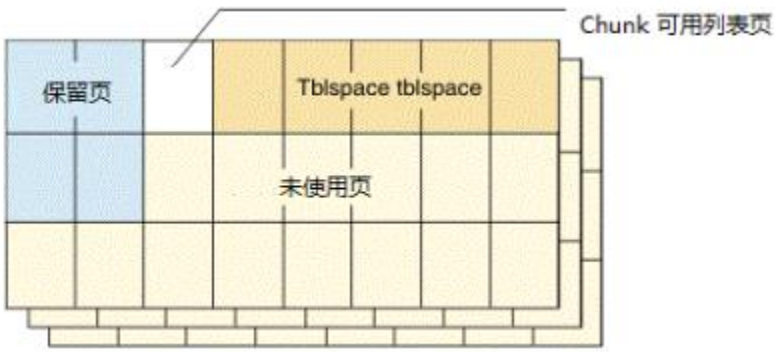
常规 Dbspace 的结构

磁盘空间初始化后，可添加新的 dbspace 。创建 dbspace 时，至少向 dbspace 指定一个 chunk（原始或格式化的磁盘空间）。该 chunk 称为 dbspace 的初始 chunk 。 图 1 说明了常规（非根）dbspace 初始 chunk 的结构。

首次创建 dbspace 时，它包含以下结构：

- 两个保留页
- chunk 中第一个 chunk 可用列表页
- 此 dbspace 的 tblspace tblspace
- 未使用页

图：常规 dbspace 的初始 chunk



附加 Dbspace Chunk 的结构

可以创建包含多于一个 chunk 的 dbspace 。Dbspace 中的初始 chunk 包含了数据库空间的表空间 tblspace 。附加 chunk 则不包含。当附加 chunk 第一次创建时，包含以下结构：

- 两个保留页
- 第一个 chunk 可用列表页
- 未使用页

图 1 说明了 Dbspace 中所有附加 chunk 的结构。（该结构也适用于 root dbspace 的附加 chunk 。）

图：附加 Dbspace Chunk

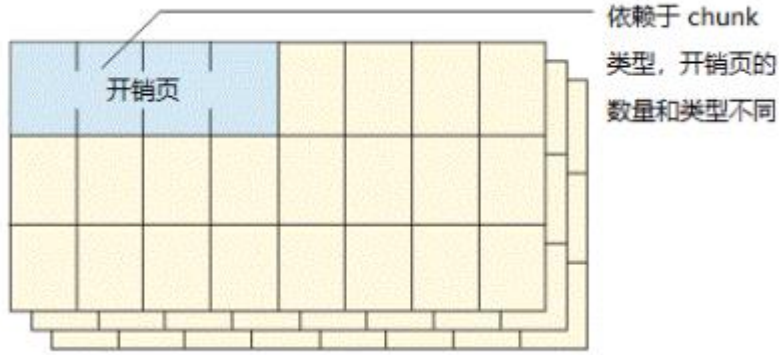
镜像 chunk 的结构

每个镜像 chunk 的大小必须与其主 chunk 相同。创建镜像 chunk 时，数据库服务器立即将主 chunk 的内容写入镜像 chunk 中。

镜像 chunk 包含于主 chunk 相同的控制结构。在数据库服务器将 blobspace 、sbspace 或 dbspace chunk 的镜像联机后，它们包含与其主对象相同的物理内容。

图 1 说明了 chunk 创建之后所显示的镜像 chunk 结构。

图：镜像 chunk 结构



由于镜像 chunk 的所有空间都保留用于镜像，所以镜像 chunk 结构总是显示没有可用空间。有关更多信息，请参阅 GBase 8s 管理员指南 中的 什么是镜像 一章。

chunk 可用列表页的结构

在每个 chunk 中，最后一个保留页后的页是跟踪 chunk 中可用空间的一个或多个 chunk 可用列表页的第一页。对于 non-root chunk ，可用空间的初始长度等于该 chunk 的大小减去 3 页。如果容纳新条目需要额外的 chunk 可用列表页，那么在 chunk 中的一个可用页中创建一个新的 chunk 可用列表页。图 1 说明了可用列表页的位置。

使用 gcheck -pe 获得 chunk 中页的物理布局。 有关更多信息，请参阅 gcheck -ce 、-pe：检查可用 chunk 列表.

图：可用列表页

tblspace tblspace 的结构

每个 dbspace 包含一个名为 tblspace tblspace 的表空间，它描述 dbspace 中的所有的 tblspace 。当数据库服务器创建 tblspace 时，它在表空间 tblspace 中放入一个条目，该条目描述新创建 tblspace 的特征。您不能删除或移动表空间 tblspace 的 chunk 。dbspace 最多可以有 2**20 个 tblspace 。

第一个 Extent 和下一个 Extent 的缺省大小取决于 dbspace 是否为 root dbspace ，如下表所示。

表 1. 每个 extent 的缺省大小和 dbspace 类型

dbspace 类型	第一个 Extent 的缺省大小	下一个 Extent 的缺省大小
root	● 500 KB （ 2 KB 页的系统） ● 1000 KB （ 4 KB 页的系统）	● 100 KB （ 2 KB 页的系统） ● 200 KB （ 4 KB 页的系统）
non-root	● 100 KB （ 2 KB 页的系统） ● 200 KB （ 4 KB 页的系统）	● 100 KB （ 2 KB 页的系统） ● 200 KB （ 4 KB 页的系统）

您可以通过以下方式为表空间 tblspace 的第一个 Extent 和下一个 Extent 指定非缺省的大小：

- 对于 root dbspace ，设置 TBLTBLFIRST 和 TBLTBLNEXT 配置参数。
- 对于 non-root dbspace ，在创建 dbspace 时请使用 gspaces 实用程序 -ef 和 -en 选项。

tblspace tblspace 条目

Tblspace tblspace 描述有关 tblspace 的特征。

要显示有关 tblspace 的信息，请使用 gcheck -pt 命令。

组件	描述
页头	24 字节，标准页头信息
页结束时间戳记	4 字节

组件	描述
Tblspace 头	136 字节，常规 tblspace 信息
Tblspace 名称	*database.owner.tablename* 或 *database.owner.indexname* 典型 30-40 字节长但是可以更长，取决于该名称的长度
列信息	8 字节 每一特定的列 特定列是指定义为 VARCHAR 、BYTE 、TEXT 或用户自定义数据类型
索引信息	对于连接的索引，该分区上的每个索引都包含 20 字节的头，头包含关于索引的一般信息。后跟索引每列组成部分的 4 字节条目 对于拆离索引，在该索引上每列包含 4 字节条目
Extent 信息	分配给该 tblspace 每个 extent : 10 字节条目 + 10 字节信息 在 tblspace 碎片整理的过程中，会使用更多的字节

tblspace 编号

tblspace tblspace 中描述的每个 tblspace 都接收到一个 tblspace 编号。该 tblspace 编号的值与 systables 系统目录表中的 partnum 字段和 sysfragments 系统目录表中 partn 字段所存储的值相同。

以下 SQL 查询检索数据数据库中每个表的 partnum（它们可位于几个不同的 dbspace 中）并将它显示为表名和 partnum 的十六进制表示法：

```
SELECT tabname, tabid, partnum, HEX(partnum) hex_tblspace_name FROM
systables
```

如果输出包括具有表名的行但 partnum 为 0，那么该表由两个或两个以上表分片组成，每分片为与其自己的 dvspace 中。例如： 图 1 显示了名为 account 但 partnum 为 0 的表。

图: 带有 partnum 值的 systables 查询的输出

tabname	tabid	partnum	hex_tblspace_name
sysfragments	25	1048611	0x00100023
branch	100	1048612	0x00100024
teller	101	1048613	0x00100025
account	102	0	0x00000000
history	103	1048615	0x00100027
results	104	1048616	0x00100028

要获取组成表的各个分片的实际 tblspace 编号，必须查询同一数据库的 sysfragments 表。 图 2 显示了 图 1 中的 account 表有三个表分片和三个索引分片。

图: 带有 partn 值的 sysfragments 表的输出

tabid	fragtype	partn	hex_tblspace_name
102	T	1048614	0x00100026
102	T	2097154	0x00200002
102	T	3145730	0x00300002
102	I	1048617	0x00100029
102	I	2097155	0x00200003
102	I	3145731	0x00300003

tblspace 编号元素

tblspace 中的第一页是逻辑页 0 。（物理页编号引用 chunk 中页的地址。）Root 空间表空间 tblspace 总是包含在第一个 dbspace 以及表空间 tblspace 的逻辑页 1 中。(位图页是页 0 。)

tblspace tblspace 大小

当初始化 dbspace 时，这些 tblspace tblspace 页作为 extent 进行分配。如果数据库服务器尝试创建表,但是 tblspace tblspace 已满,那么数据库服务器将向该 tblspace 分配下一个 extent 。

当从 dbspace 除去表时，也会删除该表在t blspace tblspace 的相应条目。

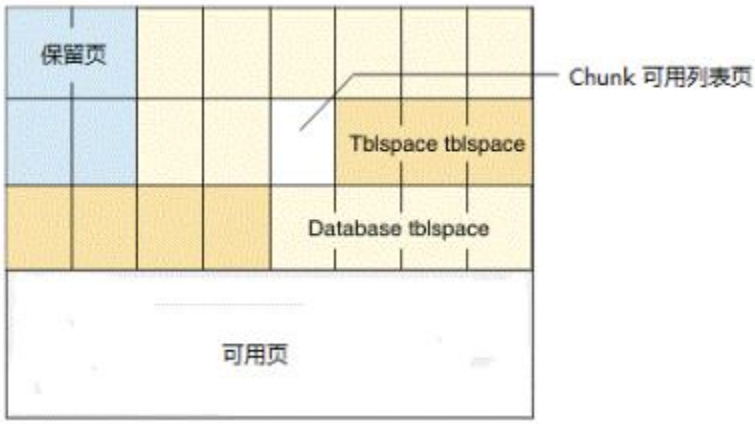
tblspace tblspace 位图页

tblspace tblspace 的第一页与任何初始 extent 的第一页一样，是描述接下来那些页的页充满度的位图。接下来的每一页在位图页上都有一个条目。如果需要，附加位图页位于分配给 tblspace 的整个邻接空间中，排列使得每个位图只描述紧跟它的页，知道下一位图或 dbspace 的结束。在 tblspace 页中，位图页以显著的间隔减少。每个位图页描述紧跟它的固定数量的页。

database tblspace 的结构

database tblspace 仅出现在 root dbspace 的初始 chunk 中。对于数据库服务器管理的每个数据库，database tblspace 都包含一个条目。图 1 说明了 database tblspace 的位置。

图: root dbspace 初始 chunk 中 database tblspace 的位置



database tblspace 编号

database tblspace 的 tblspace 编号始终是 0x100002。如果 database tblspace 是活动的，那么该 tblspace 编号出现在 gstat -t 列表中。

database tblspace 条目

每个database tblspace 条目包括以下 5 个组成部分：

- 数据库名
- 数据所有者
- 数据库创建的日期和时间
- 该数据库的 systables 系统目录表的 tblspace 编号
- 标识日志记录方式的标示

database tblspace 包含在数据库名上的唯一索引，以确保每个数据库是唯一命名的。对于任何数据库，systables 表描述数据库中每个永久表。因此，database tblspace 只指向位于任何其他地方的详细数据库信息。

当初始化 root dbspace 时，分配 database tblspace 的第一个 extent 。database tblspace 的初始 extent 大小和下一个 extent 大小为 4 页。 您不能修改这些值。

Extent 的结构和分配

本节涉及以下主题：

- Extent 结构
- Next-extent 分配

Extent 结构

Extent 是 dbspace 中邻近页的集合。每个永久数据库表都有两个 extent 大小与其相关联。初始 extent 大小是首次创建表时分配给表的千字节数。Next-extent 大小是当初始 extent 以及随后的每个 extent 变满时分配给表的千字节数。

blobspace 不使用 extent 。

有关如何指定和计算 extent 大小的特定指示信息，请参阅 GBase 8s 性能指南。

Extent 大小

第一个和下一个 extent 的缺省值是 16 千字节。如果在一个特别的 dbspace 中，它的转换比 4 页小，数据库服务器会使用最小的 extent 大小 — 4 页。 如果 dbspace 的空间为 8 千字节，而它要转换成 2 页，该数据服务器将增加 extent 大小到 32 千字节。

Extent 的最大大小是 2**31 页，等价于最大 chunk 大小。

如果 chunk 小于最大大小，那么最大 chunk 大小取决于 chunk 中可用的相邻空间。

保存索引分片的 tblspace 对 extent 大小遵循不同的规则。数据库服务器根据相应的表分片的 extent 大小来确定这些 dbspace 的 extent 大小。数据库服务器使用行大小与索引键大小的比率为索引 tblspace 指定适当的 extent 大小。（请参阅 GBase 8s 性能指南 中的估计索引页大小和分片表索引部分）。

分区中 extent 的最大数量是 32767 。

表 Extent 中的页类型

在 extent 中，个别页包含不同类型的数据。表的 Extent 页可以分成以下类别：

- 数据页

  数据页包含表的数据行。

- 位图页

  位图页包含 extent 中每页充满度的控制信息。

- blobpage

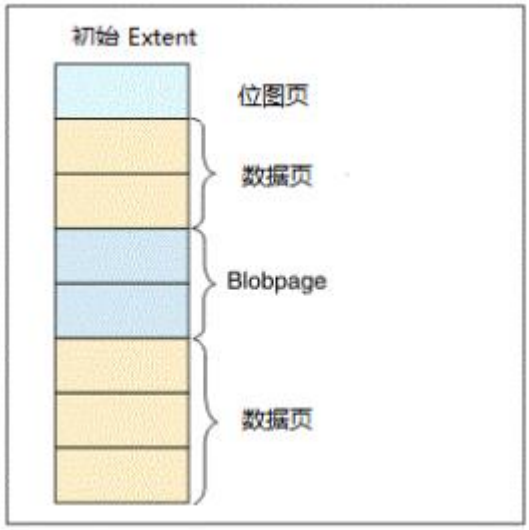
  blobpage 包含与数据行一起存储在 dbspace 中的 TEXT 和 BYTE 数据。驻留在 blobspace 中的 TEXT 和 BYTE 数据存储在 blobpage（一种与 dbspace blobpage 结构完全不同的结构）中。

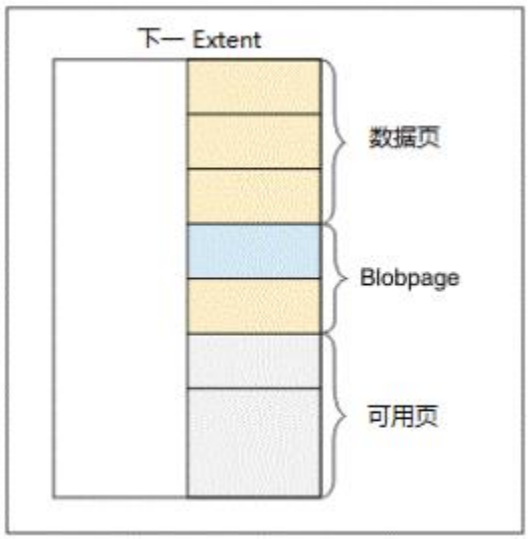
- 可用页

  可用页是 extent 中为 tblspace 使用所分配的，但其功能还未经过定义的页。可用页可用于存储任何类型的信息：数据（包括 TEXT 或 BYTE 数据类型）、索引或位图。

图 1 说明了非分片表的可能结构，它具有 8 页的初始化 extent 大小和 16 页的 next-extent 大小。

图: 表的 Extent 结构





索引 Extent 中的页类型

数据库服务器将索引页存储到不同于与其相关联的表的 tblspace 中。在 extent 中，个别索引页包含不同类型的数据。索引页可以分为以下类别：

- 索引页（根、分支和叶页）

  索引页包含表的索引信息

- 位图页

  位图页包含监视 extent 中每页充满度的控制信息

- 可用页

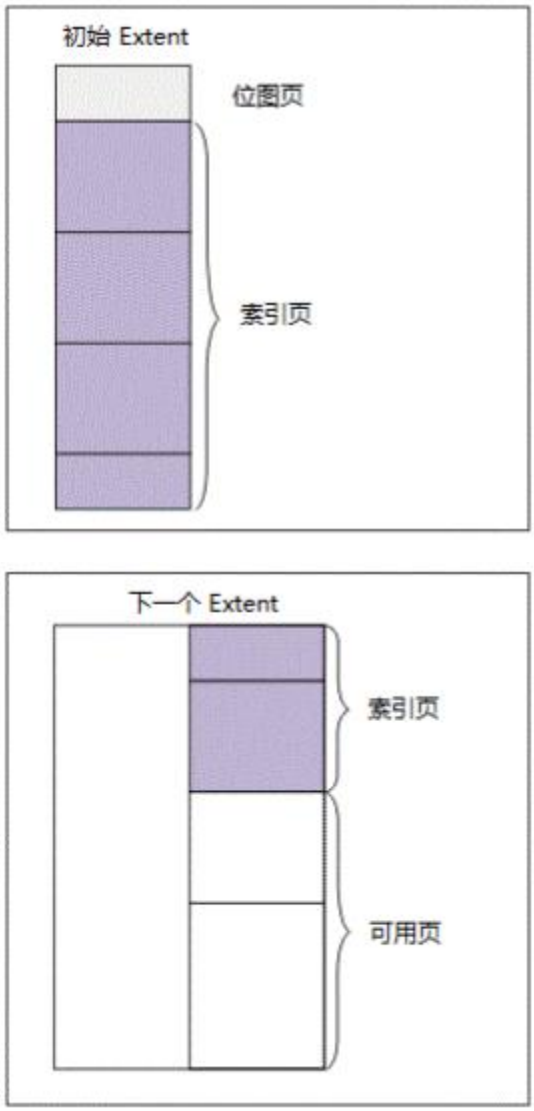
  可用页是 extent 中为 tblspace 使用所分配的，但其功能还未经过定义的页。可用页可用于存储任何类型的信息：数据、索引、 TEXT 或 BYTE 数据或位图。

除非显式指定了连接的索引，否则所有索引都是拆离的。

重要： 分配用于表分片的 Extent 不包含索引页。分片表的索引页总是驻留在单独的 tblspace 中。有关更多信息，请参阅 GBase 8s 管理员指南 中的表分片和 PDQ 一章中的表索引分片。

图 1 说明了索引的 extent 结构。

图: 索引的 Extent 结构



Next-Extent 分配

在初始化 extent 填满后，数据库服务器尝试分配相邻磁盘空间的另一个 extent 。数据库服务器所遵循的过程称为 next-extent 分配。

Tblspace 的 extent 作为表的 tblspace tblspace 信息的一个组成部分进行跟踪、分配用于任何 tblspace 的 extent 的最大数目依赖于应用程序和机器，因为它随 tblspace tblspace 条目上的可用空间量变化 。

Next-Extent 大小

数据库服务器分配用于下一个 extent 的千字节数一般等于 SQL 语句 CREATE TABLE 中所指定的下一个 extent 的大小。然而，next-extent 分配的实际大小肯会偏离指定大小，因为分配过程考虑以下三个因素：

- 该 tblspace 现有 extent 的数量
- Chunk 和 Dbspace 中的相邻空间的可用性
- 现有 tblspace extent 的位置

在以下段落和 图 1 中解释了这些因素中的每一个对 next-extent 分配的影响。

Extent 大小加倍

对于永久表或用户自定义临时表，所分配的下一个 extent 的大小都将自动加倍。将加倍到 128 千字节 (KB) 。例如：如果创建 NEXT SIZE 等于 15 KB 的表，那么数据库服务器将第一个 extent 的大小分配成 15 KB 。下一个 extent 的大小分配成 30 KB，下下个 extent 的大小分配成 60 KB。当 extent 的大小达到 128 KB ，当且仅当表中剩余空间少于表的总分配空间的 10% 时，extent 的大小才会加倍。

对于系统创建的临时表，next-extent 大小在添加了 4 个 extent 后开始加倍。

相邻空间的缺少

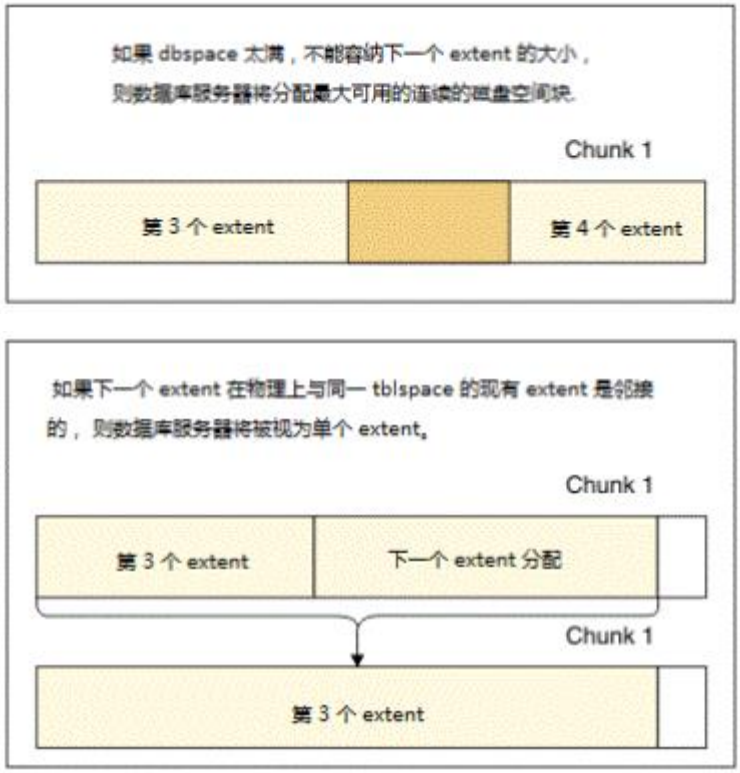
如果数据库服务器无法在第一个 chunk 中找到下一个 extent 指定大小的可用相邻空间，那么将搜索延伸到 dbspace 中的下一个 chunk 中。Extent 是不允许跨 chunk 的。

如果数据库服务器无法在 dbspace 的任何地方找到足够的相邻空间，那么它向该表分配最大可用数量的相邻空间。（最小的分配为 4 页。缺省值为 8 页）如果分配是可能的，那么即使所分配的空间量小于所请求的量，也不会返回任何错误消息。

同一表的 Extents 合并

如果分配用于下一个 extent 的磁盘空间是与已分配给相同表的磁盘空间物理相邻的，那么数据库服务器分配该磁盘空间但不将新分配视作单独 extent 。相反，数据库服务器扩展现有相邻 extent 的大小。此后，所有磁盘空间报告将该分配作为现有 extent 的扩展进行反映。即，所报告 extent 的数量始终是物理上不同 extent 的数量，而不是已分配的下一个 extent 的次数加 1 （初始 extent）。图 1 说明了 extent 分配策略。

图: Next-Extent 分配策略



在磁盘空间作为 extent 的一部分分配给 tblspace 后，该空间保持专用于该 tblspace ，即使其中包含的数据已删除也是如此。有关回收该磁盘空间的替代方法，请参阅 GBase 8s 性能指南。

dbspace 页的结构和存储

数据库服务器 I/O 的基本单位是页。页大小可由计算机而异。

在 GBase 8s 中，页大小依赖于操作系统。

非分片表中的行

数据库服务器可以存储长于一页的行。数据库服务器还支持 VARCHAR 数据类型，这导致可变长度的行。因此，行并不遵循单一格式。

如果表包含类型为 VARCHAR 的一列或多列，那么表中行的长度不一定相同。另外，在最终用户修改 VARCHAR 列中包含的数据时，此类表的行长度可能会变化。

行长度可能会大于一页。

TEXT 和 BYTE 数据并不存储在数据行中。相反，数据行包含指向数据位置的 56 字节的描述符。该描述符可以指向 dbspace 页。

描述符可以指向 blobspace blobpage。

有关如何估计固定长度和可变长度数据行的长度的指示信息，请参阅 GBase 8s 性能指南 。

rowid 的定义

GBase 8s 使用两种不同类型的 rowid 来标识表中的数据：

- 顺序 rowid

这些 rowid 是表中的字段且指定给使用 WITH ROWID 选项所创建的表。

- 内部 rowid

数据库服务器用唯一的内部 rowid 标识表中的每个数据行。该 rowid 用于标识行在数据库空间中的位置。

要获得表的内部 rowid ，请使用 gcheck -pD 选项。要了解更多信息，请参阅 gcheck -cd 和 gcheck -cD ： 检查页 。

在分片表中，术语 rowid 是制定义在表中的物理位置的唯一 4 字节整数。包含数据行第一字节的页是由 rowid 所指定的页。该页称为数据行主页。

分片表也可能有 rowid ，但它们是以另一种方式实现的。有关该主题的跟多信息，请参阅分片表中的行 。

rowid 的使用

非分片表中的每一行数据行都由一个不变的 rowid 唯一标识。创建非分片表的索引时，rowid 存储在该数据行所属的表相关联的索引页中。当数据库服务器需要数据行时，它搜索该索引来查找键值并使用相应的 rowid 来定位所请求的行。如果该表未经索引，那么数据库服务器可能会顺序读取表中的所有行。

最后，行可能超出其原始存储位置。如果发生这种情况，那么指向数据行新位置的转发指针将留在 rowid 所定义的位置。转发指针本身就是 rowid ，它定义数据行现在存储页上的页和位置。

分片表中的行

与非分片表中的行不一样，数据库服务器不 向分片表中的行指定 rowid 。如果想要按 rowid 访问数据，那么必须显式创建 rowid 列，如 GBase 8s 性能指南 中描述的那样。如果用户应用程序试图引用未包含显式创建的 rowid 的分片表中的 rowid ，那么数据库服务器向应用程序返回相应的错误代码。

以 rowid 访问分片表中的数据

从应用程序角度来看，分片表中 rowid 列与非分片表中 rowid 的功能是一样的。然而，与非分片表中 rowid 不一样，数据库服务器使用索引将 rowid 标识映射到物理位置上。当数据库服务器使用 rowid 列访问分片表中的行时，它在尝试访问该行前使用该索引查找该行的物理地址。对于非分片表，数据库服务器使用没有索引查找的直接物理访问。结果，使用 rowid 访问分片表中的行所花时间比使用 rowid 访问非分片表中的行所花时间稍长。由于维护分片表的 rowid 索引的成本，您还应期望插入和删除处理的性能影响较小。

主键访问可能导致许多情况下（特别是并行访问时）性能的显著提高。

关于使用 rowid 的建议

建议应用程序开发者使用主键而不是 rowid 作为存取方法。因为主键是在 SQL 的 ANSI 规范中定义的，所以使用它们访问数据使应用程序的可移植性更好。

有关如何定义和使用主键访问数据的完整描述，请参阅 GBase 8s SQL 参考指南 和 GBase 8s SQL 教程指南 。

数据行格式和存储

数据行的可变长度对于行存储产生以下结果：

- 一页可能包含一个或多个整行。
- 一页可能包含一个或多行的一部分。
- 一页可能包含整行和行的一部分的组合。
- 行的大小可能会在更新后增大，并变得太长而无法返回到行中原来的存储位置。

以下段落描述了在数据存储过程中数据库服务器所遵循的准则。

行的存储

为使检索时间最小，除非必要，否则不要跨页边界断行。短于一页的行总是作为整行进行存储的。当可用字节计数少于存储最大大小的行所需的字节数时，也被视为已满。

行的位置

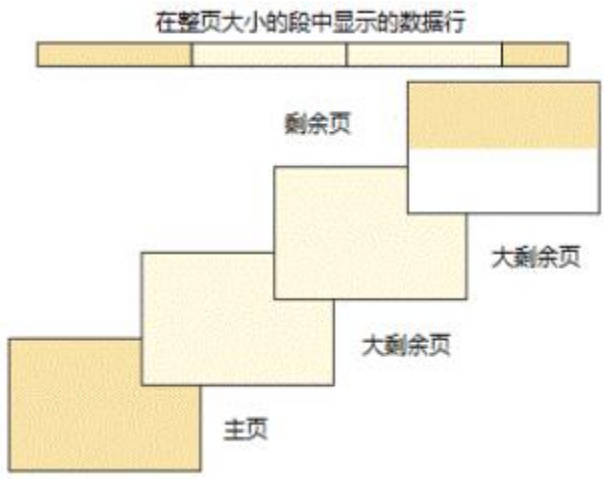
当数据库服务器接收长于一页的行时，该行存储在与所需要的同样多的整页中。然后，数据库服务器以小于一个满页的长度存储拖尾部分。

包含行的第一个字节的页是行主页。主页的编号成为 rowid 所包含的逻辑页号。主页后的每个满页称为大剩余页。如果行的拖尾部分小于一个满页，那么它存储在剩余页上。

在数据库服务器创建剩余页以容纳长行之后，它可以使用该页中剩余空间存储其他行。

图 1 说明了主页、大剩余页和剩余页的概念。

图：剩余页



页压缩

随着时间的流逝，页上的可用空间可能变成分段的。当数据库服务器试图存储数据时，它首先对照页上的可用字节数检查行长度，以确定该行是否适合。如果可以获得足够的空间，

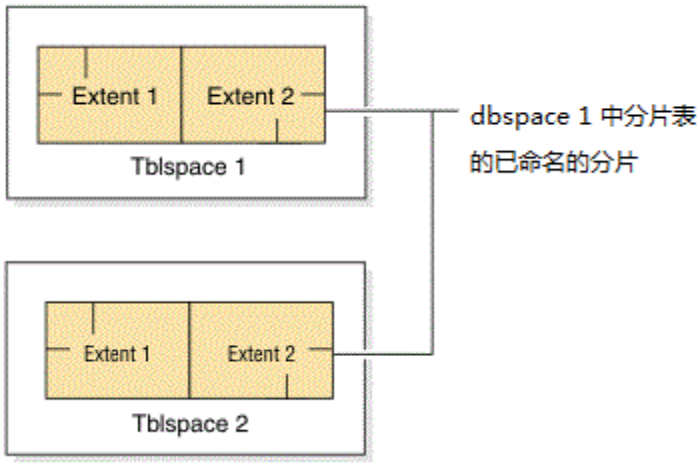
那么数据库服务器检查该页是否包含足够的相邻可用空间以容纳该行（或行的部分）。如果可用空间是不相邻的，那么数据库服务器调用页压缩。

分片表的结构

尽管表分片对应用程序是透明的，但作为数据库服务器管理员，您应知道数据库服务器是如何将磁盘空间分配于表分片以及数据库服务器是如何标识那些分片中的行的。

每个表分片都有自己的具有唯一 tblspace_id 或 fragment_id 的 tblspace 。图 1 显示了驻留在相同 dbspace 的不同分区上的分片表的磁盘分配。

图：分片表的磁盘结构



连接索引

如果使用连接索引，索引和数据的分片方式是一样的。您可以决定是将索引页与相应的数据页存储在同一个 dbspace 中，还是将它们存储在不同的 dbspace 中。有关选择分片存储策略的信息，请参阅 GBase 8s 性能指南 。

拆离索引

对于拆离索引，表分片和索引分片都存储在不同 dbspace 的 tblspace 中。

B-Tree 索引页的结构

本节提供有关 B-tree 索引页结构的一般信息。它是为对此结构感兴趣的读者编写的概述。有关 B-tree 索引的更多信息，请参阅 GBase 8s 性能指南 。

B-tree 术语的定义

数据库服务器使用 B-tree 结构组织索引信息。

图 1 显示了一个完全形态的 B-tree 索引是由以下三种不同类型的索引页或节点组成的：

- 一个根节点

  根节点包含指向分支节点的节点指针。

- 两个或多个分支节点

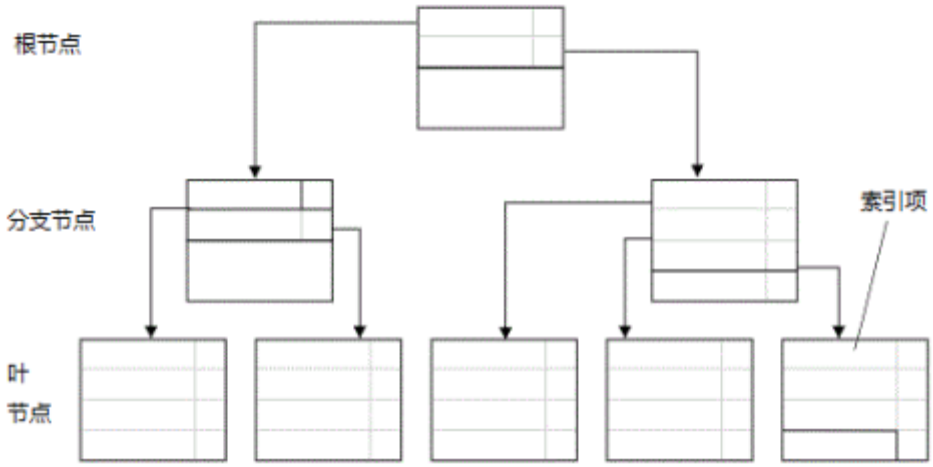
  分支节点包含指向叶节点或其他分支节点的指针。

- 多个叶节点

叶节点包含索引项和指向其他叶节点的水平指针。

每种节点都发挥不同的功能。以下各节描述了每种节点及其在索引建立中所扮演的角色。

图：完整的 B-Tree 结构



索引项

索引的基本单位是索引项 。索引项包含代表特定行的索引列值得键值。索引项还包含数据库服务器用于定位数据页中的行的 rowid 信息。

索引节点

节点是存储一组索引项的索引页。

作为代替传统 B-Tree 索引的树索引

不像传统的 B-tree 索引，一个树节点是一个更大的 B-tree 节点，它可以被划分成更小的子树与多个根节点和较少的层次。当您想减少根节点的连接并允许更多并发用户访问索引而无需等待时，您可以创建树节点来代替 B-tree 节点。

索引的逻辑存储

本节介绍数据库服务器如何创建并填充索引的概述。

根节点和叶节点的创建

当创建空表的索引时，数据库服务器分配一个索引页。该页代表根节点并保留为空，直到在该表中插入数据。

起初，根节点以与叶节点相同的方式发挥功能。对于插入表中的每一行，数据库服务器在根节点中创建并插入索引项。图 1 说明了根节点在其填充之前是什么样的。

图：根节点

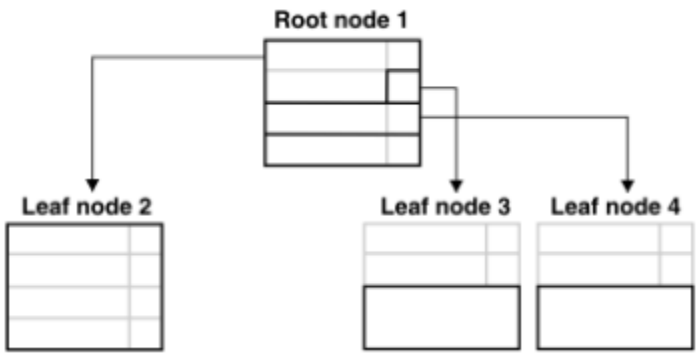
当根节点的索引项被填满时，数据库服务器通过以下步骤分割根节点：

- 创建两个叶节点
- 将大约一半的根节点条目移动至每个新创建的叶节点中
- 在根节点中放置叶节点的指针

当您将新行添加到表中时，数据库服务器将索引项添加到叶节点中。当叶节点填满时，数据库服务器创建新的叶节点，将已满索引节点的一部分内容移动到新节点中，并在根节点中添加指向新节点的节点指针，

例如：假如 图 2 图中的叶节点 3 已满。当发生这种情况时，数据库服务器添加另一个叶节点。数据库服务器从叶节点 3 将一部分记录移动至新的叶节点中，如图 图 2 所示。

图：在叶节点 3 填满后创建叶节点 4



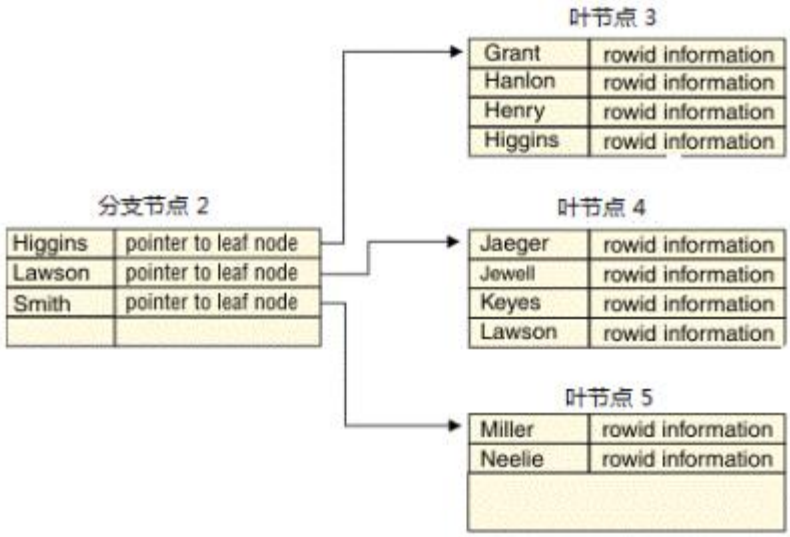
分支节点的创建

最终，当向表添加行时，数据库服务器用指向所有现有叶节点的节点指针填充根节点。当数据库服务器分割另一个叶节点，而根节点没有空间用于额外的节点指针时，那么发生以下过程。

数据库服务器分割根节点并将其内容分到两个新创建的分支节点上。随着索引项的添加，越来越多的叶节点被分割，这导致数据服务器添加更多的分支节点。最终，根节点充满了指向这些分支节点的指针。当发生这种情况时，数据库服务器重新分割根节点。然后，数据库服务器在根节点和较低的分支级别之间创建另一个分支级别。该过程导致 4 个级别的树，它带有一个根节点、两个分支级别和一个叶级别。B-tree 结构可以这种方式继续增长到最大 20 级。

分支节点可以指向其下面的其他分支节点（对于四级或更多的大索引来说）或指向叶节点。在 图 1 图中，分支节点只指向叶节点。左分支节点中的第一项包含与最左的叶节点中的最大项相同的键值以及指向它的指针。第二项包含下一叶节点中的最大项和一个指向它的节点指针。分支节点中的第三项只包含指向下一个更高叶节点的指针。根据索引的增长，在索引生命周期中的以后某个时点，除了指针之外，此第三项可能还包含实际键值。

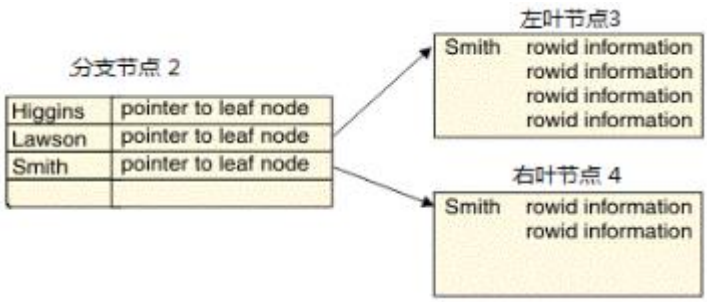
图：分支节点的典型内容



重复的键值

当多行的索引项列的值完全相同时，发生重复键值。例如：假设 B-tree 结构的第三个和第四个叶节点包含键值 Smith 。进一步假设该值重复 6 次，如 图 1 图说明的那样。

图：叶节点 3 和 4



第三个叶节点上的第一项包含重复的键值 Smith 以及包含重复键值的表中第一个物理行的 rowid 信息。为节省空间，第二项不重复键值 Smith 但只包含 rowid 信息。在整个页上继续该过程；页上不存在其他键值，只有 rowid 信息。

第四个叶页上的第一项也包含重复键值和 rowid 信息。后续项只包含 rowid 信息。

现在考虑分支节点。分支节点中的第三项包含与第三叶节点中的最大项相同的键值和 rowid ，并且有一个指向它的节点指针。第四项将只包含指向第四叶节点的节点指针，这样就节省了附加重复键值的空间。

键值锁定

为了增加并发性，数据库服务器支持 B-tree 索引中的 键值 锁定。键值锁定只锁定键的值而非 B-tree 索引中的物理位置。

键值锁定的一个最重要的用途是确保唯一键在直到删除它的事务结束时一直保持唯一。如果没有这种保护机制，用户 A 可能删除事务中的唯一键，用户 B 可能在事务提交之前插入具有相同键的行。此应用场合使得用户 A 不肯执行回滚。简直锁定阻止用户 B 插入行，直到用户 A 的事务结束。

相邻键的锁定

如果具有可重复读取的隔离级别，那么要求数据库服务器保护读取集。读取集由符合查询的 WHERE 子句中过滤条件的行组成。为保证这些行不改变，数据库服务器获得与读取集的最右端项相邻的索引项上的索。

释放的索引页

当数据库服务器从节点中物理除去索引项并释放索引页，那么释放的页是可再用的。

填充索引

创建索引时，可以指定希望的填充索引的密度。索引填充因子是索引构建过程中填充的每个索引页的百分率。使用 CREATE INDEX 语句的 FILLFACTOR 选项或 FILLFACTOR 配置参数设置填充因子。此选项对于预期在构建后不会增长的索引特别有用。有关 CREATE INDEX 语句的 FILLFACTOR 选项的其他信息，请参阅 《GBase 8s SQL 指南：语法》 。

计算索引的长度

对于非 VARCHAR 数据类型，索引项的长度的计算方法是：键值的长度加上 5 字节，这 5 字节用于描述与键值相关联的每个 rowid 信息。

索引中的键值长度通常是固定的。如果索引保存 VARCHAR 数据类型的一列或多列的值，那么键值的长度至少是键中每个 VARCHAR 值长度加一的总和。

在 GBase 8s 中，键值的最大长度是 390 字节。构成键的 VARCHAR 列的组合大小必须小于 390 减去每个 VARCHAR 列的额外字节。例如：数据库服务器构建用于以下语句的索引键长度等于 390 或 ((255+1) + (133+1))：

```
CREATE TABLE T1 (c1 varchar(255, 10), c2 varchar(133, 10));
CREATE INDEX I1 on T1(c1, c2);
```

函数索引

函数索引是一种其所有键都派生于函数结果的索引。例如：如果有一个图片列和一个标识主色的函数，那么可以对函数结果创建索引。这种索引将是您能快速检索具有相同主色的所有图片，而不用重新执行该函数。

函数型索引与任何其他 B-tree 树索引一样也使用 B-tree 结构。唯一的区别是只要是函数参数的那一列更改，就在插入或更新过程中应用确定函数。有关函数索引性质的更多相关信息，请参阅 GBase 8s 性能指南 。

要创建函数索引，请使用 CREATE FUNCTION 和 CREATE INDEX 语句。有关这些语句的更多信息，请参阅 《GBase 8s SQL 指南：语法》 。

R-Tree 索引页的结构

依赖于一维键值排序的索引结构对空间数据不起作用；例如：二维几何形状 （例如：圆、正方形和三角形）。对空间数据 （例如地理信息系统 （ GIS ）和计算机辅助设计（CAD）应用程序中使用的数据）的有效检索需要处理多维数据的存取方法。数据库服务器实现 R-tree 索引来有效存取空间数据。有关所引用结构的信息，请参阅 GBase 8s R-Tree 索引用户指南 。

2.4.2 简单大对象的存储

本节解释数据库服务器用于存取简单大对象 （TEXT 或 BYTE 数据）的结构和存储技术。

blobspace 的结构

当创建 blobspace 时，可以指定数据页的有效大小，这称为 blobpage 。Blobspace 的 blobpage 大小是在创建 blobspace 时指定的。Blobpage 大小必须是页大小的倍数。（有关确定数据库服务器页大小的信息，请参阅GBase 8s 管理员指南 中的 管理磁盘空间 一章。）Blobspace 中的所有 blobpage 的大小都是相同的，但 blobpage 的大小在 blobspace 之间可能变化。Blobpage 大小可能大于页大小，因为存储在 blobspace 中的数据从不写入共享内存中的大小为页的缓冲区中。

定制 blobpage 大小的好处是存储效率。在 blobspace 中，TEXT 和 BYTE 数据存储在一个或多个 blobpage 中，简单大对象不共享 blobpage 。当 TEXT 或 BYTE 数据等于或稍小于 blobpage 大小时，存储是最有效的。

Blobspace 自由图页和位图页是作为数据库服务器页指定的大小。这使它们能够读入共享内存和被记录下来。

当首次创建 blobspace 时，它包含以下结构：

- Blobspace 自由图页
- 用于跟踪自由图页的 blobspace 位图
- 未使用的 blobpage

dbspace blobpage 的结构

存储在 dbspace 中的 TEXT 或 BYTE 数据是存储在 blobpage 中的。Dbspace blobpage 的结构类似于 dbspace 数据页的结构。唯一的区别是可以与 TEXT 或 BYTE 数据一起存储在数据区域的额外 12 字节。

如果多个简单大对象可以适合一个页或如果简单大对象的多个拖尾部分可以适合页，那么简单大对象可以共享 dbspace blobpages 。

有关如何估计特定表所需的 dbspace blobpage 数量的讨论，请参阅 GBase 8s 性能指南 。

存储在 dbspace 页中的每段 TEXT 或 BYTE 数据前面可能有多达 12 字节的信息，这些信息不出现在任何其他 dbspace 页上。这些额外字节是开销的。

简单大对象存储和描述符

包含 TEXT 或 BYTE 数据的数据行在行本身中不包含数据。数据行包含 56 字节的描述符和一个指向数据第一个分段存储位置的转发指针（rowid）。

描述符可指向以下项之一：

- 页（如果数据存储在 dbspace 中）
- blobpage （如果数据存储在 blobspace 中）

简单大对象的创建

当插入含 TEXT 或 BYTE 数据的行时，首先创建简单大对象。在简单大对象写入磁盘之后，使用描述符更新行，并插入行。

简单大对象的删除或插入

数据库服务器无法修改简单大对象。它仅可以插入或删除它们。删除简单大对象意味着数据库服务器释放已删除对象所消耗的空间以再用。

更新 TEXT 或 BYTE 数据时，创建新的简单大对象，并使用新的 blob 描述符更新数据行。行的旧映象包含指向简单大对象的废旧值得描述符。将释放由废旧简单大对象所消耗的空间，用于提交更新之后的再使用。如果删除了包含其 blob 描述符的行，那么自动删除简单大对象。（存储已删除简单大对象的 Blobpage 不可复用，直到备份了包含已删除简单大对象的原始 INSERT 记录的逻辑日志。有关更多信息，请参阅 GBase 8s 管理员指南 中的什么是逻辑日志一章中的将逻辑日志文件被分为可用 blobpage 。）

简单大对象的大小限制

blob 描述符可以容纳的最大简单大对象是 $(2^{31} - 1)$ 或大约 2 千兆字节。

blobspace 页类型

每个 blobspace chunk 包含三种类型的项：

- Blobspace 自由图页
- 位图页
- Blobpage

blobspace自由图页

blobspace 自由图页标识未使用的 blobpage ，以便数据库服务器可作为简单大对象创建的一部分来分配它们。分配 blobpage 时，将更新该页的自由图条目。简单大对象的所有条码都是链接的。

Blobspace 自由位图页是一个数据库服务器页的大小，自由图页上的每一个条目都是 8 字节，作为两个 32 位字进行存储，如下所示：

- 第一字中的第一位指定 blobpage 是可用的还是已使用的。
- 第一字中接下来的 31 位标识写入此 blobpage 时的当前逻辑日志文件。（该信息是记录 TEXT 或 BYTE 数据所必需的。）
- 第二字包含与存储在该页上的简单大对象相关联的 tblspace 号。

可以适合自由图页的条目数依赖于您计算机的页大小。Blobspace chunk 中自由图页数依赖于 chunk 中 blobpage 的数量。

blobspace 位图页

blobspace 位图页跟踪 chunk 中 blobspace 自由图页的充满度和数目。每个 blobspace 位图页可以跟踪一定数量的位图页。Blobspace 位图页的大小取决于系统页的大小。如果系统大小为 2K ，blobspace 位图页可跟踪 2,032,128 个 blobpage 。如果系统页大小为 4K ，blobspace 位图页可以跟踪 8,258,048 个 blobpage 。

blobpage

blobpage 包含 TEXT 或 BYTE 数据。blobpage 大小由创建 blobspace 的数据库服务器管理要指定。blobpage 大小指定为页大小的倍数。

blobspace blobpage 的结构

用于在 blobspace 中存储简单大对象的存储策略不同于 dbspace 存储策略。数据库服务器不在单个 blobspace blobpage 上组合整个简单大对象或简单大对象的某些部分。例如：如果每个 blobspace blobpage 为 24 千字节，那么 26 千字节的简单大对象存储在两个 24 千字节的页上。额外的 22 千字节空间保持未使用。

blobpage 的结构包括 blobpage 头、 TEXT 或 BYTE 数据以及页结束时间戳记。blobpage 头除了其他信息还包含与数据行中转发指针相关联的页头时间戳记和 blob时间戳记。如果简单大对象存储在多个 blobpage 上，那么 blobpage 头中还包含下一 blobpage 的转发指针和另一个 blobpage 时间戳记。

2.4.3 sbspace 结构

sbspace 类似于 blobspace ，除了它拥有智能大对象外。

当 sbspace 在数据库创建时，它包含一个 sbspace 描述符。每个 sbspace chunk 包含以下结构：

- Sbspace chunk 描述符

- Chunk 可用页列表
- Sbspace元数据区域（每个 chunk 最多一个）
- 保留数据区域（每个 chunk 最多两个）
- 用户数据区域（每个 chunk 最多两个）

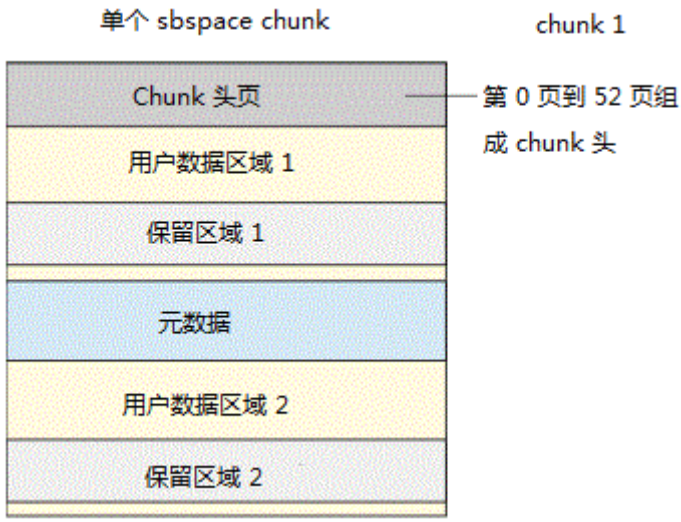
为了获得最佳性能，建议元数据区域位于 sbspace 的中间。数据库服务器自动将元数据区域置于正确的位置。然而，要指定元数据区域的位置，请在 gspaces 命令中指定 -Mo 标志。

如果您未在 gspaces 命令的 -Ms 标志中指定元数据区域的大小，那么数据库服务器使用 AVG_LO_SIZE 的值（缺省值为 8千字节）计算元数据区域的大小、有关更多信息，请参阅 使用 -Df 选项创建 sbspace 。

正常情况下，可以让系统计算元数据的大小。如果想要估计元数据区域的大小，请参阅 GBase 8s 性能指南 中的 表性能注意事项 一章。

图 1 图说明了 sbspace 中 chunk 的结构，它是在 sbspace 创建之后立即显示的，每个保留区域可以分配给用户数据或元数据区域。保留其总是在 chunk 的用户数据区域中。

图：一个简单的 sbspace chunk



由于 图 1 图中的 chunk 是 sbspace 的一个 chunk ，所以它包含一个 sbspace 描述符。chunk 1 中的 chunk 描述符 tblspace 包含有关 chunk 1 和其后添加到 sbspace 中的所有块的信息。

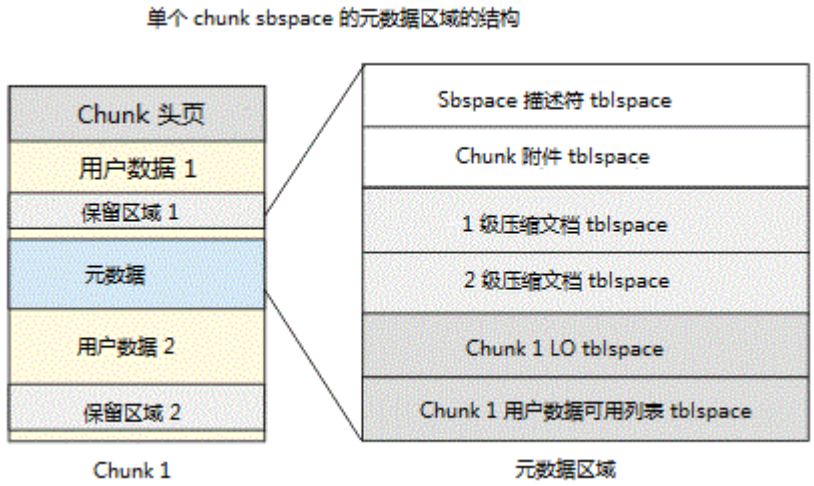
元数据区域的结构

每个 sbspace 包含在其中 chunk 的元数据区域。

对于 chunk 头页，四个区域专用于 sbspace 中的第一 chunk ：Sbspace 描述符 tblspace 、chunk 附件 tblspace 以及 1 级 和2 级压缩文档 tblspaces。Tblspace 头

部分包含这些 tblspace 中的每一个 tblspace 头（特别的，tblspace tblspace 除外）。图 1 显示了单块 sbspace 中元数据的布局。

图：单个 chunk sbspace 的元数据区域结构



在 gcheck -ps 选项中指定 Sbspace 名称时，可以显示分配并用于元数据区域中的每个 tblspace 的页数。

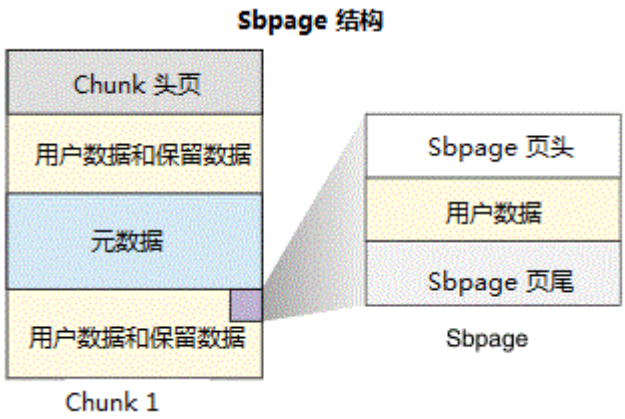
以下内容描述元数据区域是如何增长的：

- sbspace 描述符 tblspace 不增长。
- chunk 附件 tblspace 在添加 chunk 时增长。
- LO 头 tblspace 在添加 chunk 时增长。
- 如果 chunk 中的可用空间已大量分片，那么用户数据可用列表的 tblspace 增长。

sbpage 结构

每个 sbpage 由三个元素组成：sbpage 头、实际用户数据和 sbpage 跟踪器。 图 1 图显示了 sbpage 的结构。Sbpage 头由标准页头构成。sbpage 跟踪器用于检测页上的不完全写入和检测页毁坏。

图：sbpage 结构



2.4.4 时间戳记

数据库服务器使用时间戳记来标识事件对于同一类型的其他事件发生时的时间。时间戳记不是引用特定小时、分钟或秒的字面时间。它是数据库顺序指定的 4 字节整数。

2.4.5 数据库和表的创建：磁盘上发生什么

本节解释数据库服务器如何存储于数据库或表的创建相关的数据以及如何分配存储数据所必需的磁盘结构。

数据库的创建

在 root dbspace 存在之后，用户可以创建数据库。以下各段描述当数据库服务器添加新的数据库时磁盘上所发生的主要事件。

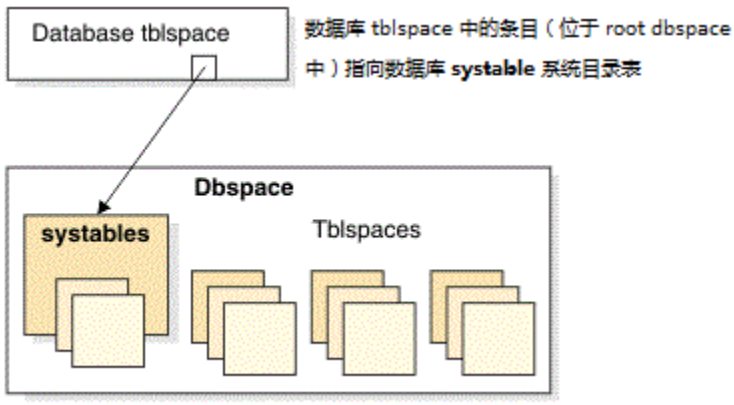
对系统目录表的磁盘空间分配

数据库服务器搜索 dbspace 中的 chunk 可用列表页，以查找要在其中创建系统目录表的可用空间。接下来，对于每个系统目录表，数据库服务器分配 8 个连续页，即每个系统目录表的初始 extent 的大小。这些表是分别创建的，不一定以彼此邻接的方式驻留在 dbspace 中。它们可以位于不同的 chunk 中。当为每张表的初始 extent 找到足够的空间时，分配这些项，并更新相关联的 chunk 可用列表页。

对系统目录表的跟踪

数据库服务器跟踪在数据库 tblspace （驻留在 root dbspace 中）中新创建的数据库，描述该数据库的条目添加到 root dbspace 中的 数据库 tblspace 中。（请参阅 database tblspace 的结构。）对于每个系统目录表，数据库服务器向构建数据库的 dbspace 中的 tblspace tblspace 添加一个一页的条目。（请参阅Tblspace Tblspace 的结构。） 图 1 图说明了数据库 tblspace 条目和该数据库的 systables 系统目录表的位置之间的关系。

图：新数据库



有关如何在创建数据库之后如何列出它们的指示信息，请参阅 GBase 8s 管理员指南 中的管理数据库日志状态一章中的监视数据库。

表的创建

在 root dbspace 存在并创建了一个数据库之后，具有必需 SQL 特权的用户可以创建数据库表。当用户创建表时，数据库服务器以名为 extent 的单位向表分配磁盘空间。（请参阅 GBase 8s 管理员指南 中的数据存储在哪里一章中什么是 extent ）。以下各段描述当数据库服务器创建表并分配磁盘空间的初始 extent 时所发生的主要事件。

磁盘空间分配

数据库服务器搜索 dbspace 中的 chunk 可用列表页，以查找等于表的初始 extent 大小的连续可用空间。当找到足够空间时，分配这些页并更新相关联的 chunk 可用列表页。

如果数据库服务器无法在 dbspace 的任何地方找到足够的相邻空间，那么它向该表分配最大可用数量的相邻空间。如果分配是可能的，那么即使所分配的空间量小于所请求的量，也不会返回任何错误信息。如果无法分配最小 extent 大小，那么返回错误。（Extent 不能跨两个 chunk 。）

tblspace tblspace 中的条目

数据库服务器向此 dbspace 中的 tblspace tblspace 添加一个该表的一页项目。指定给该表的 tblspace 编号是从描述该表的 tblspace tblspace 中的逻辑页号派生出来的。请参阅 tblspace 编号 。

tblspace 编号标识 tblspace 所位于的 dbspace 。tblspace extent 可位于任何 dbspace chunk 中。

如果必须确切知道 tblspace extent 的位置，请执行 gcheck -pe 命令，以获得按 chunk 排列的 dbspace 布局列表。

系统目录表中的条目

表本身是在数据库系统目录表中所存储的条目中完全描述的。每张表都指定由表标识或 tabid 。数据库中第一个用户定义的表对象的 tabid 值总是为 100 。（ tabid = 100 的对象也可能是一个视图、同义词或者一个序列。）有关系统目录的全面讨论，请参阅 GBase 8s SQL 参考指南 。

表可以位于与包含数据库的 dbspace 不同的 dbspace 中。Tblspace 本身是所分配 extents 的总和，而不是空间的单个连续分配，数据库服务器跟踪 tblspace 而不依赖于数据库。

临时表的创建

创建临时表所涉及的任务类似于数据库服务器添加新的永久表时所执行的任务。主要区别在于临时表不接收数据库的系统目录中的条目。有关更多信息，请参阅 GBase 8s 管理员指南 中的数据存储在哪里一章中的定义临时表一节。

2.5 解释逻辑日志记录

本章内容

要显示逻辑日志文件所包含的逻辑日志记录，请使用 glogdump 实用程序。

本章提供以下信息：

- 阅读逻辑日志记录的简要指南
- 不同逻辑日志记录类型的列表

一般来说，您无需阅读和理解逻辑日志文件。然而在调试情况下，glogdump 输出是有用的。例如：您可能想要使用 glogdump 跟踪特定事务或查看数据库服务器对特定 tblspace 进行了什么更改。也可以使用 glogdump 调查前滚过程中所发生的错误的原因。有关更多信息，请参阅 glogdump: 显示逻辑日志内容 。

2.5.1 关于逻辑日志记录

大多数 SQL 语句生成多个逻辑日志记录。当数据库服务器在逻辑日志中记录以下事件时，解释逻辑日志记录就更为复杂：

- 删除表或索引的事务
- 回滚事务
- 在其中事务任活动的检查点
- 分布式事务

以下各节讨论这些事务的逻辑日志记录。

删除表或索引的事务

一旦数据库服务器从数据库中删除了表或索引，它无法回滚该删除操作。如果事务包含 DROP TABLE 或 DROP INDEX 语句，那么数据库服务器如下处理该事务：

1. 数据库服务器完成事务的所有其他部分，并写下相关的逻辑日志记录。
2. 数据库服务器将 BEGCOM 记录写入逻辑日志和与 DROP TABLE 或 DROP INDEX 相关联的记录（例如：DINDEX ）。
3. 数据库服务器写入 COMMIT 记录。

如果事务在数据库服务器将 BEGCOM 记录写入逻辑日志之后意外终止，那么数据库服务器在恢复过程中前滚该事务，因为它无法回滚删除操作。

回滚的事务

当发生回滚时，数据库服务器为回滚的逻辑日志中的每个记录生成补偿日志记录（CLR）。如果在回滚过程中发生系统故障，那么数据库服务器使用 CLR 。CLR 向数据库服务器提供有关在故障发生前回滚进度的信息。换句话说，数据库服务器使用 CLR 来记录回滚。

如果 CLR 包含短语 includes next record ，那么打印的下一条日志记录作为补偿操作包含在 CLR 日志记录中。否则，必须假定补偿操作是 CLR 的 link 字段所指向的日志记录的逻辑撤销。

带有活动事务的检查点

如果任何事务在检查点时是活动的，那么检查点记录包含使用以下各列描述每个活动事务的子条目：

- 日志开始（十进制格式）
- 事务 ID（十进制格式）
- 唯一日志号（十进制格式）
- 日志位置（十六进制格式）
- 用户名

分布式事务

当分布式事务（跨多个数据库服务器的事务）生成日志记录，它们与非分布式事务稍有不同。如果事务提交时发生故障，那么可能需要阅读并解释它们，以确定这两个数据库服务器上事务的状态。

分布式事务中涉及以下日志记录：

- BEGPREP
- ENDTRANS
- HEURTX
- PREPARE
- TABLOCKS

有关这种类型的逻辑日志记录的更多信息，请参阅 GBase 8s 管理员指南 中有关两阶段落实和逻辑日志记录的资料。

如果正在使用 TP/XA 执行分布式事务，那么数据库服务器使用 XAPREPARE 记录而不是 PREPARE 记录。

2.5.2 逻辑日志记录的结构

每个逻辑日志记录都有头信息，根据记录类型，输出中也会出现其他信息列，就像逻辑日志记录类型和附加列中解释的那样。

逻辑日志记录头

表 1 包含了说明显示逻辑日志记录的头列的样本输出。

表 1. glogdump 的样本输出

addr	len	type	xid	id	link
2c018	32	BEGIN	6	3	0
2c038	140	HDELETE	6	0	2c018
2c0c4	64	DELITEM	6	0	2c038

addr	len	type	xid	id	link
2c104	40	DELITEM	6	0	2c0c4
2c12c	72	HDELETE	6	0	2c104
2c174	44	DELITEM	6	0	2c12c
2c1a0	72	HDELETE	6	0	2c174
2c1e8	44	DELITEM	6	0	2c1a0
2c214	64	HDELETE	6	0	2c1e8
2c254	56	DELITEM	6	0	2c214
2c28c	48	DELITEM	6	0	2c254
2c2bc	24	PERASE	6	0	2c28c
2c2d4	20	BEGCOM	6	0	2c2bc
2c2e8	24	ERASE	6	0	2c2d4
2c300	28	CHFREE	6	0	2c2e8
2c31c	24	COMMIT	6	0	2c300

表 2 定义了每个头列的内容。

头字段	内容	格式
addr	日志记录地址（日志位置）	十六进制
len	记录长度（字节）	十进制
type	记录类型的名称	ASCII
xid	事务编号	十进制
id	逻辑日志编号	十进制
link	到事务中的前一记录的链接	十六进制

逻辑日志记录类型和附加列

除了对给个记录显示 6 个头列之外，一些记录类型还显示信息的附加列。信息的显示会根据记录类型而变化。

下表列出了所有记录类型及它们附加列。

操作列标识生成日志条目的数据服务器操作类型。附加列和格式列描述逻辑日志记录头中描述的头以外每个记录类型还出现什么信息。

记录类型	操作	附加列和格式
ADDCHK	添加 chunk	● chunk 编号 - Decimal ● chunk 名称 - ASCII
ADDDBS	添加 dbspace	● dbspace 名称 - ASCII
ADDITEM	将条目添加到索引中	● tblspace ID - Hexadecimal ● rowid - Hexadecimal ● 逻辑页 - Decimal ● 键号 - Decimal ● 键长 - Decimal
ADDLOG	添加日志	● 日志编号 - Decimal ● 日志长度（页）- Decimal ● 页号 - Hexadecimal
ALLOCGENPG	分配类属页	● tblspace ID - Decimal ● rowid - Decimal ● slot 标识和长度 - Decimal ● 页的版本（如果删除的话）- Decimal ● 标识，vimage 记录 - Decimal ● 上一个 rowid - Decimal ● 数据 - ASCII
ALTERDONE	对分片的更改完成	● tblspace ID - Hexadecimal ● 上一页的物理页号 -

记录类型	操作	附加列和格式
		Hexadecimal ● 逻辑页号 - Decimal ● 更改的版本 - Decimal
ALTSPCOLSNEW	变更表中的已更改列	● 列的数量 - Decimal ● 特殊列列表 - 数组
ALTSPCOLSOLD	变更表中的已更改列	● 列的数量 - Decimal ● 特殊列列表 - 数组
BADIDX	错误的索引	● tblspace ID - Hexadecimal
BEGCOM	开始提交	● （无） - （无）
BEGIN	开始工作	● 数据 - Decimal ● 时间 - Decimal ● SID - Decimal ● 用户 - ASCII
BEGPREP	由协调者数据库服务器所写入，以记录两阶段提交协议的开始	● 标识 - Decimal（在分布式事务中值为 0） ● 参与者的数量 - Decimal
BEGWORK	开始事务	● 开始事务的时间 - Decimal ● 用户 ID - Decimal ● 进程 ID - Decimal
BFRMAP	简单大对象自由图页更改	● tblspace ID - Hexadecimal ● bpageno - Hexadecimal ● 状态 USED/FREE 日志 ID - Decimal ● 上一页 - Hexadecimal
BLDCL	构建 tblspace	● tblspace ID - Hexadecimal ● fextsize - Decimal

记录类型	操作	附加列和格式
		● nextsize - Decimal ● 行大小 - Decimal ● ncolumns - Decimal ● 表名 - ASCII
BMAPFULL	位图已修改以准备用于变更	● tblspace ID - Hexadecimal ● 位图页号 - Decimal
BMAP2TO4	2 位位图更改成 2 个 4 位位图	● tblspace ID - Hexadecimal ● 2 位位图页号 - Decimal ● 标识 - Decimal
BSPADD	添加 blobspace	● blobspace 名称 - ASCII
BTCPYBCK	将子键复制回父键	● tblspace ID - Hexadecimal ● 父逻辑页 - Decimal ● 子逻辑页 - Decimal ● slot - Decimal ● 行关闭 - Decimal ● 键号 - Decimal
BTMERGE	合并 B-tree 节点	● tblspace ID - Hexadecimal ● 父逻辑页 - Decimal ● 左逻辑页 - Decimal ● 右逻辑页 - Decimal ● 做 slot - Decimal ● 左行关闭 - Decimal ● 右 slot - Decimal ● 右行关闭 - Decimal ● 键号 - Decimal
BTSHUFFL	拖拽 B-tree 节点	● tblspace ID - Hexadecimal

记录类型	操作	附加列和格式
		● 父逻辑页 - Decimal ● 右逻辑页 - Decimal ● 左逻辑页- Decimal ● 左 slot - Decimal ● 左行关闭 - Decimal ● 键号 - Decimal ● 标识 - Hexadecimal
BTSPLIT	分割 B-tree 节点	● tblspace ID - Hexadecimal ● rowid - Hexadecimal ● 父逻辑页 - Decimal ● 左逻辑页 - Decimal ● 右逻辑页 - Decimal ● 无限逻辑页 - Decimal ● 根左逻辑页 - Decimal ● 中间分割 - Decimal ● 键号 - Decimal ● 键长 - Decimal
CDINDEX	创建拆离索引	● 数据库名 - ASCII ● 所有者 - ASCII ● 表名 - ASCII ● 索引名 - ASCII
CDR	捕获由更新语句修改过得表列组，例如：*bitvector* 。此日志记录允许 Enterprise Replication 只捕获更改过得列传输给目标站点。 在该示例中，表的前 6 列从未更改过（**bitvector** 中的最左面 6 位是 0 ），第 7 和第 8 列更新过（第7 和 8 位是 1 ）等等。	● CDR 记录的名称 - ASCII ● 分区号 - Hexadecimal ● bitvector - 二进制 CDR 日志记录的样本 **glogdump** 输出： adr len type xid id link 40 36 CDR 14 0 18

记录类型	操作	附加列和格式
	glogdump 输出显示了与输出的单行中相符的同样多的 bitvector 位。要查看以十六进制显示的整个 **bitvector** ，请使用 **glogdump -l** 命令	name partno bitvector UPDCOLS 10009a 000000110100110100
CHALLOC	Chunk extent 分配	● 页号 - Hexadecimal ● 大小 - Hexadecimal
CHCOMBINE	Chunk extent 结合	● 页号 - Hexadecimal
CHFREE	Chunk extent 释放	● 页号 - Hexadecimal ● 大小 - Hexadecimal
CHKADJUP	更新磁盘上的 chunk 附件。当从保留区域移动空间至元数据或用户数据区域或当用户添加 sbspace chunk 时，数据库服务器写入该记录	● chunk 编号 - Integer ● ud1_start_page - Integer ● ud1_size - Integer ● md_start_page - Integer ● md_size - Integer ● ud2_start_page - Integer ● ud2_size - Integer ● 标志 - Hexadecimal
CHPHYLOG	更改物理日志位置	● 页号 - Hexadecimal ● 大小（千字节） - Hexadecimal ● dbspace 名 - ASCII
CHRESERV	保留用于元数据窃取的 extent 。该记录是在添加 sbspace 块时写入的。	● chunk 编号 - Integer ● 页号 - Integer ● 长度 - Integer
CHSPLIT	Chunk extent 分割	● 页号 - Hexadecimal
CINDEX	创建索引	● tblspace ID - Hexadecimal ● 长的 rowid - Decimal ● 高的 rowid - Decimal ● 索引描述符 - ASCII

记录类型	操作	附加列和格式
COARSELOCK	粗粒度锁定	● tblspace ID - Hexadecimal ● 旧的粗粒度锁定标志值 - Decimal ● 新的粗粒度锁定标志值 - Decimal
CKPOINT	检查点	● 最大用户数 - Decimal ● 活动事务的数量 - Decimal
CLR	补偿日志记录:在回滚过程中创建	● （无） - （无）
CLUSIDX	创建群集索引	● tblspace ID - Hexadecimal ● 键号 - Decimal
COLREPAI	调整 BYTE、 TEXT 或 VARCHAR 列	● tblspace ID - Hexadecimal ● 已调整列的数量 - Decimal
COMMIT	提交工作	● 日期 - Decimal ● 时间 - Decimal
COMTAB	压缩页上的 slot 表	● 逻辑页编号 - Decimal ● 已移动 slots 的数量 - Decimal ● 已压缩的 slot 对 - ASCII
COMWORK	结束事务并提交工作	● 结束事务时间 - Decimal ● 开始事务时间 - Decimal
DELETE	删除前映象	● tblspace ID - Hexadecimal ● rowid - Hexadecimal
DELITEM	从索引中删除条目	● tblspace ID -

记录类型	操作	附加列和格式
		Hexadecimal ● rowid - Hexadecimal ● 逻辑页 - Decimal ● 键号 - Decimal ● 键长 - Decimal
DERASE	删除关闭 dbspace 中的 tblspace	● tblspace 编号 - Hexadecimal ● 表锁编号 - Decimal
DFADDEXT	已添加的下一个 extent	● partnum - Hexadecimal ● 列表中 extent 的偏移量 - Hexadecimal ● 页中的 extent 大小 - Decimal ● extent 的物理地址 - Offset and chunk no-hex
DFDRPEXT	删除源 extent	● partnum - Hexadecimal ● 列表中 extent 的偏移量 - Hexadecimal ● extent 源大小 - Decimal ● 物理地址 - offset and chunk no-hex
DFEND	结束分片的操作	● partnum - Hexadecimal
DFMVPG	将页从旧的 extent 移到新的 extent	● partnum - Hexadecimal ● 新的 extent 的偏移量 - Hexadecimal ● 资源的逻辑页号 - Hexadecimal ● 目的地的物理地址 - Offset and chunk no-hex ● 资源的逻辑地址 - Offset and chunk no-hex

记录类型	操作	附加列和格式
DFREMDUM	移除虚拟条目	● partnum - Hexadecimal
DFSTART	分片开始操作	● partnum - Hexadecimal
DINDEX	删除索引	● tblspace ID - Hexadecimal ● 键号 - Decimal
DRPBSP	删除 blobspace	● blobspace 名 - ASCII
DRPCHK	删除 chunk	● chunk 编号 - Decimal ● chunk 名 - ASCII
DRPDBS	删除 dbspace	● dbspace 名称 - ASCII
DRPLOG	删除日志	● 日志编号 - Decimal ● 日志大小（页）- Decimal ● 页号 - Hexadecimal
ENDTRANS	由协调者和参与者数据库服务器写入以记录事务的结束。ENDTRANS 指示数据库服务器从其共享内存事务表中除去事务条目并关闭该事务。 在协调者逻辑日志中，导致已提交事务的每个 BEGPREP 是与 ENDTRANS 记录配对的。如果协调者的最后决定时回滚事务，那么不写入 ENDTRANS 记录。 在参与者逻辑日志中，每个 ENDTRANS 记录是与 HEURTX 记录配对的	● （无） - （无）
ERASE	删除 tblspace	● tblspace ID -Hexadecimal
FREE_RE	将保留 extent 的 extent 分配给 sbspace chunk 的元数据或用户数据区域	● chunk 编号 - Integer ● 页号 - Integer ● 长度 - Integer ● 标志 - Hexadecimal

记录类型	操作	附加列和格式
HDELETE	删除主行	● tblspace ID - Hexadecimal ● rowid - Hexadecimal ● slotlen - Decimal
HEURTX	由参与者数据库服务器写入，以记录回滚事务的试探性决策。它应与指示事务已回滚的标准 ROLLBACK 记录相关联	● 标志 - Hexadecimal（值始终是 1 ）
HINSERT	主行插入	● tblspace ID - Hexadecimal ● rowid - Hexadecimal ● slotlen - Decimal
HUPAFT	主行更新，后映象	● tblspace ID - Hexadecimal ● rowid - Hexadecimal ● slotlen - Decimal
HUPBEF	主行更新，前映象 此外，HUPBEF 头的标志字段可能包含以下值： LM_PREVLSN 确认存在 LSN LM_FIRSTUPD 确认这是该事务对此 rowID 的首次更新	● tblspace ID - Hexadecimal ● rowid - Hexadecimal ● slotlen - Decimal ● LSN（可选）- Decimal
HUPDATE	如果主行更新前映象和后映象都可以适合单个页，那么数据库服务器写入单个 HUPDATE 记录。 此外， HUPDATE 日志的标志字段可能包含以下值： LM_PREVLSN 确认存在 LSN	● tblspace ID - Hexadecimal ● rowid - Hexadecimal ● 向前 ptr rowid - Hexadecimal ● 旧的 slotlen - Decimal ● 新的 slotlen - Decimal ● 块的数量 - Decimal

记录类型	操作	附加列和格式
	LM_FIRSTUPD 确认这是该事务对此 rowID 的首次更新	● LSN (optional) - Decimal
IDXFLAGS	索引标志	● tblspace ID - Hexadecimal ● 键号 - Hexadecimal
INSERT	插入后映象	● tblspace ID - Hexadecimal ● rowid - Hexadecimal
ISOSPCOMMIT	记录已隔离的保存点提交	● 结束事务时间 - Decimal ● 开始事务时间 - Decimal
LCKLVL	锁定方式（页或行）	● tblspace ID - Hexadecimal ● 新的锁定方式 lockmode - Hexadecimal ● 旧的锁定方式 - Hexadecimal
LG_ADDBPOOL	联机添加缓冲池	● 页大小（以字节计） - Decimal ● 池中缓冲区数 - Decimal ● lru 队列数 - Decimal ● lru_max_dirty 百分比 - Decimal ● lru_min_dirty 百分比 - Decimal
PTRUNCATE	表示意图截断表。分区标记为丢弃或复用，具体取决于指定的命令选项	● tblspace ID - Hexadecimal
TRUNCATE	TRUNCATE 已释放了 extents 且将落实该事务	● tblspace ID - Hexadecimal
MVIDXND	索引节点已移动以允许 2 位至 4 位的位图转换	● tblspace ID -

记录类型	操作	附加列和格式
		Hexadecimal ● 旧页号 - Decimal ● 新页号 - Decimal ● 父页号 - Decimal ● 父 slot 号 - Decimal ● 父 slot 偏移量 - Decimal ● 键号 - Decimal
PBDELETE	删除 tblspace blobpage	● bpageno - Hexadecimal ● 状态 USED/FREE 唯一 ID - Decimal
PBINSERT	添加 tblspace blobpage	● bpageno - Hexadecimal ● tblspace ID - Hexadecimal ● rowid - Hexadecimal ● slotlen - Decimal ● pbrowid - Hexadecimal
PDINDEX	预删除索引	● tblspace ID - Hexadecimal
PGALTER	原位更改了页	● tblspace ID - Hexadecimal ● 物理页号 - Hexadecimal
PGMODE	页模式已在位图中修改	● tblspace ID - Hexadecimal ● 逻辑页号 - Decimal ● 旧模式 - Hexadecimal ● 新模式 - Hexadecimal
PERASE	预擦除旧文件。标记要删除的表。数据库服务器释放提交的空间	● tblspace ID - Hexadecimal
PNGPALIGN8	将该 tblspace 中的页用作类属页	● 无

记录类型	操作	附加列和格式
PNLOCKID	更改 tblspaces 索标识	● tblspace ID - Hexadecimal ● 旧锁 ID - Hexadecimal ● 新锁 ID - Hexadecimal
PNSIZES	设置 tblspace extent 大小	● tblspace ID - Hexadecimal ● fextsize - Decimal ● nextsize - Decimal
PREPARE	由参与者数据库服务器写入，以记录参与者提交事务的能力（如果这样指示过）。	● 协调者的数据库服务器名称 - ASCII
PTADESC	添加更改描述信息	● tblspace ID - Hexadecimal ● 上一页的物理页号 - Hexadecimal ● 逻辑页号 - Decimal ● 已添加列的数量 - Decimal
PTALTER	分片更改已开始	● tblspace ID - Hexadecimal ● 上一页的物理页号 - Hexadecimal ● 逻辑页号 - Decimal ● 更改 desc 页号 - Decimal ● 已添加的列数 - Decimal ● 更改的版本 - Decimal ● 已添加的行大小 - Decimal
PTALTNEWKEYD	在变更表命令之后更新 tblspace 头中的键描述符	● 键描述符的字节数 - Decimal ● 键描述符中的数据 -

记录类型	操作	附加列和格式
		ASCII
PTALTOLDKEYD	在变更表命令之后更新键描述符	● 键描述符的字节数 - Decimal ● 键描述符中的数据 - ASCII
PTCOLUMN	向分片添加特殊列	● tblspace ID - Hexadecimal ● 列的数量 - Decimal
PTEXTEND	Tblspace 扩展	● tblspace ID - Hexadecimal ● 上一逻辑页 - Decimal ● 第一物理页 - Hexadecimal
PTRENAME	重命名表	● tblspace ID - Hexadecimal ● 旧表名 - ASCII ● 新表名 - ASCII
RDELETE	剩余页删除	● tblspace ID - Hexadecimal ● rowid - Hexadecimal ● slotlen - Decimal ● hrowid（可选）- Decimal ● poffset（可选）- Decimal
RENDBS	重命名 dbspace	● 新的 dbspace 名称 - ASCII
REVERT	记录 dbspace 到较早版本 dbspace 的复原	● 复原事件的类型 - Decimal ● arg1 - Decimal ● arg2 - Decimal ● arg3 - Decimal
RINSERT	剩余页插入	● tblspace ID -

记录类型	操作	附加列和格式
		Hexadecimal ● rowid - Hexadecimal ● slotlen - Decimal ● hrowid（可选）- Decimal ● poffset（可选）- Decimal
ROLLBACK	回滚工作	● 日期 - Decimal ● 时间 - Decimal
ROLWORK	结束事务并回滚工作	● 结束事务时间 - Decimal ● 开始事务时间 - Decimal
RSVEXTEND	记录对保留页的扩展	● 页数 - Decimal ● extent 的物理页号 - Hexadecimal
RTREE	记录对 R-tree 索引页的插入和删除。(R-tree 索引页上的其他操作是以物理方式记录的)记录子类型为: • LEAFINS – 在叶页中插入项 • LEAFDEL – 删除叶页中的项	● 记录子类型 - ASCII ● 索引页 rowid - Hexadecimal ● tuple 长度 - Decimal ● 基本表 rowid - Decimal ● 基本表 fragid - Decimal ● 删除标志 - Decimal
RUPAFT	剩余页更新，后映象	● tblspace ID - Hexadecimal ● rowid - Hexadecimal ● slotlen - Decimal
RUPBEF	剩余页更新，前映象	● tblspace ID - Hexadecimal ● rowid - Hexadecimal ● slotlen - Decimal ● hrowid（可选）- Decimal ● poffset （可选）- Decimal
RUPDATE	如果剩余页更新前映象和数据库都可以适合单	● tblspace ID -

记录类型	操作	附加列和格式
	个页，那么数据库服务器写入一个 RUPDATE 记录	Hexadecimal ● rowid - Hexadecimal ● 向前 ptr rowid - Hexadecimal ● 旧 slotlen - Decimal ● 新 slotlen - Decimal ● 块的数量 - Decimal ● hrowid （可选） - Decimal ● poffset （可选） - Decimal
SBLOB	指示智能大对象的子系统日志记录 各种记录子类型为： • CHALLOC • CHCOMBINE • CHFREE • CHSPLIT • CREATE • DELETES • EXTEND • HDRUPD • PDELETE • PTRUNC • REFCOUNT • UDINSERT • UDINSERT_LT • UDUPAFT • UDUPAFT_LT	● 变化 ● 有关更多信息，请参阅智能大对象的日志记录类型. ● 变化

记录类型	操作	附加列和格式
	• UDUPAFT • UDUPAFT_LT • UDWRITE • UDWRITE_LT	
SYNC	如果该日志文件是空的，且管理员指示数据库服务器切换至下一个日志文件，那么写入逻辑日志文件	• （无） - （无）
TABLOCKS	由协调者或数据库服务器写入。它与 BEGPREP 或 PREPARE 记录相关联。并包含由事务持有的已锁定 tblspace 列表（按 tblspace 编号）。（在分布式事务中，事务显示为锁的所有者。）	• 锁的数量 - Decimal • tblspace 编号 - Hexadecimal
UDINSERT	附加新的用户数据	• 锁的数量 - Decimal • tblspace 编号 - Hexadecimal
UDUPAFT	如果 UDWRITE 代价太高，那么更新用户数据后映象	• chunk - Decimal • chunk 中的页 - Hexadecimal • 页中的偏移量 - Hexadecimal • 数据长度 - Hexadecimal
UDUPBEF	如果 UDWRITE 代价太高，那么更新用户数据前映象	• chunk - Decimal • chunk 中的页 - Hexadecimal • 页中的偏移量 - Hexadecimal • 数据长度 - Hexadecimal
UDWRITE	更新用户数据（差分映象）	• chunk - Decimal • chunk 中的页 -

记录类型	操作	附加列和格式
		Hexadecimal ● chunk 中的偏移量 - Hexadecimal ● 写入前长度 - Hexadecimal ● 写入后长度 - Hexadecimal
UNDO	将要回滚的一系列事务的头记录	● 计数 - Decimal
UNDOBLDC	如果应回滚 CREATE TABLE 语句但由于相关 chunk 已关闭而无法回滚，呢么写入该记录。当重放日志文件时，将删除该表	● tblspace 编号 - Hexadecimal
UNIQID	当向行指定新的序列值时记录	● tblspace ID - Hexadecimal ● 唯一 ID - Decimal
UNIQ8ID	当向行指定新的序列8值时记录	● tblspace ID - Hexadecimal ● 唯一 ID - Decimal
UPDAFT	更新后映象	● tblspace ID - Hexadecimal ● rowid - Hexadecimal
UPDBEF	更新前映象	● tblspace ID - Hexadecimal ● rowid - Hexadecimal
XAPREPARE	参与者可以提交该 XA 事务	● （无） - （无）

智能大对象的日志记录类型

所有智能大对象日志记录都是 SBLOB 类型。每个智能大对象日志记录包含 6 个头列（在逻辑日志记录头中描述）、记录子类型和其他信息。信息的显示会根据记录子类型而变化。

表 1 列出了智能大对象的所有记录类型。子类型列描述了智能大对象的记录类型。操作列标识生成日志条目的数据库服务器操作类型。附加列和格式列描述了每个记录类型出现什么信息。

记录子类型	操作	附加列	格式
CHALLOC	分配 chunk extent	extent [chk, page, len]	Decimal
		标识	Hexadecimal
CHCOMBINE	组合用户数据 Extent 列表中的两页	chunk 编号	Decimal
		第一页	Decimal
		第二页	Decimal
CHFREE	释放 chunk extent	extent [chk, page, len]	Decimal
CHSPLIT	分割用户数据 Extent 列表中的两页	chunk 编号	Decimal
		要分割的 UDFET 页	Decimal
CREATE	创建智能大对象	智能大对象 ID [sbs, chk, page, oid]	Decimal
		lomaphdr 中 extent 的数量	Decimal
DELETE	删除提交的智能大对象	智能大对象 ID [sbs, chk, page, oid]	Decimal
EXTEND	向智能大对象的 extent 列表添加 extent	智能大对象 ID [sbs, chk, page, oid]	Decimal
		extent [chk, page, len]	Decimal
		lomap 溢出页号	Decimal
HDRUPD	更新智能大对象的头页	智能大对象 ID [sbs, chk, page, oid]	Decimal
		旧 EOF 偏移量	String
		新 EOF 偏移量	String
		旧的时间	Decimal
		新的时间	Decimal
PDELETE	对提交的待删除智能大对象进行排队	智能大对象 ID [sbs, chk, page, oid]	Decimal
PTRUNC	对提交的待截断智能大对象进行排队	智能大对象 ID [sbs, chk, page, oid]	Decimal
		旧的偏移量	String
		新的偏移量	String
REFCOUNT	增加或减少智能大对象的引用计数	智能大对象 ID [sbs, chk, page, oid]	Decimal

记录子类型	操作	附加列	格式
		如果增加，那么为 1 ；如果减少，那么为 0	Decimal
UDINSERT,	附加新的用户数据	chunk	Decimal
UDINSERT_LT		chunk 中的页	Decimal
		页中的偏移量	Decimal
		数据长度	Decimal
UDUPAFT,	如果 UDWRITE 代价太高，那么更新用户数据后映象	chunk	Decimal
UDUPAFT_LT		chunk 中的页	Decimal
		页中的偏移量	Decimal
		数据长度	Decimal
UDUPBEF,	如果 UDWRITE 代价太高，那么更新用户数据前映象	chunk	Decimal
UDUPBEF_LT		chunk 中的页	Decimal
		页中的偏移量	Decimal
		数据长度	Decimal
UDWRITE,	更新用户数据（差分映象）	chunk	Decimal
UDWRITE_LT		chunk 中的页	Decimal
		页中的偏移量	Decimal
		写入前的长度	Decimal
		写入后的长度	Decimal
		不同映象块的数量	Decimal

有关 glogdump 输出中的智能大对象示例，请参阅GBase 8s 管理员指南中的[什么是逻辑日志] 一章中的“智能大对象日志记录”。

图 1显示了 glogdump 输出中的智能大对象记录的示例。前两条记录显示 extent 已释放。下一组记录（其上下两侧分别是 BEGIN 和 COMMIT）显示存储器的分配和智能大对象的创建。

图: glogdump 输出中的智能大对象记录

```
addr    len   type    xid       id  link     subtype  specific-info

4e8428   40   SBLOB     8        0  4e7400   CHFREE   (2,53,421)
4e8450   40   SBLOB     8        0  4e8428   CHFREE   (2,579,421)

c8018    40   BEGIN     8        3  0        07/13/98 10:23:04 34       informix
c8040   264   SBLOB     8        0  c8018    CREATE   [2,2,1,900350517]  10
c8148    44   SBLOB     8        0  c8040    CHALLOC  (2,53,8) 0x1
c8174    68   SBLOB     8        0  c8148    EXTEND   [2,2,1,900350517] (2,53,8) -1
c81b8   264   SBLOB     8        0  c8174    CREATE   [2,2,2,900350518]  10
c82c0    44   SBLOB     8        0  c81b8    CHALLOC  (2,61,1) 0x1
c82ec    68   SBLOB     8        0  c82c0    EXTEND   [2,2,2,900350518] (2,61,1) -1
c8330    56   SBLOB     8        0  c82ec    REFCOUNT [2,2,1,900350517] 1
c8368    56   SBLOB     8        0  c8330    REFCOUNT [2,2,2,900350518] 1
c83a0    36   COMMIT    8        0  c8368    07/13/98 10:23:05

c83c4    40   BEGIN     8        3  0        07/13/98 10:23:05 34       informix
c83ec   264   SBLOB     8        0  c83c4    CREATE   [2,2,3,900350519]  10
c84f4    44   SBLOB     8        0  c83ec    CHALLOC  (2,62,1) 0x1
c8520    68   SBLOB     8        0  c84f4    EXTEND   [2,2,3,900350519] (2,62,1) -1
c8564    56   SBLOB     8        0  c8520    REFCOUNT [2,2,3,900350519] 1
c859c    36   COMMIT    8        0  c8564    07/13/98 10:23:05
```

3 管理实用程序

3.1 实用程序概述

GBase 8s 数据库服务器实用程序允许您直接从命令行执行管理任务。

数据库服务器实用程序支持多字节命令行参数。有关完整的多字节命令行参数的实用程序，请参阅 ../gug/ids_gug_171.html#ids_gug_171 。

在执行实用程序之前，必须使数据库服务器联机，以下情况除外：

- oninit
- 一些 glogdump 选项
- 一些 gcheck 选项

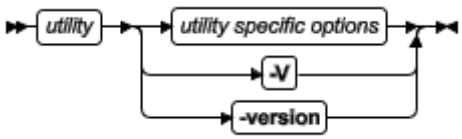
注： 在使用实用程序时，请不要使用 UNIX™ 命令 CTRL-C 向进程发送中断信号，因为这可能引起错误。

3.1.1 获取实用程序的版本信息

许多 GBase 8s 命令行实用程序都允许使用 -V 和 -version 选项来获取版本信息。这两选项主要用于调试。

-V 选项显示了软件版本号和序列号。

-version 选项扩展 -V 选项，显示关于创建操作系统、创建号码以及创建日期的更多信息。



-V 和 **-version** 选项不能与任何其他实用程序选项一起使用。例如：**gstat -version** 命令可能显示以下输出。

```
gstat -version
                Program:        gstat
                Build Version:  8.8
                Build Host:     connla
                Build OS:       SunOS 5.6
                Build Number:   009
                Build Date:     Sat Nov 20 03:38:27 CDT 2011
                GLS Version:    glslib-4.50.xC2
```

gstat -V 命令返回结果格式为：产品名称+主版本号+_+版本级别+_+版本号+专用版本代号及版次+_+迭代序号（送测序号）+补充版本标识+_+6位哈希ID，可能显示以下信息：

```
GBase8sV8.8_AEE_3.0.0G5_1P20200918_1fc8fc
```

关于格式的详细介绍，请参阅《GBase 8s SQL 指南：语法》DBINFO函数的使用 'version_gbase' 选项。

3.1.2 设定实用程序的本地环境变量

在 UNIX™ 操作系统上，无需在 shell 环境中设定本地环境变量即可启动某些实用程序。可以在 onconfig 文件中设定本地环境变量。当运行命令以启动实用程序时，使用 -FILE 选项指向 onconfig 文件。

开始之前，请确保这些前提条件得到满足：

- 该实用程序的可执行程序的路径是现有的 shell 环境的一部分。
- 如果您要在远程电脑上运行命令，必须配置远程的 shell 实用程序（例如 SSH）。

1. 向 onconfig 文件中添加一个或多个环境变量值时，每条指令请使用以下格式：

   #$variable_name value

2. 当您运行命令以开始实用程序时，使用 -FILE 选项指定到 onconfig 文件的完全或相对路径。

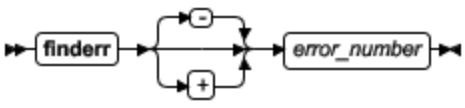
查看 -FILE 选项的参考信息中的语法、用法和示例。

实用程序读取并设定 onconfig 文件中指定的环境变量，并且这些值将优先于在本地 shell 环境中设置的值。

3.2 finderr 实用程序

使用 finderr 实用程序来查看 GBase 8s 上错误消息的额外信息。在 UNIX™ 和 Linux™ 平台上，该信息通过命令行显示。在 Windows™ 平台上，该信息在错误消息程序中显示。

语法



元素	用途	关键注意事项
error_number	错误消息编号为其提供额外	**在 UNIX 或 Linux 上：** 如果错误消息不包含减号 (-) 或加号 (+)，并且存在正版本和负版本的错误消息，那么将会显示负版本的消息。若错误消息的编号前带一个加号则显示的信息是正

元素	用途	关键注意事项
	信息	的错误消息。 在 Windows 上：如果错误消息不包含减号 (-) 或加号 (+)，并且存在正版本和负版本的错误消息，那么必须在错误消息程序中手动选择您想要查看的消息。

用法

在消息日志中输出的错误消息内容包含了消息的编号及其简要概述。使用 finderr 消息编号命令来查找错误和可能的用户操作的原因的更详细的描述，以纠正或防止错误。

在 Windows 上，您可以直接从数据库服务器程序组中选择 Error Messages 来打开错误消息程序。

示例

下面的命令显示了在 UNIX 或 Linux 平台上有关错误消息 -201 的信息：

```
finderr 201

        -201     A syntax error has occurred.

        This general error message indicates mistakes in the form of an SQL
        statement. Look for missing or extra punctuation (such as missing or
        extra commas, omission of parentheses around a subquery, and so on),
        keywords misspelled (such as VALEUS for VALUES), keywords misused (such
        as SET in an INSERT statement or INTO in a subquery), keywords out of
        sequence (such as a condition of "value IS NOT" instead of "NOT value
        IS"), or a reserved word used as an identifier.

        Database servers that provide full NIST compliance do not reserve any
        words; queries that work with these database servers might fail and
        return error -201 when they are used with earlier versions of GBase 8t
        database servers.

        The cause of this error might be an attempt to use round-robin syntax with
        CREATE INDEX or ALTER FRAGMENT INIT on an index. You cannot use round-robin
        indexes.

        The error may also occur if an SQL statement uses double quotation marks
        around input strings and the environment variable DELIMIDENT is set.
        If DELIMIDENT is set, strings that are surrounded by double quotation
        marks are regarded as SQL identifiers rather than string literals. For
        more information on the usage of DELIMIDENT, see the GBase 8t Guide to
        SQL: Reference.
```

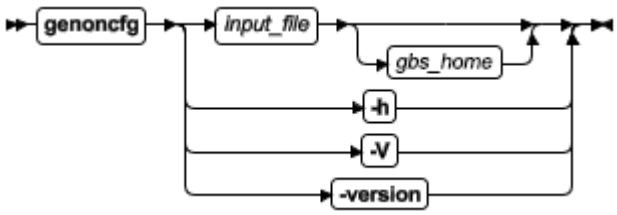
下列命令显示了有关错误消息 100 ，即对应于 SQLCODE 值为 100 的信息：

```
finderr +100

        100     No matching records found.

        The database server did not find any more data. This message is an ANSI-
        standard SQLCODE value. If you attempted to select or fetch data, you
        encountered the end of the data, or no data matched the criteria in the
        WHERE clause. Check for an empty table. Use this SQLCODE value to determine
        when a statement reaches the end of the data. For more information, see the
        discussion of SQLCODE in the GBase 8t ESQL/C Programmer's Manual. The
        database server can return this SQLCODE value to a running program.
```

3. 3 genoncfg 实用程序

使用 genoncfg 实用程序可加快根据您主机环境以及数据库服务器的预期用途对缺省的 GBase 8s 配置文件 (onconfig.std) 进行定制的过程。

语法



元素	用途	关键注意事项
input_file	包含参数设置的输入文件的名称	
gbs_home	您希望配置的 GBase 8s 安装路径	如果 GBASEDBTDIR 环境变量已被设置，那么您可以省略安装路径。如果已设置 GBASEDBTDIR 变量并且在命令行进入了该安装路径，那么实用程序会在此命令行的路径下运行 。
-h	有关 genoncfg 实用程序的帮助信息	
-V	显示短的版本信息并退出命令行实用程序	
-version	显示扩展的版本信息并退出命令行实用程序	

用法

在运行此实用程序前，请以 root 或用户 gbasedbt 的身份登入主机。

在您成功运行 genoncfg 实用程序之前，必须在输入文件中设置参数对主机环境是可用的。对于所有的环境，disk 参数在输入文件中是必不可少的。也可以在输入文件中输入指令。这些指令对于运行实用程序时是非必要的，但是它们在一些场景下会有帮助。

该实用程序不会读取和修改任何已存在的配置文件。如有您在主机环境里有一个预先存在的 ONCONFIG 文件，当您运行该实用程序，此文件中的参数值不会发生改变。因此，在将参数应用在数据库服务器实例中之前，可以查看这些建议的参数设置。

使用 genoncfg 实用程序的步骤：

1. 用文本编辑器创建包含 genoncfg 实用程序处理过程的参数值的输入文件。
2. 用输入文件运行该实用程序。该配置文件（名为onconfig）将生成并保存在该工作目录下。
3. 可选: 重命名生成的配置文件。
4. 如果您想要使用已生成的配置文件运行数据库服务器实例，将该文件复制到 $GBASEDBTDIR/etc。

genoncfg 实用程序的输入文件

使用输入文件来指定以下关于数据库服务器实例的信息：

- 预计联机事务处理系统（OLTP）的连接数
- 预计决策支持系统（DSS）的连接数
- 磁盘空间
- CPU 初始化
- 网络服务连接设置
- 恢复时间

输入文件是一个 ASCII 文本文件。对参数的排列顺序没有要求。 以下是一个输入文件的样本：

```
cpus 1
memory 1024 m
connection name demo_on onsoctcp 9088
servernum 1
oltp_connections 10
dss_connections 2
disk /opt/gbs_server/data/storage/online_root 0 k 300 m
directive one_crit
directive debug
```

表 1. genoncfg 实用程序的输入文件的参数

元素	描述
connection	服务器连接参数： ● 名称 或者 别名 ，取决于连接器的功能。是特定的服务器名（还是替代服务器名（。 ● 连接器的名称 ● 连接服务器的类型（在配置文集中等同于 NETTYPE ） ● 服务的端口号 例如：connection name demo_on onsoctcp 9088
cpus	分配给实例的中央处理单元数（CPUs） 例如： cpus 1
directive	genoncfg 实用程序可以使用的指令。 ● one_crit: 配置数据库服务器只在 root dbspace 上存储物理日志、逻辑日志和数据。 ● debug: 实时显示有关主机环境和配置文件上操作的信息。 例如： directive one_crit 此信息在解决数据库服务器配置问题时会有帮助。 一种场景是，调试指令可以导致节省时间。在该场景下，通过读取显示的信息注意到该实用程序在创建您不想要或不起作用的 onconfig 文件。您停止该实用程序（尽管它仍在运行），调整输入文件的设置，然后用修改完成后的输入文件返回实用程序。
disk	为实例设置的磁盘存储空间： ● root dbspace 的位置 ● 偏移量的大小,以兆字节（m）或千字节（k）为单位 ● root dbspace 的大小，以兆字节（m）或千字节（k ）为单位 例如： UNIX™: /opt/gbs_server/data/storage/rootdbs

元素	描述
	Windows™: d:\INFXDATA\rootdbs **重要：** 如果进入 root dbspace 工作实例的路径下，该实例将被覆盖，并不能使用。
dss_connections	预估该实例的决策处理系统（DSS）的连接数。例如：一个查询客户端或其它可以获得商业智能设置结果的应用程序可以是一个 DSS 连接。例如： dss_connections 2
memory	实例的内存量（兆字节）。例如 ： memory 1024 m
oltp_connections	估计该实例的联机处理系统（OLTP）的连接数。典型的是，在实例中修改数据库状态的应用程序时一个 OLTP 连接。例如： oltp_connections 10
rto_server_restart	指定在重启 GBase 8s 后，进入联机或静默方式，数据库服务器不得不从一个问题中恢复的时间量(以秒为单位)。该值可设置为 0 以禁用配置参数，或设置为在 60 和 1800 中的任意一值以启用参数并指示秒数。 例如： rto_server_restart 100 指定恢复时间对象为 100 秒
servernum	数据库服务器实例的唯一 ID 例如： servernum 1

3.4 gcheck 实用程序

使用 gcheck 实用程序来检查特定磁盘结构的不一致性，修复不一致的索引结构并显示有关磁盘结构的信息。

检查索引时，gcheck 实用程序需要分类空间。所需的分类空间量与创建索引所需要的空间一样大。如果收到 “no free disk space for sort ” 的错误信息，您必须重新预估使空间可用的所需的临时空间量。

可以使用等同于一些 gcheck 命令的 SQL 管理 API 命令。

3.4.1 gcheck 检查并修复

gcheck 实用程序可以修复以下类型的磁盘结构：

- 分区页统计信息
- 位图页
- 分区 blobpage

- blobspace blobpage
- 索引
- sbspace 页
- sbspace 元数据分区

如果 gcheck 检测到其他结构中的不一致性，那么向您发送消息警告以通知这些不一致性，但 gcheck 不能解决问题。有关更多信息，请参阅 GBase 8s 管理员指南 中的 一致性检查一章和 磁盘结构和存储。

每个选项执行哪些操作？

正如 表 1 所示，gcheck 选项分为三类：检查、修复并显示。 显示或打印选项（以字母 p 为前缀的那些选项）的功能与 -c 选项的功能相同，不同之处是 -p 选项显示 gcheck 实用程序执行时受检查数据的其他信息。您不能组合 gcheck 选项标志，以下段落中描述的情况除外。

一般地，-c 选项检查一致性，并且只在找到错误或不一致时才在屏幕上显示消息。

任何用户都可以执行这些检查选项。在 UNIX™ 平台上，必须是用户 gbasedbt 或 root 才可显示数据库数据或启动修复选项。在 Windows™ 上，必须是 Gbasedbt-Admin 组的成员才可显示数据库数据或启动修复选项。

表 1 将 gcheck 选项与其功能向关联。它还显示了功能等同于 gcheck -c 选项的管理 API 命令字符串。

表 1. gcheck 选项及其功能

对象	检查	SQL 管理 API *命令字符串*	修复	显示
Blobspace 简单大对象				-pB
系统目录表	-cc			-pc
数据行，没有简单大对象或智能大对象	-cd			-pd
数据行，简单大对象但没有智能大对象	-cD			-pD
带有用户定义存取方法的表	-cd, -cD	CHECK DATA		
Chunks 和 extents	-ce	CHECK EXTENTS		-pe
索引（键值）	-ci, -cix		-ci -y -pk	-pk

对象	检查	SQL 管理 API *命令字符串*	修复	显示
			-y, -pkx -y	
索引（键加 rowid）	-cI, -cIx		-cI -y -pK -y, -pKx -y	-pK
带有用户定义存取方法的索引	-ci, -cI			
索引（叶键值）			-pl -y, -plx -y	-pl
索引（叶键加 rowid）			-pL -y, -pLx -y	-pL
页（按表或分片）				-pp
页（按 chunk）				-pP
Root 保留页	-cr, -cR			-pr, -pR
智能大对象的元数据	-cs, -cS			-ps, -pS
空间使用量（按表或分片）		CHECK PARTITION PRINT PARTITION		-pt
空间使用量（按表，带索引）				-pT

使用 -y 选项执行修复

使用 -y 选项指示 gcheck 自动执行修复，如下例所示

```
gcheck -cd -y
gcheck -cD -y
gcheck -ci -y
gcheck -cI -y
```

如果不使用 -y 选项，那么 gcheck 会在遇到不一致性时提示您，并允许您请求修复。如果指定了选项 -n ，那么 gcheck 不提示您，因为该选项指示 gcheck 不执行修复。

修复 sbspace 和外部空间中的索引

如果 sbspace 和外部空间中的索引是使用支持 gcheck -y 选项的存取方法创建的，那么 gcheck 实用程序可修复 sbspace 或外部空间中的索引。尽管 gcheck 实用程序不修复分片索引，但用户定义的存取方法可以修复它们。有关存取方法所支持的 gcheck 选项的更多信息，请参阅 GBase 8s DataBlade API 程序员指南 或 GBase 8s 虚拟索引接口程序员指南。

锁定和 gcheck

以下操作期间，gcheck 实用程序在表上放置了共享锁，因此其他用户在检查完成之前将不能执行更新、插入或删除：

- 检查数据时
- 检查索引（使用 -ci 、-cI 、-pk 、-pK 、-pl 或 -pL）且表使用页锁定时
- 当指定带有 -ci 、-cI 、-pk 、-pK 、-pl 或 -pL 的 -x 选项且表使用行锁定时

如果表不使用页锁定，那么使用 gcheck -ci 、-cI 、-pk 、 -pK 、 -pl 或 -pL 选项检查索引时，数据库服务器不在表上放置共享锁。当做索引检查过程中表上没有共享锁时，其他用户可以在检查过程中更新行。

如果在索引检查过程中未在使用行锁的表上放置共享锁，那么 gcheck 实用程序无法进行准确的索引检查。要绝对确保完全的索引检查，可以用 -x 选项执行 gcheck 。使用 -x 选项， gcheck 在表上放置共享锁，这样其他用户就不能再检查完成之前执行更新、插入或删除。

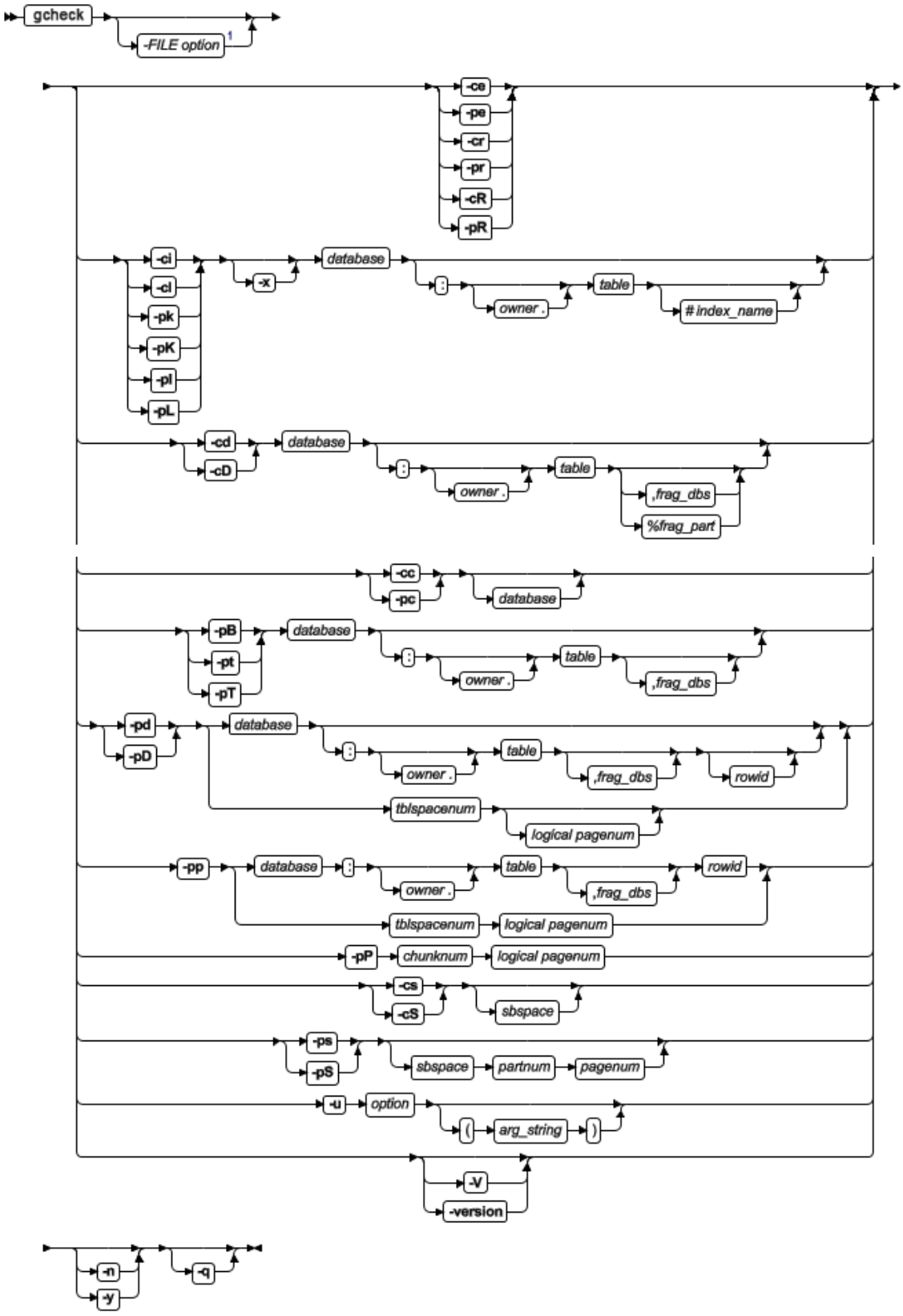
若在高可用集群的辅助服务器上运行 gcheck 实用程序，它将返回不可靠的结果。

有关 -x 选项的更多信息，请参阅 使用 -x 开启锁 。有关共享锁和意向锁的更多信息，请参阅 GBase 8s 性能指南。

在检查系统目录表时，gcheck 实用程序在这些表上放置共享锁。在执行时，它在表上放置互斥锁。

3.4.2 gcheck 实用程序语法

gcheck 检查特定磁盘结构的不一致性，修复不一致的索引结构并显示有关磁盘结构的信息。



元素	用途	关键注意事项
- cc	检查系统目录表中的指定数据库	请参阅 gcheck -cc 和 -pc：检查系统目录表

元素	用途	关键注意事项
-cd	从指定数据库、表或分片的 tblspace 中读取除简单大对象之外的所有页，并检查每页的一致性 还检查使用用户定义存取方法的表	不检查简单智能大对象 请参阅 gcheck -cd 和 gcheck -cD ：检查页
-cD	与 -cd 相同，但还读取每个 blobpage 的头并检查其一致性	检查简单大对象但不检查智能大对象 请参阅 gcheck -cd 和 gcheck -cD ：检查页
-ce	检查每个可用 chunk 列表和相应的可用空间以及每个 tblspace extent 。还检查智能大对象 extent 和 sbspace 元数据	gcheck 进程验证磁盘上的 extents 与描述它们的当前控制信息相对应。 请参阅 gcheck -ce 、-pe: 检查可用 chunk 列表。有关背景信息，请参阅 Next-Extent 分配
-ci	检查键值顺序和与指定表相关联的所有索引的水平和垂直节点链接的一致性 还检查使用用户定义存取方法的索引	请参阅 gcheck -ci 和 -cI: 检查索引节点链接
-cI	与 -ci 相同，但是还检查索引中 rowid 关联的键值是否与行中的键值相同	请参阅 gcheck -ci 和 -cI: 检查索引节点链接
-cr	检查每个 root dbspace 保留页是否存在几种情况	请参阅 gcheck -cr 和 -cR: 检查保留页
-cR	检查 root dbspace 保留页、物理日志页和逻辑日志页	请参阅 gcheck -cr 和 -cR: 检查保留页
-cs	检查 sbspace 的智能大对象和 sbspace 元数据	请参阅 gcheck -cs 、-cS 、-ps、-pS: 检查并显示 sbspace
-cS	检查 sbspace 的智能大对象和 sbspace 元数据以及 extent	请参阅 gcheck -cs 、-cS 、-ps、-pS: 检查并显示 sbspace
sbspace	指示可选的 sbspace 名称 如果未提供，那么检查所有的 sbspace	无
-n	指示不应执行任何索引修复，即使检测到错误也是如此	与索引修复选项（-ci 、-cI 、-pk 、-pK 、-pl 和 -pL）一起使用
-pB	显示描述指定表中	这些统计信息衡量数据库或表中的个别简单

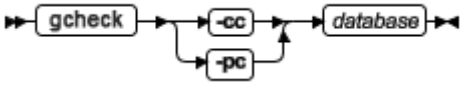
元素	用途	关键注意事项
	blobspace blobpage 的平均充满度的统计信息	大对象的存储效率。如果未指定表或分片，那么显示整个数据库的统计信息。 请参阅 gcheck -pB：显示 blobspace 统计信息。有关优化 blobspace blobpage 大小的信息，请参阅 *GBase 8s 管理员指南*中的管理磁盘空间一章
-pc	与 -cc 相同，但在检查系统目录表时还显示系统目录信息，包括每个表的 extent 使用情况	无
-pd	以十六进制格式显示行	请参阅 gcheck -pd 和 pD ：以十六进制格式显示行
-pD	显示十六进制格式的行和存储在 tblspace 中的简单大对象值或存储在sbspace sbpage 中的智能大对象和存储在 blobspace blobpage 中简单大对象的头信息	请参阅 gcheck -pd 和 pD ：以十六进制格式显示行
-pe	与 -ce 相同，但在检查可用 chunk 列表、相应可用空间以及每个 tblspace extent 时还显示 chunk 和 tblspace extent 信息	请参阅 gcheck -ce 、-pe: 检查可用 chunk 列表
-pk	与 -ci 相同，但在检查它们时还显示指定表上所有所有的键值	请参阅 gcheck -pk 、-pK 、-pl 、-pL：显示索引信息
-pK	与 -cI 相同，但在检查它们时还显示键值和 rowid	请参阅 gcheck -pk 、-pK 、-pl 、-pL：显示索引信息
-pl	与 -ci 相同，但还显示键值。只检查叶节点索引页	请参阅 gcheck -pk 、-pK 、-pl 、-pL：显示索引信息
-pL	与 -cI 相同，但还只显示叶节点索引页的键值和 rowid	请参阅 gcheck -pk 、-pK 、-pl 、-pL：显示索引信息
-pp	显示逻辑页的内容	请参阅 gcheck -pp 和 -pP: 显示逻辑页的内容
-pP	与 -pp 相同，但是需要 输入 chunk 编号和逻辑页号或内部 rowid	请参阅 gcheck -pp 和 -pP: 显示逻辑页的内容
-pr	与 -cr 相同，但在检查保留页时还显示保留页信息	请参阅 gcheck -pr 和 pR：显示保留页信息
-pR	与 -cR 相同，但还显示保留	请参阅 gcheck -pr 和 pR：显示保留页信息

元素	用途	关键注意事项
	页、物理日志页和逻辑日志页的信息	
-ps	检查和显示 sbspace 的智能大对象和 sbspace 元数据	请参阅 gcheck -cs 、-cS 、-ps、-pS: 检查并显示 sbspace
-pS	检查并显示简单大对象和 sbspace 元数据。列出个别智能大对象的 extent 和头信息	请参阅 gcheck -cs 、-cS 、-ps、-pS: 检查并显示 sbspace
-pt	显示表或分片的 tblspace 信息	请参阅 gcheck -pt 和 -pT：显示表或分片的 tblspaces
-pT	与 -pt 相同，但还显示特定于索引的信息和按页类型排列的页分配信息（对于 dbspace）	请参阅 gcheck -pt 和 -pT：显示表或分片的 tblspaces
-q	不显示所有检查和确认信息	无
-x	检查和打印索引时在表上放置共享锁	与 -ci、-cI、-pk、-pK、-pl 或 -pL 选项一起使用。请参阅 使用 -x 开启锁，获取全部信息
-y	当检测到错误时修复索引	无
-V	显示软件版本号及序列号	请参阅 获取实用程序的版本信息
-version	显示构件版本、主机、操作系统、编号、日期及 GLS 版本	请参阅 获取实用程序的版本信息
chunknum	指定用于指示特定 chunk 的十进制值	值必须是大于 0 的无符号整数。Chunk 必须存在。 执行 -pe 选项可了解哪些 chunk 号是与特定的 dbspace、blobspace 或 sbspace 相关联的
database	指定要检查其一致性的数据库名称	语法必须符合 Identifier 分段；请参阅 *《GBase 8s SQL 指南：语法》*
db1	指定包含想要检查的数据类型的本地数据库	可以选择使用格式 db1@server1 指定本地数据库服务器名
db2	指定包含想要检查的数据类型的远程数据库	可以选择使用格式 db2@server2 指定远程数据库服务器名
frag_dbs	指定包含想要检查其一致性的分片的 dbspace 名称	Dbspace 必须存在并包含要检查一致性的分片。语法必须符合 Identifier 分段；请参阅 *《GBase 8s SQL 指南：语法》*
index_name	指定想要检查其一致性的索引名称	指定的表和数据库必须存在索引。 语法必须符合 Identifier 分；请参阅

元素	用途	关键注意事项
		《GBase 8s SQL 指南：语法》
logical pagenum	指定用于指示 tblspace 中特定页的整数值	值必须是在 0 和 16,777,215（包括 0 和 16,777,215 ）之间的无符号整数。值可以表示为无符号整数或以 0x 标识符开始的十六进制数
object	指定想要检查的 DataBlade 、强制转型、运算符类、用户定义的数据类型或 UDR 的名称	如果未指定对象名称，那么数据库服务器将比较相同类型（具有相同的名称和所有者）的所有对象
owner	指定表的所有者	必须指定并的当前所有者。 语法必须符合 Owner Name 段；有关更多信息，请参阅 *《GBase 8s SQL 指南：语法》*
pagenum	标识要检查和显示的 sbspace 元数据部分的页号	无
partnum	标识要检查和显示的 sbspace 元数据分区	无
rowid	标识要显示其内容的行的 rowid 。Rowid 是作为 **gcheck -pD** 输出的一部分进行显示的	值必须是 0 和 4,277,659,295（包括 0 和 4,277,659,295）之间的无符号整数。值可以表示为无符号整数或以 0x 标识符开始的十六进制数
sbspace	指定想要检查其一致性的 sbspace 的名称	无
server	指定数据库服务器名称	如果省略数据库服务器名称，那么 **gcheck** 使用 sqlhosts 文件中的 dbservername 条目指定的名称
table	指定想要检查其一致性的表的名称	执行实用程序时，表应存在。语法必须符合 Table Name 段；有关更多信息，请参阅 *《GBase 8s SQL 指南：语法》*
tblspacenum	标识要显示其内容的 tblspace	值必须是 0 和 208,666,624（包括 0 和 208,666,624）之间的无符号整数。值可以表示为无符号整数或以 0x 标识符开始的十六进制数

3.4.3 gcheck -cc 和 -pc：检查系统目录表

语法：



-cc 选项检查指定数据库的所有系统目录表。如果未指定数据库，那么它检查所有数据库的所有系统目录表。

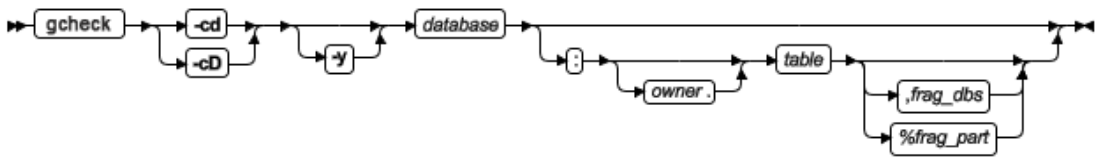
-pc 选项对系统目录表执行相同的检查，并且还会显示系统目录信息。包括每张表的物理地址、所使用的锁定类型、行大小、键的数量、extent 使用情况、已分配和使用的页数量、tblspace 分区号以及索引的使用情况。

在执行 gcheck -cc 或 gcheck -pc 之前，请执行 SQL 语句 UPDATE STATISTICS ，以确保发生准确的检查。为检查表，gcheck 将每个系统目录表与其在 tblspace 中的相应条目作比较。有关 tblspace 的更多信息，请参阅 tblspace tblspace 的结构。

3.4.4 gcheck -cd 和 gcheck -cD ：检查页

使用 gcheck -cd 和 gcheck -cD 命令检查每页的一致性。使用 gcheck -cd -y 或 gcheck -cD -y 命令修复其不一致性。

语法：



gcheck -cd 选项从指定数据库、表、分片或多个分片（碎片）的 tblspace 中读取 blobpages 和 sbpages 除外的所有页，并检查每页的一致性。它对照这些页检查位图页中的条目，以验证映射。

gcheck -cD 选项执行与 gcheck -cd 相同的检查，并检查每个 blobpage 头的一致性。gcheck -cD 选项不会比较开始时间戳记（存储在头中）和结束时间戳记（存储在 blobpage 的末尾）。使用 gcheck -cD -y 选项可以清除 blobspace 中孤立的简单大对象（它可能在跨几个日志文件执行回滚后产生）。

如果数据库包含分片表，但您未指定分片，那么 gcheck -cd 选项检查表中的所有分片。如果您未指定表，该选项会检查数据库中的所有表。通过进行比较， gcheck -pd 选项将显示指定页的十六进制转储，但不检查一致性。

对于 gcheck -cd 和 gcheck -cD 选项，gcheck 实用程序在检查表的索引时锁定每张表。要修复这些页，请使用 gcheck -cd -y 或 gcheck -cD -y。

如果表在相同的 dbspace 中的多个分区上分片，那么 gcheck -cd 和 gcheck -cD 命令将显示分区名称。以下示例显示了在相同 dbspace 中的多个分区上分片的表的典型输出：

```
TBLspace data check for multipart: t1
Table fragment partition part_1 in DBspace dbs1
Table fragment partition part_2 in DBspace dbs1
Table fragment partition part_3 in DBspace dbs1
Table fragment partition part_4 in DBspace dbs1
Table fragment partition part_5 in DBspace dbs1
```

当使用 gcheck -cd 或 gcheck -cD 命令，您可以指定 frag_dbs 或 %frag_dbs 选项，但不能都指定它们：

- 当使用 frag_dbs 选项时，该实用程序检查 frag_dbs dbspace 中所有的分片。
- 当使用 %frag_dbs 选项时，如果 PARTITION 语法在分片或表创建时被使用，该实用程序只检查名为 frag_part 的分片。

尽管可以用 PARTITION 语法分片索引，但是不能限制只对一个分片或分区检查索引。例如，您可以指示 gcheck -cDI my_db:my_tab,data_dbs1 or gcheck -cDI my_db:my_tab%part1 。该检查的 D（数据）部分根据规范限制，然而 I（索引） 检查不会被限制。

例子

以下示例检查 catalog 表中的数据行，包括简单大对象和智能大对象：

```
gcheck -cD superstores_demo:catalog
```

如果您指定一个单独的分片，gcheck 实用程序只单独显示该分片的头。对于分片表来说，每个头显示了每个分片。

```
TBLspace data check for stores_demo:tab1
                Table fragment in DBspace db1
```

消息

如果 gcheck 实用程序未找到不一致性，那么对于其检查的每张表，它显示类似于下行的头：

```
TBLSPACE data check for stores_demo:customer
```

如果 gcheck 实用程序找到不一致性，它显示类似如下的消息：

```
BAD PAGE 2:28: pg_addr 2:28 != bp-> bf_pagenum 2:69
```

物理地址 2:28 表示 chunk 号 2 的页 28 。

如果使用 DataBlade 模块提供的存取方法的索引无法找到存取方法，那么您接收到以下信息：

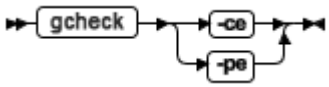
```
-9845 Access method access_method_name does not exist in database.
Ensure that the DataBlade installation was successful.
```

参考

要监视 blobspace blobpage ，请参阅 gcheck -pB：显示 blobspace 统计信息。

3.4.5 gcheck -ce 、-pe: 检查可用 chunk 列表

语法：



-ce 选项检查每个可用 chunk 列表和相应的可用空间及每个 tblspace extent 。有关更多信息，请分别参阅 Next-Extent 分配 和 Chunk 可用列表页的结构 。 gcheck 进程验证磁盘上 extent 与描述它们的当前控制信息相对应。

-pe 选项执行相同的检查，并还在检查过程中显示 chunk 和 tblspace extent 信息。-ce 和 -pe 选项还检查 sbspace chunk 中的 blobspace 、智能大对象 extent 以及用户数据和元数据信息。

有关使用 gcheck -ce 和 -pe 的信息，请参阅 GBase 8s 管理员指南 中的管理磁盘空间。

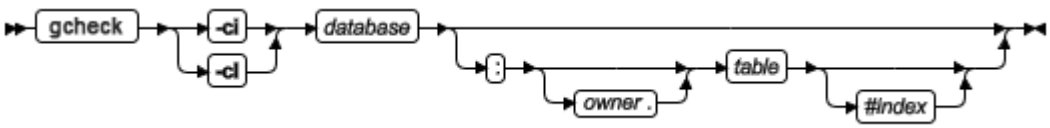
使用 CHECK EXTENTS 作为 gcheck -ce 的 SQL 管理 API 命令字符串。

3.4.6 gcheck -ci 和 -cI: 检查索引节点链接

使用 gcheck -ci 和 gcheck -cI 命令检查键值顺序和与指定表相关联的所有索引的水平和垂直节点链接的一致性。

gcheck -cI 选项还检查索引中与 rowid 相关联的键值是否与该行中的键值相同。-cI 选项不在功能检索上交叉检查数据。

语法:



如果未指定索引，该选项检查所有的索引。如果未指定表，该选项检查数据库中所有的表。

相同的 -ci 修复选项可以与 -cI 一起使用。如果 gcheck -ci 或 gcheck -cI 检测到不一致，那么它会提示您确认修复该问题索引。如果指定了 -y （是）选项，那么自动修复索引。如果指定了 -n （否）选项，那么报告该问题，但不进行修复；不出现任何提示。

如果 gcheck 未找到不一致性，那么以下消息出现：

```
validating indexes.....
```

该消息显示 gcheck 正在检查的索引的名称。

注： 使用 gcheck 重新构建索引可能会消耗很长时间。如果使用 SQL 语句 DROP INDEX 和 CREATE INDEX 删除并重建索引，那么处理速度通常会较快。

以下示例检查 customer 表上的所有索引：

```
gcheck -cI -n stores_demo:customer
```

以下示例检查 customer 表上的 zip_ix 索引：

```
gcheck -cI -n stores_demo:customer#zip_ix
```

如果索引在相同的 dbspace 中的多个分区上分片，那么 gcheck -ci 和 gcheck -cI 命令将显示分区名称。以下示例显示了在相同 dbspace 中的多个分区上分片的索引的典型输出：

```
Validating indexes for multipart:t1...
        Index idx_t1
        Index   fragment partition part_1 in DBspace dbs1
        Index   fragment partition part_2 in DBspace dbs1
        Index   fragment partition part_3 in DBspace dbs1
        Index   fragment partition part_4 in DBspace dbs1
        Index   fragment partition part_5 in DBspace dbs1
```

缺省情况下，当您使用 gcheck -ci 或 gcheck -cI 选项检查索引时，数据库服务器不在表上放置共享锁，除非该表使用页锁定。要绝对确保完全的索引检查，可以在执行 gcheck -ci 或 gcheck -cI 时使用 -x 选项。使用 -x 选项， gcheck 在表上放置共享锁，这样其他用户就不能在检查完成之前执行更新、插入或删除。有关在 gcheck -ci 和 gcheck -cI 中使用 -x 选项的更多信息，请参阅 使用 -x 开启锁 。

当在外部索引上执行 gcheck 时，用户定义的存取方法负责检查和修复索引。如果使用用户定义存取方法的索引无法找到该存取方法，那么数据库服务器报告错误。gcheck 实用程序不会修复外部索引中的不一致。对于包含多种索引类型的表，不应使用 gcheck -cI。

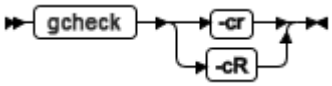
gcheck 实用程序在检查索引时需要排序空间 。所需排序的空间量与需要创建索引的空间量大小相同。有关计算所需临时空间量的更多信息，请参阅 ../prf/ids_prf_387.html#ids_prf_387。如果您收到该错误 "no free disk space for sort" ，您必须重新预估所需的临时空间量并使其可用。

重要：如果正在使用 Verity Text Search DataBlade 模块，那么 -cI 选项执行索引合并而不是通常的操作。

有关索引的更多信息，请参阅 B-Tree 索引页的结构 。

3.4.7 gcheck -cr 和 -cR: 检查保留页

语法:



-cr 选项按以下方式检查每个 dbspace 保留页：

- 它对 PAGE_CONFIG 保留页验证 ONCONFIG 文件的内容。
- 它确保所有 chunk 都可以打开、chunk 不重叠以及 chunk 大小是正确的。

-cR 选项执行相同的检查和验证，并且它还会检查所有逻辑日志和物理日志页的一致性。-cr 选项是相当快的，因为它不检查日志文件页。

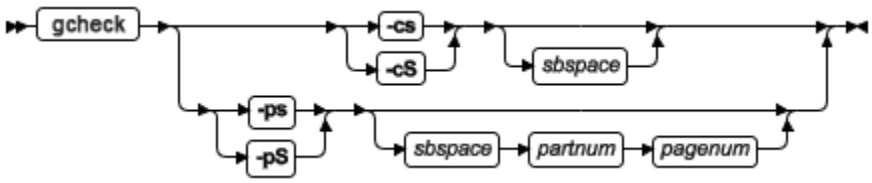
如果您已更改了配置参数值（通过 glogadmin 、onmonitor 、gspaces 或通过编辑配置文件），但还未重新初始化共享内存，那么 gcheck -cr 和 gcheck -cR 将会检测到不一致并返回错误消息。

如果 gcheck -cr 在执行之后不显示任何错误消息，那么您可以认为前述列表中的所有三项都已成功通过了检查。

有关保留页的更多信息，请参阅 保留页 。

3.4.8 gcheck -cs 、-cS 、-ps、-pS: 检查并显示 sbspace

语法:



-cs 选项检查 sbspace 。-ps 选项检查 sbspace 和 extents 。

-cS 选项验证并显示 sbspace 元数据。

-ps 选项检查 sbspace 和 extent 。如果未指定 sbspace 名，这些选项检查所有的 sbspace 。

-pS 选项验证并显示 sbspace 元数据，并且还列出智能大对象的 extent 和头信息。

如果未指定 sbspace 名，那么将会检查所有的 sbspace。以下示例检查和显示 test_sbspace 的元数据：

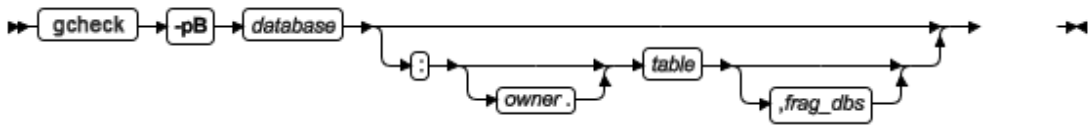
```
gcheck -ps test_sbspace
```

如果使用 -cs 或 -ps 选项将 rootdbs 指定为 sbspace 的名称，那么 gcheck 检查该 root dbspace 。

有关使用 -cs 、-cS 、-ps 和 -pS 选项的更多信息，请参阅 GBase 8s 管理员指南 。

3.4.9 gcheck -pB：显示 blobspace 统计信息

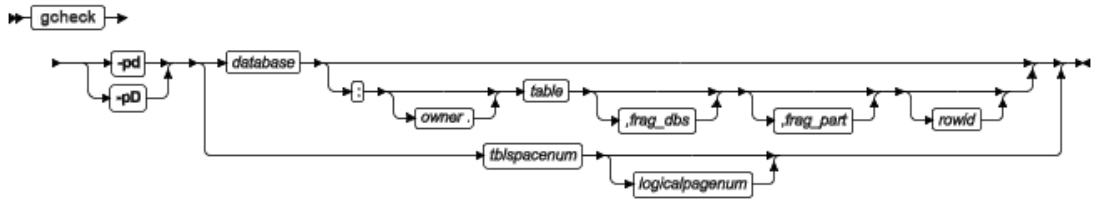
语法:



-pB 选项显示描述指定表中 blobspace blobpage 的平均充满度的统计信息。这些统计信息衡量数据库或表中的个别简单大对象的存储效率。如果未指定表或分片，那么此选项显示整个数据库的统计信息。有关更多信息，请参阅 GBase 8s 管理员指南 中的 管理磁盘空间 一章中的“优化 blobspace blobpage 大小”。

3.4.10 gcheck -pd 和 pD ：以十六进制格式显示行

语法:



-pd 将数据库、表、分片分区（碎片）和特定 rowid 或 tblspace 号以及逻辑页号作为输入。在每种情况中，-pd 会打印页头信息并显示数据库对象（数据库、表、分片、内部 rowid 或页号）的指定行，这些数据库对象是以十六进制和 ASCII 格式指定的。不执行任何一致性检查。

元素	用途	关键注意事项
database	指定要检查其一致性的数据库名称	语法必须符合 Identifier 段；请参阅《*GBase 8s SQL 指南：语法*》
frag_dbs	指定包含想要检查其一致性的分片的 dbspace 名称	Dbspace 必须存在并包含要检查一致性的分片。 语法必须符合 Identifier 段；请参阅《*GBase 8s SQL 指南：语法*》
frag_part	指定分片分区	对于使用基于表达或循环法分发计划的分片表，您可以创建多个分区，这些分区在一个 dbspace 内集合了表或索引的页面。该分区称为*分片分区*或*分片*
logical pagenum	指定用于指示 tblspace 中的特定页的整数值	值可以表示为无符号整数或以 0x 标识符开始的十六进制数。 值必须是 0 和 16,777,215（包括 0 和 16,777,215）之间的无符号整数
owner	指定表的所有者	必须指定表的当前所有者。 语法必须符合 Owner Name 段；请参阅《*GBase 8s SQL 指南：语法*》
rowid	标识要显示其内容的行的 rowid 。Rowid 是作为	值必须是 0 和 4,277,659,295（包括 0 和 4,277,659,295）之间的无符号整数。

元素	用途	关键注意事项
	gcheck -pD 输出的一部分进行显示的。	值可以表示为无符号整数或以 0x 标识符开始的十六进制数
table	指定想要检查其一致性的表的名称	执行实用程序时，表应存在。 语法必须符合 Table Name 段：请参阅《*GBase 8s SQL 指南：语法*》
tblspacenum	标识要显示其内容的 tblspace	值必须是 0 和 208,666,624（包括 0 和 208,666,624）之间的无符号整数。 值可以表示为无符号整数或以 0x 标识符开始的十六进制数

如果指定了内部 rowid（以十六进制值表示），那么该 rowid 映射到特定的页，且打印该页中的所有行。

如果指定了逻辑页号（以十进制值表示），那么打印具有该逻辑页号的 tblspace 号的所有行。

如果指定了分片，那么打印该分片中的所有行，带有其 rowid 、转发指针和页类型。

如果指定了表，那么打印该表中的所有行，带有其 rowid 、转发指针和页类型。

如果指定了数据库，那么打印该数据库中所有表的所有行。将会打印存储在数据行中的 TEXT 和 BYTE 列描述符，但不会打印 TEXT 和 BYTE 数据本身。

-pD 选项打印与 -pd 相同的信息。此外，-pD 打印存储在 tblspace 中的 TEXT 和 BYTE 值和存储在 blobspace blobpage 中简单大对象的头信息。以下示例显示了 gcheck -pd 和 gcheck -pD 命令的不同选项：

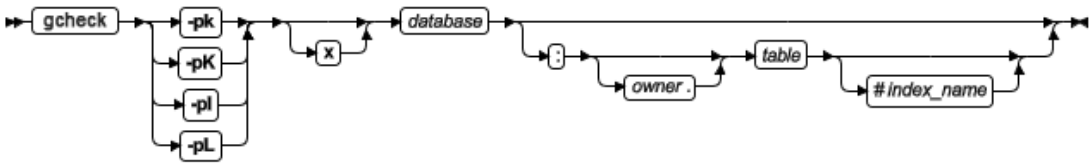
```
gcheck -pd stores_demo:customer,frgmnt1
gcheck -pd stores_demo:customer
gcheck -pD stores_demo:customer 0x101
```

以下示例显示了 gcheck -pD 命令的部分输出：

```
gcheck -pD multipart:t1 :

        TBLspace data check for multipart:t1
        Table fragment partition part_1 in DBspace dbs1
        page_type    rowid       length fwd_ptr
        HOME            101          24         0
        0:    0    0    0    a 47 48 49 20 20 20 20 20 20 20 20 20     ....GHI
        16: 20 20 20 20 20 20 20 20
        ........
```

3.4.11 gcheck -pk 、-pK 、-pI 、-pL：显示索引信息

语法：



-pk 选项执行与 -ci 选项相同的检查，此外，在进行检查时，它还会显示所有指定的表中的所有索引的键值。

-pK 选项执行与 -cI 选项相同的检查，此外，在进行检查时，它还会显示键值和 rowid 。

-pI 选项执行与 -ci 选项相同的检查，并显示键值，但它只检查叶节点索引页。它忽略根和分支节点。

-pL 选项执行与 -cI 选项相同的检查，并显示键值和 rowid ，但它只检查叶节点索引页。它忽略根和分支节点。

元素	用途	关键注意事项
database	指定要检查其一致性的数据库名称	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*
index_name	指定要检查其一致性的索引名称	索引必须存在于特定数据库中的表上。 语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*
owner	指定表的所有者	必须指定表的当前所有者。 语法必须符合 Owner Name 段；请参阅 *《GBase 8s SQL 指南：语法》*

元素	用途	关键注意事项
table	指定想要检查其一致性的表的名称	执行实用程序时，表应存在。 语法必须符合 Table Name 段；请参阅《*GBase 8s SQL 指南：语法*》
-x	检查和打印索引时在表上放置共享锁	更多完整信息，请参阅使用 -x 开启锁

如果任何一个 gcheck 选项检测到不一致，那么将会提示您确认修复问题索引。如果指定了 -y （是）选项，那么自动修复索引。如果指定了 -n （否）选项，那么报告该问题，但不进行修复；不出现任何提示。

以下示例显示有关 customer 表上所有索引的信息：

```
gcheck -pl -n stores_demo:customer
```

以下示例显示了有关索引 zip_ix 的信息，该索引时创建在 customer 表上的：

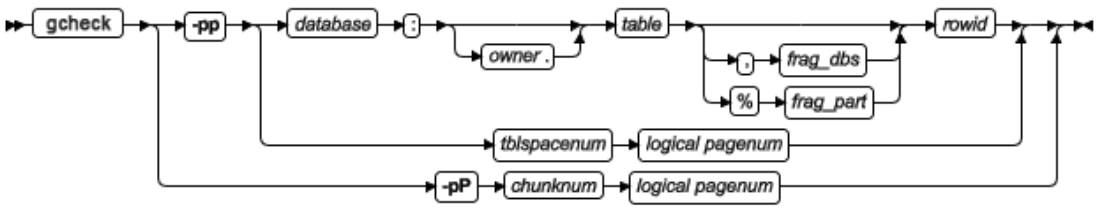
```
gcheck -pl -n stores_demo:customer#zip_ix
```

缺省情况下，当您使用 gcheck -pk 、-pK 、-pl 或 -pL 选项检查索引时，数据库服务器不在表上放置共享锁，除非该表使用页锁定。要绝对确保完全的索引检查，可以在执行 gcheck –pk 、gcheck -pK 、gcheck -pl 或 gcheck -pL 时使用 -x 选项。使用 -x 选项，gcheck 在表上放置共享锁，这样其他用户就不能在检查完成之前执行更新、删除或插入。有关使用 -x 选项的更多信息，请参阅使用 -x 开启锁 。

有关更多 gcheck -ci 的信息，请参阅 gcheck -ci 和 -cI: 检查索引节点链接。有关索引页的更多信息，请参阅 B-Tree 索引页的结构。

3.4.12 gcheck -pp 和 -pP: 显示逻辑页的内容

语法:



元素	用途	关键注意事项
database	指定要检查其一致性的数据	语法必须符合 Identifier 段；请参阅

元素	用途	关键注意事项
	库名称	*《GBase 8s SQL 指南：语法》*
chunknum	指定用于指示特定 chunk 的十进制值	值必须是大于 0 的无符号整数。Chunk 必须存在
frag_dbs	指定包含想要检查其一致性的分片的 dbspace 名称	Dbspace 必须存在并包含要检查一致性的分片。 语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*
frag_part	指定要检查分片的分区名。这会在在同一 dbspace 中创建的表上有多个分片的情况下有所帮助	对于使用基于表达或循环法分发计划的分片表，您可以创建多个分区，这些分区在一个 dbspace 内集合了表或索引的页面。该分区称为*分片分区*或*分片*
logical pagenum	指定用于指示 tblspace 中特定页的整数值	值可以表示为无符号整数或以 0x 标识符开始的十六进制数。值必须是 0 和 16,777,215（包括 0 和 16,777,215）之间的无符号整数
owner	指定表的所有者	必须指定表的当前所有者。 语法必须符合 Owner Name 段；请参阅 *《GBase 8s SQL 指南：语法》*
rowid	标识要显示其内容的行的 rowid 。Rowid 是作为 **gcheck -pD** 输出的一部分进行显示的。	值必须是 0 和 4,277,659,295（包括 0 和 4,277,659,295）之间的无符号整数。 值可以表示为无符号整数或以 0x 标识符开始的十六进制数
table	指定想要检查其一致性的表的名称	执行实用程序时，表应存在。 语法必须符合 Table Name 段；请参阅 *《GBase 8s SQL 指南：语法》*

元素	用途	关键注意事项
tblspacenum	标识要显示其内容的 tblspace	值必须是 0 和 208,666,624（包括 0 和 208,666,624）之间的无符号整数。 值可以表示为无符号整数或以 0x 标识符开始的十六进制数

-pp 选项具有以下语法变化：

调用	解释
gcheck -pp tblspc lpn <pages>	使用 tblspace 号和逻辑页号，显示逻辑页的内容，也可以指定用于指示要打印页数的可选参数
gcheck -pp *tblspc lpn -h*	使用 tblspace 号和逻辑页号，只显示逻辑页头
gcheck -pp *database:table rowid*	使用数据库名、表名和 GBase 8s 内部 rowid ，显示逻辑页的内容。可以使用 gcheck -pD 命令获得该内部 rowid 。该内部 rowid 不是在用 CREATE TABLE tabname WITH ROWIDS 语句创建的表中指定的序列 rowid 。有关更多信息，请参阅Rowid 的定义

该页内容以 ASCII 格式显示。显示还包括页上slot 表条目数。以下示例显示了gcheck -pp 命令的其他调用：

```
gcheck -pp stores_demo:orders 0x211   # database:owner.table,    # fragment rowid
gcheck -pp stores_demo:gbasedbt.customer,frag_dbspce1 0x211
gcheck -pp 0x100000a 25 # specify the tblspace number and # logical page number
```

-pP 选项具有以下语法变化：

调用	解释
gcheck -pP chunk# offset pages	使用 chunk 号和偏移量，显示逻辑页的内容。也可以指定用于指示要打印页的可选参数
gcheck -pP *chunk# offset -h*	使用 chunk 号和偏移量，只显示逻辑页头

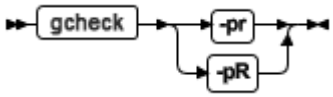
注： chunk 页的输出以十进制格式显示 start 和 length 字段。

以下示例显示使用了 gstat -pP 命令的典型输出：

```
gcheck  -pP 1 5 2
addr         stamp         nslots         flag      type         frptr      frcnt      next      prev
stamp          100005         250181          2         1000      ROOTRSV         320      1716      0
0      250181      slot      ptr      len      flg
...
addr      stamp         nslots         flag      type         frptr      frcnt      next      prev
stamp      100005   6      250182      2      1000      ROOTRSV      128      1908      0      0
250182      slot      ptr      len      flg      1      24      56      0
2      80      48      0
```

3.4.13 gcheck -pr 和 pR：显示保留页信息

语法：



-pr 选项执行与 gcheck -cr 相同的检查并显示保留页的信息。

-pR 选项执行与 onchdeck -cR 相同的检查，显示保留页的信息，并显示有关逻辑日志和物理日志页的详细信息（标记活动物理日志页的开始和结束）。

如果已更改了配置参数信息（通过编辑配置文件），但还未重新初始化共享内存，那么 gcheck -pr 和 gcheck -pR 检测到不一致性并返回错误消息。

有关 gcheck -pr 输出的列表和解释，请参阅 保留页。有关 -cr 选项的描述，请参阅 gcheck -cr 和 -cR: 检查保留页。

3.4.14 gcheck -pt 和 -pT：显示表或分片的 tblspaces

gcheck -pt 和 gcheck -pT 选项打印指定表或分片的 tblspace 报告。它们之间的仅有一个区别：gcheck -pT 打印更多的信息，包括一些特定索引的信息。

语法

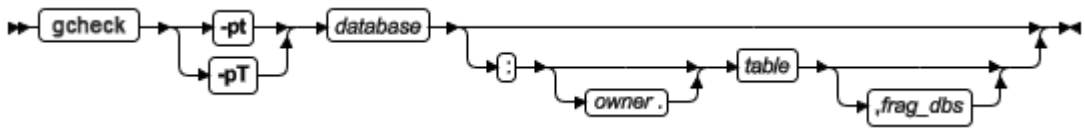


表 1. gcheck -pt 和 gcheck -pT 命令的选项

元素	用途	关键注意事项
database	指定要检查其一致性的数据库名称	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*
frag_dbs	指定包含想要检查其一致性的分片的 dbspace 名称	Dbspace 必须存在并包含要检查一致性的分片。

元素	用途	关键注意事项
		语法必须符合 Identifier 段；请参阅 Identifier
owner	指定表的所有者	必须指定表的当前所有者。 语法必须符合 Owner Name 段；请参阅 Owner name
table	指定想要检查其一致性的表的名称	表应存在。 语法必须符合 Table Name 段；请参阅 *《GBase 8s SQL 指南：语法》*

-pt 选项打印给定分片和数据库的 tblspace 报告。如果未指定表，该选项显示数据库中所有表的信息。该报告包含一般的分配信息，包括最大行大小、键数量、extent 数量、其大小、每个 extent 所分配和使用的页、当前的序列值以及表的创建日期。-pt 输出打印了 tblspace 页大小、逻辑页的页数（分配的页、使用的页和数据页）。

TBLspace Flags 字段显示了 tblspace 的配置信息，包括该 tblspace 是用于 Enterprise Replication 还是时间序列数据。

Extents 字段列出了该表中 tblspace tblspace 条目的物理地址和第一个 extent 的首页地址。该 extent 列表显示了每个事件中的逻辑页号和物理页号。

-pT 选项打印与 -pt 选项相同的信息。此外，-pT 选项还显示以下信息：

- 特定索引信息
- 按页类型分配页的信息（为 dbspace）
- 表或表分片中压缩的行数以及其压缩的百分比

如果表或分片行没有被压缩，那么“Compressed Data Summary”段落不会出现在输出中。当您想要运行选项时请做出计划，因为它要完整扫描一个分区。

-pt 和 -pT 的输出包含已使用页数的列表。输出中显示的该字段的值绝不会减小，因为作为 extent 的一部分分配给 tblspace 的磁盘空间仍保持专用于该 extent，即使在通过删除行而释放空间之后也是如此。有关当前使用页数的准确计数，请参阅 -pT 选项提供的有关 tblspace 使用情况（按页类型组织）的详细信息。

gcheck -pt 输出样本

以下示例显示了 gcheck -pt 命令输出的样本：

```
TBLspace Report for testdb:tab1

        Physical Address              2:10
        Creation date                 10/07/2004 17:01:16
        TBLspace Flags                801        Page Locking
        TBLspace use 4 bit bit-maps
        Maximum row size              14
        Number of special columns     0
        Number of keys                0
        Number of extents             1
        Current serial value          1
        Pagesize (k)                  4
        First extent size             4
        Next extent size              4
        Number of pages allocated     340
        Number of pages used          337
        Number of data pages          336
        Number of rows                75806
        Partition partnum             2097154
        Partition lockid              2097154

        Extents
        Logical Page     Physical Page        Size Physical Pages
        0              2:106         340        680
```

gcheck -pT 输出示例

以下示例显示了gcheck -pT 命令的输出：

```
TBLspace Report for database_a:nilesh.table_1a

       Table fragment partition dbspace1 in DBspace dbspace1

       Physical Address              3:5
       Creation date                 03/21/2009 15:35:47
       TBLspace Flags                8000901    Page Locking
```

```
TBLspace contains VARCHARS
TBLspace use 4 bit bit-maps
TBLspace is compressed
Maximum row size                80
Number of special columns       1
Number of keys                  0
Number of extents               1
Current serial value            100001
Current SERIAL8 value           1
Current BIGSERIAL value         1
Current REFID value             1
Pagesize (k)                    2
First extent size               8
Next extent size                8
Number of pages allocated       24
Number of pages used            22
Number of data pages            14
Number of rows                  500
Partition partnum               3145730
Partition lockid                3145730

Extents
Logical Page     Physical Page        Size Physical Pages
0           3:16053          24         24

Type                 Pages   Empty  Semi-Full    Full  Very-Full
---------------- ---------  ------- ---------- ------ --------
Free                     9
Bit-Map                  1
Index                    0
Data (Home)             14
Data (Remainder)         0        0        0       0       0
----------
Total Pages             24
```

```
Unused Space Summary

Unused data bytes in Home pages                 1177
Unused data bytes in Remainder pages               0

Home Data Page Version Summary

Version                 Count
0 (current)             14

Compressed Data Summary
Number of compressed rows and percentage of compressed rows  500 100.00
```

3.4.15 使用 -x 开启锁

-x 选项可以附加到 -ci 、-cI 、-pk 、-pK 、-pl 和 -pL 选项后，用于在受影响的表上放置共享锁。在表被锁定时，其他用户在 gcheck 检查或打印索引时无法执行插入、更新和删除。对于带有行锁定的表，在未使用 -x 选项时，gcheck 只在该表上放置 IS （意向共享）锁，该锁阻止在检查过程中执行删除表或索引之类的操作。

例如：以下样本命令指示当 gcheck 验证键值顺序、验证水平链接和确保索引中没有节点出现两次时，它锁定 customer 表的索引：

```
gcheck -cix stores_demo:customer
```

当指定选项 -x 时，gcheck 锁定使用行锁定的表的索引。如果 gcheck 检测到页锁定方式，它显示警告消息并在表上放置共享锁。

3.4.16 使用 -u 将特殊参数发送给存取方法

您可使用 -u 选项将特殊参数发送给存取方法。可能的参数视存取方法而定。例如：R-tree 存取方法支持 display 选项，如以下示例所示：

```
gcheck -pl -u "display"
```

使用逗号分隔参数串中的多个参数。

有关存取方法的有效参数信息，请您就存取方法参考用户手册。

3.4.17 退出时的返回码

gcheck 实用程序退出时返回以下代码。

```
GLS failures:-1
Invalid srial/key:2
```

```
Onconfig access error:2
Invalid onconfig settings:2
Invalid arguments to gcheck:2
Error connecting database server:1
Warning reported by gcheck:1
error detected by gcheck:2
no errors detected by gcheck:0
```

仅 Windows™ ：

```
Not properly installed:1
Authentication error:2
```

3. 5 onclean 实用程序

如果 gadmin 实用程序无法关闭数据库服务器或您无法重启服务器，可使用 onclean 实用程序强制执行立即关闭数据库服务器。onclean 实用程序试图去清除共享内存和信号量并且停止数据库服务器的虚拟进程。

语法

在 UNIX™ 和 Linux™ 上，必须以用户 root 或 gbasedbt 的身份执行 onclean 命令。在 Windows™ 上，必须以 Gbasedbt-Admin 组成员的身份执行该命令。

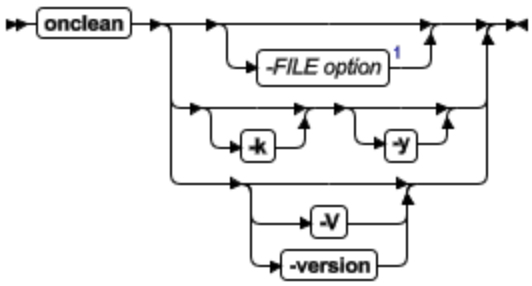


表 1. onclean 命令语法元素

元素	用途
-k	通过停止数据库服务器虚拟进程和尝试清除剩余信号量、共享内存片段（尽管它们仍在运行中）来关闭联机的服务器
-V	显示简短的版本信息
-version	显示所有的版本信息
-y	不提示输入确认

用法

只能在 gadmin 实用程序无法关闭数据库服务器或您无法重启服务器的情况下，使用 onclean 实用程序去停止数据库服务器。数据库服务器可能由于不可控的方法关闭并且无法恢复，或者它被挂载。如果数据库服务器无法重启，它以前的实例却仍会连接共享内存片段。可以检查消息日志来查看数据库服务器是否正常关闭。onclean 实用程序停止所有的 oninit 进程并尝试移除所有的共享内存片段和在 $GBASEDBTDIR/etc/.conf.dbservername 文件中记录的信号量。

注意： 请谨慎使用 onclean 实用程序。当运行 onclean 时，任意挂起的事务和进程将无法完成并且用户会话会突然断线。然而，当数据库服务器重新启动后，它会回滚这些事务。GBASEDBTDIR 环境变量必须设置可用的值以运行该实用程序。

onclean 命令在下例情况下使用：

- 如果不确定数据库服务器是否离线，可使用不带有选项的 onclean 命令。如果数据库服务器仍然在线，将会显示一个指导您运行 onclean -k 命令的消息。
- 如果数据库服务器离线，可使用 onclean 命令。
- 如果数据库服务器在线并且您确定要强制关闭它，可使用 onclean -k 命令。

使用 onclean 实用程序只能关闭本地数据库服务器；不能关闭远程数据库服务器。onclean 实用程序无法用于关闭一个整个高可用集群或远程数据库服务器。

onclean 实用程序在任何情况下可能不能清除数据库服务器使用的共享内存片段。onclean 实用程序会尝试只终止 oninit 进程。onclean 实用程序在以下情况下不会成功：

- 如果在运行 onclean 命令前有一非数据库服务器连接共享内存，onclean 实用程序将无法停止该进程以清除共享内存片段。
- 当应用程序或数据库服务器实用程序连接了网络端口时，onclean 可能无法保证一个干净的服务器启动。如果用户试图在同一网络端口初始化数据库服务器，数据库服务器会无法启动监听器进程并启动失败。onclean 实用程序无法停止释放网络端口的应用程序。

如果必要，您可以使用 onshutdown 脚本自动关闭数据库服务器，它会调用 onclean -ky 命令。

返回码

0

成功

1

因为以下其中一个问题失败：

- 错误的环境变量配置
- 运行 onclean 命令的权限错误
- 命令的语法错误

- 已损坏的信息
- 在联机的服务器上运行 onclean 命令时，没有使用 -k 选项

2

由于 onclean 使用的一个或多个操作系统的系统调用返回错误而失败。

3.5.1 onshutdown 脚本

可以使用 onshutdown 脚本自动关闭数据库服务器。该脚本尝试正常关闭服务器。如果该服务器在指定的时间后没有关闭，此脚本会强制关闭服务器。

onshutdown 脚本首先执行 gadmin -ky 命令。在等待指定时间后，该脚本运行 onclean -ky 命令。

在 UNIX™ 和 Linux™ 上，您必须是用户 root 或 gbasedbt ，才能执行 onshutdown 脚本。在 Windows™ 上，您必须是 Gbasedbt-Admin 群组的成员，才能执行 onshutdown 脚本。

语法

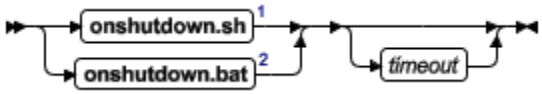


表 1. onshutdown 脚本的语法元素

元素	用途
timeout	在 gadmin -ky 命令执行之后和 onclean -ky 命令执行之前中间的等待时间（以秒为单位）。 必须是从 10 到 60 的正整数。缺省值是 30 秒

用法

在想强制关闭数据库服务器的情况下使用 onshutdown 脚本会比较合适。

注意：请谨慎使用 onshutdown 脚本。如果此脚本需要运行 onclean -ky 命令，那么任意挂起的事务和进程将无法完成并且用户会话会突然断线。然而，当数据库服务器重新启动后，它会回滚这些事务。

GBASEDBTDIR、GBASEDBTDIR 环境变量必须设置可用的值以运行该实用程序。

您只能使用 onshutdown 脚本关闭本地数据库服务器；不能关闭远程数据库服务器。

onshutdown 脚本无法用于关闭一个整个高可用集群或远程数据库服务器。

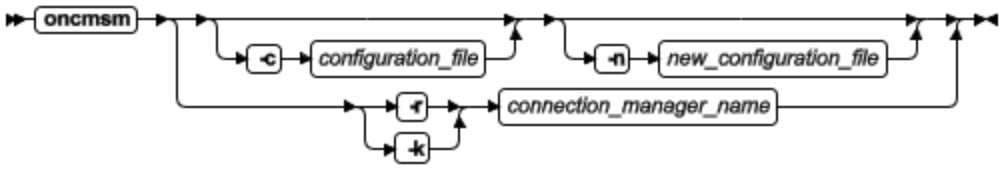
onshutdown 脚本有十秒时间段期间，在该期间内它可以被中止。

3.6 oncmsm 实用程序

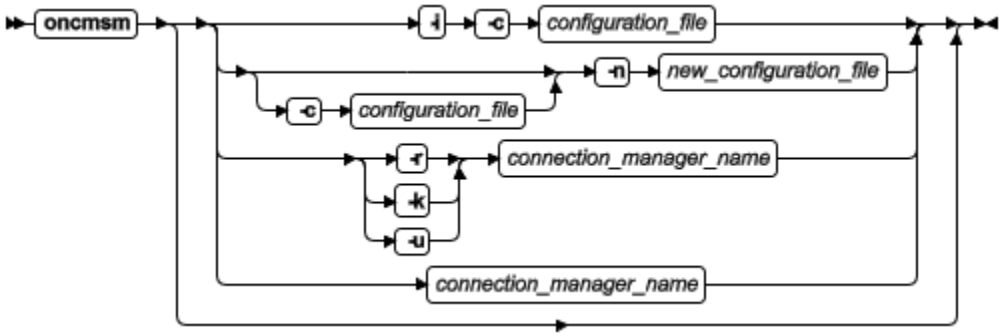
可以使用 oncmsm 实用程序启动或关闭连接管理器、加载新的配置文件到连接管理器以便修改连接管理器的设置或更新配置文件格式。

语法

UNIX 语法图:



Windows 语法图:



元素	用途	关键注意事项
-c	启动连接管理器或转换当前的连接管理器配置文件的格式	
connection_manager_name	指定连接管理器实例的名称	
-i	在 Windows™ 服务上安装连接管理器	该选项只在 Windows 平台上可用
-k	关闭连接管理器实例	
-n	指定转换配置文件的名称	
new_configuration_file	输出到 $GBASEDBTDIR/etc 目录下作为格式转换过程的一部分的文件名	

元素	用途	关键注意事项
	称	
configuration_file	位于 $GBASEDBTDIR/etc 目录下配置文件的名称	如果没有指定配置文件，连接管理器会试图加载到 $GBASEDBTDIR/etc/cmsm.cfg
-r	在不停止和重启的连接管理器的情况下重新加载连接管理器	
-u	在 Windows 服务上卸载连接管理器	该选项只在 Windows 平台上可用

用法

从命令行运行 oncmsm 实用程序来初始化连接管理器。连接管理器运行时，您可以添加、更改或删除服务等级协议（SLAs），然后重新加载该配置文件。

仅 UNIX ： 下列用户可以执行 oncmsm 实用程序：

- gbasedbt 用户
- root 用户，如果用户有连接到 sysadmin 数据库的权限
- DBSA 群组成员，如果用户有连接到 sysadmin 数据库的权限

仅 Windows ： 下列用户可以执行 oncmsm 实用程序：

- Gbasedbt-Admin 群组成员
- administrator 用户，如果用户有连接到 sysadmin 数据库的权限
- DBSA 群组成员，如果用户有连接到 sysadmin 数据库的权限

在启动它之前，您必须安装 oncmsm 实用程序作为一个服务。

oncmsm 实用程序可以按以下两种方式启动：

- 执行 oncmsm 命令。
- 点击 Start > Control Panel > Administrative Tools > Services 然后启动 oncmsm 。

如果您在使用多个连接管理器，可以执行 gstat -g cmsm 来显示连接管理器实例的名称。

示例 1: 启动连接管理器（UNIX）

以下示例中的连接管理器的 configuration_file_1 存在于 $GBASEDBTDIR/etc 目录。在已安装连接管理器的电脑上运行以下命令行，即可启动连接管理器：

```
oncmsm -c configuration_file_1
```

连接管理器启动。

示例 2: 启动连接管理器 (Windows)

以下示例中的连接管理器的 configuration_file_2 存在于 $GBASEDBTDIR/etc 目录。在已安装连接管理器的电脑上运行以下命令行，即可启动连接管理器：

```
oncmsm -i -c configuration_file_2
oncmsm connection_manager_2
```

名为 connection_manager_2 的连接管理器启动。

示例 3: 终止连接管理器

在已安装连接管理器的电脑上运行以下命令行，即可终止连接管理器：

```
oncmsm -k connection_manager_3
```

名为 connection_manager_3 的连接管理器停止。

示例 4: 重新加载连接管理器

对于以下示例，名为 connection_manager_4 管理器的 $GBASEDBTDIR\etc\configuration_file_4会被更改。要更新该连接管理器的设置，在已安装 connection_manager_4 的电脑上执行以下命令：

```
oncmsm -r connection_manager_4
```

示例 5: 转换连接管理器配置文件成当前的格式

以下示例中名为 cmsm.cfg 的文件存储在$GBASEDBTDIR/etc 目录下。要启动该连接管理器，在已安装连接管理器的电脑上执行以下命令：

```
oncmsm -n configuration_file_5
```

oncmsm 实用程序转换 cmsm.cfg 成当前配置文件的格式。然后在 $GBASEDBTDIR/etc/ 目录下输出名为 configuration_file_5 的文件。

示例 6: 转换指定的连接管理器配置文件成当前的格式

以下示例中名为 configuration_file_4 的文件存储在 $GBASEDBTDIR/etc 目录下。要启动该连接管理器，在已安装连接管理器的电脑上执行以下命令：

```
oncmsm -c configuration_file_6 -n configuration_file_7
```

oncmsm 实用程序转换 configuration_file_6 成当前配置文件格式。然后在 $GBASEDBTDIR/etc/ 目录下输出名为 configuration_file_7 的文件。

示例 7: 卸载连接管理器 (Windows)

对于该示例，您已经在 Windows 服务上安装了名为 connection_manager_4 的连接管理器。要卸载该连接管理器，在已安装连接管理器的电脑上执行以下命令：

```
oncmsm -u connection_manager_4
```

oncmsm 实用程序可用于卸载连接管理器。

3.7 onconfig_diff 实用程序

使用 onconfig_diff 实用程序比较两个 onconfig 文件。

语法

元素	用途
-d	对照当前的 onconfig 文件的设置来进行不同设置
-c	同另一个 onconfig 文件做比较
-f *filepath_1*	指定要比较的第一个文件名称。除非该文件在 $GBASEDBTDIR/bin目录下，否则需提供该文件的路径
-s *filepath_2*	指定要比较的第二个文件名称。除非该文件在 $GBASEDBTDIR/bin目录下，否则需提供该文件的路径

用法

可以执行 onconfig_diff 实用程序比较两个不同的 onconfig 文件。该 onconfig_diff 实用程序在 $GBASEDBTDIR/bin 里。

要比较的这两个文件必须在同一目录下。

以下为使用该实用程序的以下方法：

- 比较当前的 onconfig 和同版本的 onconfig。
- 比较当前的 onconfig 和新版本的 onconfig。
- 比较两个在不同服务器上的 onconfig 文件。

示例

在该示例中，onconfig.std 文件与 onconfig.production 文件作比较：

```
$ onconfig_diff -c -f onconfig.std -s onconfig.production
```

以下是该命令的输出：

```
========================================
  File 1: onconfig.std
  File 2: onconfig.production
 ========================================
 Parameters Found in File 1, not in File 2
```

```
==========================================

FULL_DISK_INIT  0

NETTYPE         ipcshm,1,50,CPU

NUMFDSERVERS    4
...
==============================================
Parameters Found in File 2, not in File 1
==============================================

JVPJAVAHOME    $GBASEDBTDIR/extend/krakatoa/jre

...
==============================================
Parameters Found in both files, but different
==============================================
ROOTPATH

File 1: $GBASEDBTDIR/tmp/demo_on.rootdbs
File 2: /usr2/support/grantf/g1150fc8/rootdbs

LOGFILES
File 1: 6
File 2: 10
LOGSIZE
File 1: 10000
File 2: 3000
...
```

3.8 gdblogmode 实用程序

本章说明如何使用 gdblogmode 实用程序。

3.8.1 gdblogmode: 更改日志记录方式

可以使用 gdblogmode 实用程序更改一个或多个数据库的日志记录方式。

此外，您也可以通过使用 alter logmode SQL 管理 API 命令字符串更改日志记录方式。

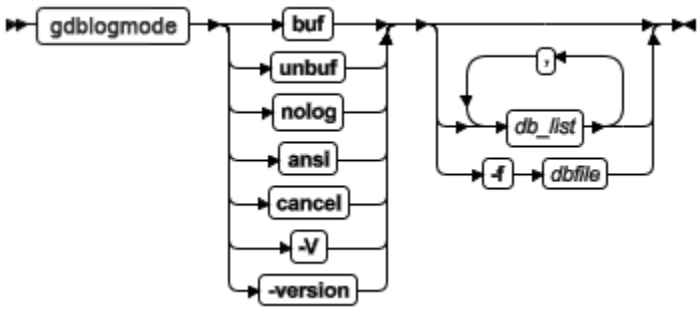
gdblogmode 实用程序将其输出记录到 BAR_ACT_LOG 文件。

如果开启数据库的事务记录日志，那么在更改生效之前，您必须对数据库中包含数据的所有存储空间创建 0 级备份。

有关日志记录方式的更多信息和示例。请参阅 GBase 8s 管理员指南 中的 数据库日志状态 一章中的以下主题：

- 修改数据库日志状态
- 修改表日志状态

您不能在高可用数据复制（HDR）辅助服务器、远程独立（RS）辅助服务器和共享磁盘（RS）辅助服务器上使用 gdblogmode 实用程序。



gdblogmode 语法

元素	用途	关键注意事项
buf	设置日志记录方式，以便事务信息在写入逻辑日志之前写入缓冲区中	无
unbuf	设置日志记录方式，以便数据在写入逻辑日志之前写入缓冲区中	无
nolog	设置日志记录方式，以便不对数据库事务进行日志记录	无
ansi	将数据库日志记录更改为服从 ANSI	一旦创建或将数据库转换为 ANSI 方式，就不能再将它更改回其他任何日志记录方式
cancel	在发生下一次 0 级备份之前取消日志记录方式更改请求	无
-f *dbfile*	更改文本文件中列出的（每行一个）数据库日志记录状态。该文本文件的路径由 *dbfile* 给出	如果数据库列表很长或经常使用，那么此命令是有用的
db_list	给出要更改其日志记录状态的数据库的空间限定列表的名称	如果未指定任何内容，那么修改数据库服务器管理的所有数据库

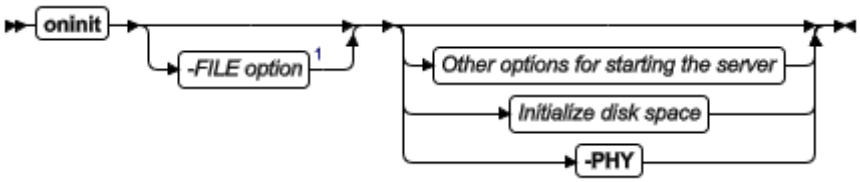
3.9 oninit 实用程序

oninit 实用程序启动数据库服务器。

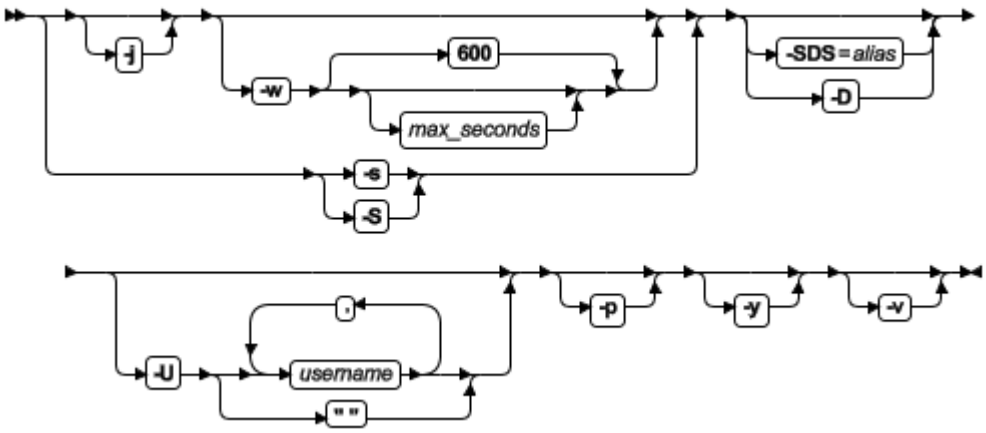
在 UNIX™ 和 Linux™ 上，必须作为用户 root 或 gbasedbt 登录才能执行 oninit ，或者非根数据库服务器的所有者运行 oninit 实用程序。用户 gbasedbt 应该是组 gbasedbt 的唯一成员。从命令行运行 oninit 命令。您可以允许属于 DBSA 群组的用户执行 oninit 命令。请参阅允许 DBSA 群组用户执行 oninit 命令 (UNIX)。

在 Windows™ 上，GBase 8s 作为 Windows 服务运行。任一具有启动 Windows 服务权限的用户都可以启动 GBase 8s 服务。该服务控制程序执行带有您提供的选项的 oninit 实用程序。

语法



Initialize disk space



元素	用途	关键注意事项
-D	用 Enterprise Replication 启动数据库服务器并禁用高可用集群复制	
-i	初始化根 dbspace 的磁盘空间，以至于它可以被	磁盘空间需要被初始化一次，以对服务器准备的数据存储。

元素	用途	关键注意事项
	数据库服务器用于执行启动操作。	缺省情况下，为了防止数据丢失，不能重新初始化磁盘空间。要重新初始化现有的根 dbspace，必须设置 FULL_DISK_INIT 配置参数为 1 并执行 oninit -i 命令。 请参阅 初始化根 sbspace 的磁盘空间
-j	在管理模式下启动服务器	请参阅 在管理模式下启动服务器
-p	在不删除临时表的条件下启动数据库服务器	如果使用该选项，数据库服务器将加快启动，但是磁盘上临时表使用的空间不会被回收
-PHY	启动服务器最新的检查点。该 -PHY 选项用于告诉服务器只做物理恢复而不需要逻辑恢复。	该选项常用于启动辅助服务器。必须使用下列命令来连接辅助服务器到主服务器： gadmin -d secondary gadmin -d RSS 如果主服务器上最新的检查点没有在辅助服务器上运行，那么上述连接失败
-s	在静态模式下启动服务器	使用该选项时，必须关闭数据库服务器。 当数据库服务器处于静态模式，只有用户 gbasedbt 可以访问数据库服务器
-S	在静态模式下，以标准方式启动数据库；禁用 HDR	当数据库服务器处于静态模式，只有用户 gbasedbt 可以访问数据库服务器
-SDS= *alias*	对于共享磁盘服务器，启动当前的服务器并用序列名指定主服务器	当主服务器和所有的 SDS 服务器都关闭时，使用 -SDS=*alias* 选项启动已设计的 SDS 服务器作为主服务器。The -SDS=*alias* 标志不能与 -i 标志结合。
-U *username*	指定哪些用户可以以管理模式访问服务器当前的会话	gbasedbt 用户和 DBSA 群组成员都是具有管理模式的用户。

元素	用途	关键注意事项
		请参阅 在管理模式下启动服务器
-v	显示服务器启动时的详细信息	
-w *max_seconds*	启动数据库服务器，并等待指示成功或失败，直到该服务器在联机模式下完全启动或经过 *max_seconds* 指定的秒数	等待时间的缺省值是 600 。 该选项在高可用集群的辅助服务器上不可用。 请参阅 使用脚本启动服务器
-y	防止验证提示	-y 选项默认所有验证提示为是

用法

缺省情况下，oninit 实用程序显示在服务器启动过程中的验证提示。可以使用 -y 选项忽略这些验证提示。使用 -v 选项查看其详细信息。在UNIX 和 Linux ， oninit 输出显示的是标准输出。在 Windows， 可以通过设置 ONINIT_STOUT 环境变量来查看 oninit 输出并保存该输出到一个文件中。

可以以不同操作方式启动服务器。缺省情况下，如果执行不带选项的 oninit 命令，那么该服务器以联机方式启动。当数据库服务器在联机模式下，所有被授权的用户都可访问此服务器。

如果执行 oninit -FILE 命令，那么不须在启动数据库服务器之前蛇者本地环境变量。数据库服务器自动使用 onconfig 文件中的值作为其环境变量。

在管理模式下启动服务器

管理模式是仅限管理员操作模式，可用来执行维护操作包括那些需要运行的 SQL 或 DDL 命令。当处于管理模式，数据库服务器只同意下列用户的请求：

- gbasedbt 用户
- DBSA 群组成员
- 对于当前会话，使用 oninit -U 命令或 gadmin -j -U 命令指定的用户。-U 选项可以重写 onconfig文件中 ADMIN_MODE_USERS 配置参数列出的任何用户。
- ADMIN_MODE_USERS 配置参数指定的用户

要使用 -U 选项添加管理方式用户时，用逗号分隔用户名 ，例如： Karin,Sarah,Andrew.

可使用 -U " " 选项移除除 gbasedbt 用户和 DBSA 群组之外的所有管理方式用户：oninit -U " "。

初始化根 sbspace 的磁盘空间

在系统上首次安装 GBase 8s 后，需要初始化数据库服务器的根 dbspace 的磁盘空间。该根 dbspace 由 ROOTPATH 配置参数指定。

如果您执行一个典型的安装并且选择创建数据库服务器或执行客户型安装，那么磁盘空间会自动初始化。否则，您必须通过执行 oninit -i 命令初始化磁盘空间。

如果必要，可以重新初始化磁盘空间。重新初始化磁盘会破坏数据库服务器当前所管理的所有数据。当重新初始化时，数据库服务器必须处于脱机方式。

缺省情况下，不能重新初始化正被数据库服务器使用的根 dbspace 。如果一个 0 页在根路径下（位于第一个 chunk 的第一页），磁盘初始化失败。可以通过设置 FULL_DISK_INIT 配置参数为 1 来运行现有的根 dbspace 的磁盘重新初始化。

使用脚本启动服务器

可以在自定义的启动脚本中使用 oninit -w 命令执行自动启动。-w 选项强制服务器等待，直到启动完全成功，并表明服务器在联机模式下，通过返回到 shell 的提示符为 0的返回码 。如果服务器不在联机模式下的超时期限内，服务器将返回代码 1 返回到 shell 提示符，并在联机日志中写入一条警告消息。

超时的缺省值为 600 秒（10 分钟），您可以修改成任一整型值。

在下列命令运行完毕之后，如果数据库在 60 秒内启动失败，会出现一个码为 1 的提示：

oninit -w 60

要确定服务器无法启动的原因，请检查联机日志。当您在脚本里使用 oninit -w 命令，可用 gstat - （打印输出头）命令检查该服务器是否处于联机模式。

允许 DBSA 群组用户执行 oninit 命令 (UNIX)

要允许属于 DBSA 群组的用户，而不是用户 gbasedbt 执行 oninit 命令，必须以用户 root 身份登录并更改 $GBASEDBTDIR/bin 目录下 oninit 实用程序的权限，将 6754 改为 6755 。

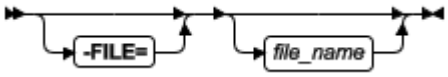
3.9.1 -FILE 选项

在 UNIX™，您可以使用 -FILE 选项运行 GBase 8s 中的某些实用程序 ，这些实用程序的本地环境变量已设置在 onconfig 文件中。并且在运行命令去启动该实用程序时，不需设置本地环境变量。

在启动以下实用程序时，您可以使用 -FILE 选项：oninit 、gcheck 、onclean 、gload 、gunload 、glogdump 、gadmin 、glogadmin 、gspaces 、gstat 和 gtape 。

语法

-FILE 选项



元素	用途	关键注意事项
-FILE= *file_name*	指定包含环境信息的 onconfig 文件的完整路径或相对路径	-FILE=*file_name* 选项必须是该命令的第一个参数

用法

在运行带有 -FILE 选项的命令之前，您必须在 onconfig 文件中按以下格式添加指令：

```
#$variable_name value
```

在 onconfig 文件中设置的环境变量优先于系统或 shell 中设置的相同的变量。

当启动带有 -FILE 选项的实用程序时，请指定 onconfig 文件的绝对路径或相对路径。例如：以下示例都启动了环境信息在 serv1 的 onconfig 中的数据库服务器。

绝对路径

```
oninit -FILE=/opt/gbs_server/data/conf/onconfig
```

相对路径

```
oninit -FILE=conf/onconfig
```

如果 GBASEDBTDIR 环境变量没有在用户的系统、shell 或 onconfig 文件中设置，GBASEDBTDIR 的值将会设置到该执行程序的路径下（如果该执行程序在的 GBASEDBTDIR 子目录下）。如果您使用远程执行的形式（例如：ssh）， 请使用 -FILE 选项指定 onconfig 文件在远程电脑上的路径。

示例

假设您在 onconfig 文件中为 js_3 实例指定了 DBDATE 和 SERVER_LOCALE 环境变量的值：

```
#onconfig for js_3
#
# *** Start environment settings for js_3
#
#$DBDATE MDY4/
#$SERVER_LOCALE en_us.utf8
#
```

```
# *** End environment settings for js_3
```

其他有关运行该实用程序的重要的环境变量（GBASEDBTDIR）在用户环境中被指定。 oninit 执行程序的路径是用户环境的一部分并且 onconfig 文件也在当前目录下。

可以从当前目录下运行 oninit -FILE=onconfig 命令去启动数据库服务器，它会自动设置 DBDATE 和 SERVER_LOCALE 环境变量的值。

3.9.2 oninit 实用程序的返回码

如果 oninit 命令发生错误，数据库服务器会返回错误消息及相应返回码值。

下表包含了 oninit 实用程序的返回码、消息内容及其用户操作。

返回码	消息内容	用户操作
0	数据库服务器初始化成功	数据库服务器启动
1	服务器初始化失败，查看写入 stderr 或联机日志的错误消息	根据写入stderr 或联机日志的错误消息，采取合适的操作
87	数据库服务器检测到违反安全或某些系统必备组件丢失或不正确	（仅 UNIX™ 上）检查用户和 gbasedbt 群组是否存在 。检查服务器配置文件 (onconfig) 和 sqlhosts 文件是否存在并有正确的权限。检查环境变量 GBASEDBTDIR、ONCONFIG 和 SQLHOSTS 的值是否有效并且其长度不能超过 255 字符数。检查环境变量 GBASEDBTDIR 是否指定了绝对路径并且没有空格、tab 、新行或其它不正确的字符。检查 $GBASEDBTDIR 目录下相关角色分离的子目录（例如：aaodir 和 dbssodir）是否有正确的所有权。运行 onsecurity 实用程序去诊断并修复这些问题。
170	数据库服务器初始化 dataskip 结构失败	释放系统上的物理内存并尝试重启数据库服务器
172	数据库服务器初始化监听器线程失败	释放一些系统资源，检查数据库服务器启动时要启动的监听器线程编号的配置参数值，并尝试再次重启数据库服务器
173	数据库服务器初始化数据复制失败	释放系统上的物理内存并尝试重启数据库服务器
174	数据库服务器启动快速恢复线程失败	释放系统上的物理内存并尝试重启数据库服务器
175	数据库服务器初始化 root dbspace 失败	检查服务器配置文件（onconfig）上的 root dbspace 的相关参数，并确保 root dbspace 的路径有效
176	共享磁盘辅助服务器初始化失败	检查 sqlhosts 文件中的条目（UNIX™）或 SQLHOSTS 注册键（Windows™），确保使用的关于主服务器的 dbdervername 配置值是正确的。检查服务器配置文件(onconfig)中 SDS_PAGING 配置参数的值是正确的。释放一些系统资源并尝试再次重启数据库服务器

返回码	消息内容	用户操作
177	数据库服务器启动 main_loop 线程失败	释放系统上的物理内存并尝试重启数据库服务器
178	数据库服务器初始化页转换所需的内存失败	释放系统上的物理内存并尝试重启数据库服务器
179	数据库服务器无法启动 CPU VPs	释放系统上的物理内存并尝试重启数据库服务器
180	数据库服务器无法启动 ADM VP	释放系统上的物理内存并尝试重启数据库服务器
181	数据库服务器初始化 kernel AIO 失败	释放系统上的物理内存并尝试重启数据库服务器
182	数据库服务器无法启动 IO VPs	释放系统上的物理内存并尝试重启数据库服务器
183	数据库服务器初始化异步 I/O 操作所需的内存失败	释放系统上的物理内存并尝试重启数据库服务器
184	数据库服务器初始化并行数据库查询（PDQ）所需的内存失败	释放系统上的物理内存并尝试重启数据库服务器
185	数据库服务器初始化各种 SQL 缓存失败	释放系统上的物理内存并尝试重启数据库服务器
186	数据库初始化全球语言支持（GLS）组件失败	释放系统上的物理内存并尝试重启数据库服务器
187	数据库服务器初始化关联服务设施（ASF）组件失败	检查 sqlhosts 文件上的条目
188	数据库服务器无法启动 CRYPTO VP	释放系统上的物理内存并尝试重启数据库服务器
189	数据库服务器初始化报警程序失败	释放系统上的物理内存并尝试重启数据库服务器
190	数据库服务器初始化审计组件失败	释放系统上的物理内存并尝试重启数据库服务器
192	数据库服务器重启 Window 工作站和桌面失败	（仅 Windows ） 在释放一些系统资源后尝试关闭数据库服务器
193	数据库服务器创建守护进程失败	（仅 UNIX ） 释放一些系统资源并尝试重启数据库服务器
194	数据库服务器重定向文件描述符失败	（仅 UNIX ）检查 /dev/null 设备的可用性并尝试重启数据库服务器
195	数据库服务器初始化当前使用的目录失败	检查数据库服务器初始化的当前工作目录的是否可用
196	数据库服务器初始化 /dev/null 设备失败	（仅 AIX®）检查 /dev/null 设备的可用性
197	数据库服务器查找要初始化该数据库服务器的用户的密码的信息失败	验证用户密码是否有效
198	数据库服务器设置资源限制失败	（仅 UNIX™ ）验证，如果必要，增加主机上限制进程的资源

返回码	消息内容	用户操作
200	数据库服务器在初始化的过程中没有足够的内存分配结构	释放系统上的物理内存并尝试重启数据库服务器
206	数据库服务器无法分配一个常驻分段	检查服务器配置文件（onconfig）上的 BUFFERPOOL 和 LOCKS 配置参数的值，确保它们可以装载在主机的可用内存上
207	数据库服务器初始化共享内存和磁盘空间失败	释放系统上的物理内存，检查该数据库服务器上所有 chunk 的有效性并且尝试再次重启数据库服务器
208	数据库服务器从共享内存分配结构失败	释放一些系统资源并尝试重启数据库服务器
209	数据库服务器在共享内存创建的过程中遭遇致命错误	释放系统上的物理内存并尝试重启数据库服务器
210	数据库服务器驻留的分段请求的内存量超出了最大允许范围	通过降低 BUFFERPOOL 和 LOCKS 配置参数的值，减少驻留分段的大小
220	数据库服务器无法读取审计配置文件	检查该神经配置文件（adtcfg）存在并有效
221	数据库服务器无法检测到 DUMPDIR 的缺省目录。它通常是 $GBASEDBTDIR/tmp 目录	创建 $GBASEDBTDIR/tmp 目录如果该目录不存在
224	数据库服务器检测到服务器配置文件中 HA_ALIAS 配置参数值错误	修正服务器配置文件(onconfig)中 HA_ALIAS 配置参数的值
226	数据库服务器无法在 sqlhosts 文件内的 DBSERVERNAME 配置参数中找到条目或者 sqlhosts 文件的内容无效	检查 sqlhosts 文件中的条目
227	错误的序列号	重新安装数据库服务器
228	用户缺少调用该执行所必需的 DBSA 权限	用户必须有 DBSA 权限或是 GBase 8s -Admin 群组成员（Windows）
229	数据库服务器无法初始化安全子系统	（仅 Windows ）用户在主机上没有必要的用户权限 或不是 GBase 8s -Admin 群组的成员
230	如果在 Windows 平台上数据库服务器作为一个进程启动，并在初始化过程中构建系统数据库超时（仅 Windows）	检查主机的事件日志确定如何打不开或无法启动的原因。数据库服务器可能在构建系统数据库时超时。释放一些系统资源并尝试重新启动数据库服务器
231	当 "oninit -w" 命令以命令行的进程运行时， GBase 8s 服务启动失败	（仅 Windows™）检查主机的事件日志，确定服务失败的原因
233	数据库服务器初始化插入式身份验证模块失败（PAM）	检查系统上的 PAM 库的配置
235	数据库服务器检测到服务器配置文件中现有配置参数有误	根据错误检查服务器配置文件（onconfig）

返回码	消息内容	用户操作
236	数据库服务器检测到其尝试限制 GBase 8s 使用版的允许值有误	检查服务器配置文件（onconfig）中 SDS_ENABLE 配置参数值为 1 。检查用 oninit -SDS 命令指定的服务器名称与 HA_ALIAS配置参数或 sqlhosts 文件中的 dbservername 条目相匹配。检查使用的共享磁盘是否为现有的共享磁盘群集的一部分
237	数据库服务器无法找到服务器配置文件	确保该服务器配置文件存在并有效
239	数据库服务器检测到 GBASEDBTDIR 环境变量的值不正确或不存在	（仅 Windows ） 检查 GBASEDBTDIR 环境变量的值
240	向数据库服务器发出不正确的命令行选项	启动时纠正向数据库服务器发出不正确的命令行选项
248	数据库服务器创建 GBase 8s 加载程序域文件失败	（仅 AIX）检查 /var/adm/ifx_loader_domain 文件是否存在
249	数据库服务器动态加载 PAM 库失败	数据库服务器的 PAM 库不可用。安装 PAM 库
250	数据库服务器动态加载 ELF 库失败	数据库服务器的 ELF 库不可用。安装 libelf 包
255	在服务器初始化过程中出现内部错误。查看写入 stderr 或联机消息日志的错误消息	根据写入 stderr 或联机消息日志的错误消息采取适当的操作

3.10 glogdump 实用程序

glogdump 实用程序显示逻辑日志文件（无论在磁盘上或备份上）的内容。

glogdump: 显示逻辑日志内容

当您想要跟踪特定事务或查看对特定 tblspace 进行了什么更改的调试情况中，glogdump 输出是有用的。（有关解释逻辑日志文件内容的信息，请参阅 解释逻辑日志记录 。）

任何用户都可以运行 -l 选项除外的所有 glogdump 选项。只有用户 gbasedbt（在 UNIX™ 上）或 Gbasedbt-Admin 群组的成员（在 Windows™ 上）可以运行 -l 选项。

如果当您执行 glogdump 时数据库服务器处于脱机模式，那么只读取磁盘上的文件。如果数据库服务器处于静态模式或联机模式，那么 glogdump 还读取存储在共享内存中逻辑日志缓冲区的逻辑日志记录（在读取磁盘上的所有记录之后）。

当数据库服务器在联机模式下从磁盘中读取具有状态 U 的逻辑日志文件时，数据库服务器拒绝对逻辑日志文件的所有访问，有效地停止所有会话的数据库活动。（有关更多信息，请参阅gstat -l 命令：打印物理和逻辑日志信息。）出于这个原因，建议您等到备份了文件之后再从备份中读取逻辑日志文件的内容。

glogdump 实用程序没有功能等同于 SQL 管理 API 命令 字符串。

glogdump 语法

元素	用途	关键注意事项
-q	不显示缺省情况下每 18 个记录出现一次的初始头和单行头	无
-V	显示软件版本号及序列号	请参阅 获取实用程序的版本信息
-version	显示构件版本、主机、操作系统、编号、日期及 GLS 版本	请参阅 获取实用程序的版本信息

当 glogdump 索引要显示的记录时，您指示它读取逻辑日志的以下部分：

- 存储在磁盘上的记录
- 存储在备份介质上的记录
- 所指定逻辑日志文件中的记录

缺省情况下，glogdump 显示逻辑日志记录头，它描述事务号和记录类型。记录类型标识所执行操作的类型。

除头之外，您可以使用读取过滤器指示 glogdump 显示以下信息:

- 逻辑日志记录头和数据（包含存储在 dbspace 或 tblspace 中的简单大对象的副本）
- blobspace 中 blobpage 的副本

它们是只从逻辑日志备份复制的，它们不可从磁盘上得到。

您可以显示每个逻辑日志记录头或可以基于以下条件指定输出：

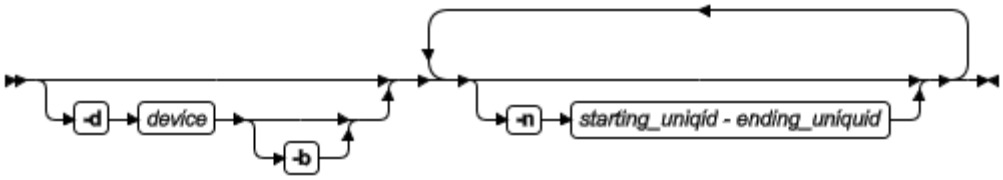
- 与特定表相关联的记录
- 特定用户启动的记录
- 与特定事务相关联的记录

如果 glogdump 在日志文件中检测到错误（例如无法识别日志类型），那么它以十六进制格式显示整个日志页并终止。

日志记录读取过滤器

glogdump 实用程序使用存储在 root dbspace 保留页中的路径名定位逻辑日志文件。如果使用 ON-Bar 备份逻辑日志，那么 glogdump 请求存储管理器从备份介质中检索想要的逻辑日志记录。

语法

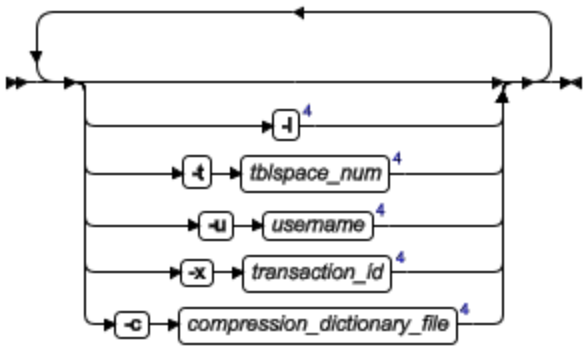


元素	用途	关键注意事项
-b	显示与 blobspace blobpage 相关联的逻辑日志记录	数据库服务器将这些记录作为 blobspace 日志记录的一部分存储在逻辑日志备份介质上
-d *device*	给出所需逻辑日志备份安装到的存储设备的路径名	如果使用 **gtape** ，那么所指定的设备必须与指定给配置参数 LTAPEDEV 的设备的路径名相同。如果未使用 **-d** 选项，那么 **glogdump** 读取存储在磁盘上的逻辑日志文件（以具有最低 *logid* 的逻辑日志文件开始）。 如果您使用 ON-Bar 来备份逻辑日志，请使用 **gbackuprestore -P** 命令来查看逻辑日志文件的内容。请参阅 *GBase 8s 备份与恢复指南* 。 有关路径名语法，请参阅操作系统文档

元素	用途	关键注意事项
-n *starting_uniqid-ending_uniqid*	指示 **glogdump** 读取包含在您指定的日志文件中从 *starting uniqid* 到 *ending uniqid* 的所有的逻辑日志记录	*starting_uniqid* 和 *ending_uniqid* 都是逻辑日志的唯一 ID 。要确定特定逻辑日志文件的 *uniqid* ，请使用 **gstat -l** 命令。 如果您没有使用 **-n** 选项，那么 **glogdump** 将读取所有可用的逻辑日志文件（在磁盘上或磁带上）。 有关 **gstat** 实用程序的信息，请参阅 监视数据库服务器状态

日志记录显示过滤器

语法



元素	用途	关键注意事项
-l	显示逻辑日志记录的长列表	日志记录的长列表包含整个日志记录的复合十六进制和 ASCII 转储。该列表不是供随意使用的。
-t *tblspace_num*	显示与所指定 tblspace 相关联的	无符号整数。编号（大于 0 ）必须在 **systables** 系统目录的 **partnum** 列中。

元素	用途	关键注意事项
	记录	将该值指定为整数或十六进制值（如果不使用 0x 前缀，那么该值作为整数解释）。要确定特定 tblspace 的 tblspace 编号，请查询 **systables** 系统目录表，它在 Tblspace 编号 中进行描述。
-u *username*	显示特定用户的记录	用户名必须是现有的登录名。用户名必须符合特定于操作系统的登录名规则
-x *transaction_id*	只显示与所指定事务相关联的记录	值必须是 0 和 TRANSACTIONS - 1（包括 0 和 TRANSACTIONS - 1）之间的无符号整数。 只在前滚过程中生成了错误的情况（不太可能发生的情况）下才能使用 **-x** 选项。当发生这种情况时，数据库服务器向消息日志发送消息，消息包含出错事务的事务 ID 。可以使用此事务 ID 和 **glogdump** 的 **-x** 选项调查错误原因
-c *compression_dictionary_file*	使用压缩字典来扩展压缩数据并显示未压缩的数据	如果 **glogdump** 命令包含 **-l** 选项和 **-c** 选项并且日志记录中有压缩图像，那么 **glogdump** 实用程序使用压缩字典扩展该日志记录中的所有可扩展的图像。 压缩图像只在压缩字典文件中具有有效压缩字典的日志记录的条件下是可扩展的。如果 **-c** 没有被指定或压缩字典文件没有一个该压缩图像的有效压缩字典，那么 **glogdump** 实用程序将显示其压缩格式中的行图像。

如果您没有压缩字典文件，可以使用 UNLOAD 命令加载该压缩字典给压缩字典文件，它将包含在 sysmaster 数据库的 syscompdicts_full 表中，示例如下：

```
UNLOAD TO 'compression_dictionary_file'
SELECT * FROM sysmaster:syscompdicts_full;
```

如果未指定任何选项，glogdump 显示日志记录中所有记录的简单列表。可以将这些选项与任何其他选项组合使用，以生成更有选择性的过滤器。例如：如果同时使用 -u 和 -x 选项，那么 glogdump 只显示在指定事务过程中指定用户启动的活动。如果同时使用 -u 和 -t 选项，那么 glogdump 只显示由指定用户启动并与指定 tblspace 相关联的活动。

3.11 gadmin 实用程序

使用 gadmin 实用程序变更数据库服务器操作方式，并在共享内容、会话、事务、参数和分段上执行其他各种操作。

这些主题展示了如何使用 gadmin 选项。如果不使用选项，那么数据库服务器返回用法语句。

在 UNIX™ 上，必须是用户 root 或用户 gbasedbt 才能执行该 gadmin 实用程序。

在 Windows™ 上，必须是 Gbasedbt-Admin 群组成员或管理员群组成员才能执行 gadmin 实用程序。

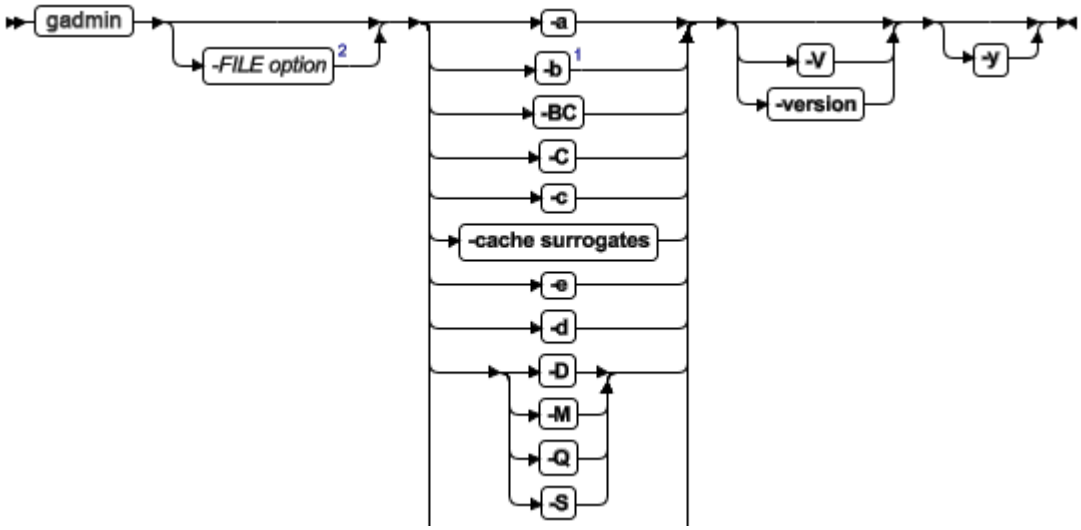
有关gadmin -b 命令的信息请参阅 GBase 8s 迁移指南 中的 gadmin -b 命令的语法 。

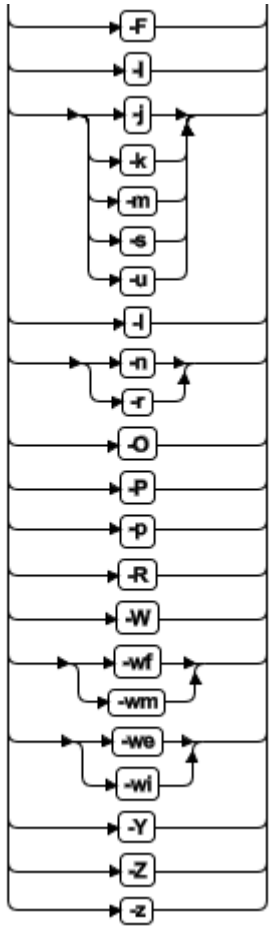
除了 gadmin -b 、gadmin -BC 和 gadmin -R.，所有的 gadmin 命令选项都有等同的 SQL 管理 API 命令 字符串。

3.11.1 gadmin 命令语法

使用 gadmin 实用程序命令执行各种数据库服务器操作。

下列语法图显示了您可使用 gadmin 命令的所有选项。语法图没有显示所有您使用的每个命令选项的元素。有关每个命令的完整的语法信息，请参阅这些命令的主题。





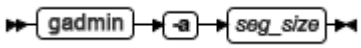
元素	用途	关键注意事项
-y	导致数据库服务器自动对所有提示响应“是”	无
-V	显示软件版本号和序列号	请参阅 获取实用程序的版本信息

元素	用途	关键注意事项
-version	显示了构件版本、主机、操作系统、数量、日期以及 GLS 版本	请参阅 获取实用程序的版本信息

有关用于还原的 gadmin -b 命令的更多信息，请参阅 GBase 8s 迁移指南 中的 gadmin -b 命令的语法 。

3.11.2 gadmin -a: 添加共享内存段

语法



元素	用途	关键注意事项
-a *seg_size*	允许添加新的虚拟共享内存段。以千字节为单位	**限制：** seg_size 的值必须是正整数。它不得超过操作系统对共享内存段大小的限制

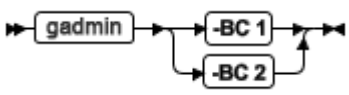
通常，您无需向共享内存的虚拟部分添加段，因为数据库服务器会在需要时自动添加段。然而，随着段的添加，数据库服务器可能在获得其需要的内存之前达到段最大数量的操作系统限制。这种情况通常在 SHMADD 配置参数设置得太小，以致数据库服务器在获得某些操作所需内存之前耗尽可用段的数量时发生。

如果手工添加的段大于 SHMADD 所指定的段，那么可以避免耗尽这些段的操作系统限制数，但仍满足数据库服务器对额外内存的需要。

该命令具有同等的 SQL 管理 API 命令。

3.11.3 gadmin -BC：允许大 chunk 方式

语法:



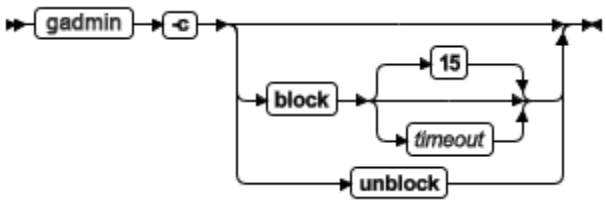
元素	用途	关键注意事项
-BC 1	启用大 chunk 、大于 2 GB 的偏移量并允许每个实例增大到	此选项允许创建大 chunk 。如果 chunk 没有均大于 2 GB ，那么可以在不删除的情况下进行转

元素	用途	关键注意事项
	32,768 个 chunk 。	换。没有大于 2 GB chunk 的 dbspace 和 blobspace 将保持旧格式。在向 dbspace 和 blobspace 添加了大于 2 GB 的 chunk 之后，随后在该dbspace 和 blobspace 中添加或更改的所有 chunk 都是新的格式。 请参阅 *GBase 8s 管理员指南*
-BC 2	对所有 dbspace 允许 “仅大 chunk ”方式	复原是不可能的。对所有 dbspace 或 blobspace 启用 9.4 大 chunk 功能。所添加或修改的任何 chunk 或偏移量都是新的格式。未更改的现有 chunk 保持旧格式。 请参阅 *GBase 8s 管理员指南*

注： 执行 gadmin -BC 命令之后，请执行一次完全的系统 0 级备份。

3.11.4 gadmin -c：强制 checkpoint

语法：



元素	用途	关键注意事项
-c	强制用于将缓冲区清仓到磁盘的 checkpoint	如果逻辑日志中的最新检查 checkpoint 记录正在阻止释放逻辑日志文件（状态 U-B-L），那么可以使用 -c 选项强制同步 checkpoint 。
block	阻塞数据库服务器运行任何事务	当数据库服务器阻塞后，用户可以以只读方式访问它。使用此选项在 GBase 8s 上执行外部备份。 有关更多信息，请参阅 *GBase 8s 备份与恢复指*

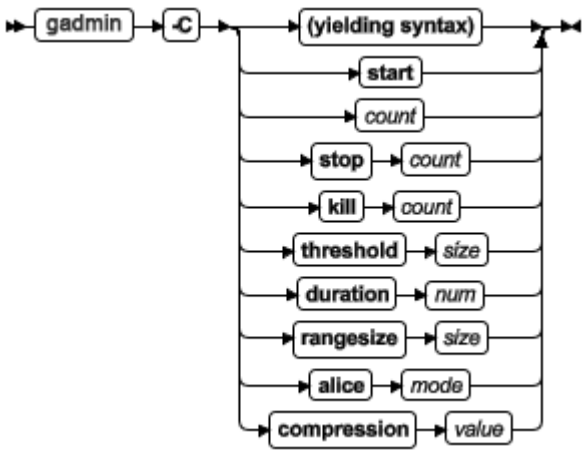
元素	用途	关键注意事项
		南
timeout	指示在返回命令提示符之前等待 checkpoint 清除的秒数	*timeout* 选项只有在配置了 DELAY_APPLY 配置参数后才能应用(请参阅 DELAY_APPLY 配置参数)。如果启用了 DELAY_APPLY 配置参数，那么主服务器请求的 checkpoint 可能在扩展的时间段内无法到达辅助服务器。也可能是没有其它 checkpoint 暂存在暂存目录中。缺省的 timeout 值为 15 秒，允许的最大 timout 值为 10 分钟（600 秒）。请参阅 *GBase 8s 备份与恢复指南*
unblock	不阻塞数据库服务器	当数据库服务器未阻塞时，可以继续数据事务和正常的数据库服务器操作。请在完成了 GBase 8s 上的外部备份之后使用此选项。 有关更多信息，请参阅 *GBase 8s 备份与恢复指南*

该命令具有等同的 SQL 管理 API 函数。

3.11.5 gadmin -C: 控制 B-tree 扫描程序

可以使用 gadmin -C 命令控制 B-tree 扫描程序并且指定关于 B-tree 扫描程序线程的信息。

语法:



元素	用途	关键注意事项

元素	用途	关键注意事项
-C	控制用于清除已删除项索引的 B-tree 扫描程序	对同时运行的线程的数量没有限制。然而，可同时启动的线程数限制为 128 个。 例如：如果您想要运行 150 个线程，那么要执行两条命令：gadmin -C 100 和 gadmin -C 50
start *count*	启动其他的 B-tree 扫描程序线程	如果没有指定 *count* ，那么假定*count* 1。对可以指定扫描程序线程数量没有限制。
stop *count*kill *count*	停止 B-tree 扫描程序线程	如果没有指定 *count* ，那么假定*count* 1 。 停止所有的所有扫描程序以阻止所有的所有被清除。 上述任一命令都可以停止 B-tree 扫描程序
threshold size *count*	设置在热列表上放置之前索引必须遇到的已删除项的最小数量	一旦所有超过阈值的索引都被清除并且 B-tree 扫描仪也没有别的工作，在该阈值之下的索引就会被添加到热列表
duration *num*	热列表处于有效状态的秒数	在秒数到期之后，热列表会由下一个可用的 B-tree 扫描程序重建，即使表上有未处理的项目。不会中断扫描程序当前正在处理的请求
rangesize *size*	在启用索引范围清除之前，确定索引的大小	大小为 -1 的值可用来禁用范围扫描
alice *num*	设置系统的 alice 方式	有效 *num* 值的范围是：0 (OFF) 到 12
compression *value*	对于数据库服务器实例，更改，合并两个部分使用的索引页的级别。如果这些页上的数据合计设定级别，那么页合并	级别的可用值为: low 、med （中等）、high 和 default 。系统的缺省值是 med

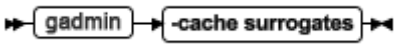
B-tree 扫描程序有追踪索引的有效性以及当前索引放在服务器上的额外工作量的统计信息。基于由删除的索引项引起的索引已完成的额外工作量，B-tree 扫描程序生成了称为“热表”

的整齐索引列表，这使得服务器做了额外的工作。将清除导致最高额外工作量的索引，然后以递减的方式清除剩下的索引。 DBA 可以动态地分配清除线程，因此允许可配置的工作负载。

该命令有等同的 SQL 管理 API 函数。

3.11.6 gadmin -cache 代理：缓存 allowed.surrogates 文件

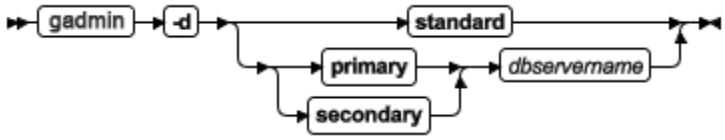
语法:



元素	用途	关键注意事项
-cache surrogates	读取 /etc/gbasedbt/allowed.surrogates 文件并将户 ID 和群组 ID 存储在共享内存缓存中 。 allowed.surrogates 文件中指定的用户名和群组名必须是有效的操作系统用户和群组，名称会转换成相应的 UID 和 GID	在会话加载 allowed.surrogates 文件时，可以使用 gadmin -cache surrogates。allowed.surrogates 文件用于指定可作为映射用户代理的用户和群组。在新的连接创建到数据库服务器之前或当创建或变更用户时，会自动检查 allowed.surrogates文件。 如果缓存刷新失败，清除现有代理缓存，即时禁用已映射的用户。服务器上现有的连接不会被共享内存缓存变化所影响。但共享内存的变化会影响新的会话

3.11.7 gadmin -d：设置数据复制类型

语法:



元素	用途	关键注意事项

元素	用途	关键注意事项
-d	用于设置服务器数据复制类型	可在数据库服务器处于静默、联机或只读方式时使用 -d standard 选项
dbservername	标识主或辅助服务器的数据库服务器名	*dbservername* 参数必须对应于目的主数据库服务器的数据库名。在共享内存重新初始化之后，数据复制对中的其他数据库服务器的名称和数据库服务器的类型（标准、主或辅助）将保留

使用 -d standard 选项

-d standard 选项删除数据复制对中数据库服务器之间的连接（如果存在一个连接）并将当前数据库服务器的数据库服务器类型设置为标准。此选项不更改对中其他数据库服务器的方式或类型。

只在从 RS 辅助服务器上断开主服务器的连接时使用 gadmin -d standard 命令。 运行该命令使 HDR 辅助服务器转变成一个独立的服务器。您不能运行 gadmin -d standard 来从 RS 辅助服务器上断开与主服务器的连接。从 RS 辅助服务器上断开主服务器的连接要运行以下命令：

在主服务器上：

```
gadmin -d delete RSS dbservername
```

在 RS 辅助服务器上：

```
gadmin -d standard
```

使用 the -d primary 选项

-d primary 选项将数据库服务器类型设置为主并尝试与 dbservername 指定的数据库服务器连接。如果连接成功，那么开启数据复制。主数据库服务器变成联机方式，而辅助数据库服务器变成只读方式。如果连接失败，那么数据库服务器变成联机方式，但不开启数据复制。

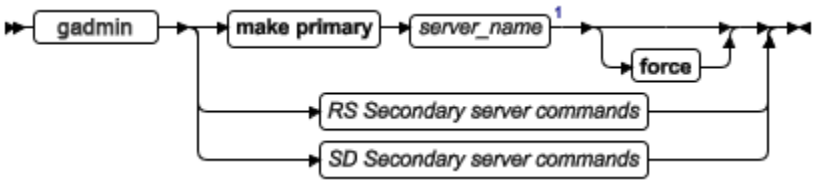
使用 the -d secondary 选项

-d secondary 选项将数据库服务器类型设置为辅助，并尝试与 dbservername 指定的数据库服务器连接。如果连接成功，那么开启数据复制。主数据库服务器变成联机方式，而辅助数据库服务器变成只读方式。如果连接失败，那么数据库服务器变成只读方式，但不开启数据复制。

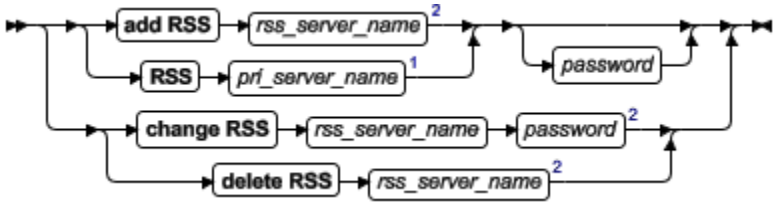
该命令有等同的 SQL 管理 API 函数。

3.11.8 gadmin -d: 设置高可用服务器的特性

语法:



RS 辅助服务器命令



SD 辅助服务器命令

元素	用途	关键注意事项
-d	用来在高可用性配置中创建、修改或删除辅助服务器	
add RSS	添加 RS 辅助服务器	该命令须在主数据库服务器上运行
rss_server_name	标识 RS 辅助服务器名称	*servername* 参数可以是 数据库服务器名称和 ER 群组名称。
password	指定辅助服务器密码	此密码只能在第一次尝试连接时使用。在主或辅助服务器建立连接后，密码将无法变更。
RSS	设置 RS 辅助服务器类型	该命令须在辅助数据库服务器上运行
pri_server_name	标识主服务器的名称	

元素	用途	关键注意事项
change RSS	变更 RS 辅助服务器	该命令须在主数据库服务器上运行
delete RSS	移除 RS 辅助服务器定义	该命令须在主数据库服务器上运行
set SDS primary	将服务器定义为共享磁盘主服务器	
server_name	数据库服务器名	当和 set SDS 或 make primary 一起使用时，它为角色变更的服务器的名称
force	用于强制变更	如果制定了 force 选项，那么无须请求将辅助服务器连接到当前服务器即可执行操作。如果没有指定 force 选项，那么必须根据当前主服务器调整操作。只能在 DBA 确定当前主服务器不在活动时使用 force 选项；否则会毁坏共享磁盘子系统
clear SDS primary	禁用共享磁盘环境。该服务器的名称不再作为 SD 主服务器	
make primary	创建主服务器	make primary 命令可用来声明辅助服务器的类型，包括 HDR 辅助服务器、RS 辅助服务器和 SD 辅助服务器。如果运行 make primary ： ● HDR Secondary：关闭当前主服务器并且将辅助服务器变为主服务器。 ● RS secondary：服务器变为标准服务器。 ● SD secondary：服务器成为新的主服务器

您也可以使用等同的SQL 管理 API 命令设置数据复制特征。有关更多信息，请参阅 SQL 管理 API 概述 和 GBase 8s 管理员指南。

有关其他 gadmin -d 的信息，请参阅 gadmin -d: 设置数据复制类型 和 gadmin -d 命令：使用数据复制来复制索引。

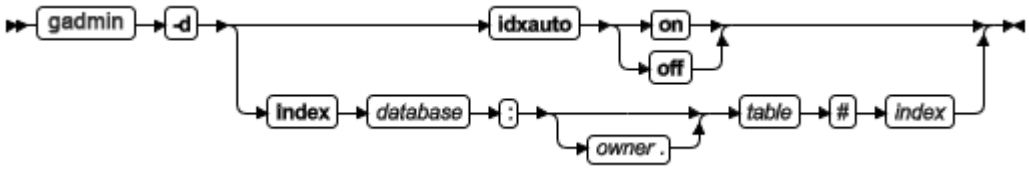
[1] 只能在辅助服务器上运行。

[2] 只能在主服务器上运行。

[3] 能在主服务器或辅助服务器上运行。

3.11.9 gadmin -d 命令：使用数据复制来复制索引

语法:



元素	用途	关键注意事项
-d	指定当辅助服务器上的索引毁坏时，如何将索引复制到 High-Availability Data-Replication (HDR) 辅助服务器	当服务器处于联机方式时,您可以使用 gadmin -d idxauto 和 gadmin -d index 命令
idxauto	当辅助服务器的索引毁坏时，启用自动索引复制	使用 gadmin -d idxauto 命令覆盖会话中 DRIDXAUTO 配置参数的值。 有关 DRIDXAUTO 的更多信息，请参阅 DRIDXAUTO 配置参数。有关复制索引的更多信息，请参阅 *GBase 8s 管理员指南* 中关于使用 HDR 一章
index	将索引从主服务器复制到辅助服务器	如果您检测到辅助服务器上索引毁坏，请使用 gadmin -d index 命令来启动将索引从主服务器复制到辅助服务器
database	指定包含要复制的索引的数	语法必须符合 Identifier 段;请参阅《*GBase*

元素	用途	关键注意事项
	据库	*8s SQL 指南：语法》*
index	指定要复制的索引的名称	指定的表和数据库中必须存在索引 语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*
owner	指定表的所有者	必须指定表的当前所有者。 语法必须符合 Table Name 段；请参阅 *《GBase 8s SQL 指南：语法》*
table	指定建立索引的表的名称	语法必须符合 Table Name 段；请参阅 *《GBase 8s SQL 指南：语法》*

gadmin -d idxauto 和 gadmin -d index 命令提供了将索引复制到包含损坏索引的辅助服务器的方法。在传送索引期间，基本表将被锁定。使用这些选项的另一种方法是在主服务器上删除并重建毁坏的索引。

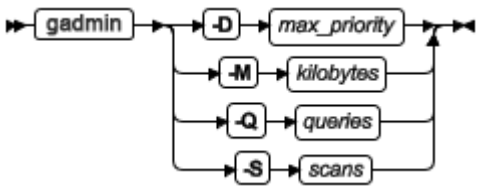
如果是分片索引的一个分片毁坏，那么 gadmin -d idxauto 命令将只传送单个受影响的分片，而 gadmin -d index 命令则传送整个索引。

该命令有等同的 SQL 管理 API 函数。 有关更多信息，请参阅 SQL 管理 API 概述 和 GBase 8s 管理员指南 。

有关 gadmin -d 命令的其他信息，请参阅 gadmin -d: 设置高可用服务器的特性 和 gadmin -d: 设置数据复制类型。

3.11.10 gadmin -D, -M, -Q, -S: 更改支持决定的参数

语法:



元素	用途	关键注意事项

元素	用途	关键注意事项
-D *max_priority*	更改 MAX_PDQPRIORITY 的值	该值必须是 0 到 100 之间的无符号整数。 指定 *max_priority* 作为调节用户对 PDQ 资源的请求的因素 有关用于控制 PDQ 的参数的信息，请参阅 MAX_PDQPRIORITY 配置参数 和 *GBase 8s 性能指南*
-M *kilobytes*	更改 DS_TOTAL_MEMORY 的值	该值的上限取决于平台。如果您输入一个大于您平台的值，那么会收到一条关于您的平台取值范围的消息。 将 *kilobytes* 指定为可用于并行查询的最大内存量。 有关更多信息，请参阅 DS_TOTAL_MEMORY 配置参数 和 *GBase 8s 性能指南*
-Q *queries*	更改 DS_MAX_QUERIES 的值	该值必须是 1 和 8,388,608 之间的无符号整数。 将 *queries* 指定为并发地执行的并行查询的最大数量。 有关用于控制 PDQ 的参数的信息，请参阅 DS_MAX_QUERIES 配置参数 和 *GBase 8s 性能指南*
-S *scans*	更改 DS_MAX_SCANS 的值	该值必须是 10 和 1,048,576 之间的无符号整数。 将 *scans* 指定为并发地执行的并行扫描的最大数量。 有关用于控制 PDQ 的参数的信息，请参阅

元素	用途	关键注意事项
		DS_MAX_SCANS 配置参数 和 *GBase 8s 性能指南*

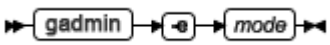
这些选项允许您在数据库服务器联机时更改配置参数。新值只影响数据库服务器的当前实例；这些值不记录在 ONCONFIG 文件中。如果关闭并重启数据库服务器，那么这些参数的值还原到 ONCONFIG 文件中的值，有关这些配置参数的更多信息，请参阅数据库配置参数。

要检查 MAX_PDQPRIORITY 、 DS_TOTAL_MEMORY 、DS_MAX_SCANS 、DS_MAX_QUERIES 和 DS_NONPDQ_QUERY_MEM 配置参数的当前值，请使用 gstat -g mgm 。请参阅gstat -g mgm 命令：打印 MGM 资源信息。

该命令有等同的 SQL 管理 API 函数。

3.11.11 gadmin -e: 更改 SQL 语句高速缓存的用途

语法:



元素	用途	关键注意事项
gadmin -e ENABLE	启用 SQL 语句高速缓存。关于更多信息，请参阅 *GBase 8s 性能指南* 中关于改进查询性能的资料	用户会话只在执行以下操作之一时才使用缓存： ● 将环境变量 STMT_CACHE 设置为 1 ● 执行 SQL 语句 SET STATEMENT CACHE ON
gadmin -e FLUSH	从 SQL 语句高速缓存中清仓不在使用的语句	gstat -g ssc ref_cnt 字段显示 0
gadmin -e OFF	关闭 SQL 语句高速缓存	不高速缓存任何语句
gadmin -e ON	开启 SQL 语句高速缓存	所有语句都是高速缓存的，除非用户使用以下操作之一关闭它： ● 将环境变量 STMT_CACHE 设置为 0

元素	用途	关键注意事项
		● 执行 SQL 语句 SET STATEMENT CACHE OFF

gadmin -e 变更只对当前数据库服务器会话生效。重新启动数据库服务器时，它使用 ONCONFIG 文件中的缺省的 STMT_CACHE 参数值。

该命令有等同的 SQL 管理 API 函数。

3.11.12 gadmin -F: 释放未使用的内存段

使用 gadmin -F 命令释放未使用的内存段，这些段不可用或进程已不再需要。

语法:

元素	用途	关键注意事项
-F	释放未使用的内存段	无

在执行 gadmin -F 时，内存管理器检查每个内存池的未使用内存。当内存管理器找到未使用内存块时，它立即释放该内存。在内存管理器检查了每个内存池之后，它开始检查内存段并释放数据库服务器不再需要的所有段。

建议在数据库服务器执行了任何创建额外内存段的功能（包括大索引构建、排序或备份）之后从操作系统调度工具定期运行 gadmin -F。

运行 gadmin -F 会造成执行该实用程序时处于活动状态的所有用户的显著性能降级。尽管执行时间很短（1 到 2 秒），但对于用户数据库服务器的降级可能达到 100%。具有多个 CPU 虚拟处理器的系统经历的降级成比例地减小。

要确认 gadmin 已释放了未使用内存，请检查消息日志。如果内存管理器释放了一个或多个段，那么它显示一条标识已释放了多少内存段和字节的消息。

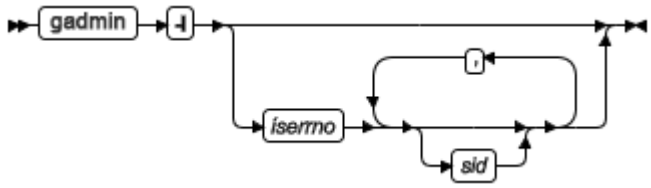
该命令有等同的 SQL 管理 API 函数。

3.11.13 gadmin -I: 控制诊断信息收集

使用 gadmin -I 选项开始和停止诊断信息的收集。

遇到错误时，可以指定 gadmin -I iserrno 选项启动收集诊断信息。也可以指定会话 ID 或收集仅指定的会话的信息。

使用 -I 选项（不需要附加其他参数）停止诊断信息的收集。

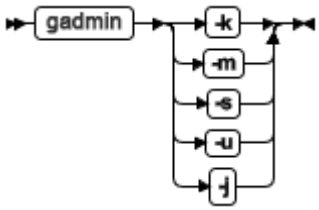


元素	用途	关键注意事项
iserrno	要收集诊断信息的错误的错误编号	无
sid	要收集诊断信息的会话的会话 ID	无

诊断信息收集程序在 GBase 8s 管理员指南 中描述。

3.11.14 gadmin -k, -m, -s, -u, -j: 更改数据库服务器方式

语法:



元素	用途	关键注意事项
-k	使数据库服务器处于脱机方式，并除去共享内存	要重新初始化共享内存，请关闭并重新启动数据库服务器 使用 -k 选项使数据库服务器处于脱机方式
-m	将数据库服务器从静默方式或管理方式更改为联机方式	请参阅 使用 -m 选项将数据库服务器处于联机方式
-s	以宽限方式关闭数据库服务器	在数据库服务器变成静默方式之前，允许正在使用数据库服务器的用户完成，但不允许新的连接，当所有处理完成时，-s 使数据库服务器成为静默方式。 -s 选项使共享内存保持不变。 请参阅 使用 -s 选项以宽限方式关闭数据库服务器
-u	立即关闭数据库服务器	此选项使数据库服务器处于静默方式，而不等待

元素	用途	关键注意事项
		用户完成其会话。它们的当前事务回滚，且它们的会话会终止。 请参阅 使用 -u 选项立即关闭数据库服务器
-j	使数据库服务器进入管理方式	该选项使得数据库服务器进入管理方式，允许 gbasedbt 用户执行所有功能，包括发出 SQL 和 DDL 命令。-j -U 选项使 DBSA 能够指定特定用户（以及 gbasedbt用户）来访问数据库服务器。 请参阅 *GBase 8s 管理员指南*

以下各节描述了使用数据库服务器从一种方式更改成另一种方式的选项。

使用 -k 选项使数据库服务器处于脱机方式

gadmin -k 选项使数据库服务器处于脱机方式，并除去数据库服务器共享内存。

请求确认的提示，在数据库服务器成为脱机方式之前，另一个提示请求对终止用户线程进行确认。如果想要消除这些提示，请执行 -y 选项和 -s 选项。

此选项不会终止所有客户机会话，请使用 -u 选项避免挂起客户机会话或虚拟服务器进程。

重要： 当使用 gadmin -k 命令关闭数据库服务器时，正在等待用户响应的实用程序可能不终止。例如：gtape 可能正在等待另一个磁带、gstat -i 可能正在等待用户响应或 gspaces 可能正在等待 y 或 n 以继续。如果发生这种问题，请使用 gadmin -uk 或 -uky ，而不是在除去共享内存之前回滚工作。有关更多信息，请参阅本页上其他选项的描述。

使用 -m 选项将数据库服务器处于联机方式

-m 选项将数据库服务器从静默方式转为联机方式。

使用 -s 选项以宽限方式关闭数据库服务器

-s 选项导致以宽限方式关闭。在数据库服务器变成静默方式之前，允许正在使用数据库服务器的用户完成，但不允许新的连接。当所有处理完成时，-s 使数据库服务器成为静默方式。-s 选项使共享内存保持不变。

请求确认的提示，如果想要消除此提示，请执行 -y 选项和 -s 选项。

使用 -u 选项立即关闭数据库服务器

-u 选项引起立即关闭，此选项使数据库服务器处于静默方式，而不等待用户完成其会话。它们的当前事务回滚，且它们的会话终止。

请求确认的提示。在数据库服务器变成静默方式之前，另一个提示请求对终止用户线程进行确认。如果想要消除这些提示，请执行 -y 选项和 -s 选项。

使用 -j 选项更改数据库服务器处于管理方式

-j 选项使数据库服务器进入管理方式，在该方式中，只允许 DBSA 群组和用户 gbasedbt 连接到服务器。-j 选项允许 DBSA 使服务器进入完全的功能方式来执行维护。

-j -U 选项使 DBSA 能够授权个别用户以管理方式访问数据库服务器。一旦连接，这些个别用户就能够执行任何 SQL 或 DDL 命令。当数据库更改为管理方式时，所有的用户会话（非 gbasedbt 用户会话）、DBSA 群组用户会话以及 gadmin -j -U 命令中标识的会话断开与数据库服务器的连接。

以下示例使三个用户连接到数据库服务器并一直持续到数据库服务器方式改为脱机、静默或联机方式：

```
gadmin -j -U karin,sarah,andrew
```

可以通过执行 gadmin -j -U 将他们的姓名从命令中的新名单中删除个别用户进行访问。例如，在下列命令中，第一个命令只授权 Karin 访问，第二个命令授权 Karin 和 Sarah 访问，第三个命令只授权 Sarah 访问（除去了 Karin 的访问）。

```
gadmin -j -U karin
gadmin -j -U karin,sarah
gadmin -j -U sarah
```

要允许用户 gbasedbt 和 DBSA 组用户在管理方式下访问数据库服务器并阻止所有单个用户访问数据库服务器，请使用以下命令

```
gadmin -j -U ''
```

关于使用配置参数在管理方式下指定用户的信息，请参阅 ADMIN_MODE_USERS 配置参数

3.11.15 gadmin -l：切换逻辑日志文件

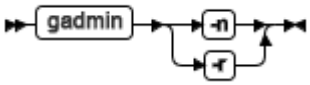
语法:

元素	用途	关键注意事项
-l	将当前逻辑日志文件切换为下一个逻辑日志文件	必须使用 gadmin 切换到下一个逻辑日志文件 有关切换到下一个逻辑日志文件的信息，请参阅 *GBase 8s 管理员指南* 中的管理逻辑日志文件一章

该命令有等同的 SQL 管理 API 函数。

3.11.16 gadmin -n, -r: 更改共享内存驻留

语法:



元素	用途	关键注意事项
-n	结束共享内存的常驻部分的强制驻留	此命令不影响 RESIDENT(ONCONFIG 文件中的强制驻留参数)
-r	开始共享内存的常驻部分的强制驻留	此命令不影响 RESIDENT(ONCONFIG 文件中的强制驻留参数)

重要： 在使用 gadmin -r 或 -n 选项之前，请将 RESIDENT 参数设置为 1 。

有关使用强制驻留参数在下一次重新启动数据库服务器时开启或关闭驻留的信息，请参阅 GBase 8s 管理员指南 中管理共享内存一章。

该命令有等同的 SQL 管理 API 函数。

3.11.17 gadmin -O: 重设 ONDBSPACEDOWN WAIT 方式

语法:

元素	用途	关键注意事项
-O	重设 ONDBSPACEDOWN 配置参数的 WAIT 方式	无

只能在以下情况中使用 gadmin -O 选项：

- ONDBSPACEDOWN 设置为 WAIT。
- 发生禁用 I/O 错误，导致数据库服务器阻塞所有正在更新的线程。
- 您无法或不想更正导致禁用 I/O 错误的问题。
- 您想要使数据库服务器将已禁用 dbspace 标记为关闭并继续进程。

当您执行此选项时，数据库服务器会将导致禁用 I/O 错误的 dbspace 标记为关闭，完成检查点，并释放已阻塞的线程。然后，gadmin 提示您一下消息：

```
This will render any dbspaces which have incurred disabling I/O errors unusable
        and require them to be restored from an archive.
```

```
        Do you wish to continue?(y/n)
```

当您运行 -O 选项时，如果 gadmin 在非临界 dbspace 上未找到任何禁用 I/O 错误，它将通知您以下消息：

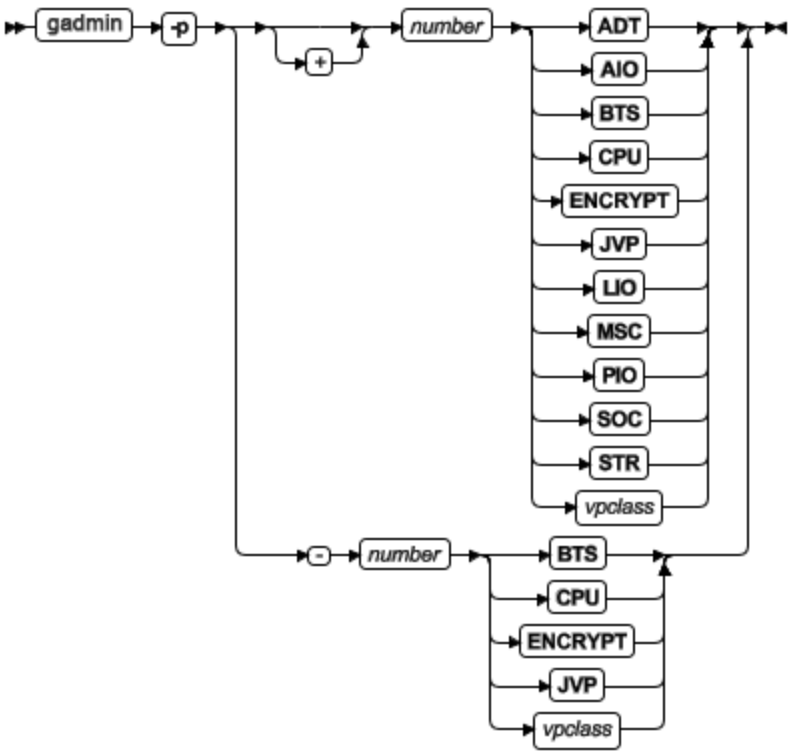
```
There have been no disabling I/O errors on any noncritical dbspaces.
```

该命令有等同的 SQL 管理 API 函数。

3.11.18 gadmin -p: 添加或删除虚拟管理器

可以使用 gadmin -p 命令动态地添加或移除数据库服务器实例的虚拟管理器。gadmin -p 命令不会修改 onconfig 文件，

语法:



元素	用途	关键注意事项
-p *number*	添加或移除虚拟处理器。*number* 参数指示要添加或删除的虚拟处理器的数量。 如果该值是负整数，那么删除处理器。如果该值时正整数，那么添加处理器。	仅当数据库服务器处于联机方式时才能使用 -p 选项，且一次只能添加一类虚拟处理器。 有关更多详细信息，请参阅 添加和删除虚拟处理器的规则。 如果正在删除虚拟处理器，那么最大数不能超过指定类型的实际处理器数量。如果正在添加

元素	用途	关键注意事项
		虚拟处理器，那么最大数依赖于操作系统。 有关更多信息，请参阅 *GBase 8s 管理员指南* 中的使用虚拟处理器一章。
ADT	运行审计处理过程	当通过设置 ONCONFIG 文件中的 ADTMODE 参数开启审计方式时，数据库服务器在审计类里开始虚拟处理器
AIO	对格式化的磁盘空间执行非日志记录磁盘 I/O	如果内核异步 I/O (KAIO) 未使用，那么还对原始磁盘空间执行非日志记录I/O
BTS	运行基本文本搜索索引操作和查询	BTS 虚拟处理器是非放弃的。如果想要同时运行多个基本文本搜索，那么指定多个 BTS 虚拟处理器。 使用 onconfig 文件中的 VPCLASS 参数创建至少一个 BTS 虚拟处理器。 更多有关基本文本搜索查询的信息，请参阅 *GBase 8s 数据库扩展用户指南*
CPU	运行所有会话线程和一些系统线程	建议 CPU VP 的数量不要大于物理处理器的数量。如果使用了 KAIO ，那么对原始磁盘空间执行 I/O ，包括对物理和逻辑日志的 I/O 。可用时运行 KAIO 线程或运行单个轮询线程。数据库服务器将 CPU VP 数用于为并行数据库查询 (PDQ) 分配资源。如果删除 CPU VP，查询运行将显著减慢。 **gstat -g mgm** 输出的 **Reinit**字段显示有关在 **gadmin -p** 命令之后等待正在运行的查询完成的查询数量的信息。另见 *GBase 8s 性能指南*
ENCRYPT	执行列级别的加密和解密例程	指定更多 ENCRYPT 虚拟处理器（如果您有多个加密的列）
JVP	在 Java™ 虚拟机（JVM）中	如果正在运行许多 Java UDR ，那么指定更多

元素	用途	关键注意事项
	执行 Java 用户定义的例程	JVP
LIO	如果在格式化的磁盘空间中，那么写入逻辑日志文件中	只有在逻辑日志位于镜像数 dbspace 中时才使用 2 个 LIO 虚拟处理器。数据库服务器最多允许 2 个 LIO 虚拟处理器
MSC	管理需要大量的堆栈空间的系统调用的请求	用于杂项内部任务
PIO	如果在格式化的磁盘空间中，那么写入物理日志文件中	只有在物理日志位于镜像数 dbspace 中时才使用 2 个 PIO 虚拟处理器。数据库服务器最多允许 2 个 PIO 虚拟处理器
SOC	使用 sockets 来执行网络通信	只能在数据库服务器通过 scokets 配置网络通信时，使用 SOC 虚拟处理器
STR	执行流管道连接	
vpclass	给出用户自定义的虚拟处理器类的名称	使用 onconfig 中的 VPCLASS 参数定义用户自定义虚拟处理器类。如果想要运行多个 UDR，那么可以指定多个用户自定义处理器。 在 Windows™ 上，同一时刻只能有一个用户自定义虚拟处理器类。省略 **gadmin -p** *vpclass* 命令中的 *number* 参数。 更多有关扩展类的信息，请参阅 VPCLASS 配置参数

添加和删除虚拟处理器的规则

可以添加和删除虚拟处理器。

请应用以下规则：

- 您不能删除最后的虚拟处理器。至少保留一台虚拟处理器。
- 您不能添加或删除 ADM 或 OPT。
- 仅限 Windows™ ：可以添加任意类的虚拟处理器，但不能删除虚拟处理器。

下列是可添加或删除的虚拟处理器：

虚拟处理器名称	添加	删除
ADT	Yes	No
AIO	Yes	No
BTS	Yes	Yes
CPU	Yes	Yes
ENCRYPT	Yes	Yes
JVP	Yes	Yes
LIO	Yes[1]	No
MSC	Yes	No
PIO	Yes[1]	No
SOC	Yes	No
STR	Yes	No
vpclass	Yes	Yes

表注：1. 您可以添加多个虚拟处理器。

使用 gstat 实用程序监视轮询线程

当数据库服务器联机时，不能删除正在运行轮询线程的 CPU 虚拟处理器。要确定在 CPU 虚拟处理器上运行的轮询线程，请使用以下命令：

```
gstat -g ath | grep 'cpu.*poll'
```

以下 gstat -g ath 输出显示了带有轮询线程的 2 个 CPU 虚拟处理。在这情况下，不能降到低于 2 个 CPU 虚拟处理器。

```
tid  tcb      rstcb  prty  status    vp-class  name
8    a362b90  0      2     running   1cpu      tlitcppoll
9    a36e8e0  0      2     cond wait arrived   3cpu
```

status 字段包含下列信息：running 、cond wait 、IO Idle 、IO Idle 、sleeping secs: number_of_seconds 或 sleeping forever。为了提高性能，可以移除或减少标识为 sleeping forever 线程的数量。

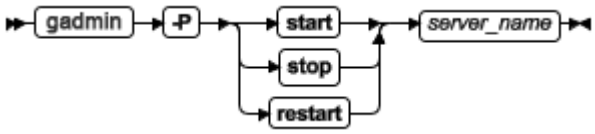
有关处理器类型的信息，请参阅 GBase 8s 管理员指南 中的虚拟处理器和线程一章。

该命令有等同的 SQL 管理 API 函数。

3.11.19 gadmin -P: 动态地启动、停止或重启监听线程

在不中断现有连接的情况下，可以使用 gadmin -P 命令动态地启动、停止或重启当前 SOCTCP 或 TLITCP 网络端口的监听线程。

语法:



元素	用途	关键注意事项
start	不中断现有连接的情况下，启动新的 SOCTCP 或 TLITCP 网络端口的监听线程	该监听线程的定义必须存在于此服务器的 sqlhosts 文件中。如果不存在，您必须在动态地启动监听线程之前添加它
stop	不中断现有连接的情况下，停止当前 SOCTCP 或 TLITCP 网络端口的监听线程	该监听线程的定义必须存在于此服务器的 sqlhosts 文件中
restart	不中断现有连接的情况下，重新启动当前 SOCTCP 或 TLITCP 网络端口的监听线程	该监听线程的定义必须存在于此服务器的 sqlhosts 文件中
server_name	要启动、停止或重启的数据库服务器名称	

这些命令不能修改 sqlhosts 文件。

这些命令有等同的带有 start listen 、stop listen 或 restart listen 参数的 SQL管理 API 函数。

示例

下列命令显示停止并重启名为 ids_serv1 的服务器的监听线程：

```
gadmin -P restart ids_serv1
```

3.11.20 gadmin -R: 重新生成 .infos.dbservername 文件

当您初始化共享内存或将数据库服务器处于脱机方式以移除文件时，数据库服务器创建 .infos.dbservername 文件。该文件在 $GBASEDBTDIR/etc 或 %GBASEDBTDIR%\etc 目录下。该文件的名称源于 sqlhosts 文件中的 dbservername 条目。

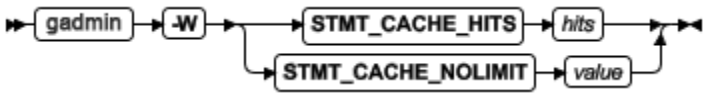
当数据库服务器访问实用程序时，它使用 .infos.dbservername 文件中的信息。 如果意外删除了，那么必须重建该文件或关闭并重启数据库服务器。

语法:

元素	用途	关键注意事项
-R	重新创建 .infos.dbservername 文件	无

3.11.21 gadmin -W: 更改 SQL 语句高速缓存的设置

语法:



元素	用途	关键注意事项
STMT_CACHE_HITS *hits*	指定语句完全插入到 SQL 语句高速缓存之前，对它的命中（引用）数。将 *hits* 设置为 1 或更大可防止特定查询进入高速缓存	仅可增加或重设STMT_CACHE_HITS 的值。新值显示在 gstat -g ssc 输出的 #hits 字段中。如果 *hits* = 0 ，数据库服务器在高速缓存中插入所有符合条件的语句及其内存结构。如果 *hits* > 0 并且 SQL 语句的已执行次数小于 STMT_CACHE_HITS ，那么数据库服务器会在高速缓存插入 *key-only* 条目。在已将语句执行了指定命中次数之后，它在高速缓存中插入符合条件的语句。ONCONFIG 参数：STMT_CACHE_HITS
STMT_CACHE_NOLIMIT *value*	控制语句是否插入 SQL 语句高速缓存	如果 *value* = 0，那么数据库服务器在高速缓存中插入语句。如果

元素	用途	关键注意事项
		value = 1，数据库服务器总是在高速缓存中插入语句。如果没有共享任何查询，那么关闭 STMT_CACHE_NOLIMIT 可防止数据库服务器为高速缓存分配大量内存。ONCONFIG 参数：STMT_CACHE_NOLIMIT

SQL 语句高速缓存示例

以下是用于更改 SQL 语句高速缓存（SSC）设置的 gadmin -W 命令的示例。更改只对当前数据库会话生效，且不能更改 ONCONFIG 值。重新启动数据库服务器时，它使用缺省的 SSC 设置（如果未在 ONCONFIG 文件中指定）或 ONCONFIG 设置。要做出永久更改，请设置相应的配置参数。

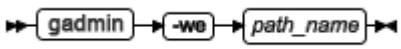
```
gadmin -W STMT_CACHE_HITS 2  # number of hits before statement is
          # inserted into SSC
gadmin -W STMT_CACHE_NOLIMIT 1  # always insert statements into
          # the cache
```

该命令有等同的 SQL 管理 API 函数。

3.11.22 gadmin -we: 导出包含当前配置参数的文件

可以使用 gadmin -we 命令创建并导出当前配置参数快照的配置文件。

语法:



元素	描述	关键注意事项
path_name	配置文件的完整或相对路径名	不能添加扩展

用法

gadmin -we 命令自动创建 ASCII 并给它命名您在命令中指定的名称。该文件的格式与 onconfig.std 文件格式相同。

如果您在当前会话中动态地变更值，导出的文件包含已更改的值而不是在 onconfig 文件中永久保存的值。

导出配置文件之后，可以导入它并作为配置文件使用。

如果运行 gadmin -we 命令并指定之前导出的文件，那么该命令导出该文件的最新版本，并重写之前导出的文件。

gadmin -we 命令与带有 gadmin 和 export 参数的 SQL 管理 API 命令具有相同的功能。

示例

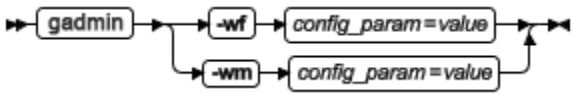
以下命令导出了 /tmp 目录下 onconfig3 文件中所有配置参数和它们当前值：

gadmin -we /tmp/onconfig3

3.11.23 gadmin -wf，-wm: 动态更改某个配置参数

使用 gadmin -wf 或 gadmin -wm 命令动态更改指定的配置参数。

语法:



元素	用途	关键注意事项
-wf	更新 onconfig 文件中指定配置参数的值	DBA 用户必须具有包含 onconfig文件的目录的写入权限
-wm	动态设置内存中指定配置参数的值	重启服务器时，指定的 *value* 不会预留
config_param = *value*	指定配置参数及其新值	请参阅 数据库配置参数

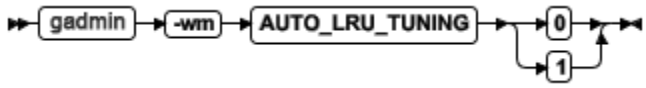
您可以动态使用 gadmin -wm 或 -wf command ，运行 gstat -g cfg tunable 命令查看配置参数列表。

gadmin -wf 和 gadmin -wm 命令有等同的 SQL 管理 API 函数。

gadmin -wm: 更改 LRU 调整状态

使用 gadmin -wm 选项更改 LRU 调整状态，不用更新 onconfig 文件。

语法:



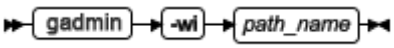
元素	用途	关键注意事项
-wm	为当前会话动态设置指定的配置参数的值	无
0	关闭当前会话的自动 LRU 调整	无
1	开启当前会话的自动 LRU 调整	无

该命令有等同的 SQL 管理 API 函数。

3.11.24 gadmin -wi: 导出配置参数文件

使用 gadmin -wi 命令导出包含多个配置参数的新值的文件。如果参数视可调整的，那么可以使用 gadmin -wm 命令单独地更新，数据库服务器将应用这些新值。

语法:



元素	用途	关键注意事项
path_name	上一次导出的配置文件的完整或相对路径	

用法

使用 gadmin -wi 导入配置文件比运行有关多个可调整配置参数的 gadmin -wm 命令更加快速和方便。

导入操作会忽略文件中不可调整的配置参数。该操作也会忽略与被当前实例使用所匹配的新的参数的值。

导出文件之后，您可以修改此导出配置参数的值。

导入操作仅会更改内存中配置参数的值。该操作不会影响 $GBASEDBTDIR/etc/$ONCONFIG 文件中的值。

gadmin -wi 命令等同于带有 wi 参数或 import 参数的 gadmin SQL 管理 API 命令。

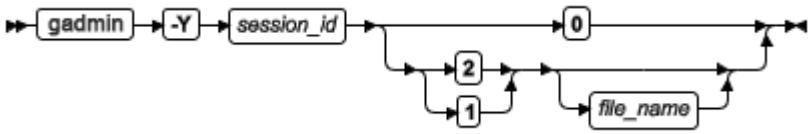
示例

以下命令导出了 /tmp 目录下名为 onconfig3 文件中的配置参数：

```
gadmin -wi /tmp/onconfig3
```

3.11.25 gadmin -Y: 动态更改 SET EXPLAIN

语法:



元素	用途	关键注意事项
file_name	解释输出文件的名称	如果不包含绝对路径，那么将在该样本输出文件的位置创建样本输出文件。如果存在该文件，解释输出会附加在其中。如果文件正用于 SET EXPLAIN 语句中，那么该文件不能使用直到动态解释关闭
session_id	指示特定会话	无
-Y	动态更改 SET EXPLAIN 语句的值	无

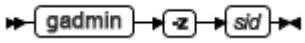
您可以使用 SET EXPLAIN 语句显示优化程序的查询计划、对返回行数的估计以及查询的相对开销。当使用 gadmin -Y 命令开启 SET EXPLAIN 时，输出显示在解释的输出文件中。对于单独的会话，gadmin -Y 命令动态更改 SET EXPLAIN 语句的值。使用该命令时，以下调用是有效的：

调用	解释
gadmin -Y *session_id* 2	开启对 *session_id* 的 SET EXPLAIN
gadmin -Y *session_id* 1	开启对 *session_id* 的 SET EXPLAIN 并且在解释输出文件中显示查询统计一节
gadmin -Y *session_id* 1 /tmp/myexplain.out	开启对 *session_id* 的 SET EXPLAIN 并将解释写入到 /tmp/myexplain.out 输出文件
gadmin -Y *session_id* 0	关闭对 *session_id* 的 SET EXPLAIN

该命令有等同的 SQL 管理 API 函数。

3.11.26 gadmin -z: 终止数据库服务器会话

语法:



元素	用途	关键注意事项
-z *sid*	终止 *sid* 中指定的会话	该值必须是大于 0 的无符号整数。且必须是当前正在运行的会话的会话标识号

要使用 -z 选项，请首先使用 gstat -u 获得会话标识（sessid），然后执行 gadmin -z ，以该会话标识号代替 sid 。

当使用 gadmin -z 时，数据库服务器尝试终止指定的会话。如果数据库服务器成功，那么它释放该会话占用的所有资源。如果数据库服务器无法释放这些资源，那么不终止该会话。如果会话不退出扇区或释放锁存器，数据库服务器管理员可以使数据库服务器脱机（如 使用 -k 选项使数据库服务器处于脱机方式 所述）以关闭所有会话。

该命令有等同的 SQL 管理 API 函数。

3.11.27 gadmin -Z: 终止分布式事务

语法:

元素	用途	关键注意事项
-Z *address*	终止与共享内存地址 *address* 相关联的分布式事务	此参数必须是已超出 TXTIMEOUT 所指定时间量的正在进行的分布式事务的地址。该地址必须符合特定于操作系统的对共享内存寻址的规则。（该地址可以从 gstat -x 输出中获得。） 直到超过 ONCONFIG 参数 TXTIMEOUT 所指定的时间量后，该选项才有效。-Z 选项应少使用，并只由分布式事务所涉的数据库服务器的管理员使用。 有关启动两阶段提交协议中的独立操作的信息，请参阅 *GBase 8s 管理员指南* 中的多阶段提交协议一章

分布式事务提供查询不同数据库服务器上的数据的能力。

注意： 如果应用程序正在执行分布式事务，那么终止分布式事务之一可使客户端/服务器数据库系统保持在不一致状态。请设法避免这种情况。

该命令有等同的 SQL 管理 API 函数。

3. 12 glogadmin 实用程序

可以使用 glogadmin 实用程序添加或删除逻辑日志文件、更改物理日志参数和添加新的缓冲池。

本章内容

本章说明如何使用以下 glogadmin 选项：

- glogadmin -a -d dbspace: 添加逻辑日志文件
- glogadmin -d -l lognum: 删除逻辑日志文件
- glogadmin -p: 更改物理日志参数
- glogadmin -b: 添加缓冲池

如果正在进行存储空间备份，那么任何 glogadmin 命令都会失败。如果不使用任何选项，那么 glogadmin 返回用法语句。

你不能在高可用数据复制（HDR）辅助服务器、远程独立（RS）辅助服务器或共享磁盘（SD）辅助服务器上使用 glogadmin 实用程序。

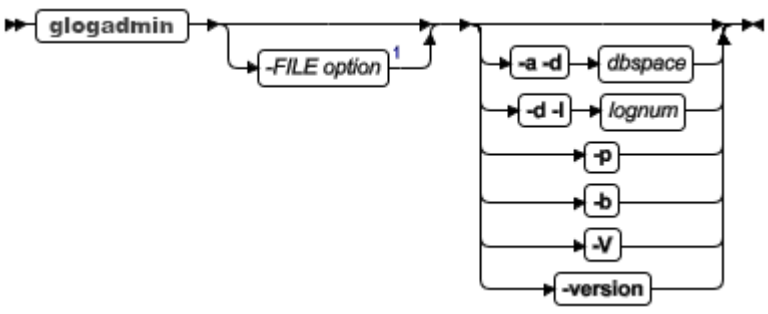
您也可以使用等同于 glogadmin 命令的 SQL 管理 API 命令添加或删除逻辑日志文件、更改物理日志参数和添加新的缓冲池。

在 UNIX™ 上，必须作为用户 root 或用户 gbasedbt 登陆才能执行 glogadmin 。只有用户 gbasedbt 才能执行 SQL 管理 API 命令字符串。

在 Windows™ 上，必须是 Gbasedbt-Admin 群组的成员才能执行 glogadmin。

3. 12. 1 glogadmin 语法

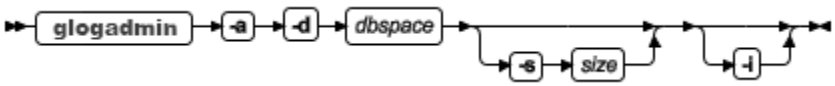
可以使用 glogadmin 实用程序修改逻辑日志或物理日志。



元素	用途	关键注意事项
-V	显示软件版本号及序列号	请参阅 获取实用程序的版本信息
-version	显示构件版本、主机、操作系统、编号、日期及 GLS 版本	请参阅 获取实用程序的版本信息

3.12.2 glogadmin -a -d dbspace: 添加逻辑日志文件

语法:



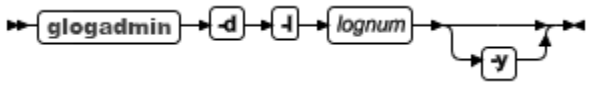
元素	用途	关键注意事项
-a -d *dbspace*	在所指定的 *dbspace* 的日志文件列表末尾添加逻辑日志文件	仅当数据库服务器具有足够的连续空间时，才可以向 dbspace 添加日志文件。新添加的日志文件具有状态 A ，并立即可以使用。可以在备份过程中添加日志文件。最多可以有 32,767 个逻辑日志文件。请使用 gstat -l 查看逻辑日志文件的状态。建议尽早对 root dbspace 和包含日志文件的 dbspace 采用 0 级备份。 不能向 blobspace 或 sbspace 添加日志文件。 语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*
-i	在当前日志文件的后面插入逻辑日志文件	在“需要日志文件”警告提示您添加逻辑日志文件时使用此项
-s *size*	指定新逻辑日志文件的大小（千字节）	该值必须是大于或等于 200 千字节的无符号整数。 如果您没有使用 -s 选项指定大小，那么数据库服务器磁盘空间初始化时，日志文件的大小将取自 ONCONFIG 文件中的 LOGSIZE 参数的值。 有关更改 LOGSIZE 的信息，请参阅 *GBase 8s 管理员*

元素	用途	关键注意事项
		指南 中的 管理逻辑日志文件 一章

该命令有等同的 SQL 管理 API 命令。

3.12.3 glogadmin -d -l lognum: 删除逻辑日志文件

语法:



元素	用途	关键注意事项
-d -l *lognum*	删除日志文件号所指定的逻辑日志文件	Restrictions: *lognum* 值必须是大于或等于 0 的无符号整数。 可以从 gstat -l 的 number 字段获得 *lognum*。*lognum* 的可能是无序的
-y	导致数据库服务器自动对所有提示响应“是”	无

用法

一次只能删除一个日志文件。

在所有时间，数据库服务器最少需要三个逻辑日志文件。如果数据库服务器配置有三个逻辑日志文件，那么您不能删除日志文件。

重要： 在删除任意前三个逻辑日志文件前，您必须添加新的逻辑日志文件并对逻辑日志文件进行备份。必须使用 gtape -a 命令或 gtape -c 命令执行备份。在添加新的逻辑日志文件并执行备份之后，您可以使用 glogadmin -d -llognum 删除前三个逻辑日志文件。

日志文件的状态取决于该日志文件是否被删除和日志文件被删除时数据库服务器采取的操作：

- 如果删除一个从未被写入的状态为|“新添加”（A）日志文件，数据库服务器删除日志文件并立即释放空间。

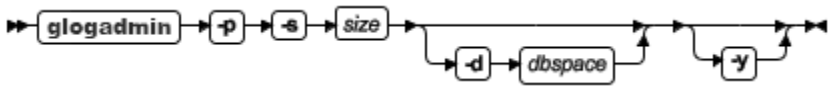
- 如果删除一个已使用的状态为“已使用”（U）或 “空闲”（F）的文件，那么数据库服务器将它标记为“已删除” （D）。并在对包含该日志文件的 dbspace 和 root dbspace 采用 0 级备份之后，数据库服务器删除此日志文件并释放空间。
- 您不能删除当前使用（C）的日志文件或包含最近检查点记录（L）的日志文件。

该命令有等同的 SQL 管理 API 命令。

将逻辑日志文件移动到另一个 dbspace 时，使用 glogadmin 命令添加或删除逻辑日志文件。请参阅 GBase 8s 管理员指南 中的 管理逻辑日志文件 一章中的移动逻辑日志文件。

3.12.4 glogadmin -p: 更改物理日志参数

语法:



元素	用途	关键注意事项
-p	更改物理日志	使用 glogadmin -p 命令时，必须包含 -s 参数。另外，您可以指定 -d 和 -y 参数。数据库服务器必须处于管理、联机或静默方式才能指定 -p 参数。数据库服务器无须重启，这些更改即可生效。
-s *size*	更改物理日志的大小（千字节）	该值必须是大于等于 200 千字节的无符号整数。 **注意：** 如果将日志移动到没有足够连续空间的 dbspace 中或将日志大小增至超出可用连续空间，那么操作将发生失败，并且物理日志不会更改。
-d *dbspace*	将物理日志的位置更改为指定的 *dbspace*	分配给物理日志的空间必须是连续的。 语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*
-y	导致数据库服务器自动对所有提示响应“是”	无

更改物理日志大小或位置之后进行备份

当您更改物理日志时，数据库服务器必须处于联机或静默方式。且无需重启数据库服务器，这些更改即可生效。

更改物理日志大小或位置后请立即创建 root dbspace 的 0 级备份。该备份对于数据库服务器的正确恢复具有至关重要的作用。

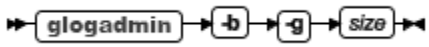
更改物理日志的大小和使用非缺省页大小

如果使用非缺省页大小，那么可能需要扩充物理日志的大小。如果要对非缺省页执行多次更新，那么您可能需要增加 150% 到 200% 物理日志大小。您可以根据物理日志触发器和检查点填充的频率调整物理日志大小。

3.12.5 glogadmin -b: 添加缓冲池

可以使用 glogadmin -b 命令创建与 dbspace 页大小相关联的缓冲池。

语法



元素	用途	关键注意事项
-b	创建缓冲池	可以在数据库服务器运行的同时添加缓冲池
-g *size*	指定要创建的缓冲区页的大小，以千字节	缓冲区页的大小必须在 2 到 16 KB 之间，并且必须是缺省页大小的倍数

您创建的缓冲池中的所有其他特性将会设置为 BUFFERPOOL 配置参数的缺省值行中的字段的值。

使用非缺省页大小创建的每个 dbspace 都必须具有相应的具有相应页大小的缓冲池。如果您创建的 dbspace 的页大小没有缓冲池，那么系统将自动使用 BUFFERPOOL 参数的缺省值行中的字段来创建缓冲池。

当您添加缓冲池时，新的 BUFFERPOOL 配置参数条目将会添加到 onconfig 文件中。

该命令有等同的 SQL 管理 API 命令。

3.12.6 glogadmin 命令的示例

以下是 glogadmin 命令的示例：

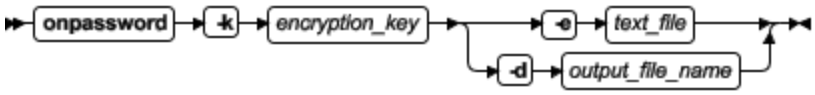
```
glogadmin -a -d rootdbs -s 1000  # adds a 1000-KB log file to rootdbs
glogadmin -a -d rootdbs -i       # inserts the log file after the current log
glogadmin -d -l 7                # drops log 7
glogadmin -p -d dbspace1 -s 3000 # resizes and moves physical-log to dbspace1
```

```
glogadmin -b -g 6 -n 3000 -r 2 -x 2.0 -m 1.0 # adds 3000 buffers of size
6K bytes each with 2 LRUS with maximum dirty of 2% and minimum dirty of 1%
```

3.13 onpassword 实用程序

使用 onpassword 实用程序加密和解密密码文件。连接管理器和 Enterprise Replication 实用程序通过非信任的网络连接数据库服务器时需要密码文件。

语法



元素	用途	关键注意事项
-k	指定秘钥	
-e	加密 ASCII 文本文件	该密码信息将加密到 $GBASEDBTDIR/etc/passwd_file
-d	解密所指定的加密文件	该 passwd_file 解密到 $GBASEDBTDIR/etc/output_file_name.
output_file_name	解密过程的输出文件名	在一类平台上创建的加密密码文件不支持别的类型的平台。在每类平台上，您必须运行 onpassword 实用程序并使用相同的文本文件和加密密钥
encryption_key	用于加密和解密密码信息的加密密钥	加密密钥可以是数字或字母的任何序列，且长度最多为 24 个字节。 要使用包含空格的加密密钥，请将该加密密钥附上引号。例如： `"my secret encryption key"`
text_file	包含用户密码信息的 ASCII 文本文件	在以下缺省位置使用 onpassword 实用程序： ● UNIX™: $GBASEDBTDIR/tmp ● Windows™: %GBASEDBTDIR%\etc

用法

只有作为用户 gbasedbt 登录，才有权限运行 onpassword 实用程序。

示例 1：加密密码文件

要使用 my_secret_encryption_key 加密 tmp/my_passwords.txt ，请运行以下命令：

```
onpassword -k my_secret_encryption_key -e my_passwords.txt
```

该密码信息已被加密到 $GBASEDBTDIR/etc/passwd_file 中。

示例 2：解密加密密码文件

要使用 my_secret_encryption_key 解密 $ GBASEDBTDIR /etc/passwd_file，请运行以下命令：

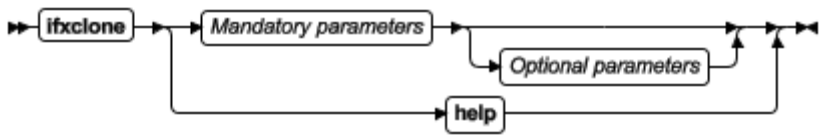
```
onpassword -k my_secret_encryption_key -d my_passwords.txt
```

该密码信息已被解密到 $GBASEDBTDIR/etc/my_passwords.txt 中。

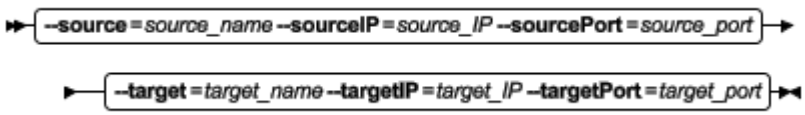
3.14 gclone 实用程序

可以使用 gclone 实用程序从当前的数据库服务器快照创建服务器。

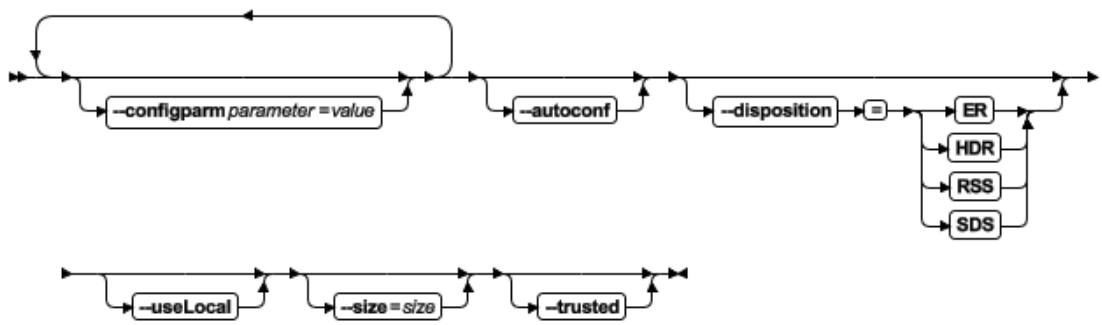
语法



Mandatory parameters



Optional parameters



元素	用途	关键注意事项
disposition	指定新服务器的最终规范	如果没有指定 --disposition (-d) 参数，那么将会创建一台标准服务器

元素	用途	关键注意事项
ER	指定新服务器实例创建为复制服务器	
HDR	指定新服务器实例创建成为 HDR 辅助服务器	
parameter=value	指定可选的配置参数并且将值设置到目标服务器	确定源服务器的配置参数必须与目标服务器相符。请参阅克隆 RS 辅助服务器的前提条件
RSS	指定新数据库实例创建为远程独立辅助服务器	
SDS	指定新数据库实例创建为共享磁盘辅助服务器	gclone 实用程序设置目标服务器的 SDS_PAGING 和 SDS_TEMPDBS 配置参数，但是全配置不在 gclone 实用程序范围之内。 如果命令中指定了 --disposition=SDS 但没指定 --useLocal ，您必须将 SD 辅助服务器的 ROOTPATH 配置参数值设置得与主服务器上 ROOTPATH 配置参数值相同
size	指定目标服务器的大小。有效的值有 tiny 、small 、medium 和 large	如果没有指定该大小参数，将会使用源实例的大小参数
source_name	指定源实例的名称	源服务器必须是主服务器并且不能是辅助服务器
source_IP	指定源服务器实例的 TCP/IP 地址	
source_port	指定源服务器实例的 TCP/IP 端口地址或与该端口关联的服务名称	

元素	用途	关键注意事项
target_name	指定目标服务器实例的名称	
target_IP	指定目标服务器实例的 TCP/IP 地址	
target_port	指定目标服务器实例的 TCP/IP 端口地址或与该端口关联的服务名称	

下表描述了有关 gclone 实用程序的选项。

长格式	短格式	含义
--autoconf	-a	新克隆服务器和高可用集群或 Enterprise Replicatio 域的其他服务器之间的自动配置连接。如果该选项用于创建复制服务器，那么 --autoconf 选项可以自动配置复制。 --autoconf 选项具有以下要求： ● 在目标服务器、源服务器和其他群集或复制服务器上，CDR_AUTO_DISCOVER 配置参数必须设置为 1。 ● REMOTE_SERVER_CFG 必须设置在所有群集或复制服务器上。 ● 源服务器的信任主机文件必须有目标服务器的主机信息 ● 如果使用 --disposition=ER 选项，并且主服务器为 Enterprise Replication 的一部分，那么在该域内的其他复制服务器必须是活动的
--configParm	-c	指定要设置到目标服务器上的配置参数的名称和值

长格式	短格式	含义
--disposition	-d	指定新服务器示例的规定
--help	-h	显示用法信息
--size	-s	指定目标实例的大小
--source	-S	指定源服务器实例的名称
--sourceIP	-I	指定源服务器实例的 TCP/IP 地址
--sourcePort	-P	指定源服务器实例的 TCP/IP 端口地址或与该端口相关联的服务名
--target	-t	指定目标服务器实例的名称
--targetIP	-i	指定目标服务器实例的 TCP/IP 地址
--targetPort	-p	指定目标服务器实例的 TCP/IP 端口地址或与该端口相关联的服务名
--trusted	-T	指定该服务器是可以信任的，并且在访问服务器时不需要获取用户 ID 和密码
--useLocal	-L	指定源服务器 onconfig 文件中的配置信息应与目标服务器 onconfig 文件合并。 确定源服务器的配置参数必须与其在目标服务器上的内容相一致。请参阅 所有服务器的前提条件

用法

可以使用 gclone 实用程序以最少的步骤或配置克隆一台服务器，或快速安装新节点到现有的 ER 复制域。当运行 gclone 实用程序时，从服务器节点获得的大部分安装信息会被克隆。要成功地克隆服务器可能仍需一些配置后步骤，以获得更好地运行的系统。

源服务器是包含希望被克隆日期的服务器。目标服务器是从源服务器中加载数据的服务器。您必须从目标服务器运行 gclone 实用程序。

要在 UNIX™ 上运行 gclone 实用程序，在目标服务器上您必须以用户 root 、gbasedbt 或 gbasedbt 群组成员的身份运行该命令。您也必须是源服务器的 DBSA。

要在 Windows™ 上运行 gclone 实用程序，在目标服务器上您必须以本地管理员群组成员的身份运行该命令。也必须是源服务器的 DBSA 并且在源服务器上您必须隶属于 Gbasedbt-Admin 群组。

gclone 实用程序使用来自源服务器的 onconfig 和 sqlhosts 配置文件配置目标服务器。

gclone 实用程序配置一些额外配置设置，不仅仅是配置克隆服务器的那些要求。--autoconf 选项提供额外配置 sqlhosts 文件记录的能力 ，之后会传播 sqlhosts 并对高可用集群服务器或 Enterprise Replication 域信任主机文件信息。Gclone 实用程序并不意味着要配置所有可能的配置选项，而是提供克隆源服务器所需的足够的配置。

可以使用大小参数来配置 CPU VP 和缓冲区的数量。 表 1 列出了创建在目标服务器上每一大小选项的 CPU VP 和缓冲池的数量。目标系统创建之后，应执行额外细化的生成配置。如果大小配置省略，则使用在源服务器上配置的参数。

表 1. 大小参数值列表

大小	CPU VP 数量	缓冲池数量
tiny	1	50,000
small	2	100,000
medium	4	250,000
large	8	500,000

可以使用 -c 选项在目标服务器上指定配置参数及其值。也可以使用已有的配置文件。如果目标服务器包含与源服务器配置文件不同的配置文件，gclone 实用程序不会重写该文件但是会修改那些克隆过程中必须与源服务器匹配的参数。

如果目标服务器位于与源服务器相同的主机上，那么需要使用 useLocal (-L) 参数。

如果指定了 useLocal 参数，gclone 实用程序会合并主服务器的 onconfig 文件和目标服务器的 onconfig 文件。所有服务器的前提条件 列出的配置参数会被 gclone 实用程序重写，其余的参数则不会受影响。

如果 useLocal 参数没有被指定为输入参数，gclone 实用程序使用源服务器的 onconfig 文件作为目标服务器的 onconfig 文件并且使用 gclone 实用程序输入的参数作为服务器的名称。

如果没有指定 useLocal 参数，gclone 实用程序使用目标服务器条目更新主机上的 sqlhosts 文件并且复制这些条目到目标服务器的 sqlhosts 文件中去。

以下是 gclone 参数选项的优先顺序：

- 在源服务器上 --configParm (-c) 参数优先于配置文件。

- --size (-s) 参数优先于合并的配置参数或本地配置文件中的设置。
- --configParm (-c) 参数优先于 --size (-s) 参数。
- 必须在每个服务器上相同的参数优先于所有其他选项。

所有服务器的前提条件

克隆服务器前请执行以下前提条件：

- 有关服务器硬件和软件的要求与与 HDR 辅助服务器的要求相同。（参考特定支持平台的machine notes）
- 源服务器和目标服务器必须在可信任的网络环境下。有关可信任环境的信息请参阅 Network security files 。
- 如果目标服务器的配置被指定为 ER 或 RSS ，那么在源服务器上您必须提供用户有连接到 sysadmin 数据库的权限。缺省情况下，仅允许用户 gbasedbt 连接 sysadmin 数据库。
- 同一时刻只能执行一次服务器克隆。直到第一次克隆完全结束才能开始第二次克隆。
- 源服务器必须将 onconfig 文件中的 ENABLE_SNAPSHOT_COPY 配置参数设置为 1 。
- 目标服务器必须没有旧 ROOTPATH 页。如果目标服务器有旧 ROOTPATH 页，那么创建一个零长度的 ROOTPATH 文件或将目标服务器的 onconfig 文件中的 FULL_DISK_INIT 配置参数设置为 1。

在克隆服务器的过程中不允许执行存档操作命令（例如：gtape 和 ON-Bar 命令）。请在开始克隆服务器之前执行存档活动。

在开始克隆服务器之前，必须在目标服务器上设置以下环境变量：

- GBASEDBTDIR
- ONCONFIG
- GBASEDBTSQLHOSTS
- GBASEDBTSERVER

必须在源和目标服务器上标识以下配置参数的值：

- DRAUTO
- DRINTERVAL
- DRTIMEOUT
- LOGBUFF
- LOGFILES
- LOGSIZE
- LTAPEBLK
- LTAPESIZE

- ROOTNAME
- ROOTSIZE
- PHYSBUFF
- PHYSFILE
- STACKSIZE
- TAPEBLK
- TAPESIZE

如果目标服务器中的 MIRROR 配置参数可用，那么以下配置参数必须在源和目标服务器之间相匹配：

- MIRRORPATH
- MIRROROFFSET

数据库服务器在源和目标服务器上允许只有某些组合 MIRROR 配置参数。请参阅 表 2.

表 2. 源和目标服务器上 MIRROR 配置参数允许的设置

设置于源服务器上的 MIRROR 配置参数	设置于目标服务器上的 MIRROR 配置参数	允许或不允许
No	No	允许
Yes	Yes	允许
Yes	No	允许
No	Yes	不允许。如果配置了该设置，那么服务器会在目标服务器的 onconfig 文件中提出警告并禁用 MIRROR 参数。

克隆 RS 辅助服务器的前提条件

1. 在目标服务器上设置一下环境变量：
   - GBASEDBTDIR
   - ONCONFIG
   - GBASEDBTSQLHOSTS
   - GBASEDBTSERVER
2. 在目标服务器上，创建存在于源服务器上所有的 chunk 和镜像 chunk 。如果目标服务器使用镜像，那么镜像 chunk 的路径必须与其在与服务器上的路径相匹配并且这些 chunk 必须存在。执行以下步骤来为目标服务器创建 chunk 和（如果必要）镜像 chunk ：
   a. 在源服务器上，运行 gstat -d 命令显示 chunk 和 镜像 chunk 的列表：

```
gstat -d
```

b. 在目标服务器上，以用户 gbasedbt 的身份登录并使用命令 touch 、chown 和 chmod 创建被 gstat -d 命令报告的 chunk 。例如，要创建名为 /usr/gbasedbt/chunks/rootdbs.chunk 的 chunk，请执行以下步骤：

```
$ su gbasedbt
Password:
$ touch /usr/gbasedbt/chunks/rootdbs.chunk
$ chown gbasedbt:gbasedbt /usr/gbasedbt/chunks/rootdbs.chunk
$ chmod 660 /usr/gbasedbt/chunks/rootdbs.chunk
```

c. 对 gstat -d 命令报告的每个 chunk 重复执行上一步中的所有命令。

3. 在目标服务器上，运行带适当参数的 gclone 实用程序。
4. 可以选择在目标服务器上创建 onconfig 和 sqlhosts 文件。

示例 1，使用源服务器配置克隆一台 RS 辅助服务器

该示例显示了如何使用来自源服务器上的 onconfig 和 sqlhosts 配置文件配置服务器。

在本示例中，忽略了 -L 选项引起的 gclone 实用程序从源服务器检索重要配置信息。该配置文件作为创建目标服务器配置的模板。使用 gclone 实用程序创建配置文件可以节省时间并减少其向配置文件中介绍错误的的机会。

对于本示例，假设源服务器 (Amsterdam)将 sqlhosts 文件配置如下：

```
#Server          Protocol        HostName          Service Group
Amsterdam     onsoctcp        192.168.0.1 123        -
```

还必须具有目标服务器的名称、IP 地址和端口号。本示例使用了以下信息：

- 源：服务器名称 Amsterdam
- 源 IP 地址：192.168.0.1
- 源端口： 123
- 目标服务器名称： Berlin
- 目标 IP 地址： 192.168.0.2
- 目标端口： 456

1. 在目标服务器上，以用户 gbasedbt 的身份登录并使用 touch 、chown 和 chmod 命令创建 chunk、更改其所有者并更改其许可权。Chunk 路径必须与 chunk 在源服务器上的路径匹配。
2. 在目标服务器上，运行 gclone实用程序：

```
gclone -T -S Amsterdam -I 192.168.0.1 -P 123 -t Berlin -i 192.168.0.2 -p 456 -d RSS
```

gclone 实用程序修改源服务器上的 sqlhosts 文件，并在新目标服务器上创建该文件的副本。目标服务器上的 sqlhosts 文件与源服务器上的相同：

```
#Server        Protocol    HostName          Service Group    DEFAULT
Amsterdam      onsoctcp    192.168.0.1 123       -            N
Berlin         onsoctcp    192.168.0.2 456                    N
```

示例 2，通过合并源服务器配置克隆 RS 辅助服务器

使用 -L (--useLocal) 选项在远程主机上创建服务器的克隆：-L 选项用于将源 onconfig 文件配置信息与目标 onconfig 文件合并。此选项还将把源 sqlhosts 文件复制到目标服务器。本示例使用以下信息：

1. 源服务器名称： Amsterdam
2. 源 IP 地址：192.168.0.1
3. 源端口： 123
4. 目标服务器名称： Berlin
5. 目标 IP 地址： 192.168.0.2
6. 目标端口： 456

1. 在目标计算机上创建 onconfig 和 sqlhosts 文件并设置环境变量。
2. 在目标服务器上，以用户 gbasedbt 的身份登录并使用 touch 、chown 和 chmod 命令创建 chunk、更改其所有者并更改其许可权。Chunk 路径必须与 chunk 在源服务器上的路径匹配。
3. 在目标服务器上 ，运行 gclone 实用程序：

```
gclone -T -L -S Amsterdam -I 192.168.0.1 -P 123 -t Berlin -i 192.168.0.2
-p 456 -d RSS
```

示例 3，向集群添加一台 RS 辅助服务器

该示例显示了如何向现有的 GBase 8s 高可用性集群添加 RS 辅助服务器。本示例使用以下信息：

- 源服务器名称： Amsterdam
- 源 IP 地址：192.168.0.1
- 源端口： 123
- 目标服务器名称： Berlin
- 目标 IP 地址： 192.168.0.2
- 目标端口： 456

1. 在目标计算机上创建 onconfig 和 sqlhosts 文件并设置环境变量。

2. 在目标服务器上，以用户 gbasedbt 的身份登录并使用 touch 、chown 和 chmod 命令创建 chunk、更改其所有者并更改其许可权。Chunk 路径必须与 chunk 在源服务器上的路径匹配。
3. 在目标服务器上 ，运行 gclone 实用程序：

```
gclone -T -L -S Amsterdam -I 192.168.0.1 -P 123 -t Berlin -i 192.168.0.2
-p 456 -s medium -d RSS
```

克隆 ER 服务器的前提条件

在尝试克隆 ER 服务器前请完成以下前提条件。

1. 源服务器（要被克隆的服务器）必须有 ER 配置并激活。
2. 对于指定目录名的配置参数，其目录名必须存在于目标服务器上。例如：如果在源服务器上要将 CDR_LOG_STAGING_DIR 配置参数设置为目录名，那么此目录也必须在目标服务器上存在。
3. 如果源服务器上的 ATS 或 RIS 可用，那么目标服务器上必须有适当的 ATS 或 RIS 目录。如果目录不存在那么 ATS/RIS spooling 将会失败。
4. 如果源服务器已设置了 CDR_SERIAL 配置参数，那么您必须在要克隆的服务器上将 CDR_SERIAL 的值设置成不同的值。在所有复制服务器上，CDR_SERIAL 的值必须都不相同。可以在 gclone 命令行中使用 --configParm (-c) 参数指定 CDR_SERIAL 配置参数的唯一值。
5. 新 ER 克隆上的时间必须适当同步。请参阅 Time Synchronization.
6. 源服务器（要被克隆的服务器）必须没有任何已停止或暂停的复制，也不能有任何影子复制定义。

在 gclone 实用程序运行过程中，避免执行更改目标服务器参与的复制集的 ER 管理任务。

示例：创建一台 ER 服务器克隆

假设有五台服务器：S1、S2、S3、S4 和 S5。它们在 ER 域中并配置为根服务器。您想要在名为 machine6 的电脑上添加新的服务器—— S6，并且希望它拥有和 S3 服务器一样的数据。

1. 在 machine6 上安装并配置 GBase 8s 数据库软件。您可以使用部署实用程序部署预配置的数据库服务器实例。
2. 将 S3 服务器上的 sqlhosts 文件辅导到 S6 服务器并修改它，为新服务器添加新条目。例如，假设新服务器的 ER 群组名为 g_S6 且ID 为 60，sqlhosts 文件将添加如下新行：

```
g_S6    group     -          -         i=60
S6      onsoctcp  machine6   service6  g=g_S6
```

3. 向其他五台服务器（ S1 到 S5）上的 sqlhosts文件中添加以上两行内容。

4. 将 S3 服务器上的 onconfig 文件复制到 S6 服务器。并将 DBSERVERNUM 配置参数更改为 S6 服务器的端口号。不要修改除路径信息以外的任意存储或 chunk 参数。
5. S6 服务器 (machine6) 提供 chunk 路径和其他存储到服务器 S3 大小相同。确保 S6 有足够的内存和磁盘空间资源。
6. 以用户 gbasedbt 的身份运行以下命令：

```
gclone -L -S S3 -I machine3 -P service3 -t S6 -i machine6 -p service6 -d ER
```

执行时，输入用户 gbasedbt 的用户名 gbasedbt 及其密码。

7. 监视 S6 和 S3 服务器的服务器日志。当克隆进程结束后，可以在服务器 S3 和 S6 上运行以下命令来检查服务器的状态：

```
cdr list server
```

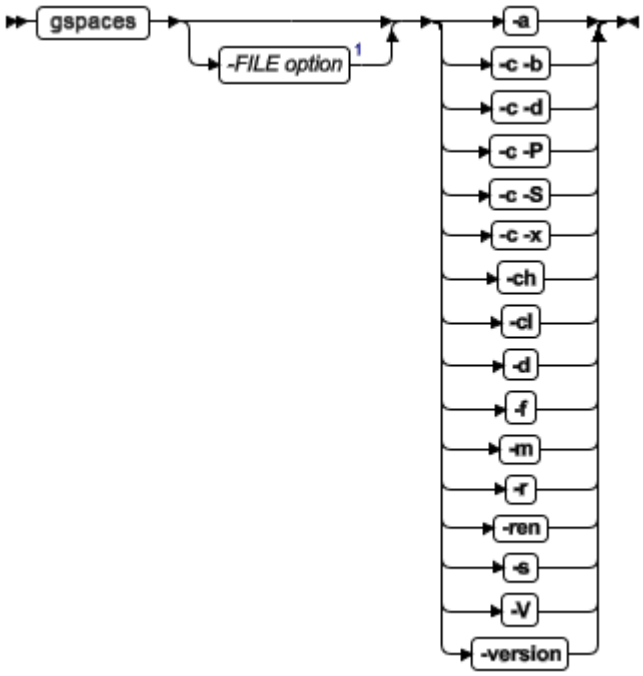
您可以看到新的 ER 服务器 g_S6 连接了所有其他五台服务器。此外，ER 节点 g_S6 将加入所有 g_S3 所加入的 ER 节点。

3.15 gspaces 实用程序

使用 gspaces 实用程序管理数据库的存储空间。

3.15.1 gspaces 语法

运行 gspaces 实用程序命令管理您的存储空间。

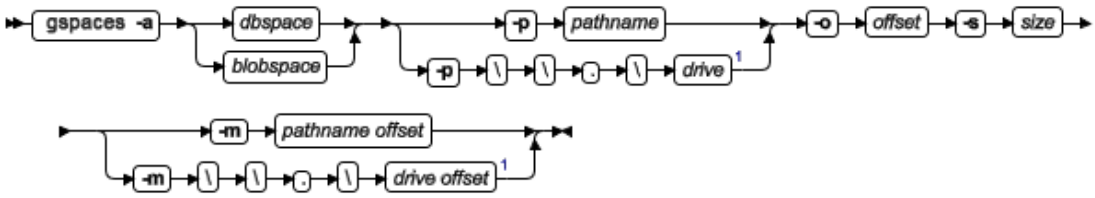


元素	用途	关键注意事项

元素	用途	关键注意事项
-V	显示软件版本号及序列号	请参阅获取实用程序的版本信息
-version	显示构件版本、主机、操作系统、编号、日期及 GLS 版本	请参阅获取实用程序的版本信息

3.15.2 gspaces -a: 向 dbspace 或 blobspace 添加 chunk

语法:



使用 gspaces -a 向 dbspace 或 blobspace 添加 chunk 。

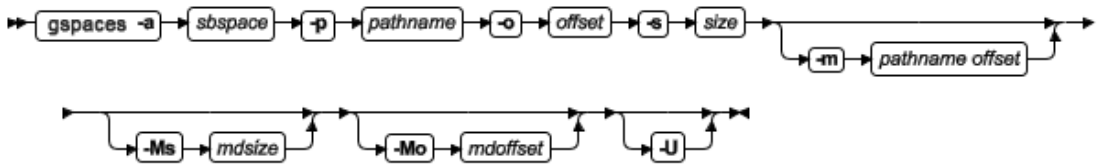
元素	用途	关键注意事项
-a	指示要添加 chunk	一个实例中至多包含 32766 个 chunk 。可以将所有的这些 chunk 放置到一个存储空间或分割在多个存储空间中
drive	指定要作为未缓冲磁盘空间分配的 Windows™ 驱动器。格式可以是 \\.*<drive>*（其中 *drive* 是指定给磁盘分区的驱动器盘符）或者 \\.*PhysicalDrive<number>*（ *PhysicalDrive* 是常量值， *number* 是物理驱动器编号）	例如： \\.\F: 有关路径名语法，请参阅操作系统文档
-m *pathname offset*	指定镜像新 chunk 的 chunk 的可选路径名和偏移量。另见此表中的 *pathname* 和 *offset* 条目	

元素	用途	关键注意事项
-o *offset*	在 -a 选项之后，*offset* 指示为到达新 blobspace 或 dbspace 的初始 chunk 所发生的磁盘分区或设备中的偏移量（千字节）	无符号整数。开始偏移量必须大于等于 0 。开始偏移量加 chunk 大小不能超过最大 chunk 大小，最大偏移量是 4 太字节。
-p *pathname*	指示正在添加的 blobspace 或 dbspace 的初始 chunk 的磁盘分区或为缓冲设备 Chunk 必须是现有的未缓冲设备或已缓冲文件	Chunk 名最多可以有 128 字节。指定路径名时，可以使用完整路径名或相对路径名。然而，如果使用相对路径名，这必须相对于是初始数据库服务器时的当前目录的目录。 UNIX™ 示例（未缓冲的设备）：/dev/rdsk/c0t3d0s4 UNIX 示例（已缓冲的设备）：/ix/ids9.2/db1chunk Windows 示例： c:\Ifmxdata\ol_icecream\mychunk1.dat 有关路径名语法，请参阅操作系统文档
-s *size*	指示新 blobspace 或 dbspace chunk 的大小（以千字节）	无符号整数。大小必须等于或大于 1000 字节，并且必须是页大小的倍数。开始偏移量加 chunk 大小不能超过最大 chunk 大小。最大偏移量是 4 太字节
blobspace	给出 chunk 将添加至的 blobspace 的名称	
dbspace	给出 chunk 将添加至的 dbspace 的名称	

该命令有等同的 SQL 管理 API 命令。

3.15.3 gspaces -a: 向 sbspace 添加 chunk

语法:



使用 gspaces -a 可向 sbspace 添加 chunk 。

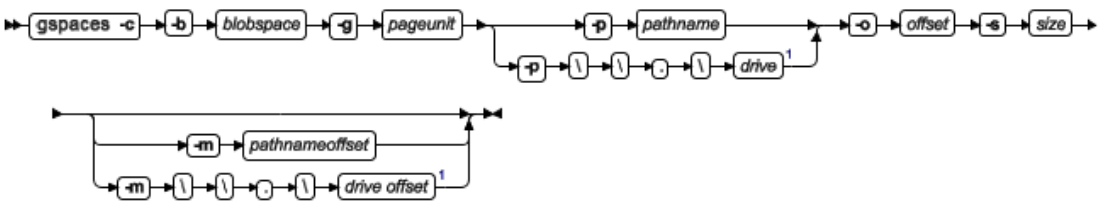
元素	用途	关键注意事项
-a	指示要添加的 chunk	一个实例中至多包含 32766 个 chunk 。可以将所有的这些 chunk 放置到一个存储空间或分割在多个存储空间中
-m *pathname offset*	指定镜像新 chunk 的 chunk 的可选路径名和偏移量。另见此表中的 *pathname* 和 *offset* 条目	有关背景信息，请参阅 *GBase 8s 管理员指南* 中的 管理磁盘空间 一章中的 向 sbspace 添加 chunk
-Mo *mdoffset*	指示应存储元数据的磁盘空间或设备中的偏移量（千字节）	值可以是 0 到 chunk 大小之间的整数。您不能指定导致元数据空间结束处超过 chunk 结束处的偏移量。 有关背景信息，请参阅 *GBase 8s 管理员指南* 中的 管理磁盘空间 一章中的 缩放 sbspace 元数据的大小
-Ms *mdsize*	指定在初始 chunk 中分配的元数据区域的大小（千字节）。剩下的是用户数据空间	值可以是 0 到 chunk 大小之间的整数。 有关背景信息，请参阅 *GBase 8s 管理员指南*中的 管理磁盘空间 一章中的 缩放 sbspace 元数据的大小
-o *offset*	在 -a 选项之后，*offset* 指示为到达新 blobspace 或 dbspace 的初始 chunk 所发生的磁盘分区或未缓冲设备中的偏移量（千字节）	无符号整数。开始偏移量必须大于等于 0 。开始偏移量加 chunk 大小不能超过最大 chunk 大小。最大偏移量为 2 或 4 千兆字节，这与平台有关。 有关更多消息，请参阅 *GBase 8s 管理员指南* 中的 管理磁盘空间 一章中的 在UNIX™ 上分配原始磁盘空间
-p *pathname*	指示正在创建的 sbspace 初始 chunk 的磁盘分区或	Chunk 名最多可以有 128 字节。指定路径名时，可以使用完整路径名或相对路径名。然

元素	用途	关键注意事项
	未缓冲设备。 Chunk 必须是现有的未缓冲设备或已缓冲文件	而，如果使用相对路径名，这必须相对于是初始化数据库服务器的当前目录。 有关路径名语法，请参阅操作系统文档
-U	指定应将整 chunk 用于存储用户数据	-M 和 -U 选项是互斥的。 有关背景信息，请参阅*GBase 8s 管理员指南*中的管理磁盘空间一章中的 向 sbspace 添加 chunk
-s *size*	指定新 sbspace chunk 的大小（千字节）	无符号整数。大小必须等于或大于 1000 字节，并且必须是页大小的倍数。开始偏移量加 chunk 大小不能超过最大 chunk 大小。 最大偏移量是 4 太字节
sbspace	给出 chunk 将添加至的 sbspace 的名称	请参阅*GBase 8s 管理员指南* 中的管理磁盘空间一章中的 向 sbspace 添加 chunk 。 语法必须符合 Identifier 段；请参阅*《GBase 8s SQL 指南：语法》*。

该命令有等同的 SQL 管理 API 命令。

3.15.4 gspaces -c -b: 创建 blobspace

语法:



可以使用 gspaces -c -b 创建 blobspace。

元素	用途	关键注意事项

元素	用途	关键注意事项
-b *blobspace*	给出要创建 blobspace 的名称	Blobspace 名称必须唯一，且不能超过 128 字节。它必须以字母或下划线开始，且必须只包含字母、数字、下划线或 $ 字符。 有关更多信息，请参阅 *GBase 8s 管理员指南* 中的 管理磁盘空间 一章中的 创建 blobspace 。语法必须符合 Identifier 段。有关更多信息，请参阅 *《GBase 8s SQL 指南：语法》*
-c	创建 dbspace 、blobspace 、sbspace 或 extspace 最多可以创建 2047 个任意类型的存储空间	创建存储空间之后，必须备份该存储空间和 root dbspace。如果所创建的存储空间名称与已删除存储空间的名称相同，那么执行另一个 0 级备份，以确保以后的恢复不混淆新存储空间和旧存储空间。 有关更多信息，请参阅 *GBase 8s 管理员指南* 中的 管理磁盘空间 一章中的 创建 dbspace 、blobspace 或 extspace
drive	指定要作为为缓冲磁盘空间分配的 Windows™ 驱动器。格式可以是 \\.*<drive>*（其中 *drive* 是指定给磁盘分区的驱动器盘符）或者 \\.*PhysicalDrive<number>*（ *PhysicalDrive* 是常量值，*number* 是物理驱动器编号）	有关分配未缓冲磁盘空间的信息，请参阅 *GBase 8s 管理员指南* 中的管理磁盘空间一章中的 在 Windows 上分配未缓冲磁盘空间。示例： ``` \\.\F: \\.\PhysicalDrive2 ``` 有关路径名语法，请参阅操作系统文档
-g *pageunit*	根据 *page_unit*（实例基本页大小的数量，2 K 或 4 K）指定 blobspace blobpage 的	无符号整数。值必须大于 0 。 Blobspace 最多能包含 2147483647 页。因此，blobspace 大小限制为 blobpage 大小

元素	用途	关键注意事项
	大小	x 2147483647。它包括组成 blobspace 的所有的 chunk 中的 blobpage。 有关更多信息，请参阅 *GBase 8s 性能指南* 中的 I/O 活动一章中的 blobpage 大小注意事项
-m *pathname offset*	指定 chunk 的可选路径名和偏移量，该 chunk 镜像新 blobspace 或 dbspace 的初始 chunk 另见此表中的 -p *pathname* 和 -o *offset* 条目	有关更多信息，请参阅 *GBase 8s 管理员指南* 中的 管理磁盘空间 一章中的 创建 dbspace 或 a blobspace
-o *offset*	指示为到达新 blobspace 、dbspace 或 sbspace 的初始 chunk 所发生的磁盘分区或设备中的偏移量（千字节）	无符号整数。无符号整数。开始偏移量必须大于等于 0 。开始偏移量加 chunk 大小不能超过最大 chunk 大小。最大偏移量为 2 或 4 太兆字节，这与平台有关。 有更多信息，请参阅 *GBase 8s 管理员指南* 中的管理磁盘空间 一章中的 分配原始磁盘空间
-p *pathname*	指示正在创建的 blobspace 或 dbspace 的初始 chunk 的磁盘分区或设备	Chunk 必须是现有的未缓冲设备或已缓冲文件。指定路径名时，可以使用完整路径名或相对路径名。然而，如果使用相对路径名，这必须相对于是初始化数据库服务器时的当前目录的目录。 UNIX™ 示例 （未缓冲的设备）： /dev/rdsk/c0t3d0s4 UNIX 示例（已缓冲的设备）

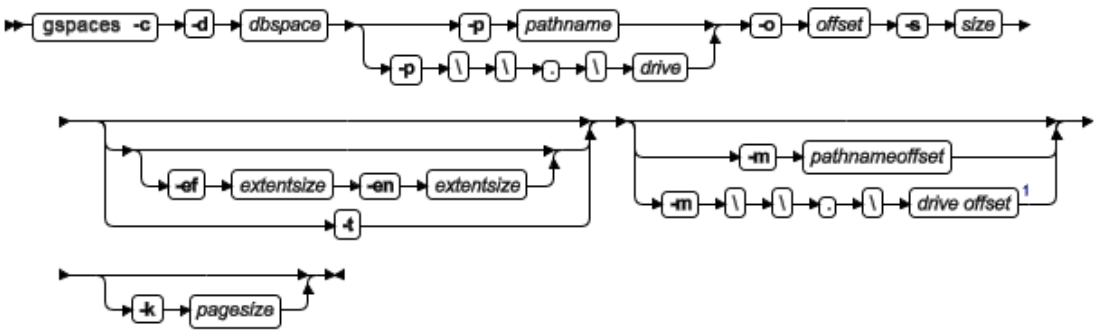
元素	用途	关键注意事项
		/ix/ids9.2/db1chunk Windows 示例： c:\Ifmxdata\ol_icecream\mychunk1.dat 有关路径名语法，请参阅操作系统文档
-s *size*	指定新 blobspace 或 dbspace 初始 chunk 的大小（千字节）	无符号整数。大小必须等于或大于 1000 千字节，并且必须是页大小的整倍数。开始偏移量加 chunk 大小不能超过最大 chunk 大小。 最大 chunk 大小为 2 或 4 太字节。这与平台有关。

该命令有等同的 SQL 管理 API 命令。

3.15.5 gspaces -c -d: 创建 dbspace

可以使用 gspaces -c -d 命令创建 dbspace 或临时 dbspace 。

语法



元素	用途	关键注意事项
-c	创建 dbspace 最多可以创建 2047 个任意类型的存储空间	创建存储空间之后，必须备份该存储空间和 root dbspace。如果所创建的存储空间名称与已删除存储空间的名称相同，那么执行另一个 0 级备份，以确保以后的恢复不混淆新存储空间和旧存储空间。

元素	用途	关键注意事项
		有关更多信息，请参管理数据库空间。
drive	指定要作为未缓冲磁盘空间分配的 Windows 驱动器。格式可以是 \\.*<drive>*（其中 *drive* 是指定给磁盘分区的驱动器盘符）或者 \\.*PhysicalDrive<number>*（ *PhysicalDrive* 是常量值，*number* 是物理驱动器编号）	示例： \\.\F: \\.\PhysicalDrive2 有关路径名语法，请参阅操作系统文档
-d *dbspace*	给出要创建的 dbspace 的名称	Dbspace 名称必须唯一，且不能超过 128 字节。它必须以字母或下划线开始，且必须只包含字母、数字、下划线或 $ 字符。 有关更多信息，请参管理数据库空间。语法必须符合 Identifier 段。有关更多信息，请参阅 标识符。
-ef *extentsize*	指定 tblspace **tblspace** 的第一个 extent 的大小(千字节)	非根 dbspace 的 tblspace **tblspace** 的第一个 extent 的最小和缺省大小等于 50 个 dbspace 页（以 KB 为单位）。例如，对于页大小为 2 KB 的dbspace，该大小为 100 KB ；对于页大小为 4 KB 的 dbspace，该大小为 200 KB ;对于页大小为 8 KB 的 dbspace，该大小为 400 KB。 Tblspace **tblspace** extent 的最大大小为 1048572 个页。在页面大小为 2 KB 的系统上，这将等于大约 2 GB 。 有关更多信息,请参阅 为表空间 tblspace 指

元素	用途	关键注意事项
		定第一个和下一个扩展数据块大小。
-en *extentsize*	指定 tblspace **tblspace** 的下一个 extent 的大小(千字节)	非根 dbspace 的 tblspace **tblspace** 的下一个 extent 的最小大小等于 4 个 dbspace 页（以 KB 为单位）。例如，对于页大小为 2 KB 的 dbspace，该大小为 8 KB ;对于页大小为 4 KB 的dbspace，该大小为 16 KB ;对于页大小为 8 KB 的 dbspace，该大小为 32 KB。 下一个 extent 的缺省大小是 50 个 dbspace 页。 Tblspace **tblspace** extent 的最大大小为 1048572 个页。在页面大小为 2 KB 的系统上，这将等于大约 2 GB 。 如果主要 chunk 中没有足够的空间分配给下一个 extent ，那么extent 将从另一个 chunk 分配。如果指定的空间不可用，那么将分配最接近的可用空间。 有关更多信息，请参阅 为表空间 tblspace 指定第一个和下一个扩展数据块大小。
-k *pagesize*	指示新的 dbspace 的非缺省的页大小（千字节） 对于具有充足的存储空间的系统，使用较大的页面大小具有以下性能优点： ● 可降低 B-tree 索引的深度，即使是对于较小的索引键 ● 您可以将当前跨越	页大小必须在 2 KB 和 16 KB 之间，并且必须是缺省页大小的倍数。例如：如果缺省页大小是 2 KB ，那么 *pagesize* 可以是 2 、4 、6 、8 、10 、12 、14 或 16 。如果缺省页大小是 4 KB (Windows) ，那么 *pagesize* 可以是 4 、8 、12 或 16 。 有关更多信息，请参阅 创建具有非缺省页大小的数据库空间。

元素	用途	关键注意事项
	多个页面(页面大小为缺省页面大小)的长行组合到同一页中 ● 检查点时间通常会随页面大小的增大而减少 ● 您可以为临时表定义不同的页面大小，这样它们将具有独立的缓冲池	
-m *pathname* *offset*	指定 chunk 的可选路径名和偏移量，该 chunk 对新的 dbspace 的初始 chunk 执行镜像操作 另见此表中的 -p *pathname* 和 -o *offset* 条目	有关更多信息，请参阅 管理数据库空间。
-o *offset*	指示为到达新的 dbspace 的初始 chunk 而发生的磁盘分区或设备中的偏移量（千字节）	无符号整数。无符号整数。开始偏移量必须大于等于 0 。开始偏移量加 chunk 大小不能超过最大 chunk 大小。 最大偏移量为 2 或 4 太兆字节，这与平台有关。
-p *pathname*	指示正在创建的 dbspace 的初始 chunk 的磁盘分区或设备	Chunk 必须是现有的未缓冲设备或已缓冲文件。指定路径名时，可以使用完整路径名或相对路径名。然而，如果使用相对路径名，这必须相对于是初始化数据库服务器时的当前目录的目录。 UNIX™ 示例 （未缓冲的设备）：

元素	用途	关键注意事项
		/dev/rdsk/c0t3d0s4 UNIX 示例（已缓冲的设备） /ix/ids9.2/db1chunk Windows 示例： c:\Ifmxdata\ol_icecream\mychunk1.dat 有关路径名语法，请参阅操作系统文档
-s *size*	指示新的 dbspace 初始 chunk 的大小（千字节）	无符号整数。大小必须等于或大于 1000 千字节，并且必须是页大小的整倍数。开始偏移量加 chunk 大小不能超过最大 chunk 大小。 最大 chunk 大小为 2 或 4 太字节。这与平台有关。
-t	创建临时 dbspace 以存储临时表	不能镜像临时 dbspace 。您不能为临时 dbspace 的 tblspace **tblspace** 指定第一个 extent 和下一个 extent 的大小。 有关更多信息，请参阅 临时数据库空间。

dbspace 的最大大小等于 chunk 的最大数量乘以 chunk 的最大大小。（每个实例中 chunk 的最大数量为 32766 。Chunk 最大大小等于 2147483647 页乘以页大小。）

该命令有等同的 SQL 管理 API 命令。

创建 dbspace 之后，您不能更改它的页大小。

在不是缺省的平台页大小的 dbspace 中，您不能存储逻辑日志或物理日志。

如果 dbspace 创建时该页大小的缓冲池不存在，那么 GBase 8s 会使用 BUFFERPOOL 参数的 default 行的字段值来创建一个缓冲池。您不能使多个缓冲池具有相同页大小。

临时 dbspace

当使用 gspaces 创建临时 dbspace 时，在您执行以下步骤之后，数据库服务器将使用新创建的临时 dbspace ：

将新的临时 dbspace 的名称添加到临时 dbspace 的列表中（在 DBSPACETEMP 配置参数和/或 DBSPACETEMP 环境变量中）。

重新启动数据库服务器。

您不能指定临时 dbspace 的第一个 extent 和下一个 extent 。临时 dbspace 的extent 大小是 100 KB （2 KB 页的系统) 或者 200 KB（ 4 KB 页的系统）。

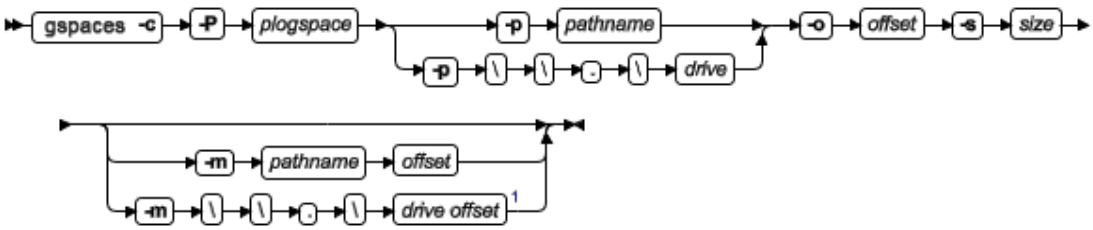
要指定 root dbspace 中 tblspace tblspace 第一个 extent 和下一个 extent 的大小，请在第一次启动数据库服务器时，在创建 root dbspace 之前使用 TBLTBLFIRST 和 TBLTBLNEXT 配置参数。

[1] 仅限于 Windows™

3.15.6 gspaces -c -P: 创建 plogspace

使用 gspaces -c -P 命令创建 plogspace 以存储物理日志。

语法



元素	用途	关键注意事项
-c	创建 plogspace	一个实例只能有一个 plogspace 。如果 plogspace 存在，那么创建新 plogspace 将物理日志移动到此新的空间中，并删除旧的 plogspace 。
-m *pathname offset*	指定 chunk 的可选路径名和偏移量，并对新的 plogspace 的初始 chunk 执行镜像操作 另见此表中的 -p *pathname* 和 -o *offset* 条目	如果镜像了该 plogspace ，那么 plogspace chunk 不能被扩展
-m \\.*drive*	指定 chunk 的 Windows™ 驱动器并镜像该新	示例：

元素	用途	关键注意事项
	plogspace 的 chunk *drive* 是指定给磁盘分区的驱动器盘符或常量值，它是物理驱动器编号	\\.\F: \\.\PhysicalDrive2 有关驱动器名语法，请参阅操作系统文档
-o *offset*	指示为到达新的 plogspace 的初始 chunk 而发生的磁盘分区或设备中的偏移量（千字节）	无符号整数。无符号整数。开始偏移量必须大于等于 0 。开始偏移量加 chunk 大小不能超过最大 chunk 大小。 偏移量必须是页大小的倍数。最大偏移量为 2 或 4 TB，这与平台有关。
-P *plogspace*	给出要创建的 plogspace 的名称	Plogspace 名称必须唯一，且不能超过 128 字节。它必须以字母或下划线开始，且必须只包含字母、数字、下划线或 **$** 字符。 语法必须符合 Identifier 段。有关更多信息，请参阅 标识符。
-p *pathname*	指示正在创建的 plogspace 的初始 chunk 的磁盘分区或设备	Chunk 必须是现有的未缓冲设备或已缓冲文件。指定路径名时，可以使用完整路径名或相对路径名。然而，如果使用相对路径名，这必须相对于是初始化数据库服务器时的当前目录的目录。 UNIX™ 示例（未缓冲的设备）： /dev/rdsk/c0t3d0s4 UNIX 示例（已缓冲的设备）： /ix/ifmx/db1chunk Windows 示例：

元素	用途	关键注意事项
		c:\Ifmxdata\ol_icecream\mychunk1.dat

-p \\.*drive*	指定要给 plogspace 分配未缓冲磁盘空间的 Windows 驱动器。
	
	drive 是指定给磁盘分区的驱动器盘符或常量值，它是物理驱动器编号
		示例：
		`\\.\F:`
		`\\.\PhysicalDrive2`
		有关驱动器名语法，请参阅操作系统文档

-s *size*	指示新的 plogspace chunk 的大小（千字节）	无符号整数。大小必须等于或大于 1000 千字节，并且必须是页大小的整倍数。开始偏移量加 chunk 大小不能超过最大 chunk 大小。 最大 chunk 大小为 2 或 4 TB。这与平台有关。

物理日志必须存储在单独的 chunk 中。缺省情况下，plogspace 的 chunk 是可扩展的，并且数据库服务器会扩展 plogspace 以提高性能。

示例

以下示例创建了一个名为 plogdbs 的 plogspace，它有 40000 KB 大小且偏移量为 0 ：

```
gspaces -c -P plogdbs -p /dev/chk1 -o 0 -s 40000
```

以下示例创建了一个已镜像的 plogspace，它的名称为 pdbs1 ，大小为 60000 KB ，偏移量为 500 KB：

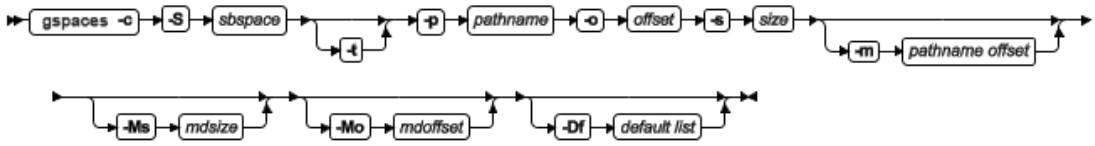
```
gspaces -c -P pdbs1 -p /dev/pchk1 -o 500 -s 60000 -m /dev/mchk1 0
```

[1] 仅限于 Windows™

3.15.7 gspaces -c -S: 创建 sbspace

可以使用 gspaces -c -S 选项创建 sbspace 或临时 sbspace 。

语法：



元素	用途	关键注意事项
-S *sbspace*	给出要创建的 sbspace 名	Sbspace 名称必须唯一，且不能超过 128 字节。它必须以字母或下划线开始，且必须只包含字母、数字、下划线或 $ 字符。 语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*
- c	创建 sbspace 最多可创建 32767 个任意类型的存储空间	无
-m *pathname offset*	指定到镜像新 sbspace 的初始 chunk 的 chunk 的可选路径名和偏移量。另见此表中的 -p *pathname* 和 -o *offset* 条目	有关更多信息，请参阅 *GBase 8s 管理员指南* 中的 数据存储一章中的 sbspace 以及管理磁盘空间 一章中的 创建sbspace
-Mo *mdoffset*	指示将存储元数据的磁盘分区或设备中的偏移量（千字节）	**限制：**值可以是 0 到 chunk 大小之间的整数。您不能指定导致元数据空间结束处超过 chunk 结束处的偏移量。 **参考：**有关更多信息，请参阅*GBase 8s 管理员指南* 中的 管理磁盘空间 一章中的 缩放 sbspace 元数据大小
-Ms *mdsize*	指定初始 chunk 中分配的元数据区域的大小（千字节） 剩下的是用户数据空间	**限制：**值可以是 0 到 chunk 大小之间的整数

元素	用途	关键注意事项
-o *offset*	指示为到达新的 sbspace 的初始 chunk 而发生的磁盘分区或设备中的偏移量（千字节）	**限制：**无符号整数。开始偏移量必须大于等于 0 。开始偏移量加 chunk 大小不能超过最大 chunk 大小。对具有 2 千字节页大小的系统，最大 chunk 大小是 4 太字节 ，对具有 4 千字节页大小的系统，最大 chunk 大小是 8 太字节 。 **参考：**有关更多信息，请参阅 *GBase 8s 管理员指南*中的 管理磁盘空间 一章中的 在 UNIX™ 上分配原始磁盘空间
-p *pathname*	指示 sbspace 初始 chunk 的磁盘分区或未缓冲设备	Chunk 必须是现有的未缓冲设备或已缓冲文件。指定路径名时，可以使用完整路径名或相对路径名。然而，如果使用相对路径名，这必须相对于是初始化数据库服务器时的当前目录的目录。 **参考：**有关路径名语法，请参阅操作系统文档
-s *size*	指示新 sbspace 初始 chunk 的大小（千字节）	**限制：**无符号整数。大小必须等于或大于 1000 千字节，并且必须是页大小的整倍数。开始偏移量加 chunk 大小不能超过最大 chunk 大小。 最大 chunk 大小为 2 或 4 太字节。这与平台有关。
-t	创建用于存储临时智能大对象的临时 sbspace 。您可以指定元数据区域的大小和偏移量	**限制：**不能镜像临时 sbspace 。您可以指定任何 -Df 选项（LOGGING=ON 选项除外，它不会生效） **参考：**有关更多信息，请参阅 使用 -t 选项

元素	用途	关键注意事项
		创建临时 sbspace
-Df *default list*	列出存储在 sbspace 中智能大对象的缺省规范	**限制**:用逗号分隔标记，如果未提供标记，那么优先使用系统缺省值。在命令行上，该列表必须括在双引号（"）中。 **参考**:有关标记及其参数的列表，请参阅 表 1

使用 -t 选项创建临时 sbspace

本示例创建 1000 千字节的临时 sbspace ：

```
gspaces -c -S tempsbsp -t -p ./tempsbsp -o 0 -s 1000
```

您可以选择在 SBSPACETEMP 配置参数中指定临时 sbspace 的名称。重新启动数据库服务器，从而使它可使用临时 sbspace 。

使用 -Df 选项创建 Sbspace

使用可选的 -Df 选项创建 sbspace 时，可以指定几个缺省规范，这些规范影响存储在 sbspace 中的智能大对象的行为。缺省规范必须用逗号分隔的列表表示。该列表无需包含所有标记。标记列表必须括在双引号（") 中。表 1 中的表描述了标记及其缺省值。

Sbspace 特征的四个继承级别是系统、sbspace、列以及智能大对象。有关更多信息，请参阅 GBase 8s 管理员指南 中 数据存储在哪里 一章中的 智能大对象。

标记	值	缺省值	描述
ACCESSTIME	ON 或 OFF	OFF	设置为 ON 时，数据库服务器跟踪对存储在 sbspace 中的所有智能大对象的访问时间。 有关更改智能大对象存储特征的信息，请参阅 *GBase 8s DataBlade API 程序员指南*
AVG_LO_SIZE	Windows™ : 4 到 2**31 UNIX™: 2 到 2**31	8	指定存储在 sbspace 中智能大对象的平均大小（千字节） 数据库服务器使用该值计算元数据区域的大小。不要一起指定 AVG_LO_SIZE 和 -Ms 。可以一起

标记	值	缺省值	描述
			指定 AVG_LO_SIZE 和元数据偏移量（-Mo） 。 如果智能大对象的大小超过 2**31，那么指定 2**31。如果智能大对象的大小小于 2 （在 UNIX 上）或小于 4 （在 Windows 中），那么指定 2 或 4 。 如果耗尽 sbspace 中元数据和保留区域中的空间，那么返回错误 131 。要将额外的 chunk 分配给仅由元数据区域构成的 sbspace，请使用 -Ms 选项。 有关更多信息，请参阅 *GBase 8s 管理员指南* 中的 管理磁盘数据 一章中的 创建智能大对象
BUFFERING	ON 或 OFF	ON	指定存储在 sbspace 中智能大对象的缓冲方式 如果设置为 ON ，那么对于智能大对象 I/O 操作，数据库服务器使用共享内存常驻部分中的缓冲池。如果设置为 OFF，那么数据库服务器使用共享内存虚拟部分中的轻量级 I/O 缓冲区（轻量级 I/O 操作）。 BUFFERING = OFF 与 LOCK_MODE = RANGE 不兼容，会产生冲突。 有关更多信息，请参阅 *GBase 8s 性能指南*中的在内存上配置影响 一章中的 轻量级 I/O
LOCK_MODE	RANGE 或 BLOB	BLOB	指定存储在 sbspace 中的智能大对象的锁定方式 如果设置为 RANGE ，那么只锁定智能大对象中一定范围内的字节。如果设置为 BLOB ，那么锁定整个智能大对象。

标记	值	缺省值	描述
			LOCK_MODE = RANGE 与BUFFERING = OFF 不兼容，会产生冲突。 有关更多信息，请参阅 *GBase 8s 性能指南* 中的锁定 一章中的 智能大对象
LOGGING	ON 或 OFF	OFF	指定存储在 sbspace 中的智能大对象的登录状态 如果设置为 ON，那么数据库服务器将更改记录到 sbspace 中的用户数据区域。在打开 sbspace 的日志记录时，对 sbspace 进行 0 级备份。 当关闭日志记录时，显示以下消息：您正在关闭智能大对象日志记录 有关更多信息，请参阅 *GBase 8s 管理员指南* 中的 数据存储和记录 一章中的 智能大对象。 有关 **gspaces -ch** 消息的信息，请参阅 数据库服务器日志中的消息
EXTENT_SIZE	4 到 2**31	无	指定创建表时首次分配给存储在 sbspace 中的智能大对象的磁盘空间的大小（千字节） 让系统选择 EXTENT_SIZE 值。为减少智能大对象中 extent 的数量，请使用 **mi_lo_specset_estbytes** (DataBlade API) 或 **ifx_lo_specset_estbytes** (GBase 8s ESQL/C) 对系统指示智能大对象的大小合计。系统尝试向智能大对象分配单个 extent。 有关更多信息，请参阅*GBase 8s 管理员指南*中的数据存储在哪里 一章中的 智能大对象。有关更改智能大对象存储特征的信息，请参阅 *GBase 8s DataBlade API 程序员指南* 或 *GBase 8s ESQL/C*

标记	值	缺省值	描述
			程序员手册。
MIN_EXT_SIZE	2 到 2**31	Windows：4UNIX: 2	指定分配给每个智能大对象的最小空间量（千字节） 将显示以下信息：正在更改 sbspace 最小 extent 大小：旧值 *value1* 新值*value2* 。 有关调整该值的信息，请参阅 *GBase 8s 性能指南* 中的 在 I/O 利用率上的配置影响 一章中的智能大对象。有关 **gspaces -ch** 消息的信息，请参阅 数据库服务器日志中的消息
NEXT_SIZE	4 到 2**31	无	指定当前 sbspace 中初始 extent 已满时，下次分配给智能大对象的磁盘空间 extent 大小（千字节）让系统选择 NEXT_SIZE 值。要减少智能大对象中 extent 的数量，请使用 **mi_lo_specset_estbytes** 或 **ifx_lo_specset_estbytes** 对系统指示智能大对象的大小合计。系统尝试向智能大对象分配单个 extent。 有关更多信息，请参阅 *GBase 8s 管理员指南* 中的 数据存储在哪里 一章中的 智能大对象。有关获取智能大对象大小的信息，请参阅 *GBase 8s DataBlade API 程序员指南* 或 *GBase 8s ESQL/C 程序员手册*。

以下示例使用以下规范创建 20 兆字节的镜像 sbspace（eg_sbsp）：

主 chunk 和镜像 chunk 的偏移量为 500 千字节

元数据区域的偏移量为 200 千字节

平均期望智能大对象大小为 32 千字节

将更改记录到 sbspace 的用户数据区域中的智能大对象中

仅限于 UNIX：

```
% gspaces -c -S eg_sbsp -p /dev/raw_dev1 -o 500 -s 20000
        -m /dev/raw_dev2 500 -Mo 200 -Df "AVG_LO_SIZE=32,LOGGING=ON"
```

更改 -Df 设置

作为数据库服务器管理员，您可以使用下列方法之一重设或更改 -Df 设置：

要更改 sbspace 的缺省值设置，使用 gspaces -ch 选项。有关更多信息，请参阅 gspaces -ch: 更改 sbspace 缺省规范。

要重设特定表的以下 -Df 缺省设置，请使用 SQL 语句 CREATE TABLE 或 ALTER TABLE：

- LOGGING
- ACCESSTIME
- EXTENT_SIZE
- NEXT_SIZE

有关 ALTER TABLE 和 CREATE TABLE 语句的更多信息，请参阅 《GBase 8s SQL 指南：语法》。

程序员可以使用 DataBlade API 和 GBase 8s ESQL/C 函数重设这些 -Df 缺省设置。有关智能大对象的存储特征的信息，请参阅 GBase 8s DataBlade API 程序员指南 和 GBase 8s ESQL/C 程序员手册。

使用 gspaces -g 选项

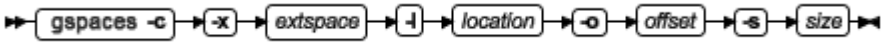
gspaces -g 选项不用于 sbspace 。数据库服务器对于 sbspace 使用不同于 blobspace 的另一种方法确定 I/O 操作中要传送的页数。数据库服务器可以自动确定在智能大对象的 I/O 操作中的传送的 chunk 大小。有关更多信息，请参阅 GBase 8s 性能指南 中 I/O 活动 一章中的 sbspace extent 大小。

该命令有等同的 SQL 管理 API 命令。

3.15.8 gspaces -c -x: 创建 extspace

使用 gspaces -c -x 选项创建 extspace。

语法:



元素	用途	关键注意事项
- c	创建 dbspace 、blobspace、 sbspace 或 extspace	创建存储空间之后，必须备份该存储空间和 root dbspace。如果所创建的存储空间名称与已删除存储空间的名称相同，那么执行另

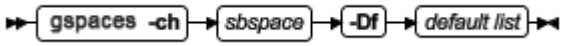
元素	用途	关键注意事项
	最多可以创建 2047 个任意类型的存储空间	一个 0 级备份，以确保以后的恢复不混淆新存储空间和旧存储空间。 有关更多信息，请参阅 *GBase 8s 管理员指南* 中 管理磁盘空间 一章中的 创建 dbspace 、blobspace 或 extspace
-l *location*	指定 extspace 的位置 存取方法决定该字符串的格式	**限制：**String 。值不得长于 255 字节。 有关更多信息，请参阅 *GBase 8s 管理员指南* 中的 管理磁盘空间 一章中的 创建 extspace
-o *offset*	指示为到达新的 blobspace、dbspace 或 sbspace 的初始 chunk 而发生的磁盘分区或设备中的偏移量（千字节）	**限制：**无符号整数。无符号整数。开始偏移量必须大于等于 0 。开始偏移量加 chunk 大小不能超过最大 chunk 大小。 最大偏移量为 2 或 4 太兆字节，这与平台有关。 有关更多信息，请参阅*GBase 8s 管理员指南* 中的 管理磁盘空间 一章中的 分配原始磁盘空间
-s *size*	指示新的 blobspace 或 dbspace 初始 chunk 的大小（千字节）	**限制**: 无符号整数。大小必须等于或大于 1000 千字节，并且必须是页大小的整倍数。开始偏移量加 chunk 大小不能超过最大 chunk 大小。 最大 chunk 大小为 2 或 4 太字节。这与平台有关。
-x *extspace*	给出要创建的 extspace 的名称	**限制：**Extspace 的名称必须唯一，且不能超过 128 字节。它必须以字母或下划线开始，且必须只包含字母、数字、下划线或 **$** 字符。

元素	用途	关键注意事项
		有关更多信息，请参阅 *GBase 8s 管理员指南* 中 管理磁盘空间 一章中的 extspace

3.15.9 gspaces -ch: 更改 sbspace 缺省规范

可以使用 gspaces -ch 选项更改 sbspace 缺省规范。

语法:



元素	用途	关键注意事项
-ch	指示要更改的一个或多个 sbspace 缺省规范	无
sbspace	给出要更改其缺省规范的 sbspace 名称	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*。有关背景信息，请参阅 *GBase 8s 性能指南* 中的 使用 gspaces 更改 sbspace 的缺省规范
-Df *default list*	列出存储在 sbspace 中的智能大对象的新缺省规范	用逗号分隔标记。如果未提供标记，那么优先使用系统缺省值。在命令行上，该列表必须括在双引号(") 中。 有关标记及其参数的列表，请参阅 表 1

使用 gspaces -ch 选项可以更改任何 -Df 标记。数据库服务器将此更改应用于在更改缺省规范之前创建的每个智能大对象。

例如：要关闭在使用 -Df 选项创建 Sbspace中创建的 sbspace 的日志记录，请使用以下命令：

gspaces -ch eg_sbsp -Df "LOGGING=OFF"

注： 在打开 sbspace 的日志记录之后，对 sbspace 进行 0 级备份以创建恢复点。

3.15.10 gspaces -cl: 清除 sbspace 中的游离智能大对象

可以使用 gspaces -cl 选项清除 sbspace 中的游离智能大对象。

语法:

元素	用途	关键注意事项
-cl	清除 sbspace 中的游离智能大对象	要查找任何游离智能大对象，请在没有用户连接到数据库服务器时使用 gcheck -pS 命令。引用计数为 0 的智能大对象为游离对象。
sbspace	给出要清除的 sbspace 的名称	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*

在正常运行过程中，不应存在任何未使用（游离）的智能大对象。删除智能大对象时，释放空间。如果在删除智能大对象时数据库服务器失败或耗尽系统内存，那么智能大对象可能作为游离对象保留。

以下是 gspaces -cl 命令的一个示例：

gspaces -cl myspace

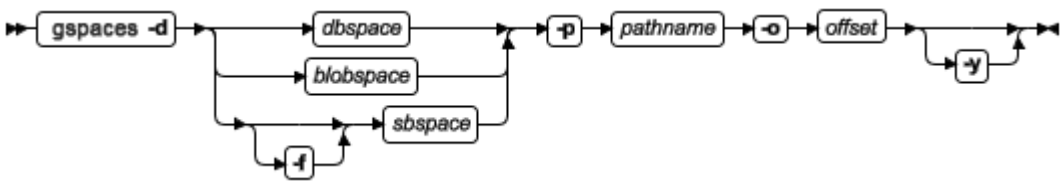
查找智能大对象引用计数的最佳方法是调用 C 程序的 mi_lo_stat 或 ifx_lo_stat函数。尽管 mi_lo_increfcount 和 mi_lo_decrefcount 函数返回引用计数，但它们增加或减少引用计数。有关这些函数的更多信息，请参阅 GBase 8s DataBlade API 函数参考。

该命令有等同的 SQL 管理 API 命令。

3.15.11 gspaces -d: 删除 dbspace 、blobspace 或 sbspace 中的 chunk

可以使用 gspaces -d 选项删除 dbspace 、blobspace 或 sbspace 中的 chunk 。

语法:



该命令有等同的 SQL 管理 API 命令。

元素	用途	关键注意事项
-d	删除 chunk	当数据库服务器处于联机或静默方式时，您可以删除 dbspace 、临时 dbspace 或 sbspace 中的

元素	用途	关键注意事项
		chunk。有关更多信息，请参阅 *GBase 8s 管理员指南* 中 管理磁盘空间 一章。 您只能在数据库服务器处于静默方式时才能删除 blobspace 中的 chunk
-f	删除包含用户数据但不包含元数据的 sbspace chunk 。如果 chunk 包含 sbspace 的元数据，那么必须删除整个 sbspace	只对 sbspace 使用 -f 选项。如果省略 -f 选项，那么不能删除包含数据的 sbspace 。 有关更多信息，请参阅 *GBase 8s 管理员指南* 中 管理磁盘空间 一章中的 从带有 gspaces 的 sbspace 中删除 chunk
-o *offset*	指示为到达您正在删除的 dbspace 、blobspace 或 sbspace 的初始 chunk 而发生的磁盘分区或设备中的偏移量（千字节）	**限制：**无符号整数。开始偏移量必须大于等于 0 。开始偏移量加 chunk 大小不能超过最大 chunk 大小。 最大偏移量是 4 太字节。 有关更多信息，请参阅 *GBase 8s 管理员指南* 中的 管理磁盘空间 一章中的 在 UNIX™ 上分配原始磁盘空间
-p *pathname*	指示您正在删除的 dbspace 、blobspace 或 sbspace 的初始 chunk 的磁盘分区或未缓冲设备	Chunk 必须是现有的未缓冲设备或已缓冲文件。指定路径名时，可以使用完整路径名或相对路径名。然而，如果使用相对路径名，这必须相对于是初始化数据库服务器时的当前目录的目录。 有关路径名语法，请参阅操作系统文档
-y	导致数据库服务器自动对所有提示响应“是”	无
blobspace	给出要删除其 chunk 的 blobspace 的名称	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*。有关从 blobspace 中删

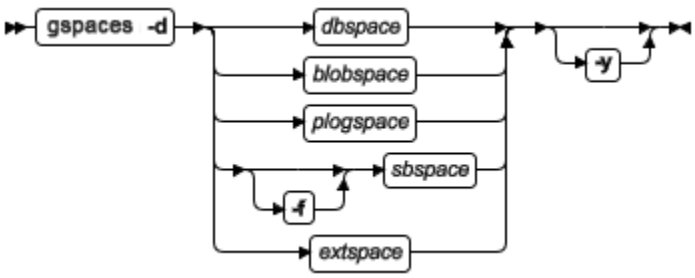
元素	用途	关键注意事项
		除 blobspace 的更多信息，请参阅 *GBase 8s 管理员指南* 中 管理磁盘空间 一章
dbspace	给出要删除其 chunk 的 dbspace 的名称	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*。有关使用 gspaces 删除 dbspace 中的 chunk，请参阅 *GBase 8s 管理员指南* 中 管理磁盘空间 一章
sbspace	给出要删除其 chunk 的 sbspace 的名称	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*。有关背景信息，请参阅 *GBase 8s 管理员指南* 中 管理磁盘空间 一章中从带有 gspaces 的 sbspace 中删除 chunk

重要： 必须指定路径名以指示数据库服务器您正在删除 chunk 。

3.15.12 gspaces -d: 删除空间

可以使用 gspaces -d 选项删除 dbspace 、blobspace 、plogspace 、sbspace 或 extspace 。

语法:



元素	用途	关键注意事项
-d	指示要删除的存储空间	可以在数据库服务器处于联机或静默方式时删除 dbspace 、blobspace 、plogspace 、sbspace 或 extspace。 删除存储空间之后，必须对它进行备份，以确保 **sysutils** 数据库和保留页时最新的。 执行 gcheck -pe ；来验证当前没有表正向 dbspace 、blobspace 或 sbspace 存储数据
-y	导致数据库服务器自动对	无

元素	用途	关键注意事项
	所有提示响应“是”	
-f	删除包含用户数据和元数据的 sbspace	必须使用 -f（强制）选项来删除含有数据的 sbspace 。 **限制：** 仅对 sbspace 使用 -f 选项。 **警告：** 如果使用 -f 选项，那么数据库服务器中的表可能含有指向已使用该选项删除的智能大对象的死指针
blobspace	给出要删除的 blobspace 的名称	在删除 blobspace 之前，请删除包含引用 blobspace 的 TEXT 或 BYTE 列的所有表
dbspace	给出要删除的 dbspace 的名称	在删除 dbspace 之前，请删除先前在 dbspace 中创建的所有数据库和表
extspace	给出要删除的 extspace 的名称	如果 extspace 与现有表或索引相关联，那么您无法删除它
plogspace	给出要删除的 plogspace 的名称	在删除之前，plogspace 必须为空
sbspace	给出要删除的 sbspace 的名称	在删除 sbspace 之前，请删除包含引用 sbspace 的 BLOB 或 CLOB 列的所有表

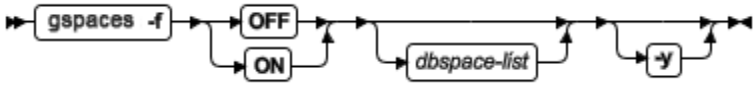
重要： 删除这些存储空间时不要指定路径名。

该命令有等同的 SQL 管理 API 命令。

3.15.13 gspaces -f: 指定 DATASKIP 参数

可以使用 gspaces -f 选项指定 DATASKIP 配置参数的值。

语法:



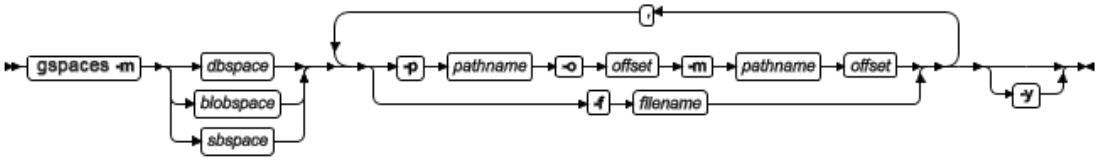
该命令有等同的 SQL 管理 API 命令。

元素	用途	关键注意事项
-f	向数据库服务器指示您想要更改指定 dbspace 或所有 dbspace 的 DATASKIP 缺省值	DATASKIP 状态中的所有更改都记录在消息日志中
-y	导致数据库服务器自动对所有提示响应“是”	无
dbspace-list	指定将打开（ON）或关闭（OFF）其 DATASKIP 的一个或多个 dbspace 的名称	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*。有关更多信息，请参阅 DATASKIP 配置参数 和 *GBase 8s 性能指南*
OFF	关闭 DATASKIP	如果使用 OFF 时未使用 *dbspace-list* ，那么对所有分片关闭 DATASKIP 。如果与 *dbspace-list* 一起使用 OFF ，那么只有指定分片被设置为 DATASKIP 关闭
ON	打开 DATASKIP	如果使用 ON 时未使用 *dbspace-list* ，那么对所有分片打开 DATASKIP 。如果与 *dbspace-list* 一起使用 ON ，那么只有指定分片被设置为 DATASKIP 打开

3.15.14 gspaces -m: 启动镜像

可以使用 gspaces -m 选项启动 dbspace 、blobspace 或 sbspace 的镜像。

语法:



该命令有等同的 SQL 管理 API 命令。

元素	用途	关键注意事项

元素	用途	关键注意事项
-f *filename*	指示 chunk 位置信息是在名为 *filename* 的文件中	该文件必须是已存在的已缓冲文件。该路径名必须符合特定于操作系统的路径名规则。 有关更多信息，请参阅 使用文件以 -f 选项指定 chunk 位置信息.
-m	添加现有 dbspace 、blobspace 或 sbspace	镜像 sbspace 中的用户数据 chunk 不需要进行镜像。 镜像 chunk 应在另一个磁盘上，必须一次镜像所有 chunk
-m *pathname offset*	*pathname* 第二次出现在语法图中，它指示执行镜像的 dbspace 、blobspace 或 sbspace 的初始 chunk 的磁盘分区或未缓冲设备。 *offset* 第二次出现在语法图中，它指示到达新镜像 dbspace 、blobspace 或 sbspace 的镜像 chunk 的偏移量。另见表中的 *pathname* 和 *offset* 条目	无
-o *offset*	*offset* 第一次出现在语法图中，它指示为未到达新镜像 dbspace 、blobspace 或 sbspace 的初始 chunk 而发生的磁盘分区或设备中的偏移量（千字节）	**限制**: 无符号整数。开始偏移量必须大于等于 0 。开始偏移量加 chunk 大小不能超过最大 chunk 大小。 最大偏移量为 4 太字节。 有关更多信息，请参阅 *GBase 8s 管理员指南*中的 管理磁盘空间 一章中的 在 UNIX™ 上分配原始磁盘空间

元素	用途	关键注意事项
-p *pathname*	*pathname* 第一次出现在语法图中，它指示您正要镜像的 dbspace 、blobspace 或 sbspace 的初始 chunk 的磁盘分区或未缓冲设备	Chunk 必须是现有的未缓冲设备或已缓冲文件。指定路径名时，可以使用完整路径名或相对路径名。然而，如果使用相对路径名，这必须相对于是初始化数据库服务器时的当前目录的目录。 有关路径名语法，请参阅操作系统文档
-y	导致数据库服务器自动对所有提示响应“是”	无
blobspace	给出想要镜像的 blobspace 的名称	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*。有关更多信息，请参阅 *GBase 8s 管理员指南* 中的 使用镜像 一章
dbspace	给出想要镜像的 dbspace 的名称	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*。有关背景信息，请参阅 *GBase 8s 管理员指南* 中的 使用镜像 一章
sbspace	给出想要镜像的 sbspace 的名称	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*。有关背景信息，请参阅 *GBase 8s 管理员指南* 中的 使用镜像 一章

使用文件以 -f 选项指定 chunk 位置信息

可以创建包含 chunk 位置信息的文件。然后，当执行 gspaces 时，使用 -f 选项向数据库服务器指示该信息位于其名称在 filename 中指定的文件。

文件的内容应遵循以下格式，选项用空格隔开，每组主 chunk 和镜像 chunk 都在单独的行上：

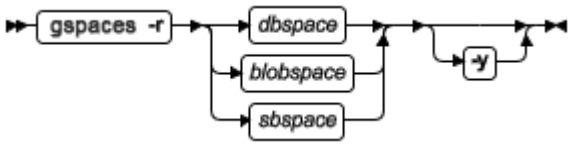
```
primary_chunk_path offset mirror_chunk_path offset
```

如果正在镜像的 dbspace 包含多个 chunk ，那么必须为要镜像的 dbspace 中的每个主 chunk 指定镜像 chunk 。有关对多 chunk dbspace 启用镜像的示例，请参阅 GBase 8s 管理员指南 中 使用镜像 一章中的 开始为带有 gspaces 的未镜像的 dbspace 镜像。

3.15.15 gspaces -r: 停止镜像

可以使用 gspaces -r 选项结束 dbspace 、blobspace 或 sbspace 的镜像。

语法:



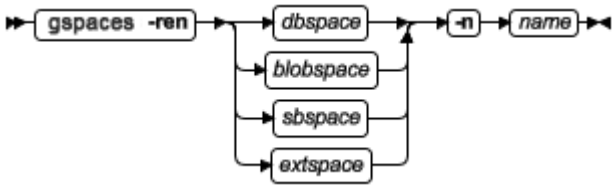
该命令有等同的 SQL 管理 API 命令。

元素	用途	关键注意事项
-r	向数据库服务器指示现有的 dbspace 、 blobspace 或 sbspace 的镜像应结束	有关背景信息,请参阅 *GBase 8s 管理员指南* 中的 使用镜像 一章
-y	导致数据库服务器自动对所有提示响应“是”	无
blobspace	给出您想要结束镜像的 blobspace 的名称	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*。有关更多信息，请参阅 *GBase 8s 管理员指南* 中的 使用镜像 一章
dbspace	给出您想要结束镜像的 dbspace 的名称	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*。有关更多信息，请参阅 *GBase 8s 管理员指南* 中的 使用镜像 一章
sbspace	给出您想要结束镜像的 sbspace 的名称	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*。有关背景信息，请参阅 *GBase 8s 管理员指南* 中的 使用镜像 一章

3.15.16 gspaces -ren: 重命名 dbspace 、blobspace 、sbspace 或 extspace

可以使用 gspaces -ren 选项重命名 dbspace 、blobspace 、sbspace 或 extspace 。

语法:



该命令有等同的 SQL 管理 API 命令。

元素	用途	关键注意事项
-ren	使得数据服务器重命名指定的 dbspace 、blobspace 、sbspace 或 extspace	**限制：**当数据处于静默方式时，您可以重命名 dbspace 、blobspace 、sbspace 或 extspace。有关更多信息，请参阅 *GBase 8s 管理员指南* 中管理磁盘空间 一章
-n *name*	指定 blobspace 、dbspace 、extspace 或 sbspace 的新名称	**限制：**Dbspace 、blobspace 、sbspace 或 extspace 的名称必须唯一，且不能超过 128 字节。它必须以字母或下划线开始，且必须只包含字母、数字、下划线或 **$** 字符。 有关更多信息，请参阅 *GBase 8s 管理员指南* 中管理磁盘空间 一章。语法必须符合 Identifier 段；有关更多信息，请参阅 *《GBase 8s SQL 指南：语法》*
blobspace	给出要重命名的 blobspace 的名称	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*。有关更多信息，请参阅 *GBase 8s 管理员指南* 中 管理磁盘空间 一章中的 重命名空间
dbspace	指定要重命名的 dbspace	**限制：**您不能重命名关键 dbspace ，例如：root dbspace 或包含物理日志的 dbspace 。 **其他信息：**如果您对包含在 DATASKIP 列表中的 dbspace 进行重命名，那么请使用 gspaces -f 命令用新的名称更新 DATASKIP 配置参数。 语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*。有关更多信息，请参阅

元素	用途	关键注意事项
		GBase 8s 管理员指南 中 管理磁盘空间 一章中的 重命名空间
extspace	指定要重命名的 extspace	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*。有关更多信息，请参阅*GBase 8s 管理员指南*中 管理磁盘空间 一章中的 重命名空间
sbspace	指定要重命名的 sbspace	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*。有关更多信息，请参阅 *GBase 8s 管理员指南* 中 管理磁盘空间 一章中的 重命名空间

当 Enterprise Replication 活动时，重命名 dbspace 、 blobspace 、sbspace 或 extspace

您可以在 Enterprise Replication 活动时重命名空间（dbspace、 blobspace 、sbspace 或 extspace ）。

当您将数据库服务器置于静默方式来重命名空间时，Enterprise Replication 将断开连接。然后您就可以重命名空间了。在将数据库服务器置于联机方式之后，服务器将会重新同步。如果想要在另一台服务器上重命名相同空间，那么必须将该服务器置于静默方式并单独重命名该空间。不同的 ER 服务器上的重命名的空间之间不传播强制关系；相同的表可以在不同的空间中。

如果 Enterprise Replication 服务器还参与 High-Availability Data Replication (HDR) ，那么您可以重命名主服务器上的 dbspace ，它将会自动传播到辅助服务器。（辅助服务器不能参与 Enterprise Replication 。）

在重命名空间之后执行归档

在重命名任何空间（extspace 或临时空间除外）之后，请对重命名的空间和 root dbspace 执行 0 级归档。这将确保您能够将空间恢复到包括重命名 dbspace 操作的状态或者该操作之后的状态。在执行任何其他类型的归档之前，也必须执行该操作。

3.15.17 gspaces -s: 更改镜像 chunk 的状态

使用 gspaces -s 选项来更改在非临界 dbspace 、blobspace 或 sbspace 中的 dbspace 、非主 chunk 的镜像 chunk 的状态。

语法：

该命令有等同的 SQL 管理 API 命令。

元素	用途	关键注意事项
-D	指示您要关闭 chunk	无
-o *offset*	指示为到达 chunk 而发生的磁盘分区或设备中的偏移量（千字节）	**限制：** U无符号整数。开始偏移量必须大于等于 0 。开始偏移量加 chunk 大小不能超过最大 chunk 大小。该偏移量必须是页大小的倍数。 最大偏移量是 4 太字节。 有关更多信息，请参阅 *GBase 8s 管理员指南* 中的 管理磁盘空间 一章中的 在 UNIX™ 上分配原始磁盘空间
-O	指示您要恢复 chunk 并使它联机	无
-p *pathname*	指示 chunk 的磁盘分区或未缓冲设备	Chunk 必须是现有的未缓冲设备或已缓冲文件。指定路径名时，可以使用完整路径名或相对路径名。然而，如果使用相对路径名，这必须相对于是初始化数据库服务器时的当前目录的目录。 有关路径名语法，请参阅操作系统文档
-s	指示您想要更改 chunk 的状态	**限制：**只能更改镜像对中的 chunk 或在非临界 dbspace 中的非主 chunk 的状态。 有关更多信息，请参阅 *GBase 8s 管理员指南* 中的 更改镜像状态
-y	导致数据库服务器自动对所有提示响应“是”	无

元素	用途	关键注意事项
blobspace	给出想要更改其状态的 blobspace 的名称	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*。有关更多信息，请参阅 *GBase 8s 管理员指南* 中的 更改镜像状态
dbspace	给出想要更改其状态的 dbspace 的名称	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*。有关更多信息，请参阅 *GBase 8s 管理员指南* 中的 更改镜像状态
sbspace	给出想要更改其状态的 sbspace 的名称	语法必须符合 Identifier 段；请参阅 *《GBase 8s SQL 指南：语法》*。有关背景信息，请参阅 *GBase 8s 管理员指南* 中的 更改镜像状态

3.15.18 避免覆盖 chunk

与一个 GBase 8s 实例相关联的 chunk 不会被其他的 GBase 8s 实例识别。它可能无意中创建被分配到另一个 GBase 8s 实例文件或设备中的 chunk ，这会导致数据损坏。

如果您尝试初始化一个 ROOTPATH 配置参数指定的文件或设备是别的实例的 root chunk 的实例，那么该命令失败并在 online.log 中显示以下消息：

DISK INITIALIZATION ABORTED: potential instance overwrite detected.

要禁用此次初始化检查，将配置文件中的 FULL_DISK_INIT 配置参数设置为 1 ，并尝试再次初始化该实例。然而，该初始化检查限制 root chunk。尽管该文件或设备被分配到其他实例，添加 dbspace 或 chunk 成功。

3.16 gstat 实用程序

gstat 实用程序读取共享内存结构，并提供有关数据库服务器在该命令执行时的统计信息。可在单个命令中组合多个 gstat 选项标志。gstat 显示输出时，共享内存的内容可能更改。gstat 实用程序不在共享内存上放置任何锁，因此运行该实用程序不会影响性能。

可使用等同于 gstat 命令的 SQL 管理 API 函数。

3.16.1 gstat 门户：按功能目录排列的 gstat 实用程序

本章列出了 gstat 命令，它们按函数类别排序。

每个目录代表不同的 GBase 8s 的功能。gstat 命令有助于提供故障转移和性能扩展信息。以粗体形式出现的命令特别有助于提供故障转移的信息。某些 gstat 命令是具体到一个目录的，而其它命令提供更一般的信息，并在一个以上的目录中列出。

目录列表

从以下列表中确定合适的目录，然后链接到那个目录的 gstat 的选项。

- gstat 实用程序归档信息选项
- gstat 实用程序高速缓存信息选项
- gstat 实用程序压缩选项
- gstat 实用程序 Debugging 选项
- gstat 实用程序 Enterprise Replication 选项
- gstat 实用程序高可用复制选项
- gstat 实用程序 I/O 选项
- gstat 实用程序锁和锁存器选项
- gstat 实用程序日志选项
- gstat 实用程序内存选项
- gstat 实用程序网络选项
- gstat 实用程序性能检查（第一层）
- gstat 实用程序性能检查（第二层）
- gstat 实用程序表选项
- gstat 实用程序线程选项
- gstat 实用程序用户/会话选项
- gstat 实用程序虚拟处理器选项
- gstat 实用程序等待选项
- 其他有用的 gstat 实用程序选项

gstat 实用程序归档信息选项

使用以下 gstat 选项显示有关归档和恢复的信息。

表 1. gstat 实用程序归档信息选项

命令	参考
gstat -D	打印 chunk I/O 活动。打印用于监视恢复过程的 dbspace 读/写活动。 gstat -D 命令：打印页读取和页写入信息
gstat -g arc	打印最后提交和任一正在进行备份的每个 dbspace 。

命令	参考
	gstat -g arc 命令：打印归档状态

gstat 实用程序高速缓存信息选项

使用以下 gstat 选项显示有关高速缓存和缓存数据（包括缓冲池）的信息。

表 2. gstat 实用程序高速缓存信息选项

命令	参考
gstat -b	打印使用中的缓冲区页。 gstat -b 命令：打印正在使用的缓冲区信息
gstat -B	打印已使用的缓冲区的信息。 gstat -B 命令：打印已使用的缓冲区信息
gstat -F	打印缓冲区队列清除程序和 I/O 的状态。 gstat -F 命令：打印计数
gstat -g cac	打印所有内存高速缓存或指定的高速缓存的概要和详细信息。 gstat -g cac 命令：打印有关缓存的信息
gstat -g dic	打印数据字典中的高速缓存，包含表的系统目录数据。打印=共享内存字典中高速缓存的每张表的一行信息。 有关更多信息，请参阅 *GBase 8s 性能指南* 。 gstat -g dic 命令：打印表信息
gstat -g dsc	为优化程序打印的表的分布统计信息。

命令	参考
	gstat -g dsc 命令：打印分布式高速缓存信息.
gstat -g prc	打印存储程序（SPL）例程的高速缓存。 打印有关 SPL 例程高速缓存的信息。 gstat -g prc 命令：打印使用 UDR 或 SPL 例程的会话
gstat -g ssc	打印数据库服务器读取高速缓存中 SQL 语句的次数。显示与 gstat -g cac 相同的输出。 有关更多信息，请参阅 *GBase 8s 性能指南* 中的提高查询性能。 gstat -g ssc 命令：打印出现的 SQL 语句
gstat -g vpcache	打印 CPU 虚拟处理器内存高速缓存。 gstat -g vpcache 命令：打印 CPU 虚拟处理器专用内存高速缓存的统计信息
gstat -h	打印缓冲区散列链信息。 gstat -h 命令：打印缓冲区头哈希链信息
gstat -p	打印缓冲池高速缓存有效性方面的全局（服务器）信息。 gstat -p 命令：打印概要文件计数
gstat -X	打印正在等待缓冲区的线程。 gstat -X 命令：打印线程信息

gstat 实用程序压缩选项

使用以下 gstat 选项打印压缩信息。

表 3. gstat 实用程序压缩选项

命令	参考
gstat -g dsk	打印当前正在运行的压缩操作的进度。 gstat -g dsk 命令：打印当前正在运行的压缩操作的进度
gstat -g ppd	打印分区压缩字典信息。 gstat -g ppd 命令：打印分区压缩字典信息

gstat 实用程序 Debugging 选项

使用以下 gstat 选项显示对服务器调试问题有用的信息。

表 4. gstat 实用程序 Debugging 选项

命令	参考
gstat -g dmp	打印在若干个给定的字节的给定的地址处原内存信息。 gstat -g dmp 命令：打印原内存
gstat -g src	搜索共享内存中的模式。注意在 Intel™ 平台上内存是以字节交互的。 gstat -g src 命令：共享内存中的模式
gstat -o	打印输出的共享缓存内容。 gstat -o 命令：输出共享内存内容。

gstat 实用程序 Enterprise Replication 选项

使用以下 gstat 选项跟踪 Enterprise Replication 统计信息并提供故障转移信息。有关 Enterprise Replication 的其他信息，请参阅 GBase 8s Enterprise Replication 指南 中 cdr view 和 cdr view profile 命令的描述 。

表 5. gstat 实用程序 Enterprise Replication 选项

命令	参考
gstat -g cat	打印来自 Enterprise Replication 全局目录的信息。该全局目录包含有关定义的服务器、复制和 enterprise 内的每个服务器的复制集的概要信息。 gstat -g cat
gstat -g cdr	打印所有 Enterprise Replication 统计信息命令的输出。 gstat -g cdr
gstat -g cdr config	打印 Enterprise Replication 配置参数和环境变量。 gstat -g cdr config
gstat -g ddr	打印 Enterprise Replication 读取和处理日志记录的组件的状态。 gstat -g ddr
gstat -g dss	打印单独数据同步（正在处理事务）线程的活动。 gstat -g dss
gstat -g dtc	打印删除表的清除程序的活动。删除或修改删除表中的行会每隔一段时间清除的。 gstat -g dtc
gstat -g grp	打印 Enterprise Replication grouper 的统计信息。此 grouper 评估日志记录、向源事务中重建单独的日志记录、打包事务并将要传输的事务列队。

命令	参考
	gstat -g grp
gstat -g nif	打印网络接口的统计信息。显示网络接口、服务器和在服务器中数据传输的状态。 gstat -g nif
gstat -g que	打印高级别队列接口（适用于 Enterprise Replication Queue Manager 的所有队列）的统计信息。 gstat -g que
gstat -g rcv	打印接收管理器统计信息。 gstat -g rcv
gstat -g rep	打印队列中用于调度管理的事件。 gstat -g rep
gstat -g rqm	打印由 Reliable Queue Manager （RQM）管理的低级别队列（发送队列、接收队列、请求发送队列、同步发送队列和控制发送队列）的统计信息。 gstat -g rqm
gstat -g sync	打印同步状态。 gstat -g sync

gstat 实用程序高可用复制选项

使用以下 gstat 选项监视高可用性集群环境和连接管理器。

表 6. gstat 实用程序高可用复制选项

命令	参考
gstat -g cluster	打印高可用性集群信息。 gstat -g cluster 命令：打印高可用集群信息
gstat -g cmsm	打印连接管理器信息。 gstat -g cmsm 命令：打印连接管理器的信息
gstat -g dri	打印数据复制信息。 请参阅 *GBase 8s 管理员指南* 中的 *监视高可用数据复制状态* gstat -g dri 命令：打印高可用性数据复制信息.
gstat -g ipl	打印索引页日志记录状态。 gstat -g ipl 命令：打印索引页日志状态信息
gstat -g laq	打印辅助服务器队列信息。 gstat -g laq 命令：打印辅助服务器队列
gstat -g proxy	打印高可用性代理分发器。 gstat -g proxy 命令：打印代理分发器信息
gstat -g rss	打印远程独立服务器（RSS）的信息。 gstat -g rss 命令：打印 RS 辅助服务器信息
gstat -g sds	打印共享磁盘辅助（SDS）服务器的信息。

命令	参考
	gstat -g sds 命令：打印 SD 辅助服务器信息
gstat -g smx	打印在高可用环境中多路复用器（SMX）的连接。打印数据传输统计信息和加密状态。打印数据传输的统计信息。 gstat -g smx 命令：打印多路复用器组信息

gstat 实用程序 I/O 选项

使用以下 gstat 选项跟踪输入和输出（读和写）活动。

表 7. gstat 实用程序 I/O 选项

命令	参考
gstat -D	打印 chunk I/O 活动。 gstat -D 命令：打印页读取和页写入信息
gstat -g cpu	打印每个线程的运行时间的统计信息。 gstat -g cpu: 打印运行时间统计信息
gstat -g ioa	打印 gstat -g ioq（队列）、gstat -g iov （虚拟处理器）和 gstat -g iob （大缓冲区）的合并信息。 gstat -g ioa 命令：打印合并的 gstat -g 信息
gstat -g iob	打印大缓冲区的使用摘要。 gstat -g iob 命令：打印大缓冲区的使用摘要
gstat -g iof	打印文件或 chunk 的 I/O 统计信息。该选项与 gstat -D 选项相似，但是它也显示有关非 chunk 、临时和排序工作文件的信息。

命令	参考
	gstat -g iof 命令：打印 asynchronous I/O 统计信息
gstat -g iog	打印 AIO 全局信息。 gstat -g iog 命令：打印 AIO 全局信息
gstat -g ioq	打印队列读取/写入的统计信息和队列长度。 gstat -g ioq 命令：打印 I/O 队列信息 。另见 *GBase 8s 性能指南* 。
gstat -g iov	打印每个虚拟处理器的异步 I/O 统计信息。 gstat -g iov 命令：打印 AIO VP 统计信息
gstat -p	打印全部磁盘活动，包括顺序扫描。 gstat -p 命令：打印概要文件计数

gstat 实用程序锁和锁存器选项

使用下列 gstat 选项显示有关锁的信息。

表 8. gstat 实用程序锁和锁存器选项

命令	参考
gstat -k	打印有关活动锁的信息。 gstat -k 命令：打印活动的锁信息
gstat -L	打印打印在锁可用列表上的可用锁的数量。 gstat -L 命令：打印可用锁的数量

命令	参考
gstat -p	打印有关锁请求、锁等待和锁存器等待的全部统计信息。 gstat -p 命令：打印概要文件计数
gstat -s	打印锁存器（互斥）的信息。 gstat -s 命令：打印锁存器信息

gstat 实用程序日志选项

使用下列 gstat 选项监视逻辑日志和物理日志。

表 9. gstat 实用程序日志选项

命令	参考
gstat -g ipl	打印在高可用环境中索引页的日志记录信息。 gstat -g ipl 命令：打印索引页日志状态信息
gstat -l	打印物理日志、逻辑日志的状态和正在缓冲的日志。 gstat -l 命令：打印物理和逻辑日志信息

gstat 实用程序内存选项

使用下列 gstat 选项监视已分配和使用的服务器内存的各个方面。

表 10. gstat 实用程序内存选项

命令	参考
gstat -g afr	打印分配给会话或共享内存池的内存段。要获得池名称，请查看 gstat -g mem 选项。 gstat -g afr 命令: 打印分配的内存分片

命令	参考
gstat -g ffr (*pool name session ID*)	打印会话或共享内存池的空间片段。 gstat -g ffr 命令：打印空闲分片
gstat -g lmm	打印有关自动低内存管理设置及其最近的活动： gstat -g lmm 命令：打印低内存管理信息
gstat -g mem	打印会话或池的虚拟共享内存的统计信息。 gstat -g mem 命令：打印池内存统计信息
gstat -g mgm	打印内存分配器（并行和排序操作）的资源的信息。 gstat -g mgm 命令：打印 MGM 资源信息。另见 *GBase 8s 性能指南*.
gstat -g nbm	打印非常驻段的 block 位图。 gstat -g nbm 命令：打印 block 位图
gstat -g rbm	打印常驻段的 block 映射。 gstat -g rbm 命令：打印共享内存的 block 映射
gstat -g seg	打印内存段的统计信息。 gstat -g seg 命令：打印共享内存段的统计信息。另见 *GBase 8s 管理员指南* 。
gstat -g ses	打印会话信息，包含已故障的内存。有关详细信息，请使用：gstat -g ses *session_id* 。

命令	参考
	gstat -g ses 命令：打印与会话有关的信息 。另见 *GBase 8s 性能指南*
gstat -g stm	打印 SQL 语句的内存的使用。 gstat -g stm 命令：打印 SQL 语句的内存使用
gstat -g stq	打印流队列缓冲区。 gstat -g stq 命令：打印队列信息
gstat -g ufr	打印正在使用共享内存或会话的内存池分段。 gstat -g ufr 命令：打印内存池片分片
gstat -R	打印缓冲池队列及其状态。 gstat -R 命令：打印 LRU 、FLRU 和 MLRU 队列信息

gstat 实用程序网络选项

使用下列 gstat 选项监视共享内存和网络连接服务。

表 11. gstat 实用程序网络选项

命令	参考
gstat -g imc	打印连接数据库服务器的 GBase 8s MaxConnect 示例的信息。如果 GBase 8s MaxConnect 未连接数据库服务器，那么此命令显示 No MaxConnect servers are connected 。
gstat -g nsc	通过 *client id* 打印共享内存的状态。如果未提供 ，那么显示所有客户端状态区域。此命令打印与 nss 命令相同的状态数据。

命令	参考
	gstat -g nsc 命令：打印当前的共享内存连接信息
gstat -g nsd	打印轮询线程的网络共享内存数据。 gstat -g nsd 命令：打印轮询线程共享内存数据
gstat -g nss	通过 *session id* 打印网络共享内存的状态。如果未提供 *session id*，那么会显示所有会话的状态区域。此命令打印与 gstat -g nsc 命令相同的状态数据。 gstat -g nss 命令：打印共享内存网络连接状态
gstat -g nta	打印来自 gstat -g ntd 、gstat -g ntm 、gstat -g ntt 和 gstat -g ntu 合并的统计信息。如果安装了 GBase 8s MaxConnect，那么此命令打印您可以用于调整 GBase 8s MaxConnect 性能的统计信息。
gstat -g ntd	通过服务打印网络统计信息。 gstat -g ntd 命令：打印网络统计信息
gstat -g ntm	打印网络邮件统计信息。 gstat -g ntm 命令：打印网络邮件的统计信息
gstat -g ntt	打印网络用户次数。 gstat -g ntt 命令：打印网络用户的次数
gstat -g ntu	打印网络用户统计信息。 gstat -g ntu 命令：打印网络用户统计信息

gstat 实用程序性能检查（第一层）

使用下列 gstat 选项监视性能并检查性能障碍。使用第二层 gstat 选项（和其它 gstat 命令）进一步缩小问题。

表 12. gstat 实用程序性能检查（第一层）

命令	参考
gstat -c	打印服务器配置。 gstat -c 命令：打印 ONCONFIG 文件内容
gstat -D	打印 chunk I/O 。 gstat -D 命令：打印页读取和页写入信息
gstat -g ath	打印所有线程的状态和统计信息。sqlexec 线程是客户端会话线程。rstcb 值与 gstat -u 命令的用户字段相关。 gstat -g ath 命令：打印所有线程的信息 。有关使用 gstat -g ath 打印 Enterprise Replication 线程的信息，请参阅 *GBase 8s Enterprise Replication 指南* 。
gstat -g ckp	打印 checkpoint 历史记录并显示建议的配置。 gstat -g ckp 命令：打印 checkpoint 历史记录和配置建议
gstat -g cpu	打印每个线程运行时间的统计信息。 gstat -g cpu: 打印运行时间统计信息
gstat -g ioq	打印 *queue name* 暂挂的 I/O 操作。 gstat -g ioq 命令：打印 I/O 队列信息

命令	参考
gstat -p	打印全局服务器性能概要文件。 gstat -p 命令：打印概要文件计数
gstat -u	打印用户线程的状态和统计信息。如果线程正在等待资源，那么该命令会识别资源的类型（flags 字段）和地址（wait 字段）。 gstat -u 命令：打印用户活动概要文件

gstat 实用程序性能检查（第二层）

使用下列 gstat 选项识别性能障碍。

表 13. gstat 实用程序性能检查（第二层）

命令	参考
gstat -b	打印活动的缓冲区。 gstat -b 命令：打印正在使用的缓冲区信息
gstat -g act	打印活动的线程。 gstat -g act 命令：打印活动线程
gstat -g glo	打印虚拟处理器和它们正在操作的系统进程（oninit 进程）。打印虚拟处理器 CPU 使用选项。在 Windows™ 上，虚拟处理器正在操作系统线程， pid 字段中的值是线程 ID 。 gstat -g glo 命令：打印全局多线程信息
gstat -g mgm	打印内存分配管理器资源信息。 gstat -g mgm 命令：打印 MGM 资源信息

命令	参考
gstat -g rah	打印预读请求信息。 gstat -g rah 命令：打印预读请求统计信息
gstat -g rea	打印在就绪队列中正在等待 CPU 资源的线程。 gstat -g rea 命令：打印准备就绪的线程
gstat -g seg	打印共享内存段统计信息。该选项显示分配到数据库服务器的共享内存段的数量和大小。 gstat -g seg 命令：打印共享内存段的统计信息.
gstat -g wai	打印正在等待的线程；所有正在等待互斥互条件或正在生成的线程。 gstat -g wai 命令：打印等待队列线程队列
gstat -k	打印活动的锁。 gstat -k 命令：打印活动的锁信息

gstat 实用程序表选项

使用下列 gstat 选项显示有关表的状态及其统计信息。

表 14. gstat 实用程序表选项

命令	参考
gstat -g buf	打印缓冲池的概要文件信息 。 gstat -g buf 命令：打印缓冲池的概要文件信息

命令	参考
gstat -g lap	打印当前活动的细体附加的状态信息（写绕过缓冲池）。 gstat -g lap 命令：打印轻量级追加(GBase_8s light append)状态信息
gstat -g opn	打印打开的分区（表）。 gstat -g opn 命令：打印打开的分区
gstat -g ppf	打印指定分区号的分区概要文件（活动数据）或打印所有分区的概要文件。 gstat -g ppf 命令：打印分区的概要文件
gstat -g scn	打印基于压缩表的行扫描、具有比页大的行的和具有 VARCHAR 、LVARCHAR 和 NVARCHAR 数据的表的扫描过程的信息，并识别该扫描时轻度扫描还是缓冲池扫描。 gstat -g scn 命令；打印扫描选项
gstat -P	打印缓冲池中的按分区（表）列出的分区（表）和 B-tree 页。 gstat -P 命令：打印分区信息
gstat -t gstat -T	打印所有活动（t）的基本 tblspace （分区）的信息或所有（T）tblspace 的信息。 gstat -t 和 gstat -T 命令：打印 tblspace 信息

gstat 实用程序线程选项

使用下列 gstat 选项显示线程的活动和状态。

表 15. gstat 实用程序线程选项

命令	参考
gstat -g act	打印活动的线程。该输出包含在 gstat -g ath 输出中。 gstat -g act 命令：打印活动线程
gstat -g ath	打印所有线程。 gstat -g ath 命令：打印所有线程的信息。有关使用 gstat -g ath 打印 Enterprise Replication 线程的信息，请参阅 *GBase 8s Enterprise Replication 指南*.。
gstat -g bth	显示正在阻塞的和正在等待的线程之间的依赖关系。 gstat -g bth 和 -g BTH：打印阻塞的和正在等待的线程
gstat -g BTH	打印正在阻塞的线程的会话和堆栈信息。 gstat -g bth 和 -g BTH：打印阻塞的和正在等待的线程
gstat -g cpu	打印每个线程运行时间的统计信息。 gstat -g cpu: 打印运行时间统计信息
gstat -g rea	打印已准备就绪的线程（正在等待 CPU 资源的线程）。该输出包含在 gstat -g ath 输出。 gstat -g rea 命令：打印准备就绪的线程.
gstat -g sle	打印正在休眠的线程信息（休眠特定时间）。不包括那些永久休眠的线程。 gstat -g sle 命令：打印所有休眠的线程
gstat -g stk	打印指定线程的堆栈或所有线程的堆栈。

命令	参考
	gstat -g stk 命令：打印线程堆栈
gstat -g sts	打印每个线程的最大和当前堆栈使用的信息。 gstat -g sts 命令：打印每个线程的堆栈用途
gstat -g tpf	打印线程活动统计信息。 gstat -g tpf 命令：打印线程概要文件
gstat -g wai	打印正在等待的线程（空闲、休眠和等待）。包含于 gstat -g ath 输出。 gstat -g wai 命令：打印等待队列线程队列
gstat -g wst	打印线程的等待统计信息 gstat -g wst 命令：打印线程的等待统计信息

gstat 实用程序用户/会话选项

使用下列 gstat 选项显示有关用户环境和活动的会话的信息。

表 16. gstat 实用程序用户/会话选项

命令	参考
gstat -g env	打印数据库服务器正在使用的环境变量值。 gstat -g env 命令：打印环境变量值
gstat -g his	打印 SQL 跟踪信息。 gstat -g his 命令：打印 SQL 跟踪信息

命令	参考
gstat -g pqs	打印当前 SQL 查询使用的运算符。 gstat -g pqs 命令：打印所有 SQL 查询的运算符
gstat -g ses	打印所有活动的会话的概要信息或单独会话的详细信息。 gstat -g ses 命令：打印与会话有关的信息
gstat -g spf	打印所有活动会话的已就绪语句的概要文件。 gstat -g spf 命令：打印已就绪语句的概要文件
gstat -g sql	打印所有活动的会话的 SQL 信息和单独的会话的详细 SQL 信息。 gstat -g sql 命令：打印与 SQL 有关的会话信息
gstat -G	打印全局事务。 gstat -G 命令：打印 TP/XA 事务信息
gstat -u	打印用户线程的状态和它们全局读/写统计信息。 gstat -u 命令：打印用户活动概要文件
gstat -x	打印有关事务的信息。 gstat -x 命令：打印数据库服务器事务信息

gstat 实用程序虚拟处理器选项

使用以下 gstat 选项显示有关虚拟处理器的信息及其统计信息。

表 17. gstat 实用程序虚拟处理器选项

命令	参考

命令	参考
gstat -g glo	打印全局多线程信息和虚拟处理器类的全局统计信息和单独虚拟处理器。在 Windows 上，虚拟处理器操作系统线程，pid 字段中的值是线程 ID。 gstat -g glo 命令：打印全局多线程信息
gstat -g sch	每个虚拟处理器的信号量操作、自旋和忙等待的数量的信息。在 Windows 上，虚拟处理器操作系统线程，pid 字段中的值是线程 ID。 gstat -g sch 命令：打印 VP 信息

gstat 实用程序等待选项

使用以下 gstat 选项显示有关线程的等待条件信息。

表 18. gstat 实用程序等待选项

命令	参考
gstat -g con	打印正在等待条件的线程的 ID 。 gstat -g ath 打印线程信息。请参阅 gstat -g con 命令：打印条件和线程信息
gstat -g lmx	打印所有锁定的互斥。 gstat -g lmx 命令：打印所有锁定的互斥
gstat -g qst	打印互斥队列和条件队列的队列等待统计信息。 gstat -g qst 命令：打印互斥队列和条件队列的等待选项
gstat -g rwm	打印读/写互斥。

命令	参考
	gstat -g rwm 命令：打印读取和写入互斥
gstat -g spi	打印使用长自旋的自旋锁及其统计信息。 gstat -g spi 命令：打印使用长自旋的自旋锁
gstat -g wai	打印正在等待的线程；所有正在等待互斥或条件或正在生成的线程。 gstat -g wai 命令：打印等待队列线程队列
gstat -g wmx	打印所有使用等待者的互斥。 gstat -g wmx 命令：使用等待者打印所有互斥

其他有用的 gstat 实用程序选项

表 19. 其他有用的 gstat 实用程序选项

命令	参考
gstat -	打印 gstat 头；包含引擎版本、状态（联机、静默等等）、自初始化后经过的时间和内存占用。 gstat - 命令：打印输出头
gstat -	打印 gstat 用法选项。 gstat -- command: 打印 gstat 选项和函数
gstat *options infile*	打印使用共享内存转储（infile）作为输入的 gstat 输出。 在共享内存转储文件中运行 gstat 命令

命令	参考
gstat -a	打印 gstat 集体输出。 gstat -a 命令：打印数据库服务器整体状态的信息
gstat -c	打印服务器配置文件。 gstat -c 命令：打印 ONCONFIG 文件内容
gstat -C	打印 B-tree 索引扫描程序信息（显示有关索引清除程序的统计信息）。 gstat -C 命令：打印 B－tree 扫描程序信息
gstat -d	打印 chunk 信息。 gstat -d 命令：打印 chunk 信息
gstat -f	打印配置数据忽略的 dbspace 。 gstat -f 命令：打印受 dataskip 影响的 dbspace 信息
gstat -g all	打印诊断信息。 gstat -g all 命令：打印诊断信息
gstat -g cfg	打印配置参数当前值列表。 gstat -g cfg 命令：打印配置参数的当前值
gstat -g dbc	打印有关 dbScheduler 和 dbWorker 线程的统计信息。 gstat -g dbc 命令：打印 dbScheduler 和 dbWorker 线程的统计

命令	参考
	信息
gstat -g dis	打印数据库服务器列表，它们的状态、目录位置、位置信息和主机名。 gstat -g dis 命令：打印数据库服务器信息
gstat -g dll	打印已加载的动态链接库文件列表。 gstat -g dll 命令：打印动态链接库文件列表
gstat -g osi	打印有关操作系统资源和参数的信息。 gstat -g osi 命令：打印操作系统的信息
gstat -g pos	打印来自 $GBASEDBTDIR/etc/.infos.servernum 文件的值。（这些值被用户使用，例如：gadmin 用户共享内存连接服务器，gadmin -R 重建 $GBASEDBTDIR/etc/.infos.servernum 文件。） gstat -g pos 命令：打印文件值
gstat -g smb	打印 sbspace 的详细信息。 gstat -g smb 命令：打印 sbspace 信息
gstat -g sym	打印 oninit 实用程序的符号表信息。 gstat -g sym 命令：打印 oninit 实用程序的符号表信息
gstat -i	将 gstat 模式变更为交互。 gstat -i 命令： 开始 交互方式

命令	参考
gstat -m	打印消息日志内容。 gstat -m 命令：打印最近的系统消息日志信息
gstat -r	打印重复的 gstat 执行。 gstat -r 命令：重复打印选择的统计信息
gstat -z	将累积统计信息重新设置为零。 gstat -z 命令：清除统计信息

3.16.2 监视数据库服务器状态

要监视数据库服务器状态，请查看 gstat 命令的标题。

每当数据库服务器阻塞时，gstat 在标题行后面显示以下行：

Blocked: reason

变量 reason 可以是下列值之一。

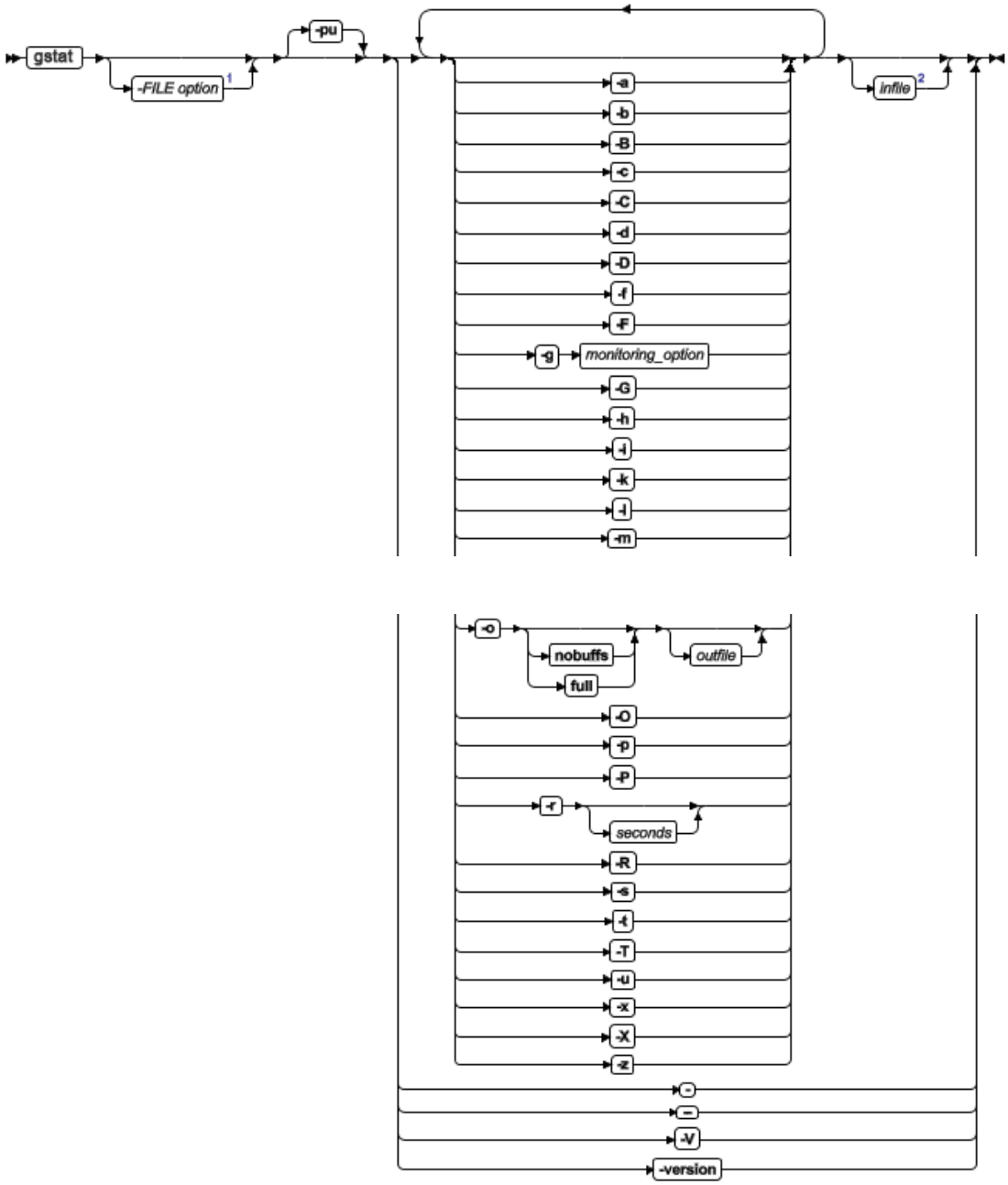
原因	描述
ADMINISTRATION	数据库处于管理方式
ARCHIVE	正在备份存储空间
ARCHIVE_EBR	由于外部备份和恢复阻塞
CHG_PLOG	正在变更物理日志时阻塞
CKPT	检查点
CKPT INP	正在进行检查点区间
DBS_DROP	正在删除 dbspace
DDR	离散数据复制
DYNAMIC_LOG	正在动态添加日志文件

原因	描述
DYNAMIC_LOG_FOR_ER	在 ER 安装中正在动态添加日志文件
FREE_LOG	正在释放日志文件
HA_CONV_STD	高可用性服务器转化为标准服务器时受阻
HA_FAILOVER	高可用性服务器故障切换时受阻
HANG_SYSTEM	数据库服务器故障
LAST_LOG_RESERVED4BACKUP	等待备份最后可用的日志文件
LBU	日志已满高水印
LOG_DROP	正在删除日志文件
LONGTX	长事务
MEDIA_FAILURE	介质故障
OVERRIDE_DOWN_SPACE	等待 dbspace 设置重写（因为 ONDBSPACEDOWN 配置参数设置为 WAIT ）

在此表中，值 CHKP INP 并不标识数据库已阻塞，而是标识在刷新缓冲池时，进程中的一个未阻塞的时间间隔 checkpoint 。CHKP INP 值出现在 gstat输出的状态行中，直到在共享内存池中的所有的页都被写入磁盘。有关设置时间间隔 checkpoint 刷新缓冲池的更多信息，请参阅 CKPTINTVL 配置参数。

3.16.3 gstat 语法

完整的 gstat 命令语法包括交互方式和如何重复执行选项的信息。



元素	用途	关键注意事项
-	只显示输出头	请参阅 gstat - 命令：打印输出头
--	显示所有 gstat 选项及其功能的列表	请参阅 gstat -- command: 打印 gstat 选项和函数 此选项不能与任何其他 gstat 选项组合
-a	解释为 gstat -cuskbtdlp。以该顺序显示输出	请参阅 gstat -a 命令：打印数据库服务器整体状态的信息
-b	显示有关当前正在使用的缓冲区的信息，包括缓冲池中常驻页的数量	请参阅 gstat -b 命令：打印正在使用的缓冲区信息

元素	用途	关键注意事项
-B	获得有关所有数据库服务器缓冲区（而不仅仅是当前正在使用的缓冲区）的信息	请参阅 gstat -B 命令：打印已使用的缓冲区信息
-c	显示 ONCONFIG 文件： ● $GBASEDBTDIR/conf/oncofig 对于 UNIX ● %GBASEDBTDIR%\etc\%ONCONFIG% 对于Windows™	请参阅 gstat -c 命令：打印 ONCONFIG 文件内容
-C	打印 B-tree 扫描程序信息	请参阅 gstat -C 命令：打印 B－tree 扫描程序信息
-d	显示每个存储空间中的 chunk 的信息	请参阅 gstat -d 命令：打印 chunk 信息.
-D	显示每个 dbspace 中前 50 个 chunk 的页读取和页写入信息	请参阅 gstat -D 命令：打印页读取和页写入信息
-f	列出当前受 DATASKIP 功能影响的 dbspace	请参阅 gstat -f 命令：打印受 dataskip 影响的 dbspace 信息
-F	显示将页清仓到磁盘上每种类型的写操作的计数	请参阅 gstat -F 命令：打印计数
-g *option*	打印监视选项	请参阅 gstat -g 监视选项
-G	打印全局事务 ID	请参阅 gstat -G 命令：打印 TP/XA 事务信息
-h	提供有关缓冲区头散列链的信	请参阅 gstat -h 命令：打印缓冲区头哈希链信息
-i	使 gstat 实用程序成为交互方式	请参阅 gstat -i 命令： 开始 交互方式.
-k	显示关于活动锁的信息	请参阅 gstat -k 命令：打印活动的锁信息
-l	显示有关物理日志和逻辑日志的信息，包括页地址	请参阅 gstat -l 命令：打印物理和逻辑日志信息
-m	显示数据库服务器消息日志中最新的 20 行	此选项的输出列出消息日志文件和完整路径名和 20 个文件条目。一个日期和时间头分隔每天的条目。时间戳记放在每天中单个条目的开始处。消息日志的名称由 ONCONFIG 文件中的 MSGPATH 指定。 请参阅 gstat -m 命令：打印最近的系统消息日志信息

元素	用途	关键注意事项
-o	将共享内存段的副本保存到 *outfile*	请参阅 gstat -o 命令：输出共享内存内容
-p	显示概要文件计数	请参阅 gstat -p 命令：打印概要文件计数
-P	显示所有分区的分区号和属于该分区的缓冲池页的拆离	请参阅 gstat -P 命令：打印分区信息
-pu	如果使用不带任何选项的 gstat 选项，那么该命令被解释为 gstat -pu（-p 选项和 -u 选项）。显示概要计数并打印用户活动的概要文件	请参阅 gstat -p 命令：打印概要文件计数 和 gstat -u 命令：打印用户活动概要文件
-r *seconds*	在每次执行之间等待指定的 *seconds* 秒之后重复伴随的 gstat 选项	请参阅 gstat -r 命令：重复打印选择的统计信息
-R	显示关于 LRU 队列、FLRU 队列和 MLRU 队列的详细信息	请参阅 gstat -R 命令：打印 LRU 、FLRU 和 MLRU 队列信息
-s	显示一般锁存器信息	请参阅 gstat -s 命令：打印锁存器信息
-t	显示活动 tblspace 的 tblspace 信息（包括驻留状态）	请参阅 gstat -t 和 gstat -T 命令：打印 tblspace 信息
-T	显示所有 tblspace 的 tblspace 信息	请参阅 gstat -t 和 gstat -T 命令：打印 tblspace 信息
-u	打印用户活动的概要文件	请参阅 gstat -u 命令：打印用户活动概要文件
-V	显示软件版本号和序列号。该选项不能与其他任何 gstat 选项组合	请参阅 获取实用程序的版本信息
-version	显示了构件版本、主机、操作系统、编号以及 GLS 版本。该选项不能与其他任何 gstat 选项组合	请参阅 获取实用程序的版本信息
-x	显示有关事务的信息	请参阅 gstat -x 命令：打印数据库服务器事务信息
-X	获取关于正在共享和等待缓冲区的线程的确切信息	请参阅 gstat -X 命令：打印线程信息
-z	将概要文件计数设置为 0	See gstat -z 命令：清除统计信息
infile	指定 gstat 命令作为所请求的源读取的文件	该文件必须包含先前存储的您使用 gstat -o 命令创建的共享内存段。 有关如何使用 gstat -o 创建*infile* 的说明，请参阅 gstat -o 命令：输出共享内存内容。 有关在资源文件中运行 gstat 的信

元素	用途	关键注意事项
		息，请参阅在共享内存转储文件中运行 gstat 命令

交互式执行

要使 gstat 实用程序处于交互方式，请使用 -i 选项。交互方式允许您输入多个选项（一个接一个）而不用退出程序。有关使用交互方式的信息，请参阅 gstat -i 命令： 开始 交互方式。

连续的 gstat 命令执行

使用组合其他 gstat 选项的 gstat -r 选项，会导致所有其他标志在每次执行之间等待指定秒数后重复执行。有关更多信息，请参阅 gstat -r 命令：重复打印选择的统计信息。

[1] 请参阅 -FILE 选项。

[2] 每项只允许出现一次，在单个 gstat 命令调用中可以指定多个选项。

3.16.4 gstat 命令：等同于 gstat -pu 命令

如果调用没有任何选项的 gstat ，那么此命令解释为 gstat -pu（ -p 选项和 -u 选项）。

语法：

3.16.5 gstat - 命令：打印输出头

所有 gstat 输出都包含一个头。gstat - 命令仅显示输出头，该命令返回的值指示数据库服务器方式。

语法:

头具有以下格式：

```
Version--Mode (Type)--(Checkpnt)--Up Uptime--Sh_mem Kbytes
```

Version

是产品名和版本号

Mode

是当前的运行方式

(Type)

如果数据库服务器使用高可用性数据复制，那么指示类型是主还是辅助

如果数据库服务器不涉及数据复制，那么此字段不出现。如果类型为主，那么显示值 P 。如果类型是辅助，那么显示值 S 。

(Checkpnt)

是 checkpoint 标志
如果设置，那么头可能在方式后面显示两个其他字段（如果时间设置是正确的话）：

(CKPT REQ)

指示用户线程已请求检查点

(CKPT INP)

指示 checkpoint 在进行中。在 checkpoint 过程中，将访问限制为只读。数据库服务器直到检查点结束才能写或更新数据

Uptime

指示数据库服务器已运行了多长时间
如果系统时间被手动修改成以前的时间，并且服务器启动时间比当前系统时间晚，那么 uptime 不可用。在此情况下，头输出 Uptime Unavailable 文本。

Sh_mem

是数据库服务器共享内存的大小（单位是千字节）
数据库服务器头的样本如下：

```
GBase 8s Version X.X.UC1--On-Line--Up 15:11:41--9216 Kbytes
```

如果数据库服务器已阻塞，那么 gstat 头的输出包含额外的行。有关该行的状态码信息，请参阅 监视数据库服务器状态。

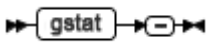
返回码

当退出 gstat 实用程序时，会显示一些有用的码。请参阅 退出 gstat 实用程序时的返回码。

3.16.6 gstat -- 命令：打印 gstat 选项和函数

使用 gstat -- 命令显示所有 gstat 选项及其功能的列表。此选项是唯一不能与任何其他标志组合的选项标记。

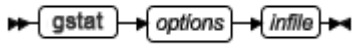
语法:



3.16.7 在共享内存转储文件中运行 gstat 命令

您可以对共享内存转储文件运行 gstat 命令。可以通过使用 gstat -o 命令显式生成共享内存转储。如果将 DUMPSHMEM 配置参数设置为 1 或 2 ，那么在声明失败时自动创建转储文件。

语法:



当使用命令行时，转储文件必须以最终参数的形式输入。以下示例打印有关名为 gstat.out 的文件中包含的共享内存转储信息，而不是尝试连接到正在运行的服务器的共享内存。

gstat -g ath gstat.out

有关如何使用 gstat -o 创建内存转储文件的说明，请参阅gstat -o 命令：输出共享内存内容。

在共享内存转储文件上交互地运行 gstat 命令

对转储文件使用 gstat -i （交互方式）运行而不是 gstat 命令。交互方式可以节省时间，这是因为它只对文件读取一次。在命令行方式，每条命令都读取文件。

以下示例读取共享内存转储文件并进入交互方式。可以在正常的交互方式中对转储执行其他 gstat 命令。

gstat -i source_file

有关交互方式的信息，请参阅 gstat -i 命令： 开始 交互方式 。

在没有创建缓冲池的共享内存转储文件中运行 gstat 命令

当您在没有创建缓冲池的共享内存转储文件（使用 gstat -o nobuffs 创建或将 DUMPSHMEM 配置参数设置为 2）中运行 gstat 命令时，它将会产生不同的输出：

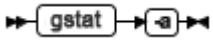
- 如果在没有创建缓存池的转储文件中运行 gstat -B ，那么输出将会在 memaddr 、nslots 和 pgflgs 列显示 0。
- 如果在没有创建缓存池的转储文件中运行 gstat -g seg ，那么输出将会显示原始的没有缓存驻留段大小。
- 如果在没有缓冲池的共享转储文件中运行 gstat -P ，那么它的输出为：

```
Nobuffs dumpfile -- this information is not available
```

3.16.8 gstat -a 命令：打印数据库服务器整体状态的信息

可以使用 gstat -a 命令显示有关数据库服务器状态的信息。该命令不仅显示有关所有 gstat 选项的信息，还显示用于初始故障排除的那些选项的信息。

语法：



3.16.9 gstat -b 命令：打印正在使用的缓冲区信息

可以使用 gstat -b 选项显示有关当前正在使用的缓冲区的信息，包括缓冲池中常驻页的总数。

语法:

可用缓冲区的最大数量以 ONCONFIG 文件中 BUFFERPOOL 配置参数的 buffers 字段进行指定。

gstat -b 命令还提供有关已修改缓冲区的数量、缓冲池中常驻页的总数。可用缓冲区的总数、可用哈希桶的数目以及以字节表示的缓冲区大小（页大小）的摘要信息。

```
123 modified, 23 resident, 2000 total, 2048 hash buckets, 2048 buffer size.
```

有关显示所有缓冲区的信息，请使用gstat -B 命令：打印已使用的缓冲区信息。

输出样本

以下是 gstat -b 命令的输出样本。有关该输出的描述，请参阅 gstat -B 命令：打印已使用的缓冲区信息。

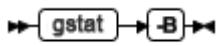
图: gstat -b 命令输出

```
Buffer pool page size: 4096

address         userthread flgs pagenum memaddr         nslots pgflgs xflgs owner waitlist
70000001097e9e8 0           c07 1:47841 7000000118e0000 10     1      0     0     0
700000010982188 0           807 1:47827 700000011939000 225    90     10    0     0
2011 modified, 50000 total, 65536 hash buckets, 4096 buffer size
```

3. 16. 10 gstat -B 命令：打印已使用的缓冲区信息

可以使用 gstat -B 选项显示那些不在可用列表中的缓冲区信息。

语法:



gstat -B 和 gstat -b 显示相同的信息。除了 gstat -b 命令只显示当前被用户线程访问的缓冲区。gstat -B 命令显示不在可用列表中的所有缓冲区。

有关在没有创建缓冲池的转储文件上运行 gstat -B 命令的更多信息，请参阅 在共享内存转储文件中运行 gstat 命令 。

输出样本

输出描述

Buffer pool page size

以字节表示的缓冲池页面大小

address

是缓冲区表中缓冲区头的地址

userthread

是访问缓冲区表的最新用户线程的地址。许多用户线程可能正在并发读取同一缓冲区

flgs

使用以下标记描述缓冲区：

0x01

已修改数据

0x02

数据

0x04

LRU

0x08

错误

pagenum

磁盘上的物理页数

memaddr

缓冲区内存地址

nslots

页中 slot 表条目的数量

该字段指示存储在该页上的行（或行的一部分）的数量

pgflgs

使用以下值（单独或组合）来描述页类型：

1

数据页

2

Tblspace 页

4

可用列表页

8

Chunk 可用列表页

9

剩余数据页

b

分区常驻 blobpage

c

Blobspace 常驻 blobpage

d

Blob chunk 可用列表位页

e

Blob chunk blob 图页

10

B-tree 节点页

20

B-tree 根节点页

40

B-tree 分支节点页

80

B-tree 叶节点页

100

逻辑日志页

200

逻辑日志的最后一页

400

逻辑日志的同步页

800

物理日志

1000

保留根页

2000

不需要物理日志

8000

带有缺省标志的 B-tree 叶

xflgs

使用以下标记来描述缓冲区访问：

0x10

共享锁

0x80

互斥锁

owner

设置 xflgs 缓冲区标记的用户线程

waitlist

正在等待访问该缓冲区的第一个用户线程的地址

有关正在等待缓冲区的所有线程的完整列表，请参阅 gstat -X 命令：打印线程信息。

3.16.11 gstat -c 命令：打印 ONCONFIG 文件内容

使用 gstat -c 命令显示 ONCONFIG 文件内容信息。

语法:

数据库服务器首先检查是否已为环境变量 ONCONFIG 指定了值。可以在数据库服务器处于任何方式（包括脱机）时使用 gstat -c 选项。

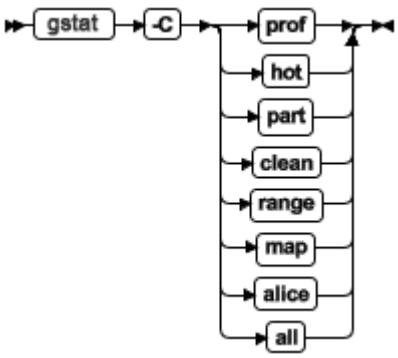
仅限 UNIX ：

在 UNIX™ 上，如果已设置了 ONCONFIG，那么 gstat -c 显示 $GBASEDBTDIR/conf/oncofig 文件的内容。如果未设置，那么缺省情况下，gstat -c 显示 $GBASEDBTDIR/conf/oncofig 的内容。

3.16.12 gstat -C 命令：打印 B－tree 扫描程序信息

使用 -C 命令显示有关 B-tree 扫描程序子系统和每个 B-tree 扫描程序线程的信息。

语法：



以下选项可用于 gstat -C 命令并且可以合并：

prof

打印系统和每个 B-tree 扫描程序线程的概要文件信息。这是缺省选项。

hot

以要清除的顺序打印 hot 列表索引键

part

打印具有索引统计信息的所有分区

clean

打印已清除或需要清除的所有分区的信息

range

打印通过使用索引范围扫描所节约的页进程数

map

为 alice 清除方法的每个索引显示了当前的 bitmap

alice

显示了 alice 清除方法选项的效率

all

打印所有 gstat -C 选项

使用 prof 选项示例输出

图: 带 prof 选项的 gstat -C 命令输出

```
Btree Cleaner Info
BT scanner profile Information
==============================
Active Threads                              1
Global Commands                       2000000   Building hot list
Number of partition scans               11003
Main Block                        0xc000000003c9dc68
BTC Admin                         0xc0000000024bc208


BTS info             id   Prio    Partnum      Key     Cmd
0xc000000003c9dee8    0   High   0x00000000     0       40  Yield N
Number of leaves pages scanned                          77
Number of leaves with deleted items                      6
Time spent cleaning (sec)                                0
Number of index compresses                               0
Number of deleted items                                113
Number of index range scans                              0
Number of index leaf scans                               0
Number of index alice scans                              2
```

使用 prof 选项输出描述

Id

BTSCANNER ID

Prio

BTSCANNER 当前的优先级

Partnum

当前正在工作线程的索引分区号

Cmd

此线程当前正在处理的命令

使用 hot 选项示例输出

图: 带有 hot 选项的 gstat -C 命令输出

```
Btree Cleaner Info

Index Hot List
==============
Current Item            5     List Created          15:29:47
List Size               4     List expires in              0 sec
Hit Threshold         500     Range Scan Threshold        -1


Partnum         Key             Hits
0x00100191        1               14 *
0x00A00022        1               13 *
0x00100191        2                8 *
0x00100150        2                7 *
```

使用 hot 选项输出的描述

Partnum

索引的分区号

Key

索引键

Hits

Hit 计数器的当前值

*****

指示在 hot 列表持续时，该分区被清除

使用 part 选项示例输出

图: gstat -C part 命令输出

```
Btree Cleaner Info


Index Statistics
================
Partnum  Key     Positions   Compress      Split
0x00100002  1           146          0          0
0x00100004  1             4          0          0
0x00100004  2            13          0          0
0x00100005  1             1          0          0
0x00100005  2             0          0          0
0x00100006  1             1          0          0
0x00100006  2             0          0          0
0x00100007  2             1          0          0
0x00100008  2             1          0          0
0x0010000a  1             0          0          0
0x0010000e  3             1          0          0
0x00100011  1             1          0          0
0x00100013  2             2          0          0
```

使用 part 选项输出描述

Partnum

索引的分区号

Key

索引键

Positions

索引的读取数

Compress

页压缩数

Split

已发生分割的数量

C

指示分区正忙于被清除

N

索引分区不再具有清除资格

使用 clean 选项示例输出

图: 带有 clean 选项的 gstat -C 命令输出

```
Btree Cleaner Info

Index Cleaned Statistics
========================
Partnum  Key      Dirty Hits  Clean Time  Pg Examined  Items Del  Pages/Sec
0x00100013  2              2           0            0          0       0.00
0x0010008b  3              1           0            0          0       0.00
0x001000c7  1              2           0            0          0       0.00
0x00100150  2              7           0            0          0       0.00
0x0010016f  2              2           0            0          0       0.00
0x00100191  1             14           0            0          0       0.00
0x00100191  2              8           0            0          0       0.00
0x00a00011  2              6           0            0          0       0.00
0x00a00013  1              0           0           24          0      24.00
0x00a00019  1              0           0          470        225     470.00
0x00a00022  1             13           0            0          0       0.00
0x00a00022  2              5           0            0          0       0.00
```

使用 clean 选项输出描述

Partnum

索引的分区号

Key

索引键

Dirty Hits

Dirty 页被扫描的次数

Clean Time

总耗时，以秒为单位

Pg Examined

被 btscanner 线程审查的页数

Items Del

从该索引移除的项的数量

Pages/Sec

每秒审查的页数

C

指示分区正在忙于被清除

N

索引分区不再具有清除的资格

示例输出

图: gstat -C range

```
Btree Cleaner Info

Cleaning Range Statistics
================================================
Partnum    Key      Low      High      Size     Saving
0x001001bc  2       36         69        96     65.6 %
0x001001be  1       16         20        48     91.7 %
0x001001cd  1        8         21        32     59.4 %
0x001001cd  2       24         25        32     96.9 %
```

输出描述

Partnum

分区号

Key

索引键

Low

范围扫描的下限

High

索引扫描的上限

Size

索引的页大小

Saving

保存相对完整扫描的时间的百分比

C

指示分区正在忙于被清除

N

索引分区不再有清除资格

输出示例

图: gstat -C map

```
Btree Cleaner Info

ALICE Bitmap of Deleted Index Items
===================================
Partnum   Key        Map
0x00100013  2 0000: 80000000 00000000
0x0010008b  3 0000: 80000000 00000000
0x001000c7  1 0000: 80000000 00000000
0x00100150  2 0000: 80000000 00000000
0x0010016f  2 0000: 80000000 00000000
0x00100191  1 0000: 80000000 00000000
0x00100191  2 0000: 80000000 00000000
0x00a00011  2 0000: 80000000 00000000
0x00a00013  1 0000: 00000000 00000000
0x00a00019  1 0000: 00000000 00000000
0x00a00022  1 0000: 80000000 00000000
0x00a00022  2 0000: 80000000 00000000
```

输出描述

Partnum

分区号

Key

索引键

Map

Alice bitmap

输出示例

图: gstat -C alice

```
Btree Cleaner Info

ALICE Cleaning Statistics
=========================

System ALICE Info: Mode =    6, Eff =    30 %, Adj =     5

Partnum    Mode BM_Sz  Used_Pg  Examined  Dirty_Pg  # I/O  Found    Eff   Adj
0x00100013    6    64       97         0         0      0      0   0.0 %    0
0x0010008b    6    64        5         0         0      0      0   0.0 %    0
0x001000c7    6    64        2         0         0      0      0   0.0 %    0
0x00100150    6    64       91         0         0      0      0   0.0 %    0
0x0010016f    6    64       91         0         0      0      0   0.0 %    0
0x00100191    6    64       26         0         0      0      0   0.0 %    0
0x00100191    6    64       26         0         0      0      0   0.0 %    0
0x001001bc    0     0       91         0         0      0      0   0.0 %    0
0x001001cd    0     0       26         0         0      0      0   0.0 %    0
0x001001cd    0     0       26         0         0      0      0   0.0 %    0
0x00a00011    6    64       91         0         0      0      0   0.0 %    0
0x00a00013    6    64       25        24         3      3      1  33.3 %    1
0x00a00019    6    64      470       470         3      3      2  66.7 %    1
0x00a00022    6    64       26         0         0      0      0   0.0 %    0
0x00a00022    6    64       26         0         0      0      0   0.0 %    0
```

输出描述

Partnum

索引的分区号

Mode

当前分区的 alice 模式

BM_Sz

向 bitmap 分配的大小

Used_Pg

索引的页大小（已使用）

Dirty_Pg

Dirty 页的数量

 # I/O

读取的页数

Found

在读取中查找到的 dirty 页数量

Eff

Bitmap 的效率

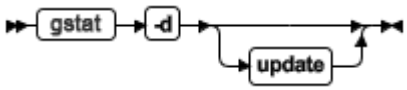
Adj

Alice 效率水平分区不充分并被调节的次数

3.16.13 gstat -d 命令：打印 chunk 信息

使用 gstat -d 命令显示每个存储空间中 chunk 的信息。

语法：



update 选项通过更新共享内存来获得可用页的准确计数。

对 sbspace 使用 gstat -d

有关使用 gstat -d 确定 sbspaces 、用户数据区域和元数据区域的大小的信息，请参阅 Monitoring sbspaces 。

对 blobspace 使用 gstat -d

如果在具有 blobspace chunk 的服务器上运行 gstat -d 命令，那么数据库服务器显示以下消息：

```
NOTE: For BLOB chunks, the number of free pages shown is out of date.
    Run 'gstat -d update' for current stats.
```

要获得 blobspace chunk 的当前统计信息，请运行 gstat -d update 命令。gstat 实用程序用每个 blobspace chunk 的可用页的准确计数更新共享空间。数据库服务器显示以下消息：

Waiting for server to update BLOB chunk statistics ...

示例输出

图: gstat -d 命令输出

```
GBase 8s Database Server Version 11.70.F       -- On-Line -- Up 00:01:27 -- 133540 Kbytes

            Dbspaces
            address   number  flags     fchunk  nchunks  pgsize  flags  owner    name
            48750028  1       0x60001   1       1        2048    N  BA  informix rootdbs
            4a0bee00  2       0x60001   2       1        2048    N  BA  informix dbspace2
            2 active, 2047 maximum

            Chunks
            address   chunk/dbs  offset  size    free   bpages flags pathname
            487501c8  1      1   0       1000000 923615         PO-B-- /dev/raw/raw1
            49f1bda0  2      2   0       5000    4972           PO-BED /work2/dbspaces/dbs2
            2 active, 32766 maximum

            NOTE: The values in the "size" and "free" columns for DBspace chunks are
            displayed in terms of "pgsize" of the DBspace to which they belong.

            Expanded chunk capacity mode: always
```

对于 dbspace 的输出描述

该显示的第一部分描述存储空间：

address

是共享内存空间表中的存储空间地址

number

是创建时指定的存储空间的唯一 ID

flags

使用十六进制值描述每个存储空间。可以累积单独的标记以显示 dbspace 的累积属性。下表描述了每个十六进制值：

表 1. 每个十六进制值的描述

标记值	描述
0x0001	允许镜像且 dbspace 是未镜像的
0x0002	允许镜像且 dbspace 是已镜像的
0x0004	Dbspace 包含禁用镜像的 chunk
0x0008	新镜像的
0x0010	Blobspace
0x0200	正在恢复空间
0x0400	空间已物理恢复
0x0800	正在恢复逻辑日志
0x2000	临时 dbspace
0x4000	正在备份 blobspace
0x8000	Sbspace
0x10000	物理或逻辑日志已更改
0x20000	Dbspace 或 chunk 表已更改
0x040000	包含大 chunk 的 blobspace
0x080000	在此 dbspace 中的 chunk 已重命名
0x00100000	仅供共享磁盘辅助服务器使用的临时 dbspace 。它是在 SD 辅助服务器中列出 SDS_TEMPDBS 配置参数的其中之一的 sbspace
0x00200000	SD 辅助服务器的临时 dbspace 。在共享磁盘辅助服务器上列出了 DBSPACETEMP 配置参数
0x00400000	该 dbspace 已被外部备份
0x00800000	Dbspace 正在进行碎片整理
0x01000000	Plogspace

fchunk

第一个 chunk 的 ID

nchunks

存储空间中的 chunk 数

pgsize

Dbspace 页的大小 （以字节为单位）

flags

使用以下字母代码描述每个存储空间：

位置 1：

标记	描述
M	已镜像
N	未镜像

位置 2：

标记	描述
X	新镜像的
P	物理恢复的，正在等待逻辑恢复
L	正在进行逻辑恢复
R	正在进行恢复
D	关闭

位置 3：

标记	描述
B	Blobspace
P	Plogspace
S	Sbspace
T	临时 dbspace
U	临时 sbspace
W	主服务器上的临时 dbspace （该标记仅在 SD 辅助服务器上显示）

位置 4：

标记	描述
B	Dbspace 具有大于 2 GB 的大 chunk

位置 5：

标记	描述

标记	描述
A	Dbspace 是自动扩展的，因为 SP_AUTOEXPAND 配置参数禁用并且 dbspace 配置了不为 0 的创建的大小或扩展大小

owner

存储空间的所有者

name

存储空间的名称

在紧随存储空间列表的行中，active 代表在数据库服务器实例中的存储空间（包括 root dbspace）的当前数量 ，maximum 代表这个数据库服务器实例总的可分配空间。

输出描述 - Chunk

gstat -d 命令输出的第二部分描述 chunk ：

address

Chunk 的地址

chk/dbs

Chunk 编号和相关联的空间编号

offset

页中文件或原始设备的偏移量

size

以 chunk 所属的 dbspace 页面大小为单位表示的 chunk 的大小

free

以 chunk 所属的 dbspace 页面大小为单位表示的 chunk 的可用页数量。值为 0 指示 chunk 中的所有空间被分配到表，但不指示表中有多少可用空间。例如：假设创建了具有 200 MB chunk 的 dbspace 并创建了具有 200 MB extent 的表。那么 free 字段的值为 0（指示 chunk 没有可用空间），然而，新的空表有 200 MB 的可用空间。

对于 blobspace ，颚化符号指示可用 blobpage 的大约数量

对于 sbspace，显示用户数据空间可用页数和用户数据空间总数

bpages

是 blobpage 中 chunk 的大小

Blobpage 可以比磁盘页大；因此，bpages 值可以小于 size 值

对于 sbspace ，是 sbpage 中 chunk 的大小。

flags

提供如下的 chunk 状态信息：

位置 1：

标记	描述
P	主
M	镜像

位置 2：

标记	描述
N	已重命名切实关闭或不一致的
O	联机
D	关闭
X	新镜像的
I	不一致的

位置 3：

标记	描述
-	Dbspace
B	Blobspace
S	Sbspace

位置 4：

标记	描述
B	Dbspace 具有大于 2 GB 的大 chunk

位置 5：

标记	描述
E	指示该 chunk 是可扩展的
-	指示该 chunk 是不可扩展的

位置 6：

标记	描述
-	为此熟的文件 chunk 不启用直接 I/O 选项或并发 I/O 选项
C	在 AIX® 上，为此熟的文件 chunk 启用并发 I/O 选项
D	为此熟的文件 chunk 启用直接 I/O 选项

pathname

物理设备的路径名

在紧随 chunk 列表的行中，active 显示活动 chunk（包括 root chunk）的数量，maximum 显示 chunk 的总数。

有关页读取和页写入的信息，请运行 gstat -D 命令。

3.16.14 gstat -D 命令：打印页读取和页写入信息

使用 gstat -D 命令显示每个空间前 50 chunk 的页读取和页写入的信息。

语法：

示例输出

图: gstat -D 命令输出

```
Dbspaces
address  number   flags      fchunk   nchunks pgsize  flags    owner    name
a40d7d8  1        0x1        1        1       2048    N        gbasedbt   rootdbs
1 active, 2047 maximum

Chunks
address  chunk/dbs  offset     page Rd  page Wr  pathname
a40d928  1      1   0          0        0        /work/11.1/dbspaces/stardbs3
1 active, 2047 maximum

Expanded chunk capacity mode: disabled
```

输出描述

gstat -D 输出几乎与 gstat -d 输出一样。以下列是 gstat -D 独有的。有关其他输出列的信息，请参阅 gstat -d 命令：打印 chunk 信息 。

page Rd

是已读取页数量

page Wr

是已写入页的数量

3.16.15 gstat -f 命令：打印受 dataskip 影响的 dbspace 信息

使用 -f 命令列出数据忽略功能当前影响的 dbspace。

语法:

-f 选项列出用 DATASKIP 配置参数和 gspaces 的 -f 选项设置的 dbspace 。执行 gstat -f 时，数据库服务器显示以下三种输出之一：

- Dataskip 对所有dbspace 都是 OFF 。

- Dataskip 对所有dbspace 都是 ON 。
- Dataskip 对以下 dbspace 都是 ON ：
  dbspace1 dbspace2...

3.16.16 gstat -F 命令：打印计数

使用 gstat -F 命令显示将页清仓到磁盘上的每种类型的写操作的计数。

语法:

示例输出

图: gstat -F 命令输出

```
Fg Writes     LRU Writes    Chunk Writes
0             330           7631

address  flusher  state  data  # LRU  Chunk  Wakeups  Idle Time
c7c8850  0        I      0     9      29     16116    16093.557
states: Exit Idle Chunk Lru
```

输出描述

可以如下解释该选项的输出：

Fg Writes

是已发生前台写入的次数

LRU Writes

是已发生 LRU 写入的次数

Chunk Writes

是已发生 chunk 写入的次数

address

是指定给该页清除程序线程的用户结构的地址

flusher

是页清除程序号

state

使用以下代码指示当前页清除程序活动：

C

Chunk 写入

E

退出

I

清除程序处于空闲状态

L

LRU 队列

退出代码指示数据库服务器正在执行关闭或页清除程序在特定时间量中还未从其写操作中返回。当操作未能在分配时间内完成时，此情况称为超时条件。数据库服务器不知道清除程序发生了什么，所以它被标记为退出。无论是两种情况中的哪一种，清除程序线程最终退出。

data

提供与 state 字段相呼应的其他信息

如果 state 为 C ，那么 data 是页清除程序正在将缓冲区写入的 chunk 编号。如果 state 为 L 那么 data 是页清除程序正从其写入的 LRU 队列。data 值显示为十进制，后跟等号，并以十六进制进行重复。

#LRU

对应于 gstat -g ath 线程 ID 输出

Chunk

已清除的 chunk 数量

Wakeups

页清除线程被唤醒的次数

Idle Time

页清除线程的空闲时间（以秒为单位）

3.16.17 gstat -g 监视选项

gstat -g 命令使用的选项只提供用于支持和调试目的。每个 gstat -g 命令只能包含这些选项中的一个。

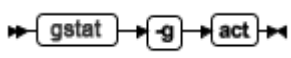
gstat -g imc 命令打印有关连接到数据库服务器的 GBase 8s MaxConnect 实例的信息。如果 GBase 8s MaxConnect 未连接数据库服务器，那么该命令显示 No MaxConnect 服务器已连接。

gstat -g nta 命令打印来自 -g ntd 、-g ntm 、-g ntt 和 -g ntu 命令的合并的网络统计信息。如果安装了 GBase 8s MaxConnect ，那么该命令打印您可以用来调整 GBase 8s MaxConnect 性能的统计信息。

gstat -g act 命令：打印活动线程

使用 gstat -g act 命令显示有关活动线程的信息。

语法:



以下是 gstat -g act 命令输出的样本。有关输出的描述，请参阅 gstat -g ath 命令：打印所有线程的信息。

示例输出

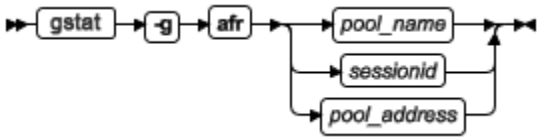
图: gstat -g act 命令输出

```
Running threads:
tid  tcb        rstcb  prty  status    vp-class    name
2    b3132d8    0      1     running   *2adm       adminthd
40   c5384d0    0      1     running   *1cpu       tlitcppoll
```

gstat -g afr 命令: 打印分配的内存分片

使用 gstat -g afr 命令显示有关指定会话或共享内存池的已分配内存分片。向每个会话分配一个共享池。

语法:



该命令需要一个额外的参数来指定池名称、会话 ID 或池地址。每个会话都分配了于会话 ID 名称相同的内存池。

pool_name 是共享内存池的名称。可运行 gstat -g mem 命令标识池的名称。

sessionid 是会话 ID 。可运行 gstat -g ses 命令标识会话 ID。

pool_address 是共享内存池的地址。可运行 gstat -g mem 命令或 gstat -g ses 命令标识池的地址。

示例输出

图: gstat -g afr 命令输出

```
Allocations for pool name global:
            addr             size       memid        fileid   location
            4b231000         3288       overhead     306      mtshpool.c:617
            4b231cd8         72         mcbmsg       1637     rldmsg.c:92
            4b231d20         160        mcbmsg       1637     rldmsg.c:92
            4b231dc0         64         osenv        2909     osenv.c:1164
            4b231e00         64         osenv        2909     osenv.c:1971
            4b231e40         64         osenv        2909     osenv.c:1164
            4b231e80         64         osenv        2909     osenv.c:1971
```

输出描述

addr (hexadecimal)

池分片的内存地址

size (decimal)

以字节表示的池分片的大小

memid (string)

池分片的内存 ID

fileid (decimal)

仅限内部使用。代码文件标识分配

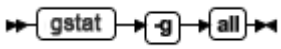
location (string)

仅限内部使用。分配代码中的行号

gstat -g all 命令：打印诊断信息

使用 gstat -g all 命令收集由 GBase Support 建议的诊断信息。为了能够正常管理，应该只使用带有个别选项的 gstat -g 命令。

语法:



gstat -g aqt 命令：打印数据集和加速查询表的信息

使用 gstat -g aqt 命令来显示有关数据集和向管理的加速查询表（AQT）的信息。

语法：

示例输出

图: gstat -g aqt 命令输出

```
AQT Dictionary Cache for database school:

            mart: school
            accelerator: DWAFINAL
            last load: 2011/07/29 07:00:39

AQT name                                FactTab #tab #matched     address

--------------------------------------------------------------------------------
aqt4d11b552-7d41-4b0c-824b-7714b6cb580a    103    1      328         0x4d187e08
aqt61498fab-3617-4c8c-ab40-fd8af4253998    103    2       42         0x4d84a448
aqtbc2da77c-bca8-4ce7-9191-8180a860da34    103    2      768         0x4d187f60
aqt88757e9d-81ee-43b4-87b2-0bf48c98fa55    103    3       15         0x4d84a190
aqta786d0dc-8e95-4de0-a1bd-773aa03a52db    103    3     1475         0x4d84a650
aqt8dd61c80-2c1c-4f0e-8f0c-91babe789f41    103    4      632         0x4d84a908

            mart: school2
            accelerator: DWAFINAL
            last load: 2011/07/29 07:01:04

AQT name                                FactTab #tab #matched     address

--------------------------------------------------------------------------------
aqt56d5aea7-32f4-44e6-8d98-02a7af37630f    103    1      845         0x4d84ac70
aqt03ec4c20-7ba8-4c3a-ae56-4134b005269d    103    2       27         0x4d95c298
aqt4ae7c2fd-5b94-423d-bc49-9ca3f5f38799    103    2     3912         0x4d84adc8
aqt5ed69a75-15e3-45cc-9892-4f5386257895    103    3       83         0x4d95c4a0
aqtdf314aa6-177d-4443-9f6d-f14ba766995a    103    3       37         0x4d95c028
```

```
aqt7e36b1f2-4646-4075-ac0b-5fdee475cd7e    103    4      518        0x4d95c758

            mart: school3
            accelerator: DWAFINAL
            last load: 2011/07/29 07:01:50

AQT name                                FactTab #tab #matched     address

--------------------------------------------------------------------------------
aqt92b36a8a-1567-4146-833c-385cd103f5d4    103    1      678        0x4d95cac0
aqt3189bec1-b6c9-417d-b969-92c687ef2e44    103    2       59        0x4d95cc18
aqt8d3b3dc8-59b6-4e34-822b-75b06b99c900    103    2     4487        0x4d90c0d8
aqt5f9c2a05-9131-4738-a929-036fcf77f65c    103    3       71        0x4d90c2e0
aqtee08ed16-6a5c-4478-ac57-fc4f99539c74    103    3      795        0x4d95ce20
aqt04d1c96a-022b-4ed7-938d-caf765bc9926    103    4      367        0x4d90c598

            18 entries
```

如果对可选 aqt_name 参数使用 AQT 名称，那么该命令打印指定的 AQT 的信息。

图: gstat -g aqt aqt_name 命令输出

```
AQT: aqt6de1afdd-f10a-45b0-93e9-0c208405fefd
   database: iwadb
   AQT tabid: 125
   Fact table: 111
   Number of times matched: 8947

   Join structure: alias(tabid)[colno,...] = alias(tabid)[colno,...] {u:unique}
            0(111)[1] = 1(110)[1] u
            1(110)[2] = 2(109)[1] u
            2(109)[5] = 3(101)[1] u
            3(101)[3] = 4(100)[1] u
            0(111)[2] = 5(106)[1] u
            5(106)[2] = 6(103)[1] u
            5(106)[3] = 7(104)[1] u
            5(106)[4] = 8(105)[1] u
            8(105)[3] = 9(101)[1] u
            9(101)[3] = 10(100)[1] u
            5(106)[5] = 11(102)[1] u
            0(111)[2,3] = 15(108)[1,2] u
            15(108)[1] = 16(106)[1] u
            16(106)[2] = 17(103)[1] u
            16(106)[3] = 18(104)[1] u
            16(106)[4] = 19(105)[1] u
            19(105)[3] = 20(101)[1] u
```

```
20(101)[3] = 21(100)[1] u
16(106)[5] = 22(102)[1] u
15(108)[2] = 23(107)[1] u
23(107)[2] = 24(101)[1] u
24(101)[3] = 25(100)[1] u
0(111)[3] = 12(107)[1] u
12(107)[2] = 13(101)[1] u
13(101)[3] = 14(100)[1] u
```

输出描述

AQTs 按照其所属数据集分组。这些组先按照加速程序名称排序，再按数据集名称排序。在数据集组中，AQTs 按以下顺序排序：Fact table tabid (FactTab)、表的数量 (#tab) 和 AQT 名称。

来自字典高速缓存的输出参考了 AQTs 的数据集。输出仅在 AQTs 加载到字典高速缓存中时显示，这种情况通常在对 AQTs 查询匹配时发生。

在数据库尝试对 AQTs 进行匹配查询之前，AQTs 在字典高速缓存中没有任何条目。gstat -g aqt 命令不会在输出中显示任何条目。当在数据库服务器启动时初始化字典高速缓存时，#matched 和 address 列得到新的值。

gstat -g aqt 命令打印以下信息：

mart

数据集的名称

accelerator

加速程序实例名称

last load

上一次数据集加载的时间戳

AQT name

系统生成的唯一的 AQT 名称

FactTab

AQT 实际数据表的 tabid

#tab

属于 AQT 表的数量

#matched

AQT 已发生的查询匹配计数

address

AQT 的内部数据库服务器内存地址

gstat -g aqt aqt_name 命令打印以下信息：

AQT

系统生成的唯一的 AQT 名称

database

AQT 所属数据库的名称

AQT tabid

构成数据库服务器中的 systables 系统目录表中的 AQT 条目的 tabid

Fact table

AQT 真实表的 tabid

Number of times matched

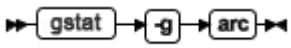
AQT 已发生的查询匹配计数

有关 AQT 信息之后是数据集的星型模式的文本表示。该文本表示显示了表的列是如何互相星型联接的。

gstat -g arc 命令：打印归档状态

使用 gstat -g arc 命令显示每个 dbspace 最后提交归档的信息，也显示正在执行归档的信息。

语法：



示例输出

图: gstat -g arc 命令输出

```
Dbspaces - Ongoing archives
number   name            Q Size Q Len  buffer partnum    size     Current-page
1        rootdbs         100    3      100    0x1001c9   0        1:128
3        datadbs01       0      0
4        datadbs02       0      0

Dbspaces - Archive Status
name               number level date             log        log-position
rootdbs            1      0     07/30/2009.09:59 28         0x320018
datadbs01          3      0     07/30/2009.09:59 28         0x320018
datadbs02          4      0     07/30/2009.09:59 28         0x320018
```

输出描述 - 正在执行归档

本节的输出代表正在归档的当前的信息。如果系统中没有归档活动，那么不会显示本节的信息。

列	描述
Number	Dbspace 编号
Name	Dbspace 名称

列	描述
Q Size	前映象队列大小。该信息主要为了 GBase 支持
Q Len	前映象队列长度。该信息主要为了 GBase 支持
Buffer	前映象缓冲区使用的页数
Partnum	前映象 bin 分区号
Size	前映象 bin 的页数
Current-page	正在执行归档的当前页

注： 前映象 bin 是在临时 dbspace 或 root dbspace 中创建的表（如果您没有临时 dbspace）。如果前映象 bin 变小，它可以扩展到其他分区，这样将会看到输出显示同一个 dbspace 具有多个 Partnum 和 Size 字段。

输出描述 - 归档状态

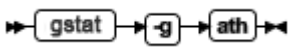
本节输出包含每个 dbspace 发生最后一次备份的信息。

列	描述
Name	Dbspace 名
Number	Dbspace 编号
Level	归档级别
Date	最后一次归档的日期和时间
Log	用来启动归档的 checkpoint 的唯一 ID （UNIQID）
Log-position	用来启动归档的 checkpoint 的日志位置（LOGPOS）

gstat -g ath 命令：打印所有线程的信息

使用 gstat -g ath 命令显示有关所有线程的信息。

语法：



示例输出

图: gstat -g ath 命令输出

```
Threads:
      tid     tcb              rstcb        prty status             vp-class      name
      2       10bbf36a8         0            1    IO Idle             3lio         lio vp 0
      3       10bc12218         0            1    IO Idle             4pio         pio vp 0
      4       10bc31218         0            1    running             5aio         aio vp 0
      5       10bc50218         0            1    IO Idle             6msc         msc vp 0
      6       10bc7f218         0            1    running             7aio         aio vp 1
      7       10bc9e540         10b231028    1    sleeping secs: 1    1cpu         main_loop()
      8       10bc12548         0            1    running             1cpu         tlitcppoll
      9       10bc317f0         0            1    sleeping forever    1cpu         tlitcplst
      10      10bc50438         10b231780    1    IO Wait             1cpu         flush_sub(0)
      11      10bc7f740         0            1    IO Idle             8aio         aio vp 2
      12      10bc7fa00         0            1    IO Idle             9aio         aio vp 3
      13      10bd56218         0            1    IO Idle            10aio         aio vp 4
      14      10bd75218         0            1    IO Idle            11aio         aio vp 5
      15      10bd94548         10b231ed8    1    sleeping forever    1cpu         aslogflush
      16      10bc7fd00         10b232630    1    sleeping secs: 34   1cpu         btscanner 0
      32      10c738ad8         10b233c38    1    sleeping secs: 1    1cpu         onmode_mon
      50      10c0db710         10b232d88    1    IO Wait             1cpu         sqlexec
```

输出描述

tid

线程 ID

tcb

线程控制 block 访问地址

rstcb

RSAM 线程控制 block 访问地址

prty

线程优先级

status

线程状态

vp-class

虚拟处理器类

name

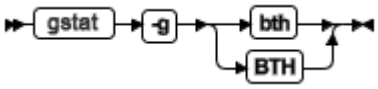
线程名称。对于参与并行存储优化操作的线程，它表示操作的名称和线程编号

- compress.number = 线程正在压缩数据
- repack.number = 线程正在重新打包数据
- uncompress.number = 线程正在解压数据
- update_ipa.number = 线程正在移除就地修改操作

gstat -g bth 和 -g BTH：打印阻塞的和正在等待的线程

使用 gstat -g bth 命令显示阻塞和正在等待线程之间的依赖关系。使用 gstat -g BTH 命令显示阻塞线程的会话和堆信息。

语法:



gstat -g bth 的示例输出

图: gstat -g bth 命令输出

```
This command attempts to identify any blocking threads.

   Highest level blocker(s)
    tid      name                session
    48       sqlexec             26

 Threads waiting on resources
  tid      name                blocking resource             blocker
   49       sqlexec             MGM                           48
   13       readahead_0         Condition (ReadAhead)          -
   50       sqlexec             Lock (0x4411e578)             49
   51       sqlexec             Lock (0x4411e578)             49
   52       sqlexec             Lock (0x4411e578)             49
   53       sqlexec             Lock (0x4411e578)             49
   57       bf_priosweep()      Condition (bp_cond)            -
   58       scan_1.0            Condition (await_MC1)          -
   59       scan_1.0            Condition (await_MC1)          -

Run 'gstat -g BTH' for more info on blockers.
```

gstat -g bth 的输出描述

tid

线程 ID

name

线程名称

session

会话 ID

blocking resource

列出的等待线程的资源类型

blocker

列出的阻塞线程的线程 ID

gstat -g BTH 的示例输出

```
Stack for thread: 48 sqlexec
base: 0x00000000461a3000
len:   69632
pc: 0x00000000017b32c3
tos: 0x00000000461b2e30
state: ready
vp: 1
```

```
        0x00000000017b32c3 (oninit) yield_processor_svp
        0x00000000017bca6c (oninit) mt_wait
        0x00000000019d4e5c (oninit) net_buf_get
        0x00000000019585bf (oninit) recvsocket
        0x00000000019d1759 (oninit) tlRecv
        0x00000000019ce62d (oninit) slSQIrecv
        0x00000000019c43ed (oninit) pfRecv
        0x00000000019b2580 (oninit) asfRecv
        0x000000000193db2a (oninit) ASF_Call
        0x0000000000c855dd (oninit) asf_recv
        0x0000000000c8573c (oninit) _iread
        0x0000000000c835cc (oninit) _igetint
        0x0000000000c72a9e (oninit) sqmain
        0x000000000194bb38 (oninit) listen_verify
        0x000000000194ab8a (oninit) spawn_thread
        0x0000000001817de3 (oninit) th_init_initgls
        0x00000000017d3135 (oninit) startup
This command attempts to identify any blocking threads.

        Highest level blocker(s)
        tid      name                session
        48       sqlexec             26

session          effective                    #RSAM    total   used    dynamic
id      user     user      tty   pid   hostname threads  memory  memory explain
26      gbasedbt  -         45   31041 mors     2        212992  186568 off

   Program : /work3/JC/VIEWS/jc_dct_phase2.view/.s/00055/80003fd351f804d3dbaccess

  tid      name     rstcb          flags    curstk   status
   48       sqlexec  448bc5e8      ---P---  4560    ready-
   58       scan_1.0 448bb478      Y------  896    cond wait  await_MC1 -

Memory pools    count 2
name         class addr          totalsize  freesize   #allocfrag #freefrag
26           V     45fcc040      208896     25616      189        16
26*00        V     462ad040      4096       808        1          1

name           free       used          name           free       used
overhead       0          6576          mtmisc         0          72
resident       0          72            scb            0          240
opentable      0          7608          filetable      0          1376
log            0          33072         temprec        0          17744
blob           0          856           keys           0          176
ralloc         0          55344         gentcb         0          2240
ostcb          0          2992          sqscb          0          21280
sql            0          11880         xchg_desc      0          1528
xchg_port      0          1144          xchg_packet    0          440
xchg_group     0          104           xchg_priv      0          336
hashfiletab    0          1144          osenv          0          2520
sqtcb          0          15872         fragman        0          1024
```

```
shmblklist     0          416            sqlj           0          72
rsam_seqscan   0          368

sqscb info
scb              sqscb          optofc   pdqpriority optcompind  directives
4499c1c0         461c1028       0        100         2           1

Sess    SQL            Current       Iso Lock       SQL  ISAM F.E.
Id      Stmt type      Database      Lvl Mode       ERR  ERR  Vers  Explain
26      SELECT         jc            CR  Not Wait   0    0    9.24  Off

Current statement name : unlcur

Current SQL statement (5) :
select * from systables,syscolumns,sysfragments

Last parsed SQL statement :
select * from systables,syscolumns,sysfragments
```

gstat -g BTH 的输出描述

tid

线程 ID

name

线程名称

session

会话 ID

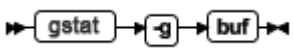
该会话信息段包含的信息与 gstat -g ses 命令输出的信息相同。请参阅 gstat -g ses 命令：打印与会话有关的信息。

剩下的信息显示了该线程的堆信息。

gstat -g buf 命令：打印缓冲池的概要文件信息

使用 gstat -g buf 命令来显示每个缓冲池的概要文件信息。

语法：



示例输出

gstat -g buf 命令的输出取决于 BUFFERPOOL 配置参数设置是否包含 memory 字段和 buffers 字段而稍有不同。输出将显示 memory 设置。buffers 设置的输出包含 max extends 和 next buffers 字段，代替了 max memory 和 next memory 字段。

图: 对于 memory 设置的 gstat -g buf 输出

```
Profile

Buffer pool page size: 2048
  dskreads   pagreads   bufreads   %cached dskwrits   pagwrits   bufwrits   %cached
```

```
1190        1773        661359      99.82    16863       83049       185805      90.92
bufwrits_sinceckpt  bufwaits   ovbuff     flushes
 11243               115         0          42
 Fg Writes     LRU Writes    Avg. LRU Time Chunk Writes  Total Mem
  0             0             nan           10883         32Mb

                                      cache
        # extends  max memory  next memory  hit ratio  last
        0          128Mb       32Mb         90         11:31:17

        Bufferpool Segments
        id segment      size     # buffs
        0  0x449f0000   32Mb     13025

        ---------------------------------

        Buffer pool page size: 8192
dskreads    pagreads    bufreads    %cached dskwrits    pagwrits    bufwrits    %cached
 0           0           11          100.00  4           16          4           0.00
bufwrits_sinceckpt  bufwaits   ovbuff     flushes
 0                   0          0          1

        Fg Writes     LRU Writes    Avg. LRU Time Chunk Writes  Total Mem
        0             0             nan           4             128Mb

        cache
        # extends  max memory  next memory  hit ratio  last
        0          1280Mb      128Mb        90         11:31:41
        Bufferpool Segments
        id segment      size     # buffs
        0  0x4928e000   128Mb    14988
        ---------------------------------
        Fast Cache Stats
        gets       hits       %hits   puts
        246854     244407     99.01   111147
```

输出描述

Buffer pool page size

缓冲池中页面的字节数

dskreads

将页面带入缓冲池的已执行你的磁盘读操作数。每次读操作读取一个或多个页面。

pagreads

从磁盘读入缓冲池的页面数

bufreads

从该缓冲池读取的页面内存映象次数

%cached

为满足高速缓存页面映象的该缓冲池读取的页面百分比（而不是执行磁盘读取）。计算为 (bufreads - dskreads) / bufreads x 100。更高的百分比说明有更好的高速缓存性能。

dskwrits

将更改的页面从缓冲池写回磁盘所执行的磁盘写入操作数。每次写入操作撰写一个或多个页面。

pagwrits

从缓冲池写入磁盘的页面数

bufwrits

写入该缓冲池的页面内存映象次数

%cached

为满足高速缓存页面映象的该缓冲池写入的页面百分比（而不是执行磁盘写入）。计算为 (bufwrits - dskwrits) / bufwrits x 100 。

bufwrits_sinceckpt

自上个 checkpiont 后页面内存映象写入该缓冲池的次数

bufwaits

在该缓冲池内线程必须等待缓冲区内锁的次数。数字越大说明在相同页面上互不兼容的锁的多个线程之间的争用越多。

ovbuff

为了创建空闲的缓冲区来读取另外一个受请求的页面而将更改了的缓冲区从该缓冲池写入磁盘的次数。如果 ovbuff 值很大，那么可能说明缓冲池还不够大，还不能容纳使用该缓冲池的应用程序所需的工作集，这可能导致性能降级。

flushes

服务器为缓冲池内所有 dirty 缓冲区执行的大清空的次数。这可能由不同的原因引起，可能是作为 checkpoint 过程的一部分而执行此操作或缓冲池在清洁的缓冲区外运行（无论 LRU 清除活动是否正常）。

Fg Writes

该缓冲池中访问缓冲区的非 I/O 清空程序线程写入磁盘的已更改缓冲区的数量。这个数字是 ovbuff 字段的超集。除了 ovbuff 字段计数的写入服务页面缺省值的次数，该值也包括为了保持数据库记录和保留页的一致性而执行的操作所做的前台写，其目的是为了保证正确的恢复。

LRU Writes

由 LRU 清除线程从该缓冲池将更改了的缓冲区写入磁盘的数量。如果缓冲池超过了指定在 lru_max_dirty 字段中 BUFFERPOOL 配置参数值或如果由于缓冲池溢出而发生前台写，那么将激活 LRU 清除程序。

Avg. LRU Time

LRU 清除程序用来清除单个 LRU 链所用的平均时间

Chunk Writes

由 Chunk 清除操作将已更改的缓冲区写入磁盘的数量。Chunk 清除程序撰写了所有在缓冲池中的某个 chunk 内的已更改的缓冲区。这项操作在需要快速清除大量缓冲区（例如 checkpoint 进程和快速恢复）的各种特殊环境中进行。

Total Mem

缓冲池的大小

extends

缓冲池被扩展的次数

max memory (memory setting)

缓冲池的目标最大大小。该缓冲池的大小可以超过该值，但是不能多于一个分段的大小。

max extends (buffers setting)

缓冲池库被扩展的至多次数。（该字段没有在示例样本中显示）

next memory (memory setting)

缓冲池的下一次要扩展的大小

next buffers (buffers setting)

为该缓冲池下一次扩展所需的缓冲区数 （该字段没有在示例样本中显示）

cache hit ratio

低于该缓冲池被扩展的读缓存的命中率

last

上次缓冲池扩展的时间

id

缓冲池段的 ID

segment

缓冲池段的内部地址

size

缓冲池段的大小

buffs

缓冲池段中缓冲区数

Fast Cache Stats

快速缓存（减少需要访问缓冲池的时间类型的缓存）的统计信息

gets

服务器在快速缓存中寻找缓冲区的次数

hits

服务器发现它正在寻找快速缓存缓冲区的次数

%hits

hits 的百分比 ， 即 hits*100/gets

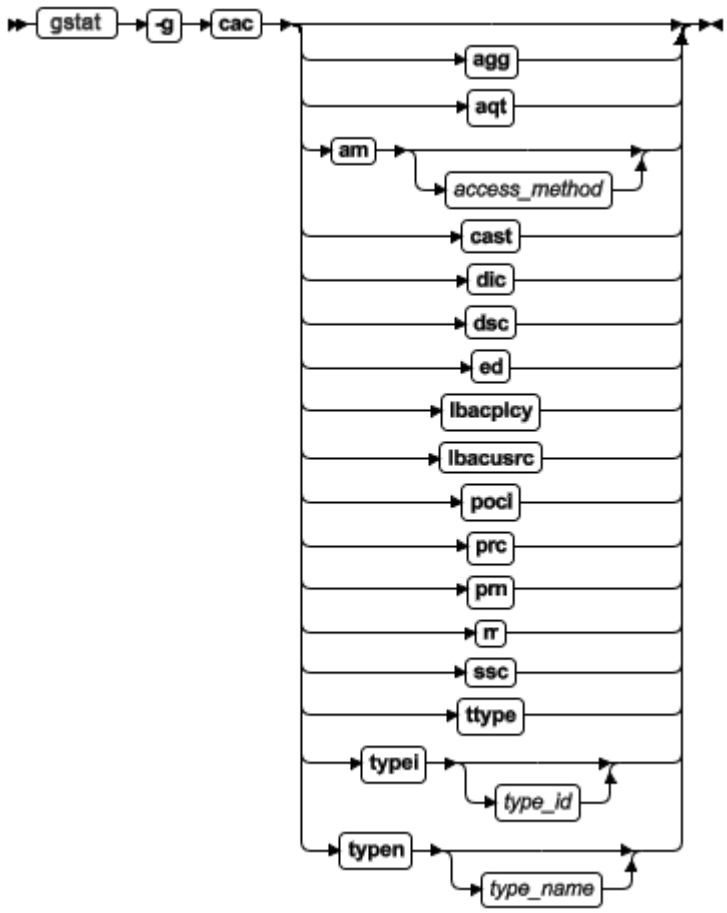
puts

服务器向快速缓存中插入缓冲区的次数

gstat -g cac 命令：打印有关缓存的信息

使用 gstat -g cac 命令来查看所有缓存或单个缓存的整体和详细信息。

语法：



可以使用不带任何选项的 gstat -g cac 命令查看所有缓存的信息。

使用以下选项来查看特定缓存的信息：

agg

打印有关聚合缓存的信息

aqt

打印有关 AQT 字典高速缓存的信息（它与 gstat -g aqt 命令输出一样的信息）。请参阅 gstat -g aqt 命令：打印数据集和加速查询表的信息

am

打印有关访问方法缓存的信息。可以看到指定访问方法的信息，包括访问方法名称。

cast

打印有关 cast 缓存的信息

dic

打印有关数据字典高速缓存的信息。也是打印 gstat -g dic 命令输出的信息。请参阅 gstat -g dic 命令：打印表信息

dsc

打印有关数据分布式缓存的信息。也是打印 gstat -g dsc 命令输出的信息。请参阅 gstat -g dsc 命令：打印分布式高速缓存信息

ed

打印有关外部命令缓存的信息

lbacplcy

打印有关 LBAC 安全政策信息缓存的信息

lbacusrc

打印有关 LBAC 证书内存缓存的信息

opci

打印有关操作类实例缓存的信息

prc

打印有关 UDR 缓存的信息。也是打印 gstat -g prc 命令输出的信息。请参阅 gstat -g prc 命令：打印使用 UDR 或 SPL 例程的会话

prn

打印有关程序名称缓存的信息

rr

打印瞬变的二级缓存的信息

ssc

打印 SQL 语句缓存信息。也是打印 gstat -g ssc 命令输出的信息。请参阅 gstat -g ssc 命令：打印出现的 SQL 语句

ttype

打印二级瞬变缓存的信息

typei

打印由其 ID 缓存的扩展类信息。可以看到指定扩展类的信息,包括扩展类 ID。

typen

打印由其名称缓存的扩展类信息。可以看到指定扩展类的信息,包括扩展类名称。

示例输出

大多数 gstat -g cac 命令的输出包含相似的格式和信息。

以下是一个 gstat -g cac lbacplcy 命令的示例输出:

```
Security Policy Info Cache:
   Number of lists          : 31
   PLCY_POOLSIZE            : 127
Security Policy Info Cache Entries:

list id   ref  drop hits       heap_ptr         item
-----------------------------------------------------------------
9    2    0    0    0          65f1b8d0         test@gbasesbt: : secpolicyid 2
15   1    0    0    0          65f1b4d0         test@gbasedbt: : secpolicyid 1
    Total number of entries  : 2
    Number of entries in use : 0
```

输出描述

大多数 gstat -g cac 命令的输出包含以下字段：

Number of lists

分布式缓存中列表数

configuration parameter name

一次可以缓存的条目数

list

分布式缓存哈希链 ID

id

该缓存条目的唯一 ID

ref

引用缓存条目的语句数

drop

当此条目被添加到缓存中后，它是否曾被删除

hits

该缓存条目被访问的次数

heap_ptr

用于存储此条目的堆地址

item name

高速缓存项的名称

Total number of entries

缓存条目的数量

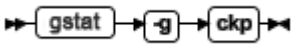
Number of entries in use

正在使用的条目数

gstat -g ckp 命令：打印 checkpoint 历史记录和配置建议

可以使用 gstat -g ckp 命令打印 checkpoint 历史记录，显示配置建议（如果检测到次优的配置）。

语法:



示例输出

图: gstat -g ckp 命令输出

```
Auto Checkpoints=On   RTO_SERVER_RESTART=60 seconds   Estimated recovery time 7 seconds
```

```
                                                    Critical Sections
Clock                          Total  Flush  Block  #      Ckpt  Wait  Long  #Dirty
Interval Time     Trigger  LSN        Time   Time   Time   Waits  Time  Time  Time  Buffers
1        18:41:36 Startup  1:f8       0.0    0.0    0.0    0      0.0   0.0   0.0   4
2        18:41:49 Admin    1:11c12cc  0.3    0.2    0.0    1      0.0   0.0   0.0   2884
3        18:42:21 Llog     8:188      2.3    2.0    2.0    1      0.0   2.0   2.0   14438
4        18:42:44*User     10:19c018  0.0    0.0    0.0    1      0.0   0.0   0.0   39
5        18:46:21 RTO      13:188     54.8   54.2   0.0    30     0.6   0.4   0.6   68232
```

```
   Physical Log    Logical Log
Dskflu   Total   Avg      Total   Avg
/Sec     Pages   /Sec     Pages  /Sec
4        3       0        1       0
2884     1966    163      4549    379
7388     318     10       65442   2181
39       536     21       20412   816
1259     210757  1033     150118  735
```

```
Max Plog   Max Llog    Max Dskflush    Avg Dskflush    Avg Dirty   Blocked
pages/sec  pages/sec   Time            pages/sec       pages/sec   Time
8796       6581        54              43975           2314         0
```

输出描述

Auto Checkpoints

标示 AUTO_CKPTS 配置参数是 on 或 off

RTO_SERVER_RESTART

显示 RTO 时间（以秒为单位）。零（0）意味着 RTO 是关闭的

Estimated recovery time ## seconds

如果数据服务器停止响应，标示评估恢复时间。该值仅在 RTO_SERVER_RESTART 被激活时出现。

Interval

Checkpoint 间隔 ID

Clock Time

Checkpoint 发生的 Clock 时间

Trigger

事件触发 checkpoint 。 星号（*）表示请求的 checkpoint 是事务阻塞的 checkpoint 。

触发器名称	描述
Admin	管理员相关任务。例如： ● 创建、删除或重命名 dbspace ● 添加或删除 chunk ● 添加或删除日志文件 ● 更改物理日志大小和位置 ● 分区上 "shrink" 操作之后 ● 打开或关闭镜像
Backup	备份相关操作。例如： ● 假备份 ● 启动归档 ● 完整的物理恢复后
CDR	第一次启动 ER 子系统或在所有复制参与者被移除后，重启 ER 子系统
CKPTINTVL	当 checkpoint 间隔过期。Checkpoint 间隔是在 onconfig 文件中 CKPTINTVL 参数指定的值
Conv/Rev	转换复原 checkpoint 。转换检查阶段后磁盘结构实际实际转换前。复原完成之后也将触发 checkpoint 。
HA	高可用性。例如： ● 新添加到高可用集群的 RSS 或 SDS 节点 ● 提升为主服务器的辅助服务器 ● 物理日志文件低的辅助服务器
HDR	高可用性数据复制。例如： ● 更改的服务器方式 ● 安装 HDR 后的启动第一次传输 ● 有潜在的主或辅助服务器上的物理日志溢出
Lightscan	在分区上关闭观察四周之前
Llog	逻辑日志资源耗尽

触发器名称	描述
LongTX	长事务。如果发现长事务未停止，那么会发起 checkpoint 停止该事务。在回滚过程中，如果在长事务终止后 checkpoint 还未准备发生，那么会在回滚阶段发起 checkpoint 。
Misc	杂项事件。例如：： ● 由于 I/O 错误 dbspace 或 chunk 关闭 ● 在回滚过程中，撤销 chunk 的附加部分时。例如：正在移动 chunk 时。 ● 聚集索引创建的过程中或变更索引为聚集索引时。
Plog	物理日志具有以下之一的条件： ● 物理日志已满 75% ● 已使用的物理日志量加上脏分区的数量已超出物理日志大小的 90%
Restore Pt	还原点。Checkpoint 在还原点开始和结束。还原点（由转换守护程序使用）是启用的 CONVERSION_GUARD 配置参数，是 RESTORE_POINT_DIR 配置参数指定的临时目录。
Recovery	在恢复过程中，快速恢复开始时
Reorg	在联机索引生成开始
RTO	维护恢复时间目标（RTO）协议。在正常操作时，在系统崩溃后重新启动的时间可能会超过 RTO_SERVER_RESTART 配置参数所设置的值。
Stamp Wrap	Checkpoint 时间戳。如果新的 checkpoint 时间戳出现在最后写入的 checkpoint 之前，那么时间戳在 checkpoint 间间隔先进先出。这样会触发另一个 checkpoint 。
Startup	数据库服务器启动时。
Uncompress	在表或分区上发生解压命令。只在表中的 checkpoint 或 没有日志记录的数据库中应用。
User	用户提交 checkpoint 请求

LSN

Checkpoint 记录在逻辑日志中的位置

Total Time

以秒表示 checkpoint 持续的总时间，从请求开始到 checkpoint 结束

Flush Time

以秒表示清除缓冲池的时间

Block Time

由于 checkpoint 被稀缺的必需资源触发而导致事务阻塞的时间（以秒表示）。例如：耗尽物理日志或环绕逻辑日志

 # Waits

由于正在等待 checkpoint 而阻塞的事务的数量

Ckpt Time

以秒表示所有事务认识到请求的 checkpoint 的时间

Wait Time

以秒表示该事务已等待 checkpoint 的平均时间

Long Time

以秒表示事务等待 checkpoint 的最长时间

 # Dirty Buffers

在 checkpoint 期间，刷新到磁盘的 dirty 缓冲区的数量

Dskflu/sec

每秒刷新的缓冲区数

Physical Log Total Pages

在 checkpoint 间隔，物理日志记录的总页数

Physical Log Avg/Sec

在 checkpoint 间隔，物理日志活动的平均率

Logical Log Total Pages

在 checkpoint 间隔，逻辑日志记录的总页数

Logical Log Avg/Sec

在 checkpoint 间隔，逻辑日志活动的平均率

Max Plog pages/sec

在 checkpoint 间隔，物理日志活动的最大速率

Max Llog pages/sec

在 checkpoint 间隔，逻辑日志活动的最大速率

Max Dskflush Time

以秒表示缓冲池刷新到磁盘的最长时间

Avg Dskflush pages/sec

缓冲池刷新到磁盘的平均速率

Avg Dirty pages/sec

在 checkpoint 之间 dirty 页的平均速率

Blocked Time

以秒表示自上次数据库服务器启动后最长阻塞时间

性能建议消息

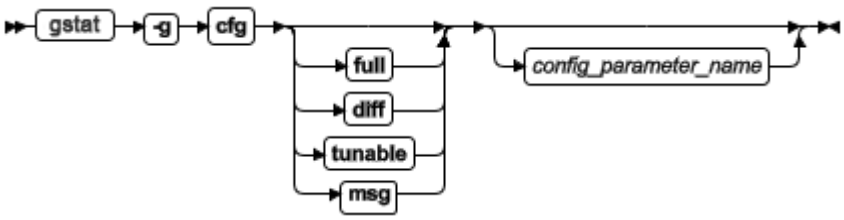
如果 GBase 8s 数据服务器检测到一个次优的配置，那么在 checkpoint 历史记录下会出现一个有关性能调整的建议消息。该性能建议也在消息日志中出现。以下是性能建议消息的示例：

```
Physical log is too small for bufferpool size. System performance may be
        less than optimal.
        Increase physical log size to at least %ldKb

        Physical log is too small for optimal performance.
        Increase the physical log size to at least $ldKb.

        Logical log space is too small for optimal performance.
        Increase the total size of the logial log space to at least %ld Kb.

        Transaction blocking has taken place. The physical log is too small.
        Please increase the size of the physical log to %ldKb

        Transaction blocking has taken place. The logical log space is too small.
        Please increase the size of the logical log space to %ldKb
```

gstat -g cfg 命令：打印配置参数的当前值

使用 gstat -g cfg 命令打印配置参数当前值列表。可以使用更多命令选项以打印更多有关配置参数的信息。

语法:



gstat -g cfg 命令具有以下格式：

gstat -g cfg 命令选项：

命令	描述
gstat -g cfg	显示配置参数当前值列表
gstat -g cfg *config_parameter_name*	只显示指定的配置参数的当前值

命令	描述
gstat -g cfg full	显示每个配置参数的所有信息，包括当前值、缺省值、onconfig 文件值和参数描述
gstat -g cfg full *config_parameter_name*	显示指定参数的所有信息
gstat -g cfg diff	显示配置参数当前值不同于其在 onconfig 文件中的永久值的信息
gstat -g cfg tunable	显示所有可调整参数的缺省值、源值和当前值。星号（*）表示您可以动态调整该配置参数
gstat -g cfg msg	显示任意消息，例如：与配置参数相关的警告或调整

示例输出

以下是 gstat -g cfg 命令显示数据库启动后 DEADLOCK_TIMEOUT 配置参数值被动态更改为 90 的输出样本：

```
id   name                 type   units   rsvd  tunable
26   DEADLOCK_TIMEOUT     INT4   Seconds       *


    min/max : 0,2147483647
    default : 60
    onconfig:
    current : 90


  Description:
   Use the DEADLOCK_TIMEOUT configuration parameter to specify the
maximum number of seconds that a database server thread can wait to acquire a lock.
ROOTNAME                rootdbs
```

以下是 gstat -g cfg diff 命令显示 TBLTBLFIRST 和 TBLTBLNEXT 配置参数的缺省值、当前值和其在 onconfig 文件中的值的输出样本：

```
id   name                    type   units   rsvd  tunable
53   TBLTBLFIRST             INT4   KB            *
        default : 500
        onconfig: 0
```

```
        current : 250


 id   name                      type   units   rsvd  tunable
 54   TBLTBLNEXT                INT4   KB            *
default : 100
onconfig: 0
current : 150
```

以下是显示有关 MSGPATH 配置参数的信息的输出样本。在这里，没有建立该配置参数的缺省值、当前值及其 onconfig 文件：

```
id   name                      type   units   rsvd  tunable
10   MSGPATH                   CHAR           *     *
default :
 onconfig: /work2/JC/online.log
 current : /work2/JC/online.log
```

以下是 gstat -g cfg msg 命令显示标识已更改值的配置参数的消息的输出样本：

```
Configuration Parameters With Messages
        name             message
        TBLTBLFIRST      Parameter's user-configured value was adjusted.
        TBLTBLNEXT       Parameter's user-configured value was adjusted.
        BUFFERPOOL       Parameter's user-configured value was adjusted.
        STACKSIZE        Parameter's user-configured value was adjusted.
        VPCLASS          Parameter's user-configured value was adjusted.
```

输出描述

name

配置参数名称

type

值的数据类型

units

值表示的单位

rsvd

标示（带星号）配置参数及其值存储在配置驻留页中

如果没有星号，那么该配置参数及其值不存储在配置驻留页中

tunable

标示（带星号）此配置参数可以动态调整。例如：使用 gadmin -wm 或 -wf 命令

如果没有星号，那么该配置参数不能被动态地修改

min/max

配置参数的最大值和最小值

default

建立到服务器中配置参数的缺省值

onconfig

配置参数值，任和值都在 onconfig.std 文件中

current

配置参数的当前值

如果动态修改了该值，其当前值是不同的。例如：使用 gadmin -wm 命令

Description

配置参数的描述

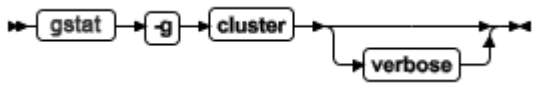
message

标识一个已更改的配置参数值的消息

gstat -g cluster 命令：打印高可用集群信息

可以使用 gstat -g cluster 命令显示有关在高可用集群环境下的服务器的信息。

语法：



gstat -g cluster 命令合并了 gstat -g dri 、gstat -g sds 和 gstat -g rss 的功能。 gstat -g cluster 命令的输出根据该命令是在主服务器或在其它任一辅助服务器上运行而稍有不同。

示例输出（主服务器）

以下 gstat -g cluster 命令的示例输出。该示例显示了命令运行在主服务器上的输出。

图: gstat -g cluster 命令输出（自主服务器上运行）

```
Primary Server:serv1
Current Log Page:16,476
Index page logging status: Enabled
Index page logging was enabled at: 2013/12/11 14:05:17


Server ACKed Log    Applied Log  Supports     Status
       (log, page)  (log, page)  Updates
serv2  16,476       16,476       Yes          SYNC(SDS),Connected,Active
serv3  16,476       16,476       Yes          ASYNC(HDR),Connected,On
serv4  16,476       16,476       Yes          ASYNC(RSS),Connected,Active
```

输出描述（主服务器）

Primary server

分配的主服务器的名称

Current log page

当前日志页的日志 ID 和页数

Index page logging status

指示索引页日志记录是否启用或禁用

Index page logging was enabled at

启用索引页日志记录的日期和时间

Server

辅助服务器的名称

ACKed Log (log, page)

最后一次应答日志传输的日志 ID 和页数

Applied Log (log, page)

最后一次应用日志传输的日志 ID 和页数

Supports Updates

显示客户端应用程序是否可以在辅助服务器上执行修改、插入和删除操作（由 UPDATABLE_SECONDARY 配置参数指定）。

Status

显示辅助服务器的连接状态

示例输出（主服务器，详细输出）

图: gstat -g cluster verbose 命令输出（自主服务器上运行）

以下是 gstat -g cluster verbose 命令的示例输出。该输出显示此命令在主服务器上运行，并带有详细的选项。

```
Primary Server:serv1
Current Log Page:16,479
Index page logging status: Enabled
Index page logging was enabled at: 2013/12/11 14:05:17



------------------------------------------------------------


server name: serv3
type: ASYNC (HDR)
control block: 0x4b673018
server status: On
connection status: Connected
Last log page sent (log id, page): 16,479
Last log page acked (log id, page); 16,479
Last log page applied (log id, page); 16,479
Approximate log page backlog: 0
SDS cycle not used
Delayed Apply Not Used
Stop Apply Not Used
```

```
        Time of last ack: 2013/12/11 14:09:12
        Supports Updates: Yes
        ------------------------------------------------------------
        server name: serv2
        type: SYNC (SDS)
        control block: 0x4c2de0b8
        server status: Active
        connection status: Connected
        Last log page sent (log id, page): 16,479
        Last log page acked (log id, page): 16,479
        Last log page applied (log id, page): 16,479
        Approximate log page backlog: 0
        SDS cycle current: 20 ACKed: 20
        Delayed Apply Not Used
        Stop Apply Not Used
        Time of last ack: 2013/12/11 14:09:13
        Supports Updates: Yes
```

输出描述（主服务器，详细输出）

Primary server

主服务器名

Current log page

当前日志页的日志 ID 和页数

Index page logging status

指示索引页日志记录是否启用或禁用

Index page logging was enabled at

启用索引页日志记录的日期和时间

Server name

辅助服务器的名称

type

显示该辅助服务器的连接是同步（SYNC）还是异步（ASYNC）。也显示辅助服务器的类型：HDR 、SDS 或 RSS

control block

线程控制 block 的在内存中的地址

server status

显示辅助服务器的当前状态

connection status

显示辅助服务器的当前网络连接状态

Last log page sent (log id, page)

最近一次主服务器发送到辅助服务器的日志页的日志 ID 和页数

Last log page acked (log id, page)

辅助服务器最近应答的日志页的日志 ID 和页数

Last log page applied (log id, page)

辅助服务器最近应用的日志页的日志 ID 和页数

Approximate log page backlog

指示还未被辅助服务器处理的日志页的大概数目

SDS cycle

指示共享磁盘辅助服务器已确认的周期数和主服务器具有先进的循环次数。由 GBase 支持用于监控主服务器和辅助服务器的协调性。

Delayed Apply

指示该辅助服务器在应用日志前是否等待特定的时间量（通过 DELAY_APPLY 配置参数指定）

Stop Apply

指示该辅助服务器是否已停止应用从主服务器接收的日志文件（通过 STOP_APPLY 配置参数指定）

Time of last ack

最后一次通知日志的日期和时间

Supports Updates

显示客户端应用程序是否可以在辅助服务器上执行修改、插入和删除操作（由 UPDATABLE_SECONDARY 配置参数指定）

示例输出（辅助服务器）

以下是 gstat -g cluster 命令的输出示例。该示例显示命令在辅助服务器上运行的输出。

图: gstat -g cluster 命令输出（自辅助服务器上运行）

```
Primary Server:serv1

Index page logging status: Enabled

Index page logging was enabled at: 2010/01/11 14:05:17

Server ACKed Log    Applied Log   Supports     Status

         (log, page)  (log, page)   Updates
   serv2  16,479       16,479        Yes     SYNC(SDS),Connected,Active
```

输出描述（辅助服务器）

Primary server

主服务器的名称

Index page logging status

指示索引页日志记录是否启用或禁用

Index page logging was enabled at

启用索引页日志记录的日期和时间

Server

辅助服务器的名称

ACKed Log (log, page)

最后一次应答日志传输的日志 ID 和页数

Applied Log (log, page)

最后一次应用日志传输的日志 ID 和页数

Supports Updates

显示客户端应用程序是否可以在辅助服务器上执行修改、插入和删除操作（由 UPDATABLE_SECONDARY 配置参数指定）

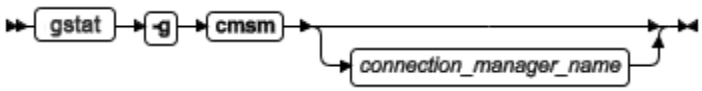
Status

显示辅助服务器的连接状态

gstat -g cmsm 命令：打印连接管理器的信息

使用 gstat -g cmsm 命令显示有关指定的连接管理器的信息，或附加在运行该命令的数据库服务器的所有连接管理器的信息。

语法:



用法

gstat -g cmsm 显示连接连接管理器的连接单元、每个连接管理器服务等级协议（SLA） 处理的连接数、SLA 定义、故障切换顺序规则、故障转移仲裁以及主服务器状态。

使用 connection_manager_name 来显示指定的连接管理器实例的信息。如果没有指定 connection_manager_name ，gstat -g cmsm 显示所有连接数据库服务器的连接管理器的信息。

示例输出 1：指定的连接管理器的输出

在以下示例中，gstat -g cmsm connection_manager_1 在主服务器的 my_cluster_1 上运行：

```
Unified Connection Manager: connection_manager_1                Hostname: my_host_1
    CLUSTER          my_cluster_1       LOCAL
    SLA           Connections    Service/Protocol    Rule
    oltp_1                 35      19910/onsoctcp    DBSERVERS=primary
    report_1               33      19810/onsoctcp    DBSERVERS=(HDR,SDS,RSS)

 Failover Arbitrator: Active Arbitrator, Primary is up ORDER=SDS,HDR,RSS PRIORITY=1
```

该命令显示 connection_manager_1 的输出。connection_manager_1 管理一个 CLUSTER 连接单元，并且是主故障转移仲裁者。

示例输出 2：高可用性集群的输出

在以下示例中，gstat -g cmsm 在主服务器的 my_cluster_2 上运行：

```
Unified Connection Manager: connection_manager_2              Hostname: my_host_2

         CLUSTER         my_cluster_2      LOCAL
         SLA          Connections   Service/Protocol   Rule
         sla_1              1535     19910/onsoctcp   DBSERVERS=primary
         sla_2              2133     19810/onsoctcp   DBSERVERS=(HDR,SDS,RSS)

         Failover Arbitrator: Active Arbitrator, Primary is up
         ORDER=SDS,HDR,RSS PRIORITY=1

         CLUSTER         my_cluster_3
         SLA          Connections   Service/Protocol   Rule
         sla_3              730     19930/onsoctcp   DBSERVERS=primary
         sla_4              901     19830/onsoctcp   DBSERVERS=(HDR,SDS,RSS)

         Failover Arbitrator: Active Arbitrator, Primary is up
         ORDER=SDS,HDR,RSS PRIORITY=1

  Unified Connection Manager: connection_manager_3               Hostname: my_host_3

         CLUSTER         my_cluster_2      LOCAL
         SLA          Connections   Service/Protocol   Rule
         sla_5              614     19920/onsoctcp   DBSERVERS=primary
         sla_6              483     19820/onsoctcp   DBSERVERS=(HDR,SDS,RSS)

         Failover Arbitrator: Failover is enabled
         ORDER=SDS,HDR,RSS PRIORITY=2

         CLUSTER         my_cluster_3
         SLA          Connections   Service/Protocol   Rule
         sla_7              678     19940/onsoctcp   DBSERVERS=primary
         sla_8              270     19840/onsoctcp   DBSERVERS=(HDR,SDS,RSS)

         Failover Arbitrator: Failover is enabled
         ORDER=SDS,HDR,RSS PRIORITY=2
```

该命令显示连接到主服务集群的两台连接管理器。connection_manager_2 和 connection_manager_3 安装在不同的主机上，并且它们一起管理两个 CLUSTER 连接单元。connection_manager_2 是这些 CLUSTER 连接单元的主故障转移仲裁者。

示例 3：复制设置的输出

在以下示例中，gstat -g cmsm 在复制服务器的 my_replicate_set_1 中运行。

```
Unified Connection Manager: connection_manager_4              Hostname: my_host_4

         REPLSET         my_replicate_set_1

         SLA          Connections   Service/Protocol   Rule

         sla_1              160     19810/onsoctcp   DBSERVERS=ANY


Unified Connection Manager: connection_manager_5              Hostname: my_host_5

         REPLSET         my_replicate_set_1

         SLA          Connections   Service/Protocol   Rule

         sla_2              240     19820/onsoctcp   DBSERVERS=ANY
```

该命令显示了两台连接到复制服务器的连接管理器。connection_manager_4 和 connection_manager_5 分别安装在不同的主机，并且它们一起管理复制服务器。

示例 4：Grid 的输出

在以下示例中，gstat -g cmsm 在 my_grid_1 的一个节点上运行。

```
Unified Connection Manager: connection_manager_6              Hostname: my_host_6
   GRID    my_grid_1
   SLA          Connections   Service/Protocol   Rule
   sla_1  456     19830/onsoctcp   DBSERVERS=(group_name_1,group_name_2) POLICY=FAILURE

Unified Connection Manager: connection_manager_7              Hostname: my_host_7
    GRID    my_grid_1
    SLA          Connections   Service/Protocol   Rule
   sla_2  785     19840/onsoctcp   DBSERVERS=(group_name_1,group_name_2)POLICY=FAILURE
```

该命令显示了两台连接到 grid 的连接管理器。它还显示了连接到该节点的两台连接管理器。connection_manager_6 和 connection_manager_7 分别安装在不同的主机，并且它们一起管理 grid 。

示例 5 ：服务器集的输出

在以下示例中，gstat -g cmsm 在一个独立服务器的服务器集上运行。

```
Unified Connection Manager: connection_manager_8          Hostname: my_host_8
   SERVERSET    server_1,server_2
   SLA          Connections   Service/Protocol   Rule
   sla_1         63     19810/onsoctcp   DBSERVERS=(server_1,server_2) POLICY=ROUNDROBIN

Unified Connection Manager: connection_manager_9          Hostname: my_host_9
   SERVERSET       server_1,server_2
   SLA          Connections   Service/Protocol   Rule
   sla_2      63     19810/onsoctcp   DBSERVERS=(server_1,server_2) POLICY=ROUNDROBIN
```

该命令显示了两台连接到此服务器集的连接管理器。connection_manager_8 和 connection_manager_9 分别安装在不同的主机，并且它们一起管理 该服务器集。。

输出描述

gstat -g cmsm 命令的输出包含每个连接管理器的章节。每一节显示连接管理器实例的名称和主机名，之后包含有关连接管理器连接的每个连接单元的子章节。

Unified Connection Manager

连接管理器实例的名称

Hostname

连接管理器的主机的名称

SLA

服务等级协议的名称，在连接管理器的配置文件中设置。

Connections

自从连接管理器启动后，每个 SLA 处理的连接数

Service/Protocol

与 SLA 相关联的端口号或服务名，随后的是连接协议类型

Rule

SLA 定义

Failover Arbitrator:

指定连接管理器是否是主故障转移仲裁者（如果主服务器是活动的，并且故障转移启用）

仅用于显示 CLUSTER 连接单元

ORDER

指定集群的故障切换顺序。仅用于显示 CLUSTER 连接单元

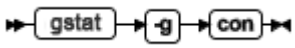
PRIORITY

指定连接管理器和集群主服务器的连接优先级。仅用于显示 CLUSTER 连接单元

gstat -g con 命令：打印条件和线程信息

使用 gstat -g con 命令显示有关条件及正在等待这些条件的线程的信息。

语法：



示例输出

图: gstat -g con 命令输出

```
Conditions with waiters:

cid      addr             name             waiter   waittime

271      c63d930          netnorm          1511     6550
```

输出描述

cid

条件标识符

addr

条件控制 block 地址

name

线程正在等待的条件的名称

waiter

正在等待条件的线程的 ID

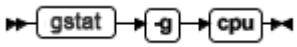
waittime

以秒表示的线程已经等待此条件的时间

gstat -g cpu: 打印运行时间统计信息

使用 gstat -g cpu 命令显示正在服务器上运行的每个线程的运行时间的统计信息。

语法:



示例输出

图: gstat -g cpu 命令输出

```
gstat -g cpu

        Thread CPU Info:
        tid     name              vp       Last Run           CPU Time     #scheds    status
        2       lio vp 0          3lio*    07/18 08:35:35       0.0000         1    IO Idle
        3       pio vp 0          4pio*    07/18 08:35:36       0.0102         2    IO Idle
        4       aio vp 0          5aio*    07/18 08:35:47       0.6876        68    IO Idle
        5       msc vp 0          6msc*    07/18 11:47:24       0.0935        14    IO Idle
        6       main_loop()       1cpu*    07/18 15:02:43       2.9365     23350    sleeping secs: 1
        7       soctcppoll        7soc*    07/18 08:35:40       0.1150         1    running
        8       soctcpio          8soc*    07/18 08:35:40       0.0037         1    running
        9       soctcplst         1cpu*    07/18 11:47:24       0.1106        10    sleeping forever
        10      soctcplst         1cpu*    07/18 08:35:40       0.0103         6    sleeping forever
        11      flush_sub(0)      1cpu*    07/18 15:02:43       0.0403     23252    sleeping secs: 1
        12      flush_sub(1)      1cpu*    07/18 15:02:43       0.0423     23169    sleeping secs: 1
        13      flush_sub(2)      1cpu*    07/18 15:02:43       0.0470     23169    sleeping secs: 1
        14      flush_sub(3)      1cpu*    07/18 15:02:43       0.0407     23169    sleeping secs: 1
        15      flush_sub(4)      1cpu*    07/18 15:02:43       0.0307     23169    sleeping secs: 1
        16      flush_sub(5)      1cpu*    07/18 15:02:43       0.0323     23169    sleeping secs: 1
        17      flush_sub(6)      1cpu*    07/18 15:02:43       0.0299     23169    sleeping secs: 1
        18      flush_sub(7)      1cpu*    07/18 15:02:43       0.0314     23169    sleeping secs: 1
        19      kaio              1cpu*    07/18 14:56:42       1.4560   2375587    IO Idle
        20      aslogflush        1cpu*    07/18 15:02:43       0.0657     23166    sleeping secs: 1
        21      btscanner_0       1cpu*    07/18 15:00:53       0.0484       784    sleeping secs: 61
        37      onmode_mon        1cpu*    07/18 15:02:43       0.3467     23165    sleeping secs: 1
        43      dbScheduler       1cpu*    07/18 14:58:14       1.6613       320    sleeping secs: 31
        44      dbWorker1         1cpu*    07/18 13:48:10       0.4264       399    sleeping forever
        45      dbWorker2         1cpu*    07/18 14:48:11       1.9346      2936    sleeping forever
        94      bf_priosweep()    1cpu*    07/18 15:01:42       0.0431        77    cond wait  bp_cond
```

输出描述

tid

线程 ID

name

线程名称

vp

正在运行线程的虚拟处理器的 ID

Last Run

线程上次运行的时间戳

CPU Time

直到现在线程运行的时间

#scheds

线程被安排运行的次数

status

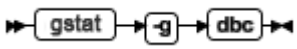
线程的状态。可能值如下：

- cond wait
- IO Idle
- join wait
- mutex wait
- ready
- sleeping
- terminated
- running
- yield

gstat -g dbc 命令：打印 dbScheduler 和 dbWorker 线程的统计信息

使用 gstat -g dbc 命令显示有关正在运行的调度任务（一些由 dbWorker 处理或计划运行，一些由 dbScheduler 线程处理。）的统计信息。

语法：



示例输出

图: gstat -g dbc 命令输出

```
Worker Thread(0)    46fa6f10
==================================
   Task:               47430c18
   Task Name:          mon_config_startup
   Task ID:            3
   Task Type:          STARTUP SENSOR
   Last Error
   Number              -310
    Message             Table (gbasedbt.mon_onconfig) already exists in database.
    Time                09/11/2007 11:41
    Task Name           mon_config_startup

Task Execution:      onconfig_save_diffs
     WORKER PROFILE
     Total Jobs Executed           10
     Sensors Executed               8
     Tasks Executed                 2
     Purge Requests                 8
     Rows Purged                    0
```

```
Worker Thread(1)    46fa6f80
==================================
    Task:               4729fc18
    Task Name:          mon_sysenv
    Task ID:            4
    Task Type:          STARTUP SENSOR
    Task Execution:     insert into mon_sysenv select 1, env_name, env_value FROM
                        sysmaster:sysenv
    WORKER PROFILE
    Total Jobs Executed            3
    Sensors Executed               2
    Tasks Executed                 1
    Purge Requests                 2
    Rows Purged                    0

Scheduler Thread    46fa6f80
==================================
    Run Queue
    Empty
    Run Queue Size        0
    Next Task             7
    Next Task Waittime    57
```

输出描述

Worker Thread

共享内存中工作线程的地址

Task

最后执行的任务的名称

Task ID

来自 sysadmin:ph_task 表中的 tk_id 列有关该任务的任务 ID

Task Type

任务的类型

Last Error

dbWorker 线程遇到的最后的错误的错误号、错误消息、时间（以秒为单位）和任务名称。它可能来自上一次执行的任务也可能来自几天前执行的任务。

Task Execution

SQL 语句或 SPL 程序或作为部分任务已执行的例程

WORKER PROFILE

dbWorker 线程概要文件数据显示总共执行的工作量、已执行的传感器的数、已执行的任务数和从通过此 dbWorker 线程执行的所有传感器的结果表中已清除的行数。

Scheduler Thread

共享内存中调度线程的地址

Run Queue

下一个要调度任务的任务 ID 。如果没有计划任务，那么该值为 Empty

Run Queue Size

正在等待通过 dbWorker 执行的任务的数量

Next Task

安排的下一个要执行的任务的任务 ID

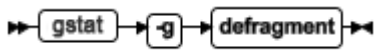
Next Task Waittime

安排的 Next Task 要执行前的秒数

gstat -g defragment 命令：打印磁盘碎片整理的分区 extent

使用 gstat -g defragment 命令显示有关磁盘碎片分区 extent 的活动请求的信息。

语法：



示例输出

图: gstat -g defragment 命令输出

```
Defrag info
id  table name                    tid  dbsnum  partnum  status               substatus  errnum
15  stores_demo:informix.stdtab2  49   2       2097155  SEARCHING_FOR_EXTENT  0          0
```

注： 该命令显示有关活动的磁盘碎片请求的信息。如果没有活动的磁盘碎片请求，那么仅返回列标题。

输出描述

id

磁盘碎片请求的 ID

table name

正在进行碎片整理的表的全名

tid

线程 ID

dbsnum

正在进行碎片整理的 dbspace 的数量

partnum

正在进行碎片整理的分区的数量

status

- SEARCHING_FOR_EXTENT
- MERGING_EXTENTS
- DEFRAG_COMPLETED
- DEFRAG_FAILED

substatus

详细的状态号，如果有的话

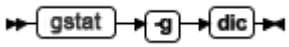
errnum

最后一次碎片整理请求返回的错误号

gstat -g dic 命令：打印表信息

使用 gstat -g dic 命令显示共享内存字典中高速缓存的每张表的一行信息。如果指定了表名，那么打印表的内部 SQL 信息。

语法:



示例输出

图: gstat -g dic 命令输出

```
Dictionary Cache:  Number of lists: 31, Maximum list size: 10
list#  size  refcnt  dirty?  heapptr        table name
---------------------------------------------------------
 1     3      1      no     14b5d890   wbe@oninit_shm:gbasedbt.t0010url
              1      no     14cbb820   wbe@oninit_shm:gbasedbt.t9051themeval
              0      no     14b63c20   wbe@oninit_shm:gbasedbt.t0060hits
 2     2      0      no     14b97420   wbe@oninit_shm:gbasedbt.t0120import
              1      no     14b6c820   wbe@oninit_shm:gbasedbt.t9110domain
 3     3      0      no     14bce020   wbe@oninit_shm:gbasedbt.t0150url
              0      no     14d3d820   contact@oninit_shm:gbasedbt.wbtags
              0      no     14c87420   wbe@oninit_shm:gbasedbt.wbtags
 4     1      0      no     14b7a420   drug@oninit_shm:abcdef.product      ..
 Total number of dictionary entries: 36
```

输出描述

list#

数据字典哈希链 ID

size

在此哈希列中的条目的数量

refcnt

当前引用高速缓存条目之一的 SQL 语句的数量

dirty?

自上次写入磁盘以来条目是否已修改

heapptr

用于存储此表的堆的地址

table name

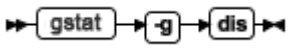
在高速缓存中的表的名称

有关 gstat -g dic 命令的更多信息，请参阅 GBase 8s 性能指南。

gstat -g dis 命令：打印数据库服务器信息

使用 gstat -g dis 命令显示数据库服务器及其状态列表和有关每个数据服务器，包括 GBASEDBTDIR 目录位置、sqlhosts 文件以及 ONCONFIG 文件的信息。可以在数据库服务器处于任何方式（包括脱机） 时使用该命令。

语法：



示例输出

图: gstat -g dis 命令输出

```
There are 2 servers found
        Server        : ol_tuxedo
        Server Number : 53
        Server Type   : IDS
        Server Status : Up
        Server Version: GBase Database Server Version 12.10.FC4G1TL
        Shared Memory : 0xa000000
        GBASEDBTDIR      :  /opt/gbs_server/home
        ONCONFIG      : #
        SQLHOSTS      : /opt/gbs_server/data/conf/sqlhosts
        Host          : avocet

        Server        : ol_9next
        Server Number : 0
        Server Type   : IDS
        Server Status : Down
        Server Version:
        Shared Memory : 0
        GBASEDBTDIR      : /local1/engines/gbs_server/home
        ONCONFIG      :
         SQLHOSTS      :
```

```
        Host            :
```

输出描述

Server

服务器名称

Server Number

服务器编号

Server Type

服务器类型

Server Status

Up 表示服务器联机，Down 表示服务器脱机

Server Version

服务器版本

Shared Memory

共享内存地址的位置

GBASEDBTDIR

在 UNIX™ 中是 $GBASEDBTDIR 目录的位置

SQLHOSTS

sqlhosts 文件的位置

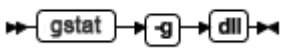
Host

服务器的主机名

gstat -g dll 命令：打印动态链接库文件列表

使用 gstat -g dll 命令显示已加载的动态链接库（DLL）文件列表。

语法：



示例输出

该输出显示仅一次每个进程组库文件的名称。该标记标示当服务器启动后，库是否已加载。

图: gstat -g dll 命令输出

```
addr            slot    vp  baseaddr        flags   filename
0x4af55310      15      1   0x2a985e3000    PM      /finance/jeffzhang/mylib.udr
0x4b6f2310              2   0x2a985e3000
0x4b71b310              3   0x2a985e3000
0x4c09f310      16      1   0x2a985e3000    M       /deptxyz/udrs/geodetic.bld
0x4c0c0310              2   0x2a985e3000
0x4c0f1310              3   0x2a985e3000
0x4c112310      17      1   0x7a138e9000            /home/gbasedbt/extend/blade.so
```

```
0x4c133310          2  0x3a421e1000
0x4c133310          3  0x3a421e1000
```

输出描述

addr

DLL 文件的地址

slot

在库表中的 Slot 编号条目

vp

虚拟处理器的 ID

baseaddr

共享库的基地址

flags

- M 标示呼叫 UDR 的线程可从从一个 CPU 虚拟处理器转移到另一个 CPU 虚拟处理器
- P 标示当数据库服务器启动后，该共享库已加载

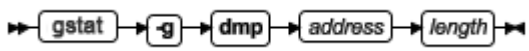
filename

DLL 文件的名称

gstat -g dmp 命令：打印原内存

使用 gstat -g dmp 命令显示有关在若干个给定的字节的给定的地址处原内存信息。

语法：



每个地址和长度必须在 gstat -g seg 输出中显示的分配内存的范围之内。指定的地址格式可以是十进制或十六进制。十六进制地址必须以 0x 开头。可以指定该地址为十进制，但是这样做需要在使用它作为命令行参数之前，将 gstat -g seg 显示的内存转换为十进制。

示例输出

图: gstat -g dmp 命令输出

```
%gstat -g dmp 0x700000011a19d48 100
address           bytes in mem
0700000011a19d48: 07000000 118e0fa8 07000000 11942b40 ........ ......+@
0700000011a19d58: 07000000 10137120 00000000 00000000 ......q ........
0700000011a19d68: 00000000 00000000 00000000 00000000 ........ ........
0700000011a19d78: 07000000 11a19d48 07000000 11a19d48 .......H .......H
0700000011a19d88: 00000000 00000000 00000000 00000000 ........ ........
0700000011a19d98 *
0700000011a19da8: 00000000 ....
```

输出描述

address

原内存的内存地址

bytes in mem

内存内容的十六进制和 ASCII 说明

该命令的输出分为三列：内存地址、以字节表示的内存十六进制值和内存的 ASCII 字节说明。内存中的字节（中间）章节显示了内存位于命令行中指定地址的前 16 个字节。第三列显示了十六进制数据的 ASCII 说明。所有没有等价的 ASCII 字符的十六进制值都会显示成间隔符。显示 ASCII 值是为了使纯文本搜寻更简便。

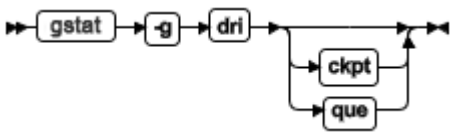
在以上显示的示例输出中，第五行数据显示零并且第六行包含星号。星号标识前一行重复的未知编号，它表示在第四行之后没有更多的数据。

gstat -g dri 命令：打印高可用性数据复制信息

使用 gstat -g dri 命令（单独使用或和 ckpt 或 que 选项一起使用），来打印有关当前服务器中的高可用性数据复制的统计信息。

可以使用 gstat -g dri 命令打印有关 HDR 服务器状态和 HRD 相关配置参数信息。

语法：



gstat -g dri 的示例输出和输出描述

图: gstat -g dri 命令输出

```
Data Replication at 0x4d676028:

   Type     State        Paired server        Last DR CKPT (id/pg)     Supports Proxy Writes
   primary   on           my_server                    4 / 5          NA

   DRINTERVAL   5
   DRTIMEOUT    30
   DRAUTO       3
   DRLOSTFOUND  /etc/dr.lostfound
   DRIDXAUTO    0
   ENCRYPT_HDR  0
   Backlog      0
   Last Send    2013/12/11 16:39:48
   Last Receive 2013/12/11 16:39:48
   Last Ping    2013/12/11 16:39:44
   Last log page applied(log id,page): 4,6
```

Type

服务器的当前类型：主服务器、辅助服务器或标准服务器

State

on 或 off

Paired server

与该服务器配对的主服务器或辅助服务器的名称

Last DR CKPT

最后 checkpoint ID 和页

Supports Proxy Writes

显示该服务器是否配置允许辅助服务器更新。Y = 支持辅助服务器更新，N = 不支持辅助服务器更新

DRINTERVAL

onconfig 文件中配置参数的值

DRTIMEOUT

onconfig 文件中配置参数的值

DRAUTO

onconfig 文件中配置参数的值

DRLOSTFOUND

onconfig 文件中配置参数的值

DRIDXAUTO

onconfig 文件中配置参数的值

ENCRYPT_HDR

onconfig 文件中配置参数的值

Backlog

在 HDR 数据复制缓冲区中还未发送到 HRD 辅助服务器中的日志页数

Last Send

最后一个消息发送至对等节点的时间

Last Receive

从对等节点接收的最后一个消息的时间

Last Ping

上次 ping 的时间

Last log page applied(log id,page)

上次应用日志的日志 ID 和页

gstat -g dri ckpt 的示例输出和输出描述

使用 gstat -g dri ckpt 命令打印在 HDR 服务器中未阻塞的 checkpoint 的信息。

图: gstat -g dri ckpt 命令输出

```
Data Replication:
          Type           State        Paired server        Last DR CKPT (id/pg)   Supports Proxy Writes
          primary        on           BB_1                        554 / 558       Y

          DRINTERVAL   30
          DRTIMEOUT    30
          DRAUTO       0
          DRLOSTFOUND  /vobs/tristarm/sqldist/etc/dr.lostfound
          DRIDXAUTO    0
            ENCRYPT_HDR  0

          DR Checkpoint processing:
          Save State              N
          Pages Saved             0
          Save Area               none
          Received log id, page   17,68
          Saved log id, page      0,0
          Drain log id, page      0,0
          Processed log id, page  17,68
          Pending checkpoints     0
```

Save State

B （buffering） 服务器正在向暂存区添加日志

D （draining） 服务器正在从暂存区移除日志

N （normal） 服务器正常工作，这意味着没有保存日志

Pages Saved

显示保存在暂存区还没应用的日志的页数

Save Area

显示暂存日志文件的位置

Received log id, page

显示从主服务器接收的最后一个日志的 ID 和页

Processed log id, page

显示排队等待恢复管道的最后一个日志的 ID 和页

Saved log id, page

显示存储在暂存区（如果暂存区状态是 B 或 D）的最后一个日志的 ID 和页

Drain log id, page

显示最后一个从暂存区移除的日志的 ID 和页

Pending checkpoints

显示暂存的还未应用的 checkpoint 数量

Pending ckpt log id, page

显示任意暂押的 checkpoint 记录的位置

gstat -g dri que 的示例输出和输出描述

使用 gstat -g dri que 命令打印几乎与 HDR 复制同步相关的信息。

图: gstat -g dri que 命令输出

```
Pending Msg to Send 1
  ACK QUEUE  5199:1256fff
  thread 0x893de6c8 (85) 5199:1258018
  thread 0x893a16b8 (83) 5199:1258048
  thread 0x89229968 (72) 5199:1258078
  thread 0x89381508 (82) 5199:12580a8
  thread 0x87e81658 (69) 5199:12580d8
  thread 0x89215968 (71) 5199:1259018
  thread 0x89336bc8 (80) 5199:1259048
  thread 0x89370018 (81) 5199:12590f8
  thread 0x892eb018 (77) 5199:125a018
  thread 0x89308018 (78) 5199:125b018
  thread 0x89290138 (75) 5199:125b048
  thread 0x893c1658 (84) 5199:125c018
  thread 0x891fe8e8 (70) 5199:125c048
  thread 0x89325018 (79) 5199:125d018
  thread 0x893ff738 (86) 5199:125d048
  thread 0x894207a8 (87) 5199:125d078

  Applied QUEUE  5199:1251018
  ----------------------------------------
```

Pending message to send

在 drprsend 线程的复制缓冲区排队的未处理的数据的数量

ACK QUEUE

日志唯一值、页编号和最近大多数 GBase_8s paged 的日志的 0xfff 值

thread

线程控制 block （TCB）的指针，括号中的数字为此线程的 ID，和由该线程执行提交的日志序列号（LSN）

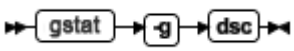
Applied QUEUE

被 HDR 辅助服务器接收的正在等待认知的提交的 LSN

gstat -g dsc 命令：打印分布式高速缓存信息

使用 gstat -g dsc 命令显示有关分布式高速缓存的信息。

语法



示例输出

图: gstat -g dsc 命令输出

```
Data Distribution Cache:
            Number of lists         : 31
            DS_POOLSIZE             : 127

            Distribution Cache Entries:

            list id   ref  drop hits       heap_ptr       distribution name
            ------------------------------------------------------------

            0    0    0    0    300        d1567a438      testdb:informix.holding.h_t_id
            0    0    0    0    6298       d0db27c38      testdb:informix.trade.th_t_id
            0    0    0    0    4499       d089d6838      testdb:informix.security.s_co_id
            0    0    0    0    4050       d086d9438      testdb:informix.customer.ca_c_id

            1    0    0    0    900        d0c01e038      testdb:informix.compet.cp_comp_id
            1    0    0    0    4049       cf99894d0      testdb:informix.customer.ca_id

            3    0    0    0    900        d15674038      testdb:informix.industry.in_name
            3    0    0    0    1794       d0db0f038      testdb:informix.news_xref.nx_id

            4    0    0    0    1800       d12663838      testdb:informix.history.hh_h_id

            5    0    0    0    900        d08cb1838      testdb:informix.watch_item.wi_id
            5    0    0    0    1800       d08b95c38      testdb:informix.address.ad_zc_code

            6    0    0    0    1050       d0b68d038      testdb:informix.ctaxrate.cx_c_id
            6    0    0    0    1050       d0b683038      testdb:informix.taxrate.tx_id

            ...

            Total number of distribution entries: 58
            Number of entries in use        : 0
```

输出描述

Number of lists

在分布式高速缓存中列表的数量

DS_POOLSIZE

每次可以高速缓存的条目的数量

list

分布式高速缓存哈希链 ID

id

哈希条目编号

ref

引用高速缓存条目的语句的数量

drop

此条目添加到高速缓存以来是否被删除

hits

高速缓存条目的访问次数

heap_ptr

用于存储此条目的堆地址

distribution name

在高速缓存中分布式的名称

Total number of distribution entries

分布式高速缓存中的条目的数量

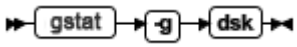
Number of entries in use

正在被使用的条目的数量

gstat -g dsk 命令：打印当前正在运行的压缩操作的进度

可以使用 gstat -g dsk 命令打印当前正在运行的压缩操作的进度，例如：压缩、重新打包和收缩。

语法：



示例输出

图: gstat -g dsk 压缩操作命令的输出

```
         Processed         Remaining Duration Remaining
Partnum   OP        Rows      Blobs  Rows      Time(s)  Time(s)  Table Name
400002    Compress  6325      1752   1497      00:00:00 00:00:00 db:sl:t1
```

图: gstat -g dsk 重新打包操作命令的输出

```
            Processed         Remaining Duration Remaining
Partnum  OP     Pass    Rows      Blobs  Rows      Time(s)  Time(s)  Table Name
400002   Repack 1       6325      1752   1497      00:00:00 00:00:00 db:sl:t1
```

输出描述

partnum

表或分片的分区号

OP

压缩操作，例如：压缩、重新打包或收缩

Pass

对于重新包装操作，1 标示第一遍读取行，2 标示第二遍读取

Processed Rows

目前为止该指定操作已处理的行数

Blobs

操作的简单大对象数

Remaining Rows

要处理的剩余行数。对于重新打包操作，是当前已过的剩余行数。

Duration Time(s)

自从该操作开始以来的持续的时间

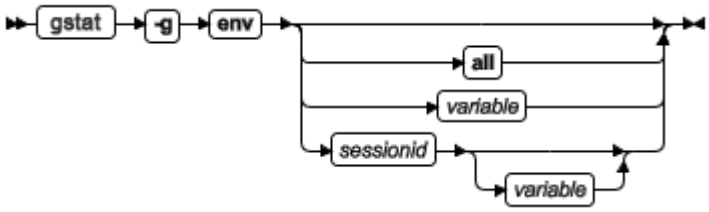
Remaining Time(s)

该操作大概剩余的时间量。对于重新包装操作，是当前已过的剩余时间。

gstat -g env 命令：打印环境变量值

可以使用 gstat -g env 命令显示数据库服务器当前使用的环境变量的值的信息。

语法：



可以指定以下调用之一。

gstat -g env	显示数据库服务器启动时变量的设置 不显示还未显式设置的变量
gstat -g env all	显示由所有会话使用的设置 此显示与 gstat -g env 和 gstat -g env*sessionid* 的输出相同（对所有当前会话重复）
gstat -g env *variable*	显示指定变量的缺省值 此 *variable* 参数使得不必将输出以管道方式运送到 grep （或某些其他实用程序）中以便在许多可能设置的变量中找到一个变量
gstat -g env *sessionid*	显示特定会话使用的设置。该显示包含以下值： ● 会话环境中的设置 ● 由数据库服务器指定，如 gstat -g env 所显示
gstat -g env *sessionid variable*	显示指定会话使用的指定变量的值 *sessionid* 和 *variable* 参数使得不必将输出以管道方式运送到 grep （或一些其他实用程序）中以便在许多可能设置的变量中找到一个变量

gstat -g env 命令显示变量的当前设置和每次在环境中设置此变量时的值的完整列表。例如：如果 PDQPRIORITY 在 .gbasedbt.rc 文件中设置为 10 ，而在 shell 环境中设置为 55 ，那么 gstat -g env 会显示两个值。

但是，如果使用 gadmin -q pdqpriority sessionid 命令更改了 PDQPRIORITY ，那么 gstat -g env 命令不会显示该会话的新值。gstat -g env 命令仅显示环境中设置的变量的值，它不显示会话正在运行时修改的值。

在以下任何情况下，您可能想要显示环境变量的值：

- 数据库服务器示例已运行了几个月，但您无法记起环境变量的设置（例如服务器语言环境设置 SERVER_LOCALE）。
- 您想要显示变量值的完整列表，以标识变量何时在多处进行了设置。
- 在这期间磁盘上的环境文件可能已更改或已丢失。
- 支持工程师想要知道特定环境变量的设置。

示例输出

下图显示了 gstat -g env 命令的输出。

图: gstat -g env 命令输出

```
Variable            Value [values-list]
DBDELIMITER         |
DBPATH              .
DBPRINT             lp -s
DBTEMP              /tmp
GBASEDBTSERVER      instance
GBASEDBTSQLHOSTS    /opt/gbs_server/data/conf/sqlhosts
GBASEDBTTERM        terminfo
GBASEDBTDIR         /opt/gbs_server/home
                    [/opt/gbs_server/home]
                    [/usr/gbasedbt]
IGNORE_UNDERFLOW    1
LANG                en_US.UTF-8
LC_COLLATE          en_US.UTF-8
LC_CTYPE            en_US.UTF-8
LC_MONETARY         en_US.UTF-8
LC_NUMERIC          en_US.UTF-8
LC_TIME             en_US.UTF-8
LKNOTIFY            yes
LOCKDOWN            no
```

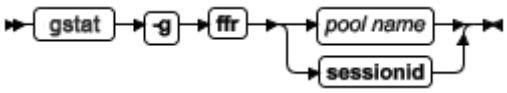
```
NODEFDAC            no
ONCONFIG            onconfig
PATH
/opt/gbs_server/home/bin:/bin:/usr/lib64/qt-3.3/bin:/usr/lo
cal/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/hom
e/gbasedbt/bin
SERVER_LOCALE       en_US.819
SHELL               /bin/bash
TERM                xterm
                    [xterm]
                    [dumb]
TERMCAP             /etc/termcap
```

gstat -g ffr 命令：打印空闲分片

可以使用 gstat -g ffr 命令显示有关特定会话或共享内存池的可用分片的信息。

该命令需要附加的参数来指定要显示内存池信息的池的名称或会话 ID 。每个会话都以其名称为会话 ID 分配到内存池中。可以使用 gstat -g mem 命令标示池的名称，使用 gstat -g ses 命令标示会话 ID 。

语法：



示例输出

图: gstat -g ffr aio 命令输出

```
Free lists for pool name aio:
addr      size      idx
165dcfa0  96        10
1659cf68  152       17
165b2f20  224       26
165c7f20  224       26
1666ec38  968       79
149f2ba0  1120      84
```

输出描述

addr (hexadecimal)

池分片的内存地址

size (decimal)

池分片大小，以字节表示

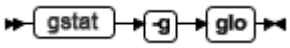
idx (decimal)

供内部使用。可用列表指针数组中的索引

gstat -g glo 命令：打印全局多线程信息

使用 gstat -g glo 命令显示有关多线程的全局信息、每个正在运行的虚拟处理器的信息以及每个虚拟处理器类的计算统计信息。该信息包括有关虚拟处理器 CPU 的使用信息、总会话数和其他多线程全局计数。

语法：



示例输出

图: gstat -g glo 命令输出

```
MT global info:
            sessions threads  vps      lngspins time
            0        23       14       0        142

            sched calls     thread switches yield 0   yield n   yield forever
            total:    85240           70451           16956     868
37319
            per sec:  0               0               0         0         0

            Virtual processor summary:
            class       vps       usercpu   syscpu    total
            cpu         1         92.12     0.59      92.71
            aio         1         0.05      0.08      0.13
            lio         1         0.00      0.00      0.00
            pio         1         0.00      0.00      0.00
            adm         1         0.00      0.01      0.01
            soc         4         0.01      0.01      0.02
            msc         1         0.00      0.00      0.00
            jvp         1         0.00      0.00      0.00
            fifo        1         0.00      0.00      0.00
            nyevp       1         0.00      0.00      0.00
            yevp        1         0.00      0.00      0.00
            total       14        92.18     0.69      92.87

            Individual virtual processors:
```

```
             vp      pid       class        usercpu    syscpu    total     Thread
Eff
             1       26328     cpu          92.12      0.59      92.71     122.65
75%
             2       26330     adm          0.00       0.01      0.01      0.00
0%
             3       26331     lio          0.00       0.00      0.00      0.00
0%
             4       26332     pio          0.00       0.00      0.00      0.00
0%
             5       26333     aio          0.05       0.08      0.13      0.28
45%
             6       26334     msc          0.00       0.00      0.00      0.19
0%
             7       26335     fifo         0.00       0.00      0.00      0.00
0%
             8       26336     nyevp        0.00       0.00      0.00      0.00
0%
             9       26337     yevp         0.00       0.00      0.00      0.00
0%
             10      26338     jvp          0.00       0.00      0.00      0.00
0%
             11      26339     soc          0.00       0.00      0.00      NA
NA
             12      26340     soc          0.00       0.00      0.00      NA
NA
             13      26341     soc          0.01       0.01      0.02      NA
NA
             14      26342     soc          0.00       0.00      0.00      NA
NA
             tot               92.18        0.69       92.87
```

输出描述

下表解释了示例输出中全局信息章节中的每个列。

表 1. 虚拟处理器摘要列的描述

列名	描述
sessions	会话数
threads	线程总数
vps	虚拟处理器的总数
lngspins	线程不得不 spin 超过 10,000 次以获得资源上的 latch 的次数
time	生成统计信息的时间。服务器启动时开始统计或通过运行 gstat -z 命令重置统计信息。
sched calls	排定呼叫的总数
thread switches	从一个线程到另一个线程切换的总次数
yield	线程收益率的统计信息（在该线程无法继续它的任务直到发生别的条件时发生）

下表解释了示例输出中 Virtual Processor Summary 章节中的每个列。

表 2. Virtual Processor Summary 列的描述

列名	描述
class	虚拟处理器的类型
vps	这个虚拟处理器类的实例的数量
usercpu	这个虚拟处理器类在 CPU 上运行所花费的总用户时间（秒）
syscpu	这个虚拟处理器类在 CPU 上运行所花费的总系统时间（秒）
total	虚拟处理器类的总 CPU 时间，它是用户时间加上系统时间的总和

下表解释了示例输出中 Individual Virtual Processor 章节中的每个列。

表 3. Individual Virtual Processor 的列描述

列名	描述
vp	虚拟处理器编号。在 Windows™ 上，是线程 ID 值。

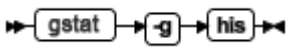
列名	描述
pid	oninit 进程的进程 ID
class	虚拟处理器类
usercpu	虚拟处理器类在 CPU 上运行的总用户时间（秒）
syscpu	虚拟处理器类在 CPU 上运行的总系统时间（秒）
total	虚拟处理器类的总 CPU 时间，它是用户时间加上系统时间的总和
Thread	线程在虚拟处理器上运行的总时间
Eff	效率。总 CPU 时间占线程在虚拟处理器上运行总时间的比率

gstat -g his 命令：打印 SQL 跟踪信息

可以使用 gstat -g his 命令显示 sysmaster 数据库中 syssqltrace 表（包括 syssqltrace 、syssqltrace_info 、syssqltrace_hvar 和 syssqltrace_itr）收集的 SQL 跟踪信息。

SQLTRACE 配置参数的 level 设置会影响由 syssqltrace 表存储和显示的 SQL 跟踪信息，并影响 gstat -g his 显示的信息。syssqltrace 表每行描述一条以前执行的 SQL 语句。缺省情况下，只有 DBSA 可以查看来自 gstat -g his 命令的 syssqltrace 信息。然而，当将 UNSECURE_ONSTAT 配置参数设置为 1 时，所有的用户都可以查看该信息。

语法：



示例输出

输出的内容依赖于跟踪的设置。

输出中的 Statement history 部分提供有关正在跟踪的当前设置的信息。

```
Statement history:

Trace Level                Low
Trace Mode              Global
Number of traces          1000
Current Stmt ID              2
Trace Buffer size         2008
Duration of buffer         293 Seconds
Trace Flags          0x00001611
Control Block        0x4c2f0028
```

下表是该输出的描述：

元素	描述
Trace Level	已跟踪的信息量。可用值为 LOW 、MED 、HIGH 和 OFF
Trace Mode	执行跟踪的类型。Global 参考系统上的所用用户。User 只参考由 SQL 管理 API 函数启用的用户。
Number of traces	跟踪的 SQL 语句的数量。该值在 onconfig 文件中进行设置，除非 SQL 管理 API 函数动态更改了 **ntraces** 参数。它的取值范围是 500 到 2147483647 。如果有 100,000 个跟踪缓冲区，每秒组织运行 1000 条 SQL 语句并要跟踪所有的语句，那么这些缓冲区在重写之前将持续 100 秒。
Current Stmt ID	当前 SQL 语句的 ID 。每条被跟踪的语句都获得一个唯一 ID 。
Trace Buffer size	每个跟踪缓冲区将获取的数据量（以字节表示）。如果将此大小设置为 2 KB ，但是有一条 12 KB 的 SQL 语句，那么该语句将会截断至少 10 KB 。更多的数据可能会被截断，它依赖于要跟踪的其它的数据。
Duration of buffer	以秒表示跟踪的数据在当前跟踪缓冲区跨度的时间量。它不是 sqltrace 功能已运行的时间。在上述示例中，**Duration of buffer** 是 293 秒，它表示跟踪的第一条和最后一条 SQL 语句之间的时间。
Trace Flags	当前设置的 SQL 跟踪标志
Control Block	该 SQL 跟踪控制 block 的内存地址

每次语句运行后，以下的信息重复显示一次。在该示例中，调用了两个变量。

```
Statement # 2:     @ 0x4c2f3028

Database:        sysmaster
Statement text:
select count(*) from systables,syscolumns where systables.tabid > ? and
systables.nrows < ?

SELECT using tables [ systables syscolumns ]
```

下表是该输出的描述：

元素	描述
Database	数据库名或该数据库的 systables 条目部分的编号
Statement text	该 SQL 语句的文本。如果该语句时存储过程，那么该语句文本将显示该过程的堆跟踪。如果声明和数字统计大于跟踪缓冲区，那么该语句文本可能会被截断。

```
Iterator/Explain
================
ID   Left  Right   Est Cost   Est Rows   Num Rows   Partnum   Type
3       0      0         17         42        146   1048579   Index Scan
4       0      0       5249       2366       2366   1048580   Seq Scan
2       3      4       5266      99372     345436         0   Nested Join
1       2      0          1          1          1         0   Group
```

下表是该输出的描述：

元素	描述
ID	SQL 迭代器 ID
Left	迭代器左侧输入的 ID
Right	迭代器右侧输入的 ID
Est Cost	本次迭代的估计成本
Est Rows	本次迭代的估计行数
Num Rows	本次迭代的实际行数
Partnum	表编号或索引分区编号
Type	操作类型

如果 SQL 语句包含一个或多个变量，并且您正在跟踪主机变量，那么输出中会显示 Host Variables 小节。

```
Host Variables
==============
1 integer    100
2 float      1000.0000000000000000
```

下表是该输出的描述：

元素	描述
Column 1	该语句中变量的位置
Column 2	该变量的数据类型
Column 3	该变量值

```
Statement information:
Sess_id  User_id  Stmt Type  Finish Time   Run Time   TX Stamp  PDQ
5        2053     SELECT     01:08:48      0.4247     340a6e9    0
```

下表是该输出的描述：

元素	描述
Sess_id	会话 ID
User_id	此操作系统用户 ID
Stmt Type	SQL 语句类型
Finish Time	当天该 SQL 语句结束的时间
Run Time	虚拟处理器或线程处理该语句所花费的总时间。例如：如果 Finish Time 是 1:15:00 ，Run Time 是 9 分钟并且启动时间不一定是 1:06:00 ，那么可能在并行语句部分调用了多个虚拟处理器或线程。
TX Stamp	该事务中记录 BEGIN WORK 语句的时间
PDQ	SQL 语句 PDQ 级别

输出中 Statement Statistics 部分提供有关该语句的特定信息。

```
Statement Statistics:
        Page        Buffer      Read        Buffer      Page        Buffer      Write
        Read        Read        % Cache     IDX Read    Write       Write       % Cache
        1285        19444       93.39       0           810         17046       95.25

        Lock        Lock        LK Wait     Log         Num         Disk        Memory
        Requests    Waits       Time (S)    Space       Sorts       Sorts       Sorts
        10603       0           0.0000      60.4 KB     0           0           0

        Total       Total       Avg         Max         Avg         I/O Wait    Avg Rows
        Executions  Time (S)    Time (S)    Time (S)    IO Wait     Time (S)    Per Sec
        1           30.8660     30.8660     30.8660     0.0141      29.2329     169.8959

        Estimated   Estimated   Actual      SQL         ISAM        Isolation   SQL
        Cost        Rows        Rows        Error       Error       Level       Memory
        102         1376        5244        0           0           CR          32608
```

元素	描述
Page Read	该 SQL 语句已从磁盘读取的页数
Buffer Read	该 SQL 语句从缓冲池读取而不是从磁盘读取页的次数
Read % Cache	应从缓冲池读取页的次数的百分比
Buffer IDX Read	尚未明确
Page Write	写入磁盘的页数
Buffer Write	修改并发送回缓冲池的页数
Write % Cache	页面写入缓冲池而不是写入磁盘的次数的百分比
Lock Requests	该语句所需的锁的总数
Lock Waits	该 SQL 语句等待锁的次数
LK Wait Time (S)	该语句等待锁的时间（以秒为单位）
Log Space	逻辑日志中该 SQL 语句已使用的存储空间量
Num Sorts	用于执行语句的排序总数
Disk Sorts	对于该 SQL 语句，对磁盘执行的排序的次数
Memory Sorts	对于该 SQL 语句，对内存执行的排序的次数

元素	描述
Total Executions	该语句已执行的总次数，或者该游标重用的次数
Total Time (S)	执行该语句的总时间（以秒表示）
Avg Time (S)	执行该语句的平均时间（以秒表示）
Max Time (S)	运行 SQL 语句的总时间（以秒为单位），不包括应用程序使用的任何时间。如果您准备一个查询然后查询 5 次，该查询每次将一个跟踪添加到跟踪缓冲区中。Max Time 是任意执行所花的最长时间
Avg IO Wait	语句等待 I/O 的时间量，不包括任何异步 I/O
I/O Wait Time (S)	语句等待 I/O 的时间量，不包括任何异步 I/O（以秒为单位）
Avg Rows Per Sec	该语句每秒产生的平均行数
Estimated Cost	与 SQL 语句关联的查询优化程序的代价
Estimated Rows	返回的估计行数，由语句的优化程序估计
Actual Rows	该语句返回的行数
SQL Error	SQL 错误编号
ISAM Error	RSAM 或 ISAM 错误编号
Isolation Level	该语句运行时使用的隔离级别
SQL Memory	该 SQL 语句需要的字节数

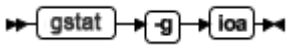
有关 syssqltrace 系统监视接口表的完整结构，请参阅 syssqltrace。

有关 SQLTRACE 配置参数的详细设置信息，请参阅 SQLTRACE 配置参数。

gstat -g ioa 命令：打印合并的 gstat -g 信息

可以使用 gstat -g ioa 命令显示来自 gstat -g iob 、gstat -g iof 、gstat -g ioq 和 gstat -g iov 命令的组合信息。

语法



示例输出

```
AIO global info:
         9 aio classes
         9 open files
         64 max global files

         AIO I/O queues:
         q name/id    len maxlen totalops  dskread dskwrite  dskcopy
         fifo    0    0      0        0        0        0        0
         drda_dbg 0    0      0        0        0        0        0
         sqli_dbg 0    0      0        0        0        0        0
         adt     0    0      0        0        0        0        0
         msc     0    0      1      231        0        0        0
         aio     0    0      5    13069    10895        0        0
         pio     0    0      1     1580        0     1580        0
         lio     0    0      1    37900        0    37900        0
         gfd     3    0     87    42115    15806    26309        0
         gfd     4    0      4        5        1        4        0
         gfd     5    0     12       35       22       13        0
         gfd     6    0     11       33       21       12        0
         gfd     7    0      1        4        3        1        0
         gfd     8    0      1        4        3        1        0

         AIO I/O vps:
         class/vp/id s  io/s totalops dskread dskwrite dskcopy wakeups io/wup
errors tempops
         fifo  7  0 i   0.0        0       0        0       0       1  0.0
0        0
```

```
          msc  6  0 i    0.0      231        0        0        0      221   1.0
0       231
          aio  5  0 i    0.0    39285    26358    10793        0    37531   1.0
0         5
          aio  9  1 i    0.0     5770     3795     1944        0     5926   1.0
0         0
          aio 10  2 i    0.0     2308      717     1585        0     1953   1.2
0         0
          aio 11  3 i    0.0     1463      166     1295        0     1166   1.3
0         0
          aio 12  4 i    0.0     1219       46     1172        0      943   1.3
0         0
          aio 13  5 i    0.0     1041       34     1007        0      805   1.3
0         0
          aio 15  6 i    0.0      425        2      423        0      438   1.0
0         0
          aio 16  7 i    0.0      342        5      337        0      395   0.9
0         0
          pio  4  0 i    0.0     1580        0     1580        0     1581   1.0
0      1580
          lio  3  0 i    0.0    37900        0    37900        0    29940   1.3
0     37900

          AIO global files:
          gfd pathname         bytes read     page reads   bytes write    page
writes io/s
          3   ./rootdbs        85456896       41727       207394816
101267       572.9
          op type     count          avg. time
          seeks       0              N/A
          reads       13975          0.0015
          writes      51815          0.0018
          kaio_reads  0              N/A
          kaio_writes 0              N/A
```

```
        4    tempsbs.chunk    2048          1          8192          4
113.6
        op type      count          avg. time
        seeks        0              N/A
        reads        1              0.0131
        writes       3              0.0074
        kaio_reads   0              N/A
        kaio_writes  0              N/A

        5    sbs1.chunk       45056         22         26624         13
173.4
        op type      count          avg. time
        seeks        0              N/A
        reads        22             0.0063
        writes       6              0.0038
        kaio_reads   0              N/A
        kaio_writes  0              N/A

        6    sbs2.chunk       43008         21         24576         12
76.1
        op type      count          avg. time
        seeks        0              N/A
        reads        21             0.0148
        writes       6              0.0072
        kaio_reads   0              N/A
        kaio_writes  0              N/A

        7    qhdr.chunk       6144          3          2048          1
550.5
        op type      count          avg. time
        seeks        0              N/A
        reads        3              0.0019
        writes       1              0.0016
```

```
          kaio_reads  0              N/A
          kaio_writes 0              N/A

          8   ./dbs1           6144           3           2048           1
403.0
          op type     count          avg. time
          seeks       0              N/A
          reads       3              0.0027
          writes      1              0.0018
          kaio_reads  0              N/A
          kaio_writes 0              N/A


          AIO big buffer usage summary:
          class                 reads
writes
          pages    ops  pgs/op  holes  hl-ops hls/op      pages    ops
pgs/op
          fifo      0       0   0.00      0       0   0.00           0
0    0.00
          drda_dbg  0       0   0.00      0       0   0.00           0
0    0.00
          sqli_dbg  0       0   0.00      0       0   0.00           0
0    0.00
          kio       0       0   0.00      0       0   0.00           0      0
0.00
          adt       0       0   0.00      0       0   0.00           0      0
0.00
          msc       0       0   0.00      0       0   0.00           0      0
0.00
          aio 228709   20228  11.31   1005     203   4.95      213272  18556
11.49
          pio       0       0   0.00      0       0   0.00       19672   1580
12.45
```

```
          lio       0        0    0.00       0        0    0.00        55287   37900
1.46
```

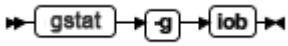
输出描述

有关每个输出列的描述，请分别参阅 gstat -g iob 命令：打印大缓冲区的使用摘要 、gstat -g ioq 命令：打印 I/O 队列信息 和 gstat -g iov 命令：打印 AIO VP 统计信息 命令。

gstat -g iob 命令：打印大缓冲区的使用摘要

使用 gstat -g iob 命令显示大缓冲区的使用摘要。

语法：



示例输出

图: gstat -g iob 命令输出

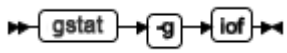
```
AIO big buffer usage summary:
                 reads                                    writes
       pages     ops  pgs/op  holes  hl-ops hls/op      pages     ops  pgs/op
fifo       0       0    0.00      0       0   0.00          0       0    0.00
 kio       0       0    0.00      0       0   0.00          0       0    0.00
 adt       0       0    0.00      0       0   0.00          0       0    0.00
 msc       0       0    0.00      0       0   0.00          0       0    0.00
 aio       0       0    0.00      0       0   0.00        607     607    1.00
 pio       0       0    0.00      0       0   0.00          0       0    0.00
 lio       0       0    0.00      0       0   0.00          0       0    0.00
```

gstat -g iof 命令：打印 asynchronous I/O 统计信息

使用 gstat -g iof 命令显示按 chunk 或文件的 asynchronous I/O 统计信息。

这个命令与 gstat -D 命令相似，除了 gstat -g iof 也显示非 chunk 文件的信息。它包含临时文件和排序工作文件的信息。

语法：



示例输出

图: gstat -g iof 命令输出

```
AIO global files:
           gfd pathname      bytes read   page reads   bytes write   page writes  io/s
           3   rootdbs       1918976      937          145061888     70831        36.5

           op type     count         avg. time
           seeks       0             N/A
           reads       937           0.0010
           writes      4088          0.0335
           kaio_reads  0             N/A
           kaio_writes 0             N/A
```

输出描述

gfd

该 chunk 或文件的全局文件描述符编号

pathname

Chunk 或文件的路径名

bytes read

已经对 chunk 或文件执行的读取的字节数

page reads

已经对 chunk 或文件执行的页读取数

bytes write

已经对 chunk 或文件执行写入的字节数

page writes

已经对 chunk 或文件执行的页写入数

io/s

每妙执行的 I/O 操作数，该值代表 chunk 或文件的 I/O 性能

op type

操作类型

count

此操作发生的次数

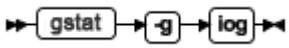
avg time

操作结束所花的平均时间

gstat -g iog 命令：打印 AIO 全局信息

使用 gstat -g iog 命令显示有关 AIO 的全局信息。

语法：



示例输出

图: gstat -g iog 命令输出

```
AIO global info:
   8 aio es
   5 open files
  64 max global files
```

gstat -g ioq 命令：打印 I/O 队列信息

可以使用 gstat -g ioq 命令显示关于由 I/O 队列执行的操作的数量和类型的统计信息。

语法：

如果给定一个 queue_name ，那么只显示具有该名称的队列。如果未给定 queue_name ，那么显示所有队列的信息。

示例输出

图: gstat -g ioq 命令输出

```
AIO I/O queues:
            q name/id    len maxlen totalops  dskread dskwrite  dskcopy
            sqli_dbg   0      0      0        0        0        0        0
            fifo   0      0      0        0        0        0        0
            adt    0      0      0        0        0        0        0
            msc    0      0      1      537        0        0        0
            aio    0      0      3     6537      238     5777        0
            pio    0      0      2     1103        0     1102        0
            lio    0      0      2    11795        0    11794        0
            gfd    3      0     17    17489     1526    15963        0
            gfd    4      0     17    18347     2384    15963        0
            gfd    5      0     16      220       41      179        0
            gfd    6      0      4        4        0        4        0
            gfd    7      0      4        4        0        4        0
            gfd    8      0      4        4        0        4        0
            gfd    9      0      9       54       24       30        0
            gfd   10      0     16      149       40      109        0
            gfd   11      0     16      621      128      493        0
            gfd   12      0     16     1953     1146      807        0
            gfd   13      0     16      409       71      338        0
            gfd   14      0     16      378       60      318        0
```

输出描述

q name/id

I/O 队列的名称和编号。名称说明队列的类型。编号用来区分具有相同名称的队列。

以下是可能的队列名称以及每个类型的队列处理的对象的列表：

sqli_dbg

处理 GBase 技术支持的 SQL 接口调试功能的 I/O

fifo

处理 FIFO VPs 的 I/O

adt

处理审计 I/O

msc

处理杂项 I/O

aio

处理 GBase 8s 异步 I/O

kio

处理内核 AIO

pio

处理物理日志记录 I/O

lio

处理逻辑日志记录 I/O

gfd

全局文件描述符 - 为每个主 chunk 和镜像 chunk 分配单独的全局文件描述符。每个 gfd 队列的使用取决于 kaio 是否开启以及关联 chunk 是格式化的还是原始的

len

在队列中暂挂 I/O 请求的数量

maxlen

队列中同时存在的 I/O 请求的最大数量

totalops

队列中已经完成的 I/O 操作的总数

dskread

队列已完成的读操作的总数

dskwrite

队列已完成的写操作的总数

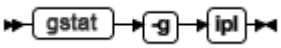
dskcopy

队列已完成的复制操作的总数

gstat -g ipl 命令：打印索引页日志状态信息

可以使用 gstat -g ipl 命令显示索引页日志的状态信息。

语法：



示例输出

图: gstat -g ipl 命令输出

```
Index page logging status: Enabled
            Index page logging was enabled at: 2008/12/20 16:01:02
```

输出描述

Index page logging status

索引页日志记录状态：已启用或禁用

Index page logging was enabled at

启用索引页日志记录的日期和时间

gstat -g iov 命令：打印 AIO VP 统计信息

可以使用 gstat -g iov 命令显示每个虚拟处理器的异步 I/O 统计信息。

语法：

gstat -g iov

示例输出

图: gstat -g iov 命令输出

```
AIO I/O vps:
class/vp/id  s  io/s  totalops  dskread dskwrite  dskcopy  wakeups  io/wup  errors tempops
fifo   7  0  i  0.0         0        0        0        0        1     0.0       0       0
msc    6  0  i  0.1      9988        0        0        0     7833     1.3       0    9988
aio    5  0  i  0.0      4894     3341     1426        0     4393     1.1       0       0
aio    9  1  i  0.0        41        0       41        0       33     1.2       0       0
pio    4  0  i  0.0       199        0      199        0      200     1.0       0     199
lio    3  0  i  0.0      6344        0     6344        0     6344     1.0       0    6344
```

输出描述

class

虚拟处理器类

vp

虚拟处理器在类中的 ID 编号

s

AIO 虚拟处理器的当前状态

f

派生

i

空闲

s

搜素

b

正忙

o

打开

c

关闭

io/s

自数据库服务器启动以来或自 gstat -z 上次运行以来（看着两者哪个发生在后）虚拟处理器的平均 I/O 速度（以每秒操作数衡量）

totalops

自数据库服务器启动以来或自 gstat -z 上次运行以来（看着两者哪个发生在后）虚拟处理器执行的 I/O 操作总数（以每秒操作数衡量）

dskread

自数据库服务器启动以来或自 gstat -z 上次运行以来（看着两者哪个发生在后）虚拟处理器执行的读操作总数（以每秒操作数衡量）

dskwrite

自数据库服务器启动以来或自 gstat -z 上次运行以来（看着两者哪个发生在后）虚拟处理器执行的写操作总数（以每秒操作数衡量）

dskcopy

自数据库服务器启动以来或自 gstat -z 上次运行以来（看着两者哪个发生在后）虚拟处理器执行的复制操作总数（以每秒操作数衡量）

wakeups

对于 AIO VPs ，是自数据库服务器启动以来或自 gstat -z 上次运行以来（看着两者哪个发生在后）虚拟处理器处于空闲状态的次数

io/wup

对于 AIO VPs ，是自数据库服务器启动以来或自 gstat -z 上次运行以来（看着两者哪个发生在后）该虚拟处理器每次唤醒执行的 I/O 操作平均数

errors

KAIO 超出资源的错误总数

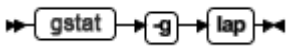
tempops (decimal)

仅供内部使用。是用来确定何时添加新 AIO VP 的 I/O 操作计数。它只在 AUTO_AIOVPS 配置参数启用时应用。

gstat -g lap 命令：打印轻量级追加(GBase_8s light append)状态信息

可以使用 gstat -g lap 命令显示有关系统内的轻量级追加的状态信息。

语法：



示例输出

图: gstat -g lap 命令输出

```
Light Append Info
session id  address   cur_ppage   la_npused   la_ndata    la_nrows    bufcnt
31          b60a5e8   ffbff494    2938        2937        93990       4
```

输出描述

Session id (decimal)

执行轻量级追加操作的会话 ID

address (hexadecimal)

轻量级追加缓冲区的地址

cur_ppage (hexadecimal)

当前的物理页面地址

la_npused (decimal)

分配的页面数

la_ndata (decimal)

附加的数据的页面数

la_nrows (decimal)

附加的行数

bufcnt (decimal)

轻量级追加缓冲区数

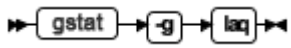
gstat -g laq 命令：打印辅助服务器队列

使用 gstat -g laq 命令打印有关从主服务器接收的应用日志信息的辅助服务器队列的信息。在高可用性集群中，主服务器通过网络向一台或多台辅助服务器发送日志记录。每个辅助服务器持续重播来自主服务器的事务日志，以确保数据复制到辅助服务器中。主服务器中的每个 tblspace 被分配到辅助服务器的队列中去接收日志记录。称作 apply thread 的线程，将应用该日志存储到辅助服务器的队列中。这些日志以它们接收时的顺序来应用。

可以使用 gstat -g laq 命令监视该队列在辅助服务器上的性能。如果您怀疑由于辅助服务器上的日志没有及时重播而造成主服务器性能降低，那么可以使用该命令。Avg Depth（平均深度）列标示了上次队列插入操作发生时队列中日志的平均数量

gstat -g laq 命令只在辅助服务器上可用。在主服务器上运行该命令只会返回 gstat 头的输出。

语法：



示例输出

图: gstat -g laq 命令输出

```
Log Apply Info:
          Thread         Queue    Total      Avg
          Size   Queued    Depth
          xchg_1.0           0        9     5.00
          xchg_1.1           0        0     0.00
          xchg_1.2           0        7     3.43
          xchg_1.3           0        8     1.12
          xchg_1.4           0        4     1.50
          xchg_1.5           0        4     1.50
          xchg_1.6           0        6     1.33
          xchg_1.7           0       47     3.11
          xchg_1.8           0       37     9.46
          xchg_1.9           0       13     3.69

          Secondary Apply Queue:         Total Buffers:12 Size:64K Free Buffers:11
          Log Recovery Queue:            Total Buffers:4 Size:32K Free Buffers:2
          Log Page Queue:                Total Buffers:32 Size:2K Free Buffers:32
          Log Record Queue:              Total Buffers:50 Size:64K Free Buffers:49
```

输出描述

Thread

分配接收日志记录的应用线程的名称

Queue Size

等待该应用线程的日志记录队列数

Total Queued

给出的应用线程队列中的日志记录的总数

Avg Depth

上次队列插入操作发生时队列中日志的平均数量

Secondary Apply Queue

该辅助应用队列接收来自主服务器的日志缓冲区。显示的值代表分配给接收日志缓冲区记录的缓冲区总数、队列的大小及可用缓冲区数

Log Recovery Queue

该日志恢复队列接收来自辅助应用队列的输出。日志缓冲区被转换成兼容 gtape 实用程序的格式。显示的值代表恢复队列中流缓冲区的总数、流缓冲区的大小及可用缓冲区数

Log Page Queue

该日志页队列接收来自日志恢复队列的输出。gtape 格式被移除并且数据分割成独立的日志页。显示的值代表恢复队列中的日志页的总数、队列的大小以及可用缓冲区数

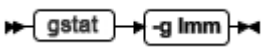
Log Record Queue

该日志队列接收来自日志页队列的输出。此日志页被分割成独立的日志记录。显示的值代表恢复队列中的日志记录数、队列的大小以及可用缓冲区数

gstat -g lmm 命令：打印低内存管理信息

使用 gstat -g lmm 命令显示有关自动低内存管理设置及其近期活动。

语法：



示例输出

图: gstat -g lmm 命令输出

```
Low Memory Manager


            Control Block      0x4cfca220
            Memory Limit       300000 KB
            Used               149952 KB
            Start Threshold    10240 KB
            Stop Threshold     10 MB
```

```
Idle Time          300 Sec
Internal Task      Yes
Task Name          'Low Memory Manager'
Low Mem TID        0x4cfd7178
# Extra Segments   0

Low Memory Manager Tasks

Task               Count    Last Run
Kill User Sessions 267      04/04/2011.16:57
Kill All Sessions  1        04/04/2011.16:58
Reconfig(reduce)   1        04/04/2011.16:59
Reconfig(restore)  1        04/04/2011.17:59

Last 20 Sessions Killed

Ses ID Username Hostname PID   Time
194    sfisher  host01   13433 04/04/2011.16:57
201    sfisher  host01   13394 04/04/2011.16:57
198    sfisher  host01   13419 04/04/2011.16:57
190    sfisher  host01   13402 04/04/2011.16:57
199    sfisher  host01   13431 04/04/2011.16:57

Total Killed 177
```

输出描述

Control Block

自动低内存管理的内部控制结构地址

Memory Limit

服务器可以附加的内存量

Used

服务器当前已使用的内存量

Start Threshold

自动低内存管理的启动阈值

Stop Threshold

自动低内存管理的停止阈值

Idle Time

自动低内存管理认为一个会话空闲后的时间量

Internal Task

Yes = 使用 GBase 8s 程序

No = 使用用户自定义程序

Task Name

用户自定义程序名称

Low Mem TID

自动低内存管理线程的地址

Task

Kill = 自动运行进程并终止会话

Reconfig（减少） = 自动运行进程并释放 block 未使用的内存

Reconfig（恢复） = 自动运行进程并恢复服务和配置

Count

执行该任务的次数

Last Run

最近一次执行任务的日期和时间

Ses ID

终止的会话的 ID（使用 gadmin －z 命令）

Username

会话所有者的用户名

Hostname

发起会话的主机的名称

PID

进程 ID

Time

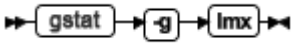
会话终止的日期和时间

可以使用 LOW_MEMORY_MGR 配置参数来启用自动低内存管理。

gstat -g lmx 命令：打印所有锁定的互斥

可以使用 gstat -g lmx 命令显示所有锁定的互斥的信息。

语法：



示例输出

图: gstat -g lmx 命令输出

```
Locked mutexes:
            mid      addr              name               holder   lkcnt  waiter   waittime
            119006   7000001e684b928   td_mutex           298      0
            134825   7000002043a9148   free_lock          11009    0      200      22921
            11010    22918
            587817   70000022ddb3268   sync_lock1         200      0
            593614   700000239ce7b68   SB_LTH_LATCH       875      0

            Number of mutexes on VP free lists: 49
```

输出描述

mid

内部互斥标识符

addr

锁定的互斥的地址

name

互斥的名称

holder

持有这个互斥的线程的 ID

0 = 在共享模式下持有的读/写互斥

lkcnt

对于读/写互斥，是线程在共享模式下锁定的此互斥的数量。对于重新锁定互斥，它是此线程持有的互斥被锁定或重新锁定的次数

waiter

等待这个互斥的线程 ID 的列表

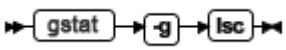
waittime

此线程已经等待的时间（以秒为单位）

gstat -g lsc 命令：打印活动的轻度扫描（GBase_8s light scan）状态（不推荐使用）

gstat -g lsc 命令已被 gstat -g scn 命令取代。

语法：



示例输出

图: gstat -g lsc 命令输出

```
Light Scan Info
descriptor  address          next_lpage  next_ppage       ppage_left  bufcnt  look_aside
3           474b74b0         4a0         7e2c80           416         1            N
```

输出描述

descriptor (decimal)

轻度扫描 ID

address (hex)

轻度扫描描述符的内存地址

next_lpage (hex)

下一个要扫描的逻辑页地址

next_ppage (hex)

下一个要扫描的物理页地址

ppage_left (decimal)

在当前 extent 中要扫描的剩余物理页数

#bufcnt (decimal)

用于此轻度扫描的细体扫描缓冲区数

#look_aside (char)

此轻度扫描是否要观察周边（Y = yes ， N = no）。当线程需要在缓冲池中检查现有页以获得正在被轻度扫描的页的最新图像时，发生观察周边操作

使用 gstat -g scn 命令显示当前扫描的状态、基于压缩表行扫描、行大于页的表、具有 VARCHAR 、LVARCHAR 和 NVARCHAR 数据类型的表。有关更多信息，请参阅 gstat -g scn 命令；打印扫描选项。

gstat -g mem 命令：打印池内存统计信息

可以使用 gstat -g mem 命令显示某个池的内存统计信息。

如果运行来自 PER_STMT_EXEC 和 PER_STMT_PREPf 内存持续时间池的分配内存的 SQL 查询，那么 gstat -g mem 命令显示有关 PRP.sessionid.threadid 池和 EXE.sessionid.threadid 池的信息。

语法：

会话池以会话编号命名。如果未提供参数，那么将显示所有池的信息。

示例输出

图: gstat -g mem 命令输出

```
Pool Summary:
            name                addr             totalsize freesize #allocfrag #freefrag
            resident      R     10a001028        2420736   7960     2          2
            res-buff      R     10a250028        8269824   7960     2          2
            global        V     10aac0028        9351168   32648    650        11
            ...
            ...
            ...
            onmode_mon    V     10b983028        20480     2752     108        1
            13            V     10bd5d028        16384     5200     12         2
            Blkpool Summary:
            name                addr             size      #blks      pre-hint   szavail
            global        V     10aac8920        0         0          0          0
            xmf_msc_pl    V     10ac84ca0        954368    73         0          0
```

输出描述

Pool Summary

name

池的名称

在创建池的位置的共享内存段类型

addr

池内存地址

totalsize

池大小，以字节表示

freesize

在池中的可用内存量

#allocfrag

在池中的已分配的分片

#freefrag

在池中的可用分片

Blkpool Summary

name

池的名称

在创建池的位置的共享内存段类型

addr

池内存地址

size

池大小，以字节表示

#blks

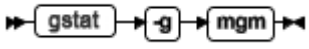
池中的 block 数

gstat -g mgm 命令：打印 MGM 资源信息

可以使用 gstat -g mgm 命令显示有关内存分配管理器（MGM）资源信息。

可以使用 gstat -g mgm 命令监视 MGM 如何协调内存使用和扫描线程。此命令读取共享内存结构并提供在命令执行时的精确统计信息。

语法：



gstat -g mgm 输出显示称作 quantum 的内存单位。memory quantum 代表内存单位，如下所示：

```
memory quantum = DS_TOTAL_MEMORY / DS_MAX_QUERIES
```

以下算式显示 gstat -g mgm 输出的值的 memory quantum：

```
memory quantum = 4000 kilobytes / 31
          = 129 kilobytes
```

数据库服务器根据分配内存的需求调整 quantum 的值。因此，gstat -g mgm 命令显示的 quantum 值并非一直精确。

scan thread quantum 总是等于 1。

示例输出

图: gstat -g mgm 命令输出

```
            Memory Grant Manager (MGM)
            ------------------------

            MAX_PDQPRIORITY:  100
            DS_MAX_QUERIES:    31
            DS_MAX_SCANS:      1048576
            DS_NONPDQ_QUERY_MEM: 128 KB
            DS_TOTAL_MEMORY:   4000 KB

            Queries:   Active     Ready   Maximum
            0          0        31
            Memory:     Total      Free   Quantum
            (KB)         4000      4000       128

            Scans:      Total      Free   Quantum
            1048576   1048576         1

            Load Control:    (Memory)      (Scans)  (Priority)  (Max Queries)
(Reinit)
            Gate 1       Gate 2      Gate 3         Gate 4     Gate 5
            (Queue Length)          0            0           0
0          0

            Active Queries:  None
            Ready Queries:  None
            Free Resource        Average #        Minimum #
            --------------    ---------------     ---------
            Memory               0.0 +- 0.0           500
```

```
Scans                0.0 +- 0.0          1048576

Queries              Average #        Maximum #    Total #
-------------    ---------------    ---------    -------
Active               0.0 +- 0.0             0          0
Ready                0.0 +- 0.0             0          0

Resource/Lock Cycle Prevention count:  0
```

输出描述

输出的第一部分显示 PDQ 配置参数的值。

输出的第二部分描述 MGM 内部控制信息。它包括四组的信息。第一组是 Queries：

Active

当前正在执行的 PDQ 查询的数量

Ready

已准备好运行但数据库服务器由于装入控制原因而延迟查询执行的用户查询的数量

Maximum

数据库服务器允许处于活动状态的查询的最大数量。反映 DS_MAX_QUERIES 配置参数的当前值

下一组是 **Memory**：

Total

可由 PDQ 查询使用的可用内存的 KB （DS_TOTAL_MEMORY 指定这个值）

Free

用于 PDQ 查询的当前未使用的内存的 KB

Quantum

在 memory quantum 中的内存的 KB

下一组是 **Scans**：

Total

由 DS_MAX_SCANS 配置参数指定的扫描线程的总数

Free

当前可用于决策支持查询的扫描线程的数量

Quantum

在扫描线程量子中的扫描线程的数量

该部分输出的最后一组描述 MGM **Load Control**：

Memory

正在等待内存的查询的数量

Scans

正在等待扫描的查询的数量

Priority

正在等待具有更高 PDQ 优先级的查询运行的查询的数量

Max Queries

正在等待查询 slot 的查询数量

Reinit

在 gadmin -M 或 -Q 命令之后，正等待正在运行的查询完成的查询的数量

输出的下一部分 Active Queries 描述 MGM 活动的和就绪的队列。这部分输出显示在每个入口等待的查询的数量：

Session

启动查询的会话的会话 ID

Query

与查询相关联的内部控制 block 的地址

Priority

分配给查询的 PDQ 优先级

Thread

向 MGM 注册查询的线程

Memory

当前分配给查询或为查询保留的内存量（单位是 MGM 页，即 8 KB）

Scans

有查询当前使用的扫描线程的数量，或者分配给查询的扫描线程的数量

Gate

查询正在该处等待的入口编号

输出的下一部分 Free Resource 提供 MGM 可用资源的统计信息。这个部分和最后部分的数值反映自系统初始化或自上次 gadmin -Q 、-M 或 -S 命令以来的统计信息。这部分输出包含以下信息：

Average

平均内存量和扫描数量

Minimum

最小可用内存量和扫描数量

输出的下一部分 Queries 提供关于 MGM 查询的统计信息：

Average

活动且就绪队列的平均长度

Maximum

活动且就绪队列的最大长度

Total

活动且就绪队列的总长度

Resource/Lock Cycle Prevention count

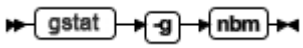
系统为了避免潜在的死锁而直接激活一个查询的次数。（数据库服务器可以察觉如果没有立即运行查询的话，其队列内一些查询什么时候可能发生死锁情况。）

gstat -g nbm 命令：打印 block 位图

可以使用 gstat -g nbm 命令显示非常驻段的 block 位图。

位图的每个位代表一个 4 KB 的 block。如果 block 正在使用，那么该位设置为 1 。如果 block 是空闲可用的，那么该位设置为 0 。位图以一系列的十六进制数字显示。位从 0 开始编号，从而 block 也从 0 开始编号，所以第一个 block 是 block 0，第二个是 block 1 ，以此类推。

语法：



示例输出

这个示例显示在 0x10CC00000 的虚拟内存段的位图。位图自身是在 0x10CC00290 。这个段的全部 1792 个 block 都可用。除了 block 0 和 block 1023 。

图: gstat -g nbm 命令输出

```
Block bitmap for virtual segment address 0x10cc00000:
            address = 0x10cc00290, size(bits) = 1792
            used = 1, largest_free = -1
            0:8000000000000000 0000000000000000 0000000000000000 0000000000000000
            256:0000000000000000 0000000000000000 0000000000000000 0000000000000000
            512:0000000000000000 0000000000000000 0000000000000000 0000000000000000
            768:0000000000000000 0000000000000000 0000000000000000 0000000000000001
            1024:0000000000000000 0000000000000000 0000000000000000 0000000000000000
            1280:0000000000000000 0000000000000000 0000000000000000 0000000000000000
            1536:0000000000000000 0000000000000000 0000000000000000 0000000000000000
```

输出描述

address

位图的起始地址

size

位图中的位数。这也是在内存段中的 4 KB block 的数量

used

在位图中设置为 1 的位的总数。这也是在内存段中的使用中的 4 KB block 的数量

largest free

如果这个值不是 -1 ，那么它是连续可用位的最大数量。它也是在内存段的最大邻接 block 集合中 4 KB block 的数量 。

值 -1 表示还未计算最大可用空间。数据库服务器只在尝试分配从 lastalloc block 开始的 block 集合却没有足够可用空间时计算最大可用空间。一旦在段中分配了另一个 block ，这个值就再次设置为 -1 。

gstat -g nsc 命令：打印当前的共享内存连接信息

可以使用 gstat -g nsc 命令显示有关当前所有连接或特定连接 ID 的共享内存连接的信息。

语法：

如果未提供 client_id ，那么给出关于到数据库服务器的所有当前共享内存连接的信息。如果提供了 client_id ，那么此命令给出具有该 ID 的共享内存连接的更详细的信息。

示例输出

以下是不带有 client_id 的 gstat -g nsc 输出。它显示当前只有一个用户通过共享内存连接到数据库服务器。这个连接的 ID 是 0 。

图: gstat -g nsc 命令输出

```
clientid  clientPID      state #serverbufs #clientbufs    #rdwrts
0       6031  Connected           4           4         12
```

此示例显示使用 client_id 0 运行命令的输出：

图: 带有 client id 的 gstat -g nsc 命令输出

```
Network Shared Memory Status for Client: 0

           clientid    clientPID        state  #serverbufs  #clientbufs      #rdwrts
           0        18949    Connected            4            4       447048

           needbuf         segid        semid       semnum     be_semid    be_semnum
           0          1303       851969            0       851969           10

           be_curread  be_curwrite   fe_curread  fe_curwrite
           -1            1            0           2

           be_nextread be_nextwrite  fe_nextread fe_nextwrite
           2            2            4           3

           readyqueue
           -1   -1   -1   -1   -1   -1   -1   -1   -1   -1   -1

           Server Buffers                            Client Buffers
           i: bufid     status   offset  fe_addr         bufid     status   offset  fe_addr
           0:     4      inuse     4474   804474             0      avail     3424   803424
           1:     5      inuse     4888   804888             1      avail     3838   803838
           2:     6      avail     4c9c   804c9c             2      inuse     3c4c   803c4c
           3:     7      avail     50b0   8050b0             3      avail     4060   804060
           4:    -1       free        0        0            -1       free        0        0
           5:    -1       free        0        0            -1       free        0        0
```

输出描述

clientid

服务器分配的 ID

clientPID

客户端进程 ID

state

连接状态

Connected

客户端已和服务器建立了连接

Con1

服务器已经成功建立与客户端的连接，但客户端尚未得到通知

Waiting

服务器正处于与客户机建立连接的过程中

Reject

服务器已经拒绝客户机连接，通常是由于服务器正在关闭或还未处于联机方式

Closed

服务器已关闭与客户端的连接。客户端可能还不知道这个情况

Not connected

服务器正在初始化连接的内部结构

Unknown

连接已关闭，并且客户端知道这个情况。服务器正在消除内部结构

#serverbufs

当前分配的数据库服务器缓冲区

#clientbufs

当前分配的客户端缓冲区

#rdwrts

自连接创建以来通过这个连接执行的读和写的总数

仅当带 client_id 运行 gstat -g nsc 命令时，以下项才显示在输出中：

needbuf

标示服务器是否在等待释放某个缓冲区

0

False

1

True

segid

共享内存段 ID

semid

信号量 ID

semnum

在信号量 ID 中的信号量编号

be_semid

后端信号量 ID

be_semnum

在信号量 ID 中的后端信号量编号

be_curread

正被读取的后端缓冲区的 ID

be_curwrite

正被写入的后端缓冲区的 ID

fe_curread

正被读取的前端缓冲区的 ID

fe_currwrite

正被写入的前端缓冲区的 ID

be_nextread

要读取的下一个后端缓冲区的 ID

be_nextwrite

要写入的下一个后端缓冲区的 ID

fe_nextread

要读取的下一个前端缓冲区的 ID

fe_nextwrite

要写入的下一个前端缓冲区的 ID

readyqueue

共享缓冲区标识队列

Buffers

i

消息缓冲区的内部位置键

bufid

消息缓冲区 ID

status

消息缓冲区的状态

offset

共享内存段中的内存缓冲区的偏移量

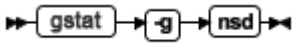
fe_addr

消息缓冲区的前端地址

gstat -g nsd 命令：打印轮询线程共享内存数据

可以使用 gstat -g nsd 命令显示轮询线程的共享内存数据。

语法：



示例输出

图: gstat -g nsd 命令输出

```
Network Shared Memory Data for Poll Thread: 0
Free Message Buffer Bitmap
(bitmap address = 10b9eef80, bitmap size 480)
000000010b9eef80:ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
000000010b9eefa0:ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
Free Message Buffer Status Bitmap
(bitmap address = 10ca0a9b0, bitmap size 50)
000000010ca0a9b0:ffffffff ffffff
Message Buffer Table
bufid   clientid            addr
Message Buffer Status Table
clientid      netscb addr             addr           offset
```

gstat -g nss 命令：打印共享内存网络连接状态

可以使用 gstat -g nss sessionid 命令显示有关共享内存网络连接状态的信息。

语法：

如果未提供 sessionid ，那么会为每个共享内存连接列出一行摘要。

示例输出

图: gstat -g nss 命令输出

```
clientid  clientPID   state    #serverbufs #clientbufs    #rdwrts
           1      14018  Connected        4           4        331
           0      12398  Connected        4           4        294
           2      14036  Connected        4           4         59
```

输出描述

clientid (decimal)

服务器为查找分配的值

clientPID (decimal)

客户端进程 ID

state (string)

连接的当前状态

- Connected
- Con1
- Waiting
- Reject

- Bedcover
- Closed
- Not connected
- Unknown

#serverbufs (dec)

当前分配的数据库服务器的缓冲区数

#clientbufs (dec)

当前分配的客户端缓冲区数

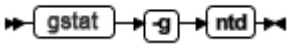
#rdwrts (dec)

使用中的缓冲区数

gstat -g ntd 命令：打印网络统计信息

可以使用 gstat -g ntd 命令按服务显示网络统计信息。

语法：



示例输出

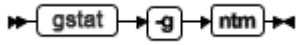
图: gstat -g ntd 命令输出

```
global network information:
  #netscb connects     read    write  q-limits  q-exceed alloc/max
   4/   5       11        0     3546 3549/  10   10/   0    0/   0

Client Type     Calls   Accepted   Rejected       Read      Write
sqlexec         yes           11          0       3531       3540
srvinfx         yes            0          0          0          0
onspace         yes            0          0          4          9
onlog           yes            0          0          0          0
onparam         yes            0          0          0          0
oncheck         yes            0          0          0          0
onload          yes            0          0          0          0
onunload        yes            0          0          0          0
onmonitor       yes            0          0          0          0
dr_accept       yes            0          0          0          0
cdraccept       no             0          0          0          0
ontape          yes            0          0          0          0
srvstat         yes            0          0          0          0
asfecho         yes            0          0          0          0
listener        yes            0          0         11          0
crsamexec       yes            0          0          0          0
onutil          yes            0          0          0          0
drdaexec        yes            0          0          0          0
smx             yes            0          0          0          0
safe            yes            0          0          0          0
Totals                        11          0       3546       3549
```

gstat -g ntm 命令：打印网络邮件的统计信息

可以使用 gstat -g ntm 命令打印网络邮件的统计信息。

语法：



示例输出

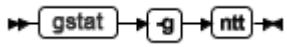
图: gstat -g ntm 命令输出

```
global network information:
  #netscb connects      read    write  q-limits  q-exceed alloc/max
   4/   5       11         0     3546 3549/  10   10/   0    0/   0

Network mailbox information:
 box    netscb thread name     max received   in box   max in box full signal
   5  f07e8b0 soctcppoll        10        24        0         1        0    yes
   6  f0b6ad8 soctcplst         10         0        0         0        0     no
   7  f0e8b18 soctcplst         10         0        0         0        0     no
```

gstat -g ntt 命令：打印网络用户的次数

可以使用 gstat -g ntt 命令显示有关网络用户时间的信息。

语法：



示例输出

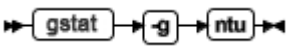
图: gstat -g ntt 命令输出

```
global network information:
  #netscb connects      read    write    q-limits  q-exceed alloc/max
   3/   3        0        0        0    135/  10    0/   0    2/   0
Individual thread network information (times):
 netscb   thread name   sid  open     read     write     address
c76ea28   ontape        61   14:34:48 14:34:50 14:34:50
c63e548   tlitcplst      4   14:30:43 14:34:48           server.ibm.com|5006|tlitcp
c631028   tlitcppoll     3   14:32:32
```

gstat -g ntu 命令：打印网络用户统计信息

可以使用 gstat -g ntu 命令显示有关网络用户的统计信息。

语法：



示例输出

图: gstat -g ntu 命令输出

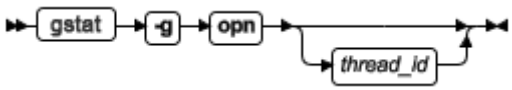
```
global network information:
  #netscb connects     read    write    q-free  q-limits  q-exceed alloc/max
   2/   3       16     2611     2603    1/   1  135/  10    0/   0    1/   1

Individual thread network information (basic):
  netscb type   thread name    sid   fd poll    reads   writes q-nrm q-pvt q-exp
 d1769f0 soctcp soctcplst        3    1    5       16        0  0/ 0  0/ 0  0/ 0
 d1199f0 soctcp soctcppoll       2    0    5     2595        0  0/ 0  0/ 0  0/ 0
```

gstat -g opn 命令：打印打开的分区

可以使用 gstat -g opn 命令显示通过线程 ID 打印系统中当前打开的分区（表和索引）的列表。

使用 thread_id 选项来限制指定 ID 的列表。

语法：



示例输出

图: gstat -g opn 命令输出

```
            tid  rstcb              isfd  op_mode    op_flags   partnum
ucount ocount lockmode
            38   0x00000000460db7b0 0     0x00000400 0x00000397 0x001000af 2
2      0
            38   0x00000000460db7b0 1     0x00000002 0x00000117 0x001000af 2
2      0
            38   0x00000000460db7b0 2     0x00000440 0x00000797 0x0010010c 2
0      0
            38   0x00000000460db7b0 3     0x00000400 0x00000407 0x0010010a 2
0      0
            38   0x00000000460db7b0 4     0x00000400 0x00000407 0x0010010a 2
0      0
            38   0x00000000460db7b0 5     0x00000002 0x00000003 0x00100003 2
2      0
            38   0x00000000460db7b0 6     0x00000400 0x00000397 0x00100003 2
2      0
            38   0x00000000460db7b0 7     0x00000400 0x00000413 0x0010010f 2
0      0
```

```
                38    0x00000000460db7b0 8     0x00000440 0x00000797 0x0010010c 2
0       0
                38    0x00000000460db7b0 9     0x00000402 0x00000403 0x0010010f 2
0       0
                38    0x00000000460db7b0 10    0x00000442 0x00000403 0x00100111 1
0       0
                38    0x00000000460db7b0 11    0x00000442 0x00000403 0x00100110 1
0       0
                38    0x00000000460db7b0 12    0x00000442 0x00000403 0x00100112 1
0       0
                38    0x00000000460db7b0 15    0x00000400 0x00004407 0x00000006 1
0       0
                38    0x00000000460db7b0 16    0x00000400 0x00000413 0x00100119 1
0       0

                36    0x00000000460dbf98 0     0x00000400 0x00000397 0x001000af 2
2       0
                36    0x00000000460dbf98 1     0x00000002 0x00000003 0x001000af 2
2       0
                36    0x00000000460dbf98 3     0x00000402 0x00000407 0x0010010a 2
0       0
                36    0x00000000460dbf98 4     0x00000400 0x00000413 0x0010010a 2
0       0
                36    0x00000000460dbf98 6     0x00000442 0x00000797 0x0010010c 1
0       0

                37    0x00000000460dc780 0     0x00000400 0x00000397 0x001000af 2
2       0
                37    0x00000000460dc780 1     0x00000002 0x00000117 0x001000af 2
2       0
                37    0x00000000460dc780 2     0x00000400 0x00000407 0x0010010a 2
0       0
                37    0x00000000460dc780 3     0x00000440 0x00000797 0x0010010c 2
0       0
```

```
            37    0x00000000460dc780 4     0x00000400 0x00000413 0x0010010f 2
0       0
            37    0x00000000460dc780 5     0x00000400 0x00000407 0x0010010a 2
0       0
            37    0x00000000460dc780 6     0x00000440 0x00000797 0x0010010c 2
0       0
            37    0x00000000460dc780 7     0x00000400 0x00000397 0x00100003 2
2       0
            37    0x00000000460dc780 8     0x00000002 0x00000003 0x00100003 2
2       0
            37    0x00000000460dc780 9     0x00000442 0x00000403 0x00100111 1
0       0
            37    0x00000000460dc780 10    0x00000442 0x00000403 0x00100110 1
0       0
            37    0x00000000460dc780 11    0x00000402 0x00000403 0x0010010f 2
0       0
            37    0x00000000460dc780 12    0x00000400 0x00000413 0x00100119 1
0       0
            37    0x00000000460dc780 13    0x00000442 0x00000403 0x00100112 1
0       0
            37    0x00000000460dc780 14    0x00000400 0x00004407 0x00000006 1
0       0
```

输出描述

tid (decimal)

当前访问分区资源（表和索引）的线程 ID

rstcb (hexadecimal)

该线程的 RSAM 线程控制 block 的内存地址

isfd (decimal)

与打开分区关联的 ISAM 文件描述符

op_mode (hexadecimal)

使用以下十六进制值组合的分区锁定方式的当前状态：

```
0x000000 Open for input only
               0x000001 Open for output only
               0x000002 Open for input and output
```

```
0x000004 System catalog
0x000008 No logical logging
0x000010 Open if not already opened for alter
0x000020 Open all fragments data and index
0x000040 Do not allocate a blob descriptor
0x000080 Open for alter
0x000100 Open all data fragments
0x000200 Automatic record lock
0x000400 Manual record lock
0x000800 Exclusive ISAM file lock
0x001000 Ignore dataskip - data cannot be ignored
0x002000 Dropping partition - delay file open
0x004000 Do not drop blobspace blobs when table dropped
(alter fragment)
0x010000 Open table for DDL operations
0x040000 Do not assert fail if this partnum does not exist
0x080000 Include fragments of subtables
0x100000 Table created under supertable
0x400000 Blob in use by CDR
```

op_flags (hexadecimal)

分区的当前状态使用以下十六进制值组合：

```
0x0001  Open data structure is in use
0x0002  Current position exists
0x0004  Current record has been read
0x0008  Duplicate created or read
0x0010  Skip current record on reverse read
0x0020  Shared blob information
0x0040  Partition opened for rollback
0x0080  Stop key has been set
0x0100  No index related read aheads
0x0200  isstart called for current stop key
0x0400  Pseudo-closed
0x0800  Real partition opened for SMI query
0x1000  Read ahead of parent node is done
```

```
0x2000  UDR keys loaded
0x4000  Open is for a pseudo table
0x8000  End of file encountered when positioning in table
```

partnum (hexadecimal)

已打开资源（表和索引）的分区数

ucount (decimal)

当前访问该分区的用户线程数

ocount (decimal)

打开该分区的次数

lockmode (decimal)

使用以下代码值之一保存锁定类型：

```
0   No locks
1   Byte lock
2   Intent shared lock
3   Shared lock
4   Shared lock by repeatable read (only on items)
5   Update lock
6   Update lock by repeatable read (only on items)
7   Intent exclusive lock
8   Shared, intent exclusive lock
9   Exclusive lock
10  Exclusive lock by repeatable read (only on items)
11  Inserter's repeatable read test lock
```

gstat -g osi 命令：打印操作系统的信息

可以使用 gstat -g osi 命令显示您的操作系统和参数的信息，包括共享内存和信号量参数、电脑上当前配置的内存量和未使用的内存量。

示例输出

gstat -g osi 命令也显示电脑上硬件程序的统计信息。

在服务器未联机时使用该命令。

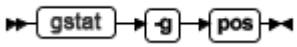
图: gstat -g osi 命令输出

```
Machine Configuration....
OS Name                                 Linux
OS Release                              2.6.9-34.ELsmp
OS Node Name                            idas
OS Version                              #1 SMP
OS Machine                              x86_64
Number of processors                    4
Number of online processors             4
System memory page size                 4096 bytes
System memory                           7970 MB
System free memory                      1536 MB
Number of open files per process        1024
shmmax              33554432
shmmin              1
shmids              4096
shmNumSegs          2097152
semmap              << UnSupported >>
semids              128
semnum              32000
semundo             << UnSupported >>
semNumPerID         250
semops              32
semUndoPerProc      << UnSupported >>
semUndoSize         20
semMaxValue         32767
```

gstat -g pos 命令：打印文件值

可以使用 gstat -g pos 命令显示 $GBASEDBTDIR/etc/.infos.DBSERVERNAME 文件中的值。

语法：



示例输出

图: gstat -g pos 命令输出

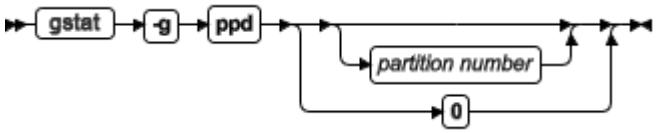
```
 1   2 -32750 shm  32786 5279c803         4ee94000    135364608 V
 2   2  19 shm     19 5279c804         56fac000    830087168 V
 3   7   0 infos ver/size 3 272
 4   1   0 snum  9088 5279c801         44000000 22148 instance
 5   4   0 onconfig path   #
 6   5   0 host localhost.localdomain
 7   6   0 oninit ver GBase Database Server Version 12.10.FC4G1TL
 8   8   0 sqlhosts path /opt/gbs_server/data/conf/sqlhosts
 9   3   2 sema 5111810
10   2 -32752 shm  32784 5279c801         44000000     51986432 R
11   2  17 shm     17 5279c802         47194000    131072000 V
```

gstat -g ppd 命令：打印分区压缩字典信息

可以使用 gstat –g ppd 命令显示为压缩表和表分片或压缩 B-tree 索引创建的活动压缩字典的信息。可以选择打印特定编号分区或所有打开分区的信息。

gstat –g ppd 命令打印的信息与在 sysmaster 数据库中 syscompdicts_full 表和 syscompdicts 视图显示的信息相同。唯一的区别是 syscompdicts_full 表和 syscompdicts 视图不仅仅只显示活动字典，还显示所有压缩字典的信息。

语法：



如果指定了分区号，那么 gstat -g ppd 打印该分区的分区概要文件信息。如果指定 0 ，该选项打印所有分区的概要文件信息。

示例输出

图: gstat –g ppd Output

partnum	ColOffset	DbsNum	CrTS	CrLogID	CrLogPos	DrTS	DrLogID	DrLogPos
0x1001d5	-1	1	1393371661	4	16339024	0	0	0
0x1001d5	4	1	1393371661	4	16355408	0	0	0

输出描述

partnum

应用压缩字典的分区号

ColOffset

压缩分区 blob 列的字节偏移量。-1 表示只有该行是压缩的

DbsNum

字典驻留的 dbspace 号

CrTS

显示创建该字典的时间戳

CrLogID

当创建字典时，创建的逻辑日志的唯一 ID

CrLogPos

创建字典时，逻辑日志的位置

DrTS

显示何时清除该字典的时间戳

DrLogID

当清除字典时，创建的逻辑日志的唯一 ID

DrLogPos

清除字典时，逻辑日志的位置

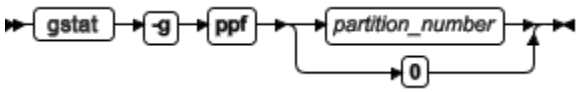
gstat -g ppf 命令：打印分区的概要文件

可以使用 gstat -g ppf partition_number 命令显示指定分区号的分区概要文件。

使用 gstat -g ppf 或 gstat -g ppf 0 命令显示所有分区的概要文件。如果 TBLSPACE_STATS 配置参数设置为 0 ，那么 gstat -g ppf 命令显示：Partition profiles disabled。

有关 gstat -g ppf 命令的更多信息，请参阅 GBase 8s 性能指南 。

语法：



示例输出

图: gstat -g ppf 命令输出

```
Partition profiles
        partnum    lkrqs lkwts dlks  touts isrd  iswrt isrwt isdel bfrd  bfwrt seqsc rhitratio
        0x100001   0     0     0     0     0     0     0     0     0     0     0     0
        0x100002   1506  0     0     0     416   4     0     4     1282  20    0     97
        0x100003   15    0     0     0     5     0     0     0     20    0     0     75
        0x1000a5   0     0     0     0     0     0     0     0     12    0     0     67
        0x1000e3   4     0     0     0     1     0     0     0     4     0     0     25
        0x200001   0     0     0     0     0     0     0     0     0     0     0     0
        0x300001   0     0     0     0     0     0     0     0     0     0     0     0
        0x400001   0     0     0     0     0     0     0     0     0     0     0     0
```

输出描述

partnum (hex)

分区号

lkrqs (decimal)

请求分区的锁的数量

lkwts (decimal)

等待分区的锁的数量

dlks (decimal)

分区的死锁数

touts(decimal)

分区的远程死锁超时

isrd (decimal)

分区的读取数

iswrt (decimal)

分区的写入数

isrwt (decimal)

分区的重写或修改数

isdel (decimal)

分区的删除数

bfrd (decimal)

缓冲区的读取数，以页表示

bfwrt (decimal)

缓冲区的写入数，以页表示

seqsc (decimal)

分区的顺序扫描数

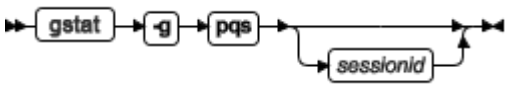
rhitratio (percentage)

磁盘读和缓冲区读的比率

gstat -g pqs 命令：打印所有 SQL 查询的运算符

可以使用 gstat -g pqs 命令显示当前正在运行的 SQL 查询中的使用的运算符的信息。可以使用该命令对应用程序镜像故障排除、找到该查询所使用的运算符以及每个运算符返回行数及长度。 EXPLAIN 文件包含的信息显示此查询计划的概览，gstat -g pqs 命令显示该查询和查询计划的运行时的操作者。

语法：



可以指定以下之一的调用：

表 1. 每个 gstat -g pqs 命令调用的描述

调用	解释
gstat -g pqs	显示每个会话的一行摘要
gstat -g pqs *sessionid*	显示指定会话的信息

示例输出

以下示例显示了在不同会话中三个单独的 SQL 语句运行的结果。这些语句是：

```
Query Operators:
addr     ses-id   opname    phase   rows    time          in1         in2     stmt-type
ae50b3a  23       scan      open    0       00:00.00         0           0    SELECT
af269d0   5       nljoin    next    224717  00:01.82   af26a90     aeb4478    SELECT
af26a90   5       scan      next    472     00:00.20         0           0    SELECT
aeb4478   5       scan      next    50      00:01.63         0           0    SELECT
ad3c530  26       scan      open    0       00:00.00         0           0    UPDATE (all)
```

图: gstat -g pqs 命令输出

```
Query Operators:
addr     ses-id   opname    phase   rows    time          in1         in2    stmt-type
ae50b3a  23       scan      open    0       00:00.00        0           0    SELECT
af269d0   5       nljoin    next    224717  00:01.82   af26a90     aeb4478    SELECT
af26a90   5       scan      next    472     00:00.20        0           0    SELECT
aeb4478   5       scan      next    50      00:01.63        0           0    SELECT
ad3c530  26       scan      open    0       00:00.00        0           0    UPDATE (all)
```

输出描述

addr

运算符在内存中的地址。可以使用该地址跟踪属于每个 JOINT 运算符的 SCAN 运算符

ses-id

运行 SQL 语句的会话的 ID

opname

运算符名称

phase

已使用的运算符的阶段，例如 OPEN 、NEXT 、CLOSE

rows

该运算符执行的行数

time

执行该运算符的时间量。以毫秒显示。 01:20.10 是 1 分钟 20 秒 10 毫秒

in1

该连接中的第一个运算符（outer）

in2

该连接中的第二个运算符（inner）

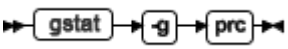
stmt-type

SQL 语句的类型，例如 SELECT 、UPDATE 、DELETE

gstat -g prc 命令：打印使用 UDR 或 SPL 例程的会话

可以使用 gstat -g prc 命令显示当前使用 UDR 或 SPL 例程的会话数。

语法：



示例输出

图: gstat -g prc 命令输出

```
UDR Cache:
        Number of lists          : 31
        PC_POOLSIZE              : 127

        UDR Cache Entries:

        list id   ref  drop hits      heap_ptr  udr name
        ---------------------------------------------------------
        0    80   0    0    702       4c589020  syscdr@amsterdam:.ifx_allow_newline
        0    494  1    0    3         4c1e6820  syscdr@amsterdam:.compare
        0    219  0    0    2         4bfd1020  syscdr@amsterdam:.streamread
        0    297  0    0    8         4bb99020  syscdr@amsterdam:.ifx_checksum
        0    134  0    0    10214     4bb5f020  syscdr@amsterdam:.destroy
        0    232  0    0    34        4bd62820  syscdr@amsterdam:.cdrcmd
        0    364  0    0    1         4c345020  syscdr@amsterdam:.rci_insert
        0    180  0    0    1         4bcba020  syscdr@amsterdam:.gist_drop
        0    91   0    0    9         4bd2e020  sysha@amsterdam:.rlt_open
        0    500  0    0    76        4bb9f020  sysadmin@amsterdam:.admin
        0    27   0    0    1478      4c0ec020  sysadmin@amsterdam:.destroy
        ...

        Total number of udr entries : 254
        Number of entries in use    : 9
```

输出描述

Number of lists

UDR 高速缓存中的列表数

PC_POOLSIZE

一次可以高速缓存的条目数

list

UDR 高速缓存哈希链 ID （bucket number）

id

此例程的唯一 ID

ref

从高速缓存访问 UDR 或 SPL 例程的会话数

drop

该例程是否标记已删除

hits

访问该高速缓存条目的次数

heap_ptr

用于存储该条目的堆地址

udr_name

高速缓存中 UDR 或 SPL 例程的名称

Total number of udr entries

高速缓存中的条目数

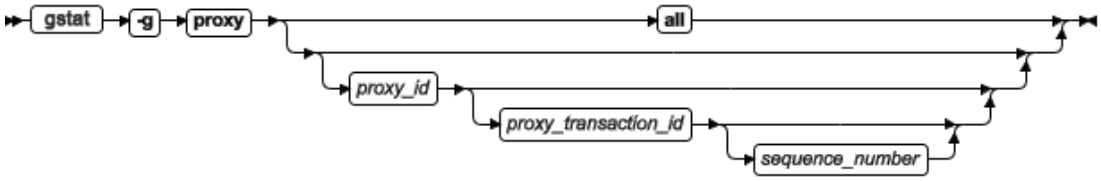
Number of entries in use

正在使用的条目数

gstat -g proxy 命令：打印代理分发器信息

可以使用 gstat -g proxy 命令显示有关代理分发器的信息。gstat -g proxy 命令的输出根据该命令是否在主服务器或辅助服务器上运行而稍有不同。

语法：



调用	解释
gstat -g proxy	显示代理分发器信息
gstat -g proxy all	当在主服务器上运行时，显示有关代理分发器和代理线程的信息。当在辅助服务器上运行时。显示有关所有当前执行更新到辅助服务器的会话的信息
gstat -g proxy *proxy_id* *proxy_transaction_id* *sequence_number*	该选项只在辅助服务器上可用。显示当前由给出的代理分发器产生的工作的详细信息。*proxy_transaction_id* 和 *sequence_number* 是可选的参数。当提供时，第一数字作为 *proxy_transaction_id* ，第二个数字解释为 *sequence_number*。如果提供的 *proxy_transaction_id* 或 *sequence_number* 不存在，那么该命令的输出与 gstat - 输出一样

在主服务器上使用 gstat -g proxy 命令的示例输出

图: gstat -g proxy 命令输出（自主服务器上运行）

```
Secondary          Proxy      Reference Transaction  Hot Row
Node               ID         Count     Count        Total
nagpur_sdc1        2619       0         2            0
nagpur_c2          2632       0         1            0
nagpur_sec         2633       0         1            0   I
```

输出描述

Secondary Node

被主服务器所知的辅助服务器名称

Proxy ID

代理分发器的 ID 。在高可用性集群中代理 ID 是唯一的

Reference Count

标示当前事务中正在使用该信息的线程数。当计数为 0 时，该事务处理完毕（无论成功或不成功）

Transaction Count

当前代理分发器正在处理的事务数

Hot Row Total

代理分发器已处理的热行总数

在辅助服务器上使用 gstat -g proxy 命令的示例输出

图: gstat -g proxy 命令输出（自辅助服务器上运行）

```
Primary             Proxy      Reference Transaction  Hot Row
Node                ID         Count     Count        Total
nagpur              2619       5         2            0
```

输出描述

Primary Node

主服务器名

Proxy ID

代理分发器的 ID 。在高可用性集群中代理 ID 是唯一的

Reference Count

标示当前事务中正在使用该信息的线程数。当计数为 0 时，该事务处理完毕（无论成功或不成功）

Transaction Count

当前代理分发器正在处理的事务数

Hot Row Total

代理分发器已处理的热行总数

在主服务器上使用 gstat -g proxy all 命令的示例输出

图: gstat -g proxy all 命令输出（自主服务器上运行）

```
Secondary           Proxy      Reference Transaction  Hot Row
Node                ID         Count     Count        Total
nagpur_sdc1         2619       0         2            0
nagpur_c2           2632       0         1            0
nagpur_sec          2633       0         1            0

TID      Flags      Proxy  Source   Proxy    Current  sqlerrno iserrno
                    ID     SessID   TxnID    Seq
94       0x00000224 2619   21       1        29       0        0
95       0x00000224 2619   22       2        68       0        0
93       0x00000224 2632   21       2        2        0        0
91       0x00000224 2633   25       1        6        0        0
```

输出描述

Secondary Node

被主服务器所知的辅助服务器名

Proxy ID

代理分发器的 ID 。在高可用性集群中代理 ID 是唯一的

Reference Count

标示当前事务中正在使用该信息的线程数。当计数为 0 时，该事务处理完毕（无论成功或不成功）

Transaction Count

当前代理分发器正在处理的事务数

Hot Row Total

代理分发器已处理的热行总数

TID

正在主服务器上运行的代理线程的 ID 。该 ID 是代理分发器创建用于处理来自辅助服务器会话的工作

Flags

代理线程的标志

Proxy ID

代表正在运行的代理线程（TID）的代理分发器 ID

Source SessID

辅助服务器上用户会话的 ID

Proxy TxnID

当前事务的编号。这些编号对于代理分发器是唯一的

Current Seq

当前事务中当前操作的顺序号

sqlerrno

任一 SQL 错误的错误号（或者 0 如果没有错误）

iserrno

任一 ISAM 或 RSAM 错误的错误号 （或者 0 如果没有错误）

在辅助服务器上使用 gstat -g proxy all 命令的示例输出

图: gstat -g proxy all 命令输出（自辅助服务器上运行）

```
Primary             Proxy       Reference Transaction  Hot Row
Node                ID          Count     Count        Total
nagpur              2619        5         2            0

Session  Session  Proxy    Proxy    Proxy    Current  Pending  Reference
         Ref      Proxy_id TID      TxnID    Seq      Ops      Count
21       2        2619     94       1        29       1        1
22       2        2619     95       2        68       1        1
```

输出描述

Primary Node

主服务器名

Proxy ID

代理分发器的 ID 。在高可用性集群中代理 ID 是唯一的

Reference Count

标示当前事务中正在使用该信息的线程数。当计数为 0 时，该事务处理完毕（无论成功或不成功）

Transaction Count

当前代理分发器正在处理的事务数

Hot Row Total

由代理分发器处理的总行数。热行是辅助服务器上被一个或多个客户端修改过多次的行。当行被多次修改时，如果最近来自不同会话的更新操作不在辅助服务器上重演，辅助服务器会读取之前从主服务器在该行上放置一个更新锁的视图。

Session

会话 ID

Proxy ID

代表正在运行的代理线程（TID）的代理分发器 ID

Proxy TID

正在主服务器上运行的代理线程的 ID 。该 ID 是代理分发器创建用于处理来自辅助服务器会话的工作

Proxy TxnID

当前事务的编号。这些编号对于代理分发器是唯一的

Current Seq

当前事务中当前操作的顺序号

Pending Ops

辅助服务器上还未发送到主服务器的已缓冲的操作数

Reference Count

标示当前事务中正在使用该信息的线程数。当计数为 0 时，该事务处理完毕（无论成功或不成功）

在辅助服务器上使用 proxy_id 选项的示例输出

该命令只在辅助服务器上才返回信息。

图: gstat -g proxy proxy_id 命令输出（自辅助服务器上运行）

```
Proxy    Reference Pending  ProxySID
TxnID    Count     Ops
1        1         1        3
2        1         1        4
```

输出描述

Proxy TxnID

当前的事务的编号。这些编号对于代理分发器是唯一的

Reference Count

标示当前事务中正在使用该信息的线程数。当计数为 0 时，该事务处理完毕（无论成功或不成功）

Pending Ops

辅助服务器上还未发送到主服务器的已缓冲的操作数

Proxy SID

代理会话 ID

在辅助服务器上使用 proxy_id proxy_transaction_id 选项的示例输出

该命令只在辅助服务器上才返回信息。

图: gstat -g proxy_id proxy_transaction_id 命令输出（自辅助服务器上运行）

```
Sequence Operation rowid    Table                          sqlerrno
Number   Type              Name
28       *Update   526      stores_demo:nilesho.customer   0
```

输出描述

Sequence Number

操作数

Operation Type

已执行的操作的类型。可以是：插入、修改、删除和其它

rowid

应用该操作的行的行 ID

Table Name

表的全名，修改以适应合理的长度。格式为：database.owner.tablename

sqlerrno

任一 SQL 错误的错误号（或者 0 如果没有错误）

在辅助服务器上运行 proxy_id proxy_transaction_id sequence_number 选项的示例输出

该命令只在辅助服务器上才返回信息。

输出的字段与 gstat -g proxy_id proxy_transaction_id 命令输出的字段相同。不同的是，gstat -g proxy_id proxy_transaction_id 命令显示该事务的详细信息，gstat -g proxy_id proxy_transaction_id sequence_number 显示所有事务的详细信息。

图: gstat -g proxy_id proxy_transaction_id sequence_number 命令输出（自辅助服务器上运行）

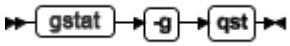
```
s
Proxy    Reference Pending  ProxySID
TxnID    Count     Ops
61       0         3        22

onstat -g proxy 2788 61


Sequence Operation rowid    Table                          sqlerrno
Number   Type               Name
960      Update    264      stores_demo:nilesho.customer      0
961      Update    265      stores_demo:nilesho.orders        0
962      Update    266      stores_demo:nilesho.items         0

onstat -g proxy 2788 61 962


Sequence Operation rowid    Table                          sqlerrno
Number   Type               Name
962      Update    266      stores_demo:nilesho.items         0
```

gstat -g qst 命令：打印互斥队列和条件队列的等待选项

可以使用 gstat -g qst 命令显示互斥队列和条件队列（互斥或条件的等待者的队列）的等待统计信息。

QSTATS 配置参数必须设为 1 来启用统计信息的采集。有关更多信息，请参阅 QSTATS 配置参数

语法：



示例输出

图: gstat -g qst 命令输出

```
Mutex Queue Statistics
name     nwaits   avg_time max_time avgq maxq nservs   avg_time

ddh chai 1        1354863  1354863  1    1    56       1690

Condition Queue Statistics
name     nwaits   avg_time max_time avgq maxq nservs   avg_time

arrived  1        110008   110008   1    1    0        0
logbf0   21       642      4431     1    2    0        0
logbf1   15       475      2519     1    2    0        0
logbf2   19       596      3274     1    2    0        0
bp_cond  1        0        0        1    1    0        0
```

输出描述

name (string)

等待的互斥或条件资源的名称

nwaits (decimal)

等待此资源的次数

avg_time (decimal)

等待的平均时间（微秒）

max_time (decimal)

等待的最长时间（微秒）

avgq (decimal)

队列的平均长度

maxq (decimal)

队列的最长长度

nservs (decimal)

获取此资源的次数

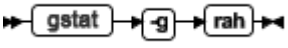
avg_time (decimal, microsecond)

每次获取资源时挂起的平均时间（微秒）

gstat -g rah 命令：打印预读请求统计信息

可以使用 gstat -g rah 命令显示有关预读请求的信息。

语法：



示例输出

图: gstat -g rah 命令输出

```
Read Ahead

           # Qs                1
           # threads           2
           # Requests          58690
           # Continued               0
           # Memory Failures   0
           Last Thread Add     04/06/2013.14:34
           Way behind          0

           Partition ReadAhead Statistics

           Buffer   Disk   Hit    Data       Index        Idx/Dat     Log/PageList Last Committed
           Partnum   Reads    Reads  Ratio  # Reqs Eff # Reqs  Eff  # Reqs  Eff # Pages Eff  # Reqs  Eff   # Resch
           0x200003  4312677  110    99     0      0   0       0    0       0    0      0     12906  100   0
           0x300002  23740584 1427   99     0      0   0       0    0       0    0      0     6681   100   7
           0x400002  17818942 966    99     0      0   0       0    0       0    0      0     25849  100   57
```

输出描述

Qs

预读请求队列数

 # threads

预读线程数

 # Requests

预读请求数

 # Continued

持续发生预读请求的次数

 # Memory Failures

由于内存不满足而请求失败的次数

Last Thread Add

上一次添加预读线程的日期和时间

Way behind

由于预读的保护进程太落后而删除的页请求列

Partnum

分区号

Buffer reads

已读取的缓冲池数和磁盘页数

Disk Reads

从磁盘读取的页数

Hit Ratio

分区的缓存命中率

 # Reqs

预读请求数。（该实例有五个输出字段：数据、索引、索引数据、日志页和上次提交的行）

Eff

预读请求的效率。它是预读操作请求的页数与不需要预读操作的已缓存页数的比率。值位于 0 和 100 之间。数字越大表示该预读效率更高。该实例有五个输出字段：数据、索引、索引数据、日志页和上次提交的行）

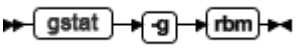
Resch

因为对多片行的更新不完整而要重新安排的最后提交的行的请求数

gstat -g rbm 命令：打印共享内存的 block 映射

可以使用 gstat -g rbm 命令显示共享内存常驻段中可用和已使用的 block 的十六进制位图。

语法：



示例输出

图: gstat -g rbm 命令输出

```
Block bitmap for resident segment address 0x44000000:
 address = 0x440003bc, size(bits) = 3035
 used = 3031, largest_free = 4

      0:ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
    256:ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
    512:ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
    768:ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
   1024:ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
   1280:ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
   1536:ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
   1792:ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
   2048:ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
   2304:ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
   2560:ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
   2816:ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff fffffe00
```

输出描述

Header

address (hex)

在段内已用/空闲的 block 的内存中起始地址

size (bits)

block 位图中的位数；每个位代表一个 block

used (blocks)

位图中已用的 blocks

largest_free (blocks)

可用 block 的最大运行

Data

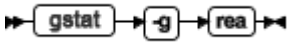
Bit number (decimal): data (hex)

位数，后面是 32 个字节的数据（十六进制）

gstat -g rea 命令：打印准备就绪的线程

可以使用 gstat -g rea 命令显示当前已就绪的虚拟处理器线程的信息。

语法：



示例输出

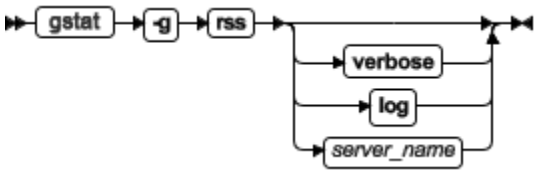
图: gstat -g rea 命令输出

```
Ready threads:
tid     tcb      rstcb    prty    status    vp-class    name
6       536a38   406464   4       ready        3cpu     main_loop()
28      60cfe8   40a124   4       ready        1cpu     onmode_mon
33      672a20   409dc4   2       ready        3cpu     sqlexec
```

gstat -g rss 命令：打印 RS 辅助服务器信息

可以使用 gstat -g rss 命令显示有关远程独立辅助服务器的信息。

语法：



gstat -g rss 命令的输出因命令在主服务器上还是 RS 辅助服务器上运行而稍有不同。

调用	解释
gstat -g rss	显示 RS 辅助服务器的简要信息
gstat -g rss verbose	显示 RS 辅助服务器的详细信息
gstat -g rss log	显示日志信息。该命令只能在主服务器上应用。
gstat -g rss *server_name*	显示指定的 RS 辅助服务器的信息。该命令只能在主服务器上应用。

示例输出（主服务器）

图: gstat -g rss verbose 命令输出，自主服务器上运行

```
Local server type: Primary
Index page logging status: Enabled
Index page logging was enabled at: 2009/08/31 09:35:22
Number of RSS servers: 1
```

```
RSS Server information:

RSS Server control block: 0x5fdd9740
RSS server name: serv3
RSS server status: Active
RSS connection status: Connected
RSS flow control:576/528
Log transmission status: Active
Next log page to send(log id,page): 53,117632
Last log page acked(log id,page): 53,115615
Last log page applied (log id, page); 53,115615
Time of Last Acknowledgment: 2009-08-31.14:14:09
Pending Log Pages to be ACKed: 1984
Approximate Log Page Backlog:97104
Sequence number of next buffer to send: 3676
Sequence number of last buffer acked: 3612
Supports Proxy Writes: Y
```

输出描述（主服务器）

Local server type

主或 RSS（远程独立辅助）服务器类型

Index page logging status

显示是否启用或禁用主服务器和辅助服务器之间的索引页日志记录

Index page logging was enabled at

启用索引页日志记录的日期和时间

Number of RSS servers

连接到主服务器的 RS 辅助服务器数

RSS Server control block

RS 辅助服务器控制 block

RSS Server name

RS 辅助服务器名称

RSS Server status

显示 RS 辅助服务器是否活动

RSS flow control

逻辑日志页中的数值，代表确定何时启用或禁用流控制

RSS Connection status

RS 辅助服务器的连接状态

Log transmission status

显示日志事务是否激活

Next log page to send (log id, page)

要发送的下一个日志页的日志 ID 和页编号

Last log page acked (log id, page)

应答的上一个日志的日志 ID 和页编号

Last log page applied (log id, page)

最后应用日志的日志 ID 和页编号

Time of Last Acknowledgment

最后一次应答日志的时间

Pending Log pages to be ACKed

已发生但还未应答的日志数

Approximate Log Page Backlog

已发送的日志数和最后逻辑日志数的不同

Sequence number of next buffer to send

要发送的下一个缓冲区的序号

Sequence number of last buffer acked

应答的上一个缓冲区的序号

Supports Proxy Writes

显示该服务器当前是否配置了允许更新辅助服务器。 Y = 支持更新辅助服务器，N = 不支持更新辅助服务器

使用 log 选项的示例输出（主服务器）

图: gstat -g rss log 命令输出，自主服务器上运行

```
        Log Pages Snooped:
        RSS Srv          From      From      Tossed
        name              Cache     Disk      (LBC full)


        cdr_ol_nag_1_c1    1368       1331       0
        cdr_ol_nag_1_c2    1357       1342       0
        cdr_ol_nag_1_c3    1356       1343       0
```

使用 log 选项的输出描述（主服务器）

Log Pages Snooped

每台 RS 辅助服务器的统计信息

RSS Srv name

RS 辅助服务器名称

From Cache

来自高速缓存

From Disk

来自磁盘的日志

Tossed (LBC full)

由于 LBC 变满而抛弃的日志页数

示例输出（RS 辅助服务器）

图: gstat -g rss 命令输出，自 RS 辅助服务器上运行

```
        Local server type: RSS
        Server Status: Active
        Source server name: cdr_ol_nag_1
        Connection status: Connected
        Last log page received(log id,page): 7,877
```

输出描述（RS 辅助服务器）

Local server type

主或 RSS（远程独立辅助）服务器类型

Server Status

显示 RS 辅助服务器是否活动

Source server name

主服务器名称

Connection status

RS 辅助服务器的连接状态

Last log page received (log id,page)

最近接收到的日志 ID 和页面

使用 verbose 选项的示例输出（RS 辅助服务器）

图: gstat -g rss verbose 命令输出，自 RS 辅助服务器上运行

```
        RSS Server control block: 0x45a3fe58
        Local server type: RSS
        Server Status: Active
        Source server name: my_server
        Connection status: Connected
        Last log page received(log id,page): 10,1364
        Sequence number of last buffer received: 489
```

```
        Sequence number of last buffer acked: 489
        Delay Apply: Configured (3)
        Stop Apply: Not configured.
        Delay or Stop Apply control block: 0x45a40ba8
        Pending pages: 7
        Last page written: (10:1372).
        Next page to read: (10:1366).
        Delay or Stop Apply thread: Running.
```

使用 verbose 选项的输出描述（RS 辅助服务器）

RSS Server control block

服务器控制 block

Local server type

本地服务器的类型

Server Status

RS 辅助服务器的状态

Source server name

在 RS 辅助服务器的高可用性集群中的主服务器名称

Connection status

RS 辅助服务器和集群的主服务器之间的连接状态

Last log page received (log id,page)

RS 辅助服务器上一次应答的日志的日志 ID 和页编号

Sequence number of last buffer received

RS 辅助服务器接收的上一个缓冲区的序号

Sequence number of last buffer acked

RS 辅助服务器应答的上一个缓冲区的序号

Delay Apply

是否设置了延迟应用。延迟值，以秒表示，用括号括起

Stop Apply

是否设置了停止应用。停止值，用括号括起，它是 1 或 Unix 时间

Delay or Stop Apply control block

延迟或停止应用的控制 block

Pending pages

等待写入日志暂存目录的页数

Last page written

上一个写入日志暂存目录的日志的日志 ID 和页编号

Next page to read

下一个要写入日志暂存目录的日志的日志 ID 和页编号

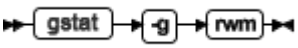
Delay or Stop Apply thread

延迟应用或停止应用线程的状态

gstat -g rwm 命令：打印读取和写入互斥

可以使用 gstat -g rwm 命令显示读、写和等待互斥线程的信息和这些线程获得的凭单的地址列表。

语法：



示例输出

图: gstat -g rwm 命令输出

```
MUTEX   NAME     write/read/wait    tcb list
            <address> <name>     first mutex
            Writer     ticket = <ticket address>   tcb=<thread address> <thread name>
            Readers    ticket = <ticket address>   tcb=<thread address> <thread name>
            Waiters    ticket = <ticket address>   tcb=<thread address> <thread name>
            <address> <name>        second mutex
            Writer     ticket = <ticket address>   tcb=<thread address> <thread name>
            Readers     ticket = <ticket address>  tcb=<thread address> <thread name>
            Waiters     ticket = <ticket address>  tcb=<thread address> <thread name>
            ....
            ....
            ....
            <address> <name>       last mutex
            Writer     ticket = <ticket address>   tcb=<thread address> <thread name>
            Readers    ticket = <ticket address>   tcb=<thread address> <thread name>
            Waiters    ticket = <ticket address>   tcb=<thread address> <thread name>
```

输出描述

tcb

线程地址列表

Writer

写线程列表

Readers

读线程列表

Waiters

正在等待的线程列表

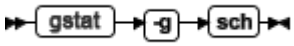
ticket

线程获得的凭单的地址

gstat -g sch 命令：打印 VP 信息

可以使用 gstat -g sch 命令显示线程迁移和每个虚拟处理器的信号量操作、自旋和忙等待的数量的信息。

语法：



示例输出

图: gstat -g sch 命令输出

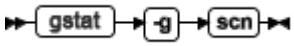
```
VP Scheduler Statistics:
 vp    pid       class       semops    busy waits  spins/wait
 1     3284      cpu         23997     0           0
 2     1340      adm         0         0           0
 3     4624      lio         2         0           0
 4     3320      pio         2         0           0
 5     6076      aio         7710      0           0
 6     4580      msc         46        0           0
 7     3428      soc         7         0           0
 8     2308      soc         1         0           0

Thread Migration Statistics:
 vp  pid      class steal-at steal-sc idlvp-at idlvp-sc inl-polls Q-ln
 1   3284     cpu   0        0        0        0        0         0
 2   1340     adm   0        0        0        0        0         0
 3   4624     lio   0        0        0        0        0         0
 4   3320     pio   0        0        0        0        0         0
 5   6076     aio   0        0        0        0        0         0
 6   4580     msc   0        0        0        0        0         0
 7   3428     soc   0        0        0        0        0         0
 8   2308     soc   0        0        0        0        0         0
```

gstat -g scn 命令；打印扫描选项

可以使用 gstat -g scn 命令显示当前扫描的状态和该扫描的信息。

如果您有一个长时间运行的扫描，那么可以使用该命令检查扫描的进程、在扫描结束前确定扫描所需的时间并可以查看该扫描的信息。对于表来说，gstat -g scn 命令的输出标识该扫描是轻度扫描还是缓冲池扫描。

语法：



示例输出

图: gstat -g scn 输出显示表信息

```
Light Scan Info
descriptor  address           next_lpage  next_ppage        ppage_left  bufcnt  look_aside

RSAM batch sequential scan info

SesID Thread Partnum  Rowid    Rows Scan'd Scan Type Lock Mode  Notes
48    68     10016e   12bb09   43146       Light     Table      Look aside,
40    47     100106   101      0           Buffpool  +Test      Must copy
```

有关运行扫描时索引扫描可用的信息。

图: gstat -g scn 输出显示索引扫描信息

```
RSAM batch index scan info

            SesID Thread Partnum  Scan Type Lock Mode  Notes
            136   156    100197             SLock+Test
            Start Key   GT   :-2147483648:
            Stop Key    EQ   :1500:
            Current key      :170:
            Current position: buffp 0x10a4bc0c8 pagenum 2 slot 17 rowoff 4 flags 0
```

输出描述

descriptor (decimal)

轻度扫描 ID

address (hex)

轻度扫描描述符的内存地址

next_lpage (hex)

要扫描的下一个逻辑页地址

next_ppage (hex)

要扫描的下一个物理页地址

ppage_left (decimal)

当前 extent 中剩余的物理页数

bufcnt

此次轻度扫描使用的轻度扫描缓冲区数

look_aside

此次轻度扫描是否要观察周边（Y = yes, N = no）。当线程需要在缓冲池中检查现有页以获得正在被轻度扫描的页的最新图像时，发生观察周边操作

SesID

会话 ID

Thread

线程 ID

Partnum

分区号

Rowid

当前行 ID

Rows Scan'd

已扫描的行数

Scan Type

对于表来说，是以下其中之一：

- Bufferpool
- Light（轻度扫描）

对于索引来说，是以下其中之一：

- key only
- 没有值，如果扫描不是 key only 扫描

Lock Mode

获得的锁的类型或没有锁：

- Table（表级别锁定）
- Slock（共享锁）
- Ulock（更新锁）
- blank（没有锁）

该列也显示了以下之一的值：

- +Test （扫描测试带有特定锁类型的冲突；没有获得锁）
- +Keep （获取的锁将保持到会话结束而不是事务结束）

Notes®

该列显示了以下之一的值：

- Look aside

  轻度扫描正在观察周边。

  轻度扫描直接从磁盘读取页的 block 到大缓冲区，而不是从缓冲区管理器获得每个页。在一些情况下，该进程要求此次轻度扫描检查缓冲池中每个数据页的情况下，来自它的一个大缓冲区的过程；这个过程称为 look aside。如果页在缓冲池中，那么此次轻度扫描会使用副本而不是轻度扫描大缓冲区中的部分。如果页不在缓冲池中，那么此次轻度扫描将会使用从磁盘读取到它的大缓冲区的副本。如果轻度扫描正在观察周边，那么会此次扫描的性能会稍微降低。

  在很多情况中，轻度扫描会检测到缓冲池不可能具有新版本的页。在这种情况下，此次轻度扫描将不会检查缓冲池也不会观察周边。

- Forward row lookup

  服务器正在对一个具有跨页的表执行轻度扫描。该轻度扫描必须访问和使用缓冲池来得到任何主页上不完整的行的剩余的片段。

Start key

扫描的起始键

Stop key

扫描的结束键

Current key

该扫描中的当前键

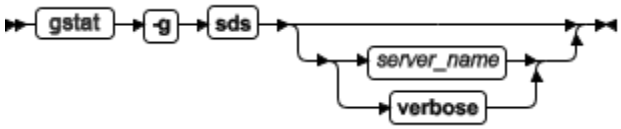
Current position

扫描在索引中的当前位置。例如：页、slot 和偏移量

gstat -g sds 命令：打印 SD 辅助服务器信息

可以使用 gstat -g sds 命令显示有关共享磁盘辅助服务器的信息。

语法：



gstat -g sds 命令的输出因命令在主服务器上还是 SD 辅助服务器上运行而稍有不同。

调用	解释
gstat -g sds	显示 SD 辅助服务器的简要信息
gstat -g sds verbose	显示 SD 辅助服务器的详细信息
gstat -g sds *server_name*	显示指定的 SD 辅助服务器的信息。当指定 *server_name* 时，该命令必须从主服务器声明

示例输出（主服务器）

图: gstat -g sds 命令输出，自主服务器上运行

```
Local server type: Primary
Number of SDS servers:1

SDS server information

SDS srv      SDS srv      Connection        Last LPG sent  Supports
name         status       status            (log id,page)  Proxy Writes
C_151162     Active       Connected             554,4998   Y
```

输出描述（主服务器）

Local server type

主或 SDS （共享磁盘辅助）服务器类型

Number of SDS servers

连接到主服务器的 SD 辅助服务器数

SDS Srv name

SD 辅助服务器的名称

SDS Srv status

显示 SD 辅助服务器是否活动

Connection status

显示 SD 辅助服务器是否连接

Last LPG sent (log id, page)

最近的 LPG 日志 ID 和页

Supports Proxy Writes

显示该服务器当前是否配置了允许更新辅助服务器。 Y = 支持更新辅助服务器，N = 不支持更新辅助服务器

使用 verbose 选项的示例输出（主服务器）

图: gstat -g sds server_name 命令输出，自主服务器上运行

```
Number of SDS servers:1
Updater node alias name: server_1
SDS server control block: 0x4d6a5e08
server name: server_2
server type: SDS
server status: Active
connection status: Connected
Last log page sent(log id,page):5,1829
Last log page flushed(log id,page):5,1829
Last log page acked (log id, page):5,1829
Last LSN acked (log id,pos):5,7492024
Last log page applied(log id,page): 5,1829
Approximate Log Page Backlog:0
Current SDS Cycle:1054
Acked SDS Cycle:1054
Sequence number of next buffer to send: 84329
Sequence number of last buffer acked: 84326
Time of last ack:2013/12/12 09:13:49
Supports Proxy Writes: N
Time of last received message: 2013/12/12 09:13:49
Time of last alternate write: N/A
Time of last alternate read : N/A
```

使用 verbose 选项的输出描述（主服务器）

Number of SDS servers

与主服务器共享磁盘空间的 SD 辅助服务器数

Updater node alias name

主服务器名

SDS server control block

SD 辅助服务器控制 block

server name

服务器名

server type

服务器的类型

server status

显示该服务器是否被激活

connection status

主服务器和辅助服务器之间连接的状态

Last log page sent (log id, page)

最近发送的日志页面的日志 ID 和页面

Last log page flushed (log id, page)

最近清空的日志页面的日志 ID 和页面

Last log page acked (log id, pos))

最近应答的日志页

Last LSN acked (log id, pos)

最近应答的日志序列号

Last log page applied(log id,page)

最近应用的日志的日志 ID 和页编号

Approximate Log Page Backlog

等待发送的日志数

Current SDS Cycle

由 GBase 支持内部使用来监视主服务器和 SDS 服务器之间的协调性

Acked SDS Cycle

由 GBase 支持内部使用来监视主服务器和 SDS 服务器之间的协调性

Sequence number of next buffer to send

要发送的下一个缓冲区的序号

Sequence number of last buffer acked

应答的下一个缓冲区的序号

Time of last ack

上一个日志确认的日期和时间

Supports Proxy Writes

显示该服务器当前是否配置了允许更新辅助服务器。 Y = 支持更新辅助服务器，N = 不支持更新辅助服务器

Time of last received message:

当前服务器最近一次从另一个服务器接收消息的时间戳

Time of last alternate write

当前服务器最近一次写入由 SDS_ALTERNATE 配置参数指定的 blobspace 的时间戳

Time of last alternate read

当前服务器最近一次读取由 SDS_ALTERNATE 配置参数指定的 blobspace 的时间戳

使用 verbose 选项的示例输出 （SD 辅助服务器）

图: gstat -g sds verbose 命令输出，自 SD 辅助服务器上运行

```
SDS server control block: 0xb299880
Local server type: SDS
Server Status : Active
Source server name: my_source_server
Connection status: Connected
Last log page received(log id,page): 7,884
Next log page to read(log id,page):7,885
Last LSN acked (log id,pos):7,3621272
Sequence number of last buffer received: 0
Sequence number of last buffer acked: 0
Current paging file:/dbspaces/page_my_source_server_sdc1_
Current paging file size:2048
Old paging file:/dbspaces/page_my_source_server_sdc1_
Old paging file size:10240
```

使用 verbose 选项的输出描述 （SD 辅助服务器）

SDS server control block

SD 辅助服务器控制 block

Local server type

主或 SDS（共享磁盘辅助）服务器类型

Server status

显示 SD 辅助服务器是否活动

Source server name

主服务器名

Connection status

显示 SD 辅助服务器是否已连接

Last log page received (log id, page)

最近接收的日志页面

Next log page to read (log id,page)

要读取的下一个日志页的序号

Last LSN acked (log id,pos)

最近应答的 LSN

Sequence number of last buffer received

接收的上一个缓冲区的序号

Sequence number of last buffer acked

应答的上一个缓冲区的序号

Current paging file

当前 GBase_8s paging 文件的名称

Current paging file size

当前 GBase_8s paging 文件的大小

Old paging file

上一个 GBase_8s paging 文件的名称

Old paging file size

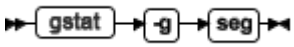
上一个 GBase_8s paging 文件的大小

gstat -g seg 命令：打印共享内存段的统计信息

可以使用 gstat -g seg 命令显示共享内存段的统计信息。

该命令显示附加了多少段以及它们的大小。您可以在转储文件（创建时不包含缓冲池）中运行 gstat -g seg命令。

语法：



示例输出

图: gstat -g seg 命令输出

```
Segment Summary:
            id         key        addr     size       ovhd     class blkused  blkfree
            720914    52e44801  44000000 4390912    248812  R    1072     0
            753683    52e44802  44430000 131072000  769136  V    22573    9427
            819221    52e44803  4c130000 66027520  1        B    16120    0
            851990    52e44804  50028000 83648512  1        B    20422    0
            Total:    -          -       285138944 -        -    60187    9427
            Virtual segment low memory reserve (bytes):4194304
            Low memory reserve used 0 times and used maximum block size 0 bytes
```

输出描述

id

共享内存分段 ID

key

与共享内存分段 ID 相关联的共享内存键

addr

共享内存分段地址

size

共享内存分段大小（字节）

ovhd

共享内存分段控制信息（开销）大小（字节）

class

共享内存的类型（B 代表共享内存池、V 代表虚拟、VX 代表虚拟扩展、M 代表消息）

blkused

已使用内存的 block 数

blkfree

空闲内存的 block 数

Virtual segment low memory reserve (bytes)

当需要必要活动且服务器限制了可用内存时，供使用的保留内存的大小（以字节表示）（可在 LOW_MEMORY_RESERVE 配置参数中指定保留内存）

Low memory reserve used 0 times and used maximum block size 0 bytes)

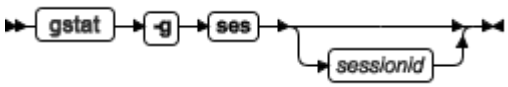
服务器使用保留内存的次数和最大的必需内存

gstat -g ses 命令：打印与会话有关的信息

可以使用 gstat -g ses 命令显示会话的信息。

缺省情况下，只有 DBSA 可以查看 gstat -g ses 信息。然而，当 UNSECURE_ONSTAT 配置参数设置为 1 时，所有的用户都可以查看该信息。

语法：



可以指定以下调用之一。

gstat -g ses

显示每个会话的单行摘要

gstat -g ses sessionid

显示特定会话的信息

所有会话的示例输出

图: gstat -g ses 命令输出

session					#RSAM	total	used	dynamic
id	user	tty	pid	hostname	threads	memory	memory	explain
24	gbasedbt	-	0	-	0	12288	7936	off
23	gbasedbt	-	17602	carson	1	57344	48968	off
3	gbasedbt	-	0	-	0	12288	9168	off
2	gbasedbt	-	0	-	0	12288	7936	off

特定会话的示例输出

图: gstat -g sessessionid 命令输出

session	effective					#RSAM	total	used	dynamic
id	user	user	tty	pid	hostname	threads	memory	memory	explain
53	gbasedbt	-	36	18638	apollo11	1	73728	63048	off

```
Program :
 /usr/gbasedbt/bin/dbaccess
 tid      name     rstcb            flags    curstk   status
 77       sqlexec  4636ba20         Y--P---  4240     cond wait  sm_read   -
 Memory pools    count 1
 name    class addr          totalsize  freesize   #allocfrag #freefrag
 53       V     4841d040       73728      10680      84         6

         name           free       used           name           free       used
         overhead       0          3288           scb            0          144
         opentable      0          2904           filetable      0          592
         log            0          16536          temprec        0          2208
         gentcb         0          1656           ostcb          0          2920
         sqscb          0          21296          sql            0          72
         hashfiletab    0          552            osenv          0          2848
         sqtcb          0          7640           fragman        0          392

  sqscb info
   scb              sqscb            optofc   pdqpriority optcompind  directives
   481b70a0         483e2028         0        0           0           1

   Sess       SQL            Current            Iso Lock       SQL  ISAM F.E.
    Id         Stmt type      Database           Lvl Mode       ERR  ERR  Vers  Explain
    53         -              sysmaster          CR  Not Wait   0    0    9.24  Off

         Last parsed SQL statement :
         Database 'sysmaster@lxl'

         Xadatasources participated in this session :
         Xadatasource name                                      RMID       Active
         xabasicdb@atmol10:sitaramv.xads_t3_i1                     6        YES
         xabasicdb@atmol10:sitaramv.xads_t2_i1                     4        YES
         xabasicdb@atmol10:sitaramv.xads_t1_i3                     3        YES
         xabasicdb@atmol10:sitaramv.xads_t1_i2                     2        YES
         xabasicdb@atmol10:sitaramv.xads_t1_i1                     1        YES
         xabasicdb@atmol10:sitaramv.xads_t2_i2                     5        NO

   DRDA client info
```

```
    Userid:
    Wrkstnname:   nemea
    Applname:     db2jcc_application
    Acctng:       JCC03510nemea
    Programid:
    Autocommit:
    Packagepath:
```

输出描述：session 部分

Session id

会话 ID

user

启动该会话的用户名

tty

与此会话的前端关联的 tty

pid

与此会话的前端关联的进程 ID

hostname

此会话已连接的主机名

#RSAM threads

为此会话分配的 RSAM 线程的数量

total memory

为此会话分配的内存量

used memory

此会话实际使用的内存量

dynamic explain

生成会话的 SQL 语句的说明输出（on 或 off）

输出描述：program 部分

显示在您的会话中使用的客户端程序的完整路径。使用客户端程序信息来监视或停止访问数据库。

输出描述：threads 部分

尽管该部分没有标题，但是以下输出显示有关线程的信息。

tid

线程 ID

name

线程的名称

rstcb

RSAM 控制 block

flags

使用以下代码描述线程的状态：

位置 1

B

正在等待缓冲区

C

正在等待 checkpoint

G

正在等待逻辑日志缓冲区写入

L

正在等待锁定

S

正在等待互斥

T

正在等待事务

X

正在等待事务清除

Y

正在等待条件

位置 2

*

此位置中的星号表示线程在事务中遇到 I/O 失败

位置 3

A

归档线程

B

开始工作

P

开始准备或已准备好工作

X

XA 已准备好

C

正在提交或已提交

R

正在异常终止或已异常终止

H

尝试异常终止或尝试回滚

位置 4

P

主线程

位置 5

R

正在读取

X

临界区

位置 6

R

恢复线程

位置 7

M

监视线程

D

守护线程

C

清除程序

F

清空程序

B

B-tree 扫描程序

curstk

当前堆栈大小

status

当前线程状态

输出描述： memory pools 头部分

每个会话池重复这些信息。

name

池名称

class

池所分配位置的内存类。R 代表常驻、V 代表虚拟、M 代表消息

addr

池结构的地址

totalsize

池获得的内存的总的大小，以字节表示

freesize

在池中的可用字节数量

#allocfrag

在池中已分配内存分片的数量

#freefrag

在池中的可用分片数量

输出描述：Memory pools 部分

name

已经从池中分配内存的组件的名称

free

释放的字节数量

used

分配的字节数量

输出描述：sqscb info 部分

scb

会话控制 block。这是共享内存中主会话结构的地址

sqscb

会话的 SQL 级别控制 block

optofc

OPTOFC 环境变量或 ONCONFIG 配置文件设置的当前值

pdqpriority

PDQPRIORITY 环境变量或 ONCONFIG 配置文件设置的当前值

optcompind

OPTCOMPIND 环境变量或 ONCONFIG 配置文件设置的当前值

directives

DIRECTIVES 环境变量或 ONCONFIG 配置文件设置的当前值

输出描述：SQL 部分

显示特定会话的 SQL 信息。该部分具有和 gstat -g sql 命令相同的信息。请参阅 gstat -g sql 命令：打印与 SQL 有关的会话信息。

输出描述：Last parsed SQL statement 部分

Last parsed SQL statement 部分具有和 gstat -g sql 命令相同的信息。请参阅 gstat -g sql 命令：打印与 SQL 有关的会话信息。

输出描述：Xadatasources participated in this session 部分

Xadatasources participated in this session 部分显示有关在会话期间可用的 XA 数据源的信息。它们的资源管理器标识，以及它们当前是否是活动的。

Xdatasource name

参与会话的 XA 数据源

RMID

对应的 XA 数据源的资源管理器的标识

Active

XA 数据源是否依然是活动的

输出描述：DRDA client info 部分

DRDA client info 显示连接客户端的分布式关系数据库体系结构（DRDA） 的信息。

Userid

客户端用户的用户 ID

Wrkstnname

客户端工作站名称

Applname

客户端应用程序名称，例如：db2jcc_application

Acctng

来自客户端的审计字符串，例如：JCC03510nemea

Programid

客户端程序标识（ GBase 8s 没有使用）

Autocommit

GBase 8s 数据源缺省的事务自动提交方式

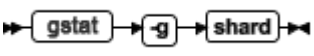
Packagepath

客户端包装路径（ GBase 8s 没有使用）

gstat -g shard 命令：打印有关分片高速缓存的信息

可以使用 gstat -g shard 命令显示有关分片高速缓存的信息。

语法：



示例 1：使用基于散列分片的分片定义的输出

对于本示例，使用以下命令创建一个分片定义：

```
cdr define shardCollection collection_1 database_1:josh.customers_1
          --type=delete --key=column_2 --strategy=hash --versionCol=column_3
          g_shard_server_A
          g_shard_server_B
          g_shard_server_C
          g_shard_server_D
```

以下示例显示了当 gstat -g shard 命令在 g_shard_server_A 、g_shard_server_B 、g_shard_server_C 或 g_shard_server_D 上运行的输出。

图: 使用哈希算法将数据分布到多个数据库服务器的分片定义的 gstat -g shard 命令的输出。

```
GBase 8s Database Server Version 8.8 -- On-Line -- Up 00:00:20 -- 162316 Kbytes
collection_1 database_1:josh.customers_1 key:column_2 HASH:DELETE SHARD
OPTIMIZATION:ENABLED
Matching for delete:column_3
g_shard_server_A (65545) mod(ifx_checksum(column_2::LVARCHAR, 0), 4) = 0
g_shard_server_B (65546) mod(ifx_checksum(column_2::LVARCHAR, 0), 4) in (1, -1)
g_shard_server_C (65547) mod(ifx_checksum(column_2::LVARCHAR, 0), 4) in (2, -2)
g_shard_server_D (65548) mod(ifx_checksum(column_2::LVARCHAR, 0), 4) in (3, -3)
```

示例 1 的输出描述

Sharding definition name

分片定义的名称。在此示例中该值为 collection_1

Database name

包含分布在多个分片的集合或表的数据库的名称。在此示例中该值为 database_1

Table owner name

分布在多个分片的集合或表的所有者名称。在此示例中该值为 josh

Table name

分布在多个分片的集合或表的名称。在此示例中该值为 customers_1

Shard key

分片键用于分配行或文档。值可以是表的一个列、文档字段或一个表达式。在此示例中该值为 column_2

Sharding strategy

确定新行或文档存储在哪个数据库服务器的方法。值可以是 HASH （哈希算法）或 EXPRESSION（表达式）。在此示例中该值为 HASH

Sharding type

在行或文档复制到目标服务器后指定源服务器的动作。值可以是 DELETE 、KEEP 或 INFORMATIONAL 。 在此示例中该值为 DELETE

Shard optimization

指定查询是否可以跳过不包含相关数据的分片服务器。值可以是 ENABLED 或 NOT ENABLED 。在此示例中该值为 ENABLED

Version column

指定 Enterprise Replication 尝试验证源行或源文件没有被更新时使用的列或键。值可以是一列或文档字段。 在此示例中该值为 column_3

Sharding rule

复制数据到指定的数据库服务器的规则。在上一示例中，g_shard_server_A （服务器编号 65545）基于该规则来发送数据：

mod(ifx_checksum(col2::LVARCHAR,0),4)=0

示例 2：使用基于分片表达式的分片定义的输出

对于本示例，使用以下命令创建一个分片定义：

```
cdr define shardCollection collection_2 database_2:john.customers_2
          --type=keep --key=state --strategy=expression --versionCol=version_column
          g_shard_server_F "IN ('AL','MS','GA')"
          g_shard_server_G "IN ('TX','OK','NM')"
          g_shard_server_H "IN ('NY','NJ')"
          g_shard_server_I REMAINDER
```

以下示例显示了运行在 g_shard_server_F 、g_shard_server_G 、g_shard_server_H 或 g_shard_server_I 上的 gstat -g shard 命令的输出。

图: 使用一个表达式将数据分布到多个数据库服务器的分片定义的 gstat -g shard 命令的输出。

```
GBase 8s Database Server Version 8.8.U -- On-Line -- Up 00:19:07 -- 162316 Kbytes
collection_2 database_2:john.customers_2 key:state EXPRESSION:KEEP SHARD
OPTIMIZATION:ENABLED
Matching for delete:version_column
g_shard_server_F (65564) state IN ('AL','MS','GA')
g_shard_server_G (65565) state IN ('TX','OK','NM')
g_shard_server_H (65566) state IN ('NY','NJ')
g_shard_server_I (65567) not ((state IN ('AL','MS','GA')) or (state IN('TX','OK','NM'))
or (state IN ('NY','NJ')))
```

示例 2 的输出描述

Sharding definition name

分片定义的名称。在此示例中该值为 collection_2

Database name

包含分布在多个分片的集合和表的数据库的名称。在此示例中该值为 database_2

Table owner name

分布在多个分片的集合或表的所有者名称。在此示例中该值为 john

Table name

分布在多个分片的集合或表的名称。在此示例中该值为 customers_2

Shard key

分片键用于分配行或文档。值可以是表的一个列、文档字段或一个表达式。在此示例中该值为 state

Sharding strategy

确定新行或文档存储在哪个数据库服务器的方法。值可以是 HASH （哈希算法）或 EXPRESSION（表达式）。在此示例中该值为 EXPRESSION

Sharding type

在行或文档复制到目标服务器后指定源服务器的动作。值可以是 DELETE 、KEEP 或 INFORMATIONAL 。 在此示例中该值为 KEEP

Shard optimization

指定查询是否可以跳过不包含相关数据的分片服务器。值可以是 ENABLED 或 NOT ENABLED 。在此示例中该值为 ENABLED

Version column

指定 Enterprise Replication 尝试验证源行或源文件没有被更新时使用的列或键。值可以是一列或文档字段。 在此示例中该值为 version_column

Sharding rule

复制数据到指定的分片的规则。在本示例中 g_shard_server_F（服务器编号为 65564 ）基于该规则接收数据：

state in ('AL','MS','GA')

g_shard_server_I（服务器编号为 65567） 基于该规则接收数据：

```
not ((state in ('AL','MS','GA'))
or (state in ('TX','OK','NM'))
or (state in ('NY','NJ')))
```

示例 3：使用 BSON 分片键和基于分片表达式的分片定义的输出

对于本示例，使用以下命令创建一个分片定义：

```
cdr define shardCollection collection_3 database_3:susan.customers_3
          -t delete -k bson_value_lvarchar(data,'age') -s expression -v version
          g_shard_server_J "BETWEEN 0 and 20"
          g_shard_server_K "BETWEEN 21 and 62"
          g_shard_server_L "BETWEEN 63 and 100"
          g_shard_server_M  REMAINDER
```

以下示例显示运行在 shard_server_J 、shard_server_K 、shard_server_L 或 shard_server_M 上的 gstat -g shard 命令的输出。

图: 使用 BSON 分片键和表达式将数据分布到多个数据库服务器的分片定义的 gstat -g shard 的命令的输出。

```
GBase 8s Database Server Version 8.8 -- On-Line -- Up 01:34:01 -- 354721 Kbytes
          collection_3 database_3:susan.customers_3 key:bson_value_lvarchar(data,'age')
          EXPRESSION:DELETE SHARD OPTIMIZATION:ENABLED
```

```
        Matching for delete:version
        g_shard_server_J (65568) bson_value_lvarchar(data,'age') BETWEEN 0 and 20"
        g_shard_server_K (65569) bson_value_lvarchar(data,'age') BETWEEN 21 and 62"
        g_shard_server_L (65570) bson_value_lvarchar(data,'age')BETWEEN 63 and 100"
        g_shard_server_M (65571) not((bson_value_lvarchar(data,'age') BETWEEN 0 and 20)
        or (bson_value_lvarchar(data,'age') BETWEEN 21 and 62) or (bson_value_lvarchar
        (data,'age') BETWEEN 63 and 100))
```

示例 3 的输出描述

Sharding definition name

分片定义的名称。在此示例中的该值为 collection_3

Database name

包含分布在多个分片的集合或表的数据库的名称。在此示例中该值为 database_3

Table owner name

分布在多个分片的集合或表的所有者名称。在此示例中该值为 susan

Table name

分布在多个分片的集合或表的名称。在此示例中该值为 customers_3

Shard key

分片键用于分配行或文档。值可以是表的一个列、文档字段或一个表达式。在此示例中该值为查询 BSON age 键作为该分片键的表达式 bson_value_lvarchar(data,'age')

Sharding strategy

确定新行或文档存储在哪个数据库服务器的方法。值可以是 HASH （哈希算法）或 EXPRESSION（表达式）。在此示例中该值为 EXPRESSION

Sharding type

在行或文档复制到目标服务器后指定源服务器的动作。值可以是 DELETE 、KEEP 或 INFORMATIONAL 。 在此示例中该值为 DELETE

Shard optimization

指定查询是否可以跳过不包含相关数据的分片服务器。值可以是 ENABLED 或 NOT ENABLED 。在此示例中该值为 ENABLED

Version column

指定 Enterprise Replication 尝试验证源行或源文件没有被更新时使用的列或键。值可以是一列或文档字段。 在此示例中该值为 version

Sharding rule

复制数据到指定的分片的规则。在本示例中 g_shard_server_J（服务器编号为 65568）基于该规则接收数据：

bson_value_lvarchar(data,'age') BETWEEN 0 and 20

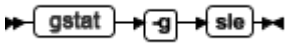
g_shard_server_M（服务器编号为 65571） 基于该规则接收数据：

```
not((bson_value_lvarchar(data,'age') BETWEEN 0 and 20)
  or (bson_value_lvarchar(data,'age') BETWEEN 21 and 62)
  or (bson_value_lvarchar(data,'age') BETWEEN 63 and 100))
```

gstat -g sle 命令：打印所有休眠的线程

可以使用 gstat -g sle 命令打印所有休眠的线程。

语法：



示例输出

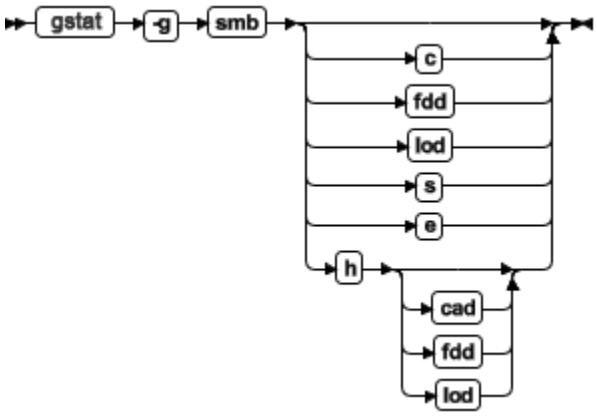
图: gstat -g sle 命令输出

```
Current Admin VP sleep period:  10  millisecs
Sleeping threads with timeouts: 21  threads
   tid v_proc            rstcb             name      time
    49      1          b3b13a8       onmode_mon      0.02
     5      1                0  Cosvr Avail Mgr      0.05
    42      1          b3ad028      main_loop()      0.08
     9      3          b3ad6e8         xtm_svcc      0.64
    14      5                0       mgmt_thd_5      0.65
    13      4                0       mgmt_thd_4      0.65
     4      1                0       mgmt_thd_1      0.65
     6      3                0          dfm_svc      0.98
    33     13                0      mgmt_thd_13      1.54
    27     10                0      mgmt_thd_10      1.54
    21      7                0       mgmt_thd_7      1.54
    12      3                0       mgmt_thd_3      1.76
    29     11                0      mgmt_thd_11      1.76
    23      8                0       mgmt_thd_8      2.08
    31     12                0      mgmt_thd_12      2.08
    35     14                0      mgmt_thd_14      2.98
    19      6                0       mgmt_thd_6      3.00
    25      9                0       mgmt_thd_9      3.00
    37      3                0          sch_rgm      3.48
    44      5          b3af8a8      btscanner 0      7.31
    46      3          b3b0628        bum_sched     41.26
```

gstat -g smb 命令：打印 sbspace 信息

可以使用 gstat -g smb 命令显示有关 sbspace 的详细信息。

语法：



命令	解释
gstat -g smb c	列出 sbspace 中的所有的 chunk
gstat -g smb e	列出所有智能大对象表类型的条目
gstat -g smb e cad	列出智能大对象 chunk 头表条目
gstat -g smb e fdd	列出智能大对象文件描述符条目
gstat -g smb e lod	列出智能大对象头表中的条目
gstat -g smb fdd	列出智能大对象文件描述符
gstat -g smb h	列出所有智能大对象表类型头
gstat -g smb h cad	列出智能大对象 chunk 头表头
gstat -g smb h fdd	列出智能大对象文件描述符表头
gstat -g smb h lod	列出智能大对象文头表的表头
gstat -g smb lod	列出在智能大对象头表中的头和条目
gstat -g smb s	列出 sbspace 属性（所有者、名称、页大小、-Df 标识设置）Sbspace 创建过程中不初始化值为 0 或 -1 的字段

gstat -g smb c 命令的示例输出

使用 gstat -g smb c 命令监视每个 sbspace chunk 中的可用空间量，以及用户数据、元数据和保留区域以页面为单位的大小。gstat -g smb c 命令显示每个 sbspace chunk 的以下信息：

- Chunk 编号和 sbspace 名称
- Chunk 大小和路径名
- 总用户数据页和可用的用户数据页
- 在每个用户数据和元数据区域中的页面的位置和数据

在以下示例中，sbspace 的 chunk 2 具有 2253 个源可用页（orig fr） 2253 个用户页（usr pgs）和 2245 个可用页（free pg）。对于第一个用户数据区域 （Ud1），起始页面偏移量为 53 ，页面数量为 1126。对于元数据区域（Md），起始页的偏移量为1179 ，页面数量为 194。对于第二个用户数据区域（Ud2），起始页的偏移量为 1373 ，页面数量为 1127 。

```
Chunk Summary:

         sbnum 2  chunk 2
         chunk:   address   flags    offset   size   orig fr   usr pgs   free pg
         303cf2a8  F------- 0        2500   2253      2253      2245
         path: /usr11/myname/sbspace1

         start pg   npages
         Ud1  :   53         1126
         Md   :   1179       194
         Ud2  :   1373       1127
```

gstat -g smb s 命令的输出

gstat -g smb s 命令显示系统中的所有 sbspace 的存储属性：

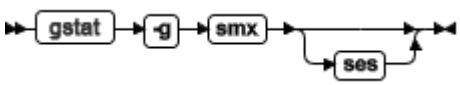
- sbspace 名称、标志、所有者
- 日志记录状态
- 平均智能大对象大小
- 第一个 extent 大小、下一个 extent 大小和最小 extent 大小
- 最大 I/O 访问时间
- 锁定方式

有关 gstat -g smb 命令的更多信息，请参阅 GBase 8s 性能指南 。

gstat -g smx 命令：打印多路复用器组信息

可以使用 gstat -g smx 命令显示有关使用 SMX 服务器的多路复用器组信息。

语法：



命令	解释
gstat -g smx	显示 SMX 连接的统计信息
gstat -g smx ses	显示 SMX 会话的统计信息

示例输出

图: gstat -g smx 命令输出

```
SMX connection statistics:
     SMX control block: 0x47d5e028
     Peer server name: lx1
```

```
        SMX connection address: 0x47d60d10
        Encryption status: Disabled
        Total bytes sent: 27055
        Total bytes received: 2006989
        Total buffers sent: 782
        Total buffers received: 7090
        Total write calls: 782
        Total read calls: 7090
        Total retries for write call: 0
        Data compression level: 1
        Data sent: compressed 40760 bytes by 33%
        Data received: compressed 12579324 bytes by 84%
```

输出描述

SMX control block

SMX 控制 block

Peer server name

显示同级服务器的名称

SMX connection address

显示 SMX 连接地址的信息

Encryption status

显示是否已启用或禁用加密

Total bytes sent

显示已发送的字节总数

Total bytes received

显示已接收的字节总数

Total buffers sent

显示已发送的缓冲区总数

Total buffers received

显示已接收的缓冲区总数

Total write calls

显示写入调用的总数

Total read calls

显示读取调用的总数

Total retries for write call

显示写入调用的重试总数

Data compression level

显示由 SMX_COMPRESS 配置参数设置的 SMX 压缩级别

Data sent: compressed x bytes by y%

显示已发送的数据的已解压的字节数和压缩率

Data received: compressed x bytes by y%

显示已接收的数据的已解压的字节数和压缩率

示例输出

图: gstat -g smx ses 输出

```
SMX session statistics:
       SMX control block: 0x17c69028
       Peer        SMX session        client        reads        writes
       name        address            type
       delhi_sec     19022050    smx Clone Send            6          183
```

输出描述

SMX control block

SMX 控制 block

Peer name

显示同级别的服务器名称

SMX session address

SMX 会话地址

Client type

显示辅助服务器的类型

reads

显示会话读取的总数

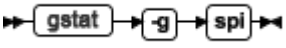
writes

显示会话写入的总数

gstat -g spi 命令：打印使用长自旋的自旋锁

可以使用 gstat -g spi 命令显示有关使用长自旋的自旋锁的信息。

语法：



服务器中的许多资源由两个或更多的线程访问。在一些访问（诸如更新共享值）中，服务器必须确保每次只有一个线程在访问该资源。spin lock 是用来为一些资源提供互斥存取的机制。有了这类锁，在第一次尝试时由于另一个线程占用锁而没有成功获取锁的线程重复尝试获取锁，直至成功为止。

自旋锁的成本很低，而且自旋锁通常用于在短期内获取互斥的资源。然而，如果自旋锁被高度使用，那么循环重试机可能会变得更贵。

gstat -g spi 命令对于帮助识别由于高度使用自旋锁而形成的性能瓶颈很有帮助。该选项列出了带有等待的自旋锁，在线程第一次尝试时没有成功为其获取锁而循环重试的自旋锁。

示例输出

图: gstat -g spi 命令输出

```
Spin locks with waits:

Num Waits   Num Loops   Avg Loop/Wait    Name
114          117675      1032.24          lockfr3
87          256461      2947.83          fast mutex, lockhash[832]
1             11         11.00           fast mutex, 1:bhash[16668]
4           51831     12957.75           fast mutex, 1:lru-4
1             490        490.00           fast mutex, 1:bf[994850] 0xe00002 0x14eb32000
```

输出描述

Num Waits (decimal)

为该自旋锁等待的线程总数

Num Loops (decimal)

在线程成功获取自旋锁前的尝试总数

Avg Loop/Wait (floating point)

用来获取自旋锁的尝试的平均数。以 Num Loops / Num Waits 计算

Name (string)

使用以下代码命名自旋锁：

lockfr

锁可用列表。lockfr 后的数字是锁可用列表数组中的索引。

lockhash[]

锁哈希桶。方括号内的字段是该锁哈希桶存储数组的索引

:bhash []

缓冲区哈希桶。冒号之前的字段是缓冲池的索引；方括号内在 bhash 之后的字段是该缓冲区哈希桶数组的索引

:lru-

LRU latch 。冒号之前的字段是缓冲池的索引；lru- 之后的字段标识缓冲区链对正在被使用

:bf[]

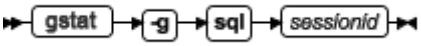
缓冲区 latch 。冒号之前的字段是缓冲池的索引；方括号内 bf 后的字段是该缓冲区数组的位置。接下来的两个字段分别是以十六进制表示的缓冲区内存的分区号和页头地址

gstat -g sql 命令：打印与 SQL 有关的会话信息

可以使用 gstat -g sql 命令显示有关会话的 SQL 相关信息。

缺省情况下，只有 DBSA 可以查看 gstat -g sql syssqltrace 信息。然而当 UNSECURE_ONSTAT 配置参数设置为 1 时，所有的用户都可以查看该信息。

语法：



可以指定以下调用之一。

调用

解释

gstat -g sql

显示每个会话的单行摘要

gstat -g sqlsessionid

显示特定会话的 SQL 信息

注： 不显示加密函数中的加密密码和密码提示参数。下图显示了在 Last parsed SQL statement 字段显示加密密码。

图: gstat -g sql 命令输出

```
gstat -g sql 22

           Sess  SQL            Current            Iso Lock       SQL  ISAM F.E.          Current
           Id    Stmt type      Database           Lvl Mode       ERR  ERR  Vers Explain    Role
           22     -             test               CR  Not Wait    0    0   9.03 Off         hr
           Last parsed SQL statement :
           select id, name, decrypt_char(ssn, 'XXXXXXXXXX') from emp
```

输出描述

Sess id

会话标识

SQL Stmt type

SQL 语句类型

Current Database

会话的当前数据库的名称

ISO Lvl

隔离级别

DR

Dirty 读取

CR

已提交读取

CS

游标锁定

DRU

Dirty 读取，保留更新锁

CRU

已提交读取，保留更新锁

CSU

游标锁定，保留更新锁

LC

已提交读取，最后提交

LCU

已提交读取，最后提交，保留更新锁

RR

可重复的读取

NL

没有事务的数据库

Lock mode

当前会话的锁定方式

SQL Error

当前语句遇到的 SQL 错误号

ISAM Error

当前语句遇到的 ISAM 错误号

F.E. Version

当前客户端程序使用的 SQLI 协议的版本

Explain

SET EXPLAIN 设置

Current Role

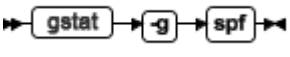
当前用户的角色

gstat -g spf 命令：打印已就绪语句的概要文件

可以使用 gstat -g spf 命令显示有关 SQL 查询的当前统计信息。

可以使用该统计信息确定每条语句的成本。

语法：



如果启用了 SQL 跟踪，该信息显示的是由此语句完成的工作的快照并且它可能随着语句继续运行而改变。例如：要监视一条活动的语句中缓冲区读或写的增长率，可以在 2 秒的间隔中发起三个 gstat -g spf 运行。

如果禁用了 SQL 跟踪，那么 "Statistics disabled" 会声明一个警告消息：

示例输出

图: gstat -g spf 命令输出

```
Statement profiles
      sid  sdb      tottm  execs runtm  pdq scans sorts bfrd  pgrd  bfwrt  pgwrt  lkrqs  lkwts
      35   4de84028 0.01   0     0.01   0   0     0     301   352   0      512    2998   0
      25   4dc0b028 0.00   0     0.00   0   0     0     0     0     0      0      0      0
      ...
```

输出描述

sid

会话 ID

sdb

该语句指针的后 8 位数字

tottm

以秒表示所有语句当前运行的总时间

execs

已运行完成的语句的当前数量。该值不包含正在运行的语句。

runtm

以秒表示该语句当前的运行时间

pdq

当前并发数据库查询（PDQ）优先级别。 PDQ 优先值可以是 0 到 100 之间的任意整数。

有关更多信息，请参阅 Managing PDQ queries

scans

当前已分配的 PDQ 扫描的数量

sorts

当前已完成排序的数量

bfrd

当前的缓冲区读取数

pgrd

当前的页读取数

bfwrt

当前的缓冲区写入数

pgwrt

当前的页写入数

lkrqs

当前的锁请求数

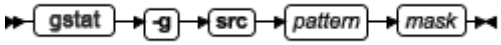
lkwts

当前的锁等待数

gstat -g src 命令：共享内存中的模式

可以使用 gstat -g src 命令搜索共享内存中的模式。

语法：



示例输出

以下示例显示了 gstat -g srcpattern mask 命令的输出，这里 pattern = 0x123 而 mask = 0xffff 。

图: gstat -g src 命令输出

```
Search Summary:
 addr             contents
 000000000ad17a50: 01090000 00000000 00000000 00000123   ........ .......#
 000000000ad7dec0: 00000001 014e3a0c 00000000 0ade0123   .....N:. .......#
```

输出描述

addr (hexadecimal)

找到搜索模式的共享内存地址

contents (hexadecimal)

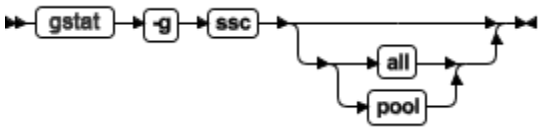
给出地址上的内存内容

gstat -g ssc 命令：打印出现的 SQL 语句

可以使用 gstat -g ssc 命令监视数据库服务器读取高速缓存中的 SQL 语句的次数。

缺省情况下，只有 DBSA 可以查看 gstat -g ssc syssqltrace 信息。然而当 UNSECURE_ONSTAT 配置参数设置为 1 时，所有的用户都可以查看该信息。

语法；



all 选项报告 key-only 高速缓存的条目和完全高速缓存的语句 。如果 hits 列中的值比 STMT_CACHE_HITS 值小，该条目是 key-only 高速缓存条目。有关更多信息，请参阅 GBase 8s 性能指南 中的内存利用率。

pool 选项报告该 SQL 语句高速缓存的所有内存池的用法。该输出显示内存池的名称、类型、地址和总大小的信息。有关更多信息，请参阅 GBase 8s 性能指南 中的提高查询性能。

示例输出

图: gstat -g ssc 命令输出

```
      Statement Cache Summary:
      #lrus   currsize  maxsize   Poolsize  #hits   nolimit
```

```
4      117640    524288    139264    0      1
Statement Cache Entries:
lru hash ref_cnt hits flag heap_ptr    database         user
---------------------------------------------------------------
0  262      0    7   -F  aad8038     sscsi007         admin
INSERT INTO ssc1 ( t1_char , t1_short , t1_key , t1_float , t1_smallfloat
, t1_decimal , t1_serial ) VALUES ( ? , ? , ? , ? , ? , ? , ? )
0  127      0    9   -F  b321438     sscsi007         admin
INSERT INTO ssc2 ( t2_char , t2_key , t2_short ) VALUES ( ? , ? , ? )
1  134      0   15   -F  aae0c38     sscsi007         admin
SELECT t1_char , t1_short , t1_key , t1_float , t1_smallfloat ,
t1_decimal , t1_serial FROM ssc1 WHERE t1_key = ?
1  143      0    3   -F  b322c38     sscsi007         admin
INSERT INTO ssc1 ( t1_char , t1_key , t1_short ) SELECT t2_char , t2_key
+ ? , t2_short FROM ssc2
2   93      0    7   -F  aae9838     sscsi007         admin
DELETE FROM ssc1 WHERE t1_key = ?
2  276      0    7   -F  aaefc38     sscsi007         admin
SELECT count ( * ) FROM ssc1
2  240      1    7   -F  b332838     sscsi007         admin
SELECT COUNT ( * ) FROM ssc1 WHERE t1_char = ? AND t1_key = ? AND
t1_short = ?
3   31      0    7   -F  aaec038     sscsi007         admin
SELECT count ( * ) FROM ssc1 WHERE t1_key = ?
3   45      0    1   -F  b31e438     sscsi007         admin
DELETE FROM ssc1
3  116      0    0   -F  b362038     sscsi007         admin
SELECT COUNT ( * ) FROM ssc1
Total number of entries: 10.
```

输出描述 - Statement Cache Summary 部分

#lrus

最近最少使用的队列（LRUS）的次数

currsize

当前高速缓存的大小

maxsize

总高速缓存内存量限制

Poolsize

总的池大小

#hits

在插入前的命中的数量。这个数量等于 STMT_CACHE_HITS 配置参数的值r

nolimit

STMT_CACHE_NOLIMIT 配置参数的值

输出描述 - Statement Cache Entries 部分

Statement Cache Entries 部分显示已全部插入高速缓存的条目。

lru

高速缓存条目所属的 lru 队列的索引

hash

高速缓存条目的哈希值

ref_count

引用语句的线程的数量

hits

语句与高速缓存中的语句匹配的次数。匹配可用于 key-only 或完全高速缓存的条目

flag

高速缓存条目标志；-F 说明语句是完全高速缓存的；-D 说明语句已删除

heap_ptr

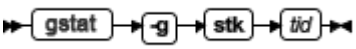
高速缓存条目的内存堆的地址

gstat -g stk 命令：打印线程堆栈

可以使用 gstat -g stk tid 命令显示由线程 ID 指定的线程堆栈。

所有的平台都不支持该选项，并且该选项并非一直精确。

语法：



示例输出

图: gstat -g stk tid 命令输出

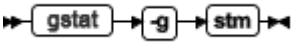
```
Stack for thread: 2 adminthd
 base: 0x000000010aad5028
  len:   33280
   pc: 0x00000001002821e8
  tos: 0x000000010aadc621
state: running
   vp: 2

0x1002821e8 oninit :: yield_processor + 0x260 sp=0x10aadce20(0x10ac834d0, 0x0, 0x1,
   0x100000000, 0xc8a000, 0x100c8a000)
0x100274e38 oninit :: wake_periodic + 0xdc sp=0x10aadced0 delta_sp=176(0x41b0, 0xc7a024bc,
   0x0, 0x41c4, 0x10aacf598, 0x90)
0x100274fcc oninit :: admin_thread + 0x108 sp=0x10aadcf80 delta_sp=176(0x0, 0x2328,
   0xd26c00, 0x5, 0xc8a000, 0x156c)
0x1002484ec oninit :: startup + 0xd8 sp=0x10aadd050 delta_sp=208(0xa, 0x10aad47d0,
   0x10aad47d0, 0x100db1988, 0xd1dc00, 0x1)
```

gstat -g stm 命令：打印 SQL 语句的内存使用

可以使用 gstat -g stm 命令显示每个准备好的 SQL 语句所使用的内存。

缺省情况下，只有 DBSA 可以查看 gstat -g stm syssqltrace 信息。然而，当 UNSECURE_ONSTAT 配置参数设置为 1 时，所有的用户都可以查看此信息。

语法：



要只显示一个会话的内存，请在 gstat -g stm 命令中指定会话 ID。

示例输出

图: gstat -g stm 命令输出

```
session   65 ------------------------------------------------------------
sdblock  heapsz  statement ('*' = Open cursor)
aad8028   16544   SELECT COUNT ( * ) FROM ssc1 WHERE t1_char = ?
AND t1_key = ? AND t1_short = ?
```

输出描述

sdblock

语句描述符 block 的地址

heapsz

语句内存堆的大小

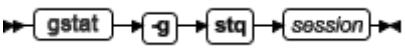
statement

查询文本

gstat -g stq 命令：打印队列信息

可以使用 gstat -g stq 命令显示有关队列的信息。

语法：



要查看特定的会话信息，请指定 session 选项。要查看所有的会话信息，请不要指定 session 选项。

示例输出

图: gstat -g stq 命令输出

```
Stream Queue: (session 25 cnt 4)  0:db12400  1:db18400  2:dcf0400  3:dcf6400
Full Queue:  (cnt 2  waiters 0)  0:0  1:db12400
Empty Queue: (cnt 0  waiters 0)
```

输出描述

session

会话 ID

cnt

流队列缓冲区的数量

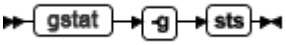
waiters

正在等待流队列缓冲区的线程数量

gstat -g sts 命令：打印每个线程的堆栈用途

可以使用 gstat -g sts 命令显示有关每个线程的最大和当前堆栈使用的信息。

语法：



示例输出

图: gstat -g sts 命令输出

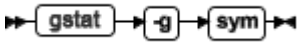
```
Stack usage:

TID     Total   Max             Current         Thread Name
                bytes   %       bytes   %
2       32768   3124    9       3079    9       adminthd
3       32768   2870    8       2871    8       childthd
5       32768   14871   45      2871    8       Cosvr Avail Mgr
6       32768   2870    8       2871    8       dfm_svc
7       131072  3190    2       3191    2       xmf_svc
9       32768   3126    9       3127    9       xtm_svcc
10      32768   3580    10      3335    10      xtm_svcp
11      32768   3238    9       3239    9       cfgmgr_svc
12      32768   6484    19      2871    8       lio vp 0
14      32768   6484    19      2871    8       pio vp 0
16      32768   6484    19      2871    8       aio vp 0
18      131072  10391   7       2871    2       msc vp 0
20      32768   4964    15      2871    8       fifo vp 0
22      32768   4964    15      2871    8       fifo vp 1
24      32768   6028    18      2871    8       aio vp 1
26      32768   5444    16      2951    9       dfmxpl_svc
27      32768   2886    8       2887    8       sch_svc
28      32768   7812    23      5015    15      rqm_svc
29      32768   7140    21      3079    9       sm_poll
30      32768   11828   36      6439    19      sm_listen
31      32768   2870    8       2871    8       sm_discon
32      32768   14487   44      4055    12      main_loop()
33      32768   4272    13      2903    8       flush_sub(0)
34      32768   2902    8       2903    8       flush_sub(1)
35      32768   2870    8       2871    8       btscanner 0
36      32768   3238    9       3239    9       aslogflush
37      32768   3055    9       2887    8       bum_local
38      32768   3238    9       3239    9       bum_rcv
39      32768   4902    14      4903    14      gadmin_mon
```

```
42      32768   4964    15      2871    8       lio vp 1
44      32768   5136    15      2871    8       pio vp 1
```

gstat -g sym 命令：打印 oninit 实用程序的符号表信息

可以使用 gstat -g sym 命令显示 oninit 实用程序符号表的信息。

语法：



示例输出

图: gstat -g sym 命令输出

以下示例显示了该输出的前几行：

```
Table for oninit has 23378 entries
Initial value for -base-: 0x0
0x3451e0 _start
0x345300 .ld_int
0x345348 .ld_llong
0x3453dc .ld_float
0x345428 .ld_double
0x3454c4 .st_int
0x3454fc .st_llong
0x34556c .st_float
0x3455c0 .st_double
0x34565c .st_float_foreff
0x345694 .st_double_foreff
0x345718 main
0x34c2ac get_cfgfile
0x34c2fc is_server_alias
```

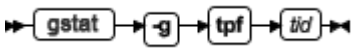
输出描述

gstat -g sym 命令显示 oninit 实用程序中符号的相对内存地址和名称（函数和变量）。

gstat -g tpf 命令：打印线程概要文件

可以使用 gstat -g tpf 命令显示线程概要文件。

语法：



要打印指定线程的概要文件，请指定 tid 线程 ID 。将 tid 设置为 0 来显示所有线程的概要文件。

示例输出

图: gstat -g tpf 命令输出

```
gstat -g tpf 945

Thread profiles

tid lkreqs lkw dl to lgrs isrd iswr isrw isdl isct isrb lx bfr bfw lsus lsmx seq
```

```
945 1969    0    0    0    6181 1782 2069 13    0    0    0    0    16183 7348 743580 0    6
```

输出描述

tid

线程 ID

lkreqs

锁请求数

lkw

锁等待数

dl

死锁数

to

远程死锁超时

lgrs

日志记录

isrd

读取数

iswr

写入数

isrw

重新写入数

isdl

删除数

isct

提交数

isrb

回滚数

lx

长事务

bfr

缓冲区读取数

bfw

缓冲区写入数

lsus

当前使用的日志空间

lsmx

使用的最大日志空间

seq

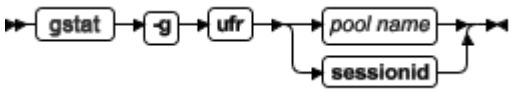
顺序扫描数

gstat -g ufr 命令：打印内存池片分片

可以使用 gstat -g ufr 命令显示当前在特定的内存池中使用的分片列表。

该命令需要一个附加的参数来指定池名或要显示池的会话 ID 。每个会话都会分配一个与其会话 ID 名称相同的内存池。使用 gstat -g mem 命令标识池名，使用 gstat -g ses 命令标识会话 ID 。

语法：



内存池分为很多分片用于不同的用途。使用 gstat -g ufr 命令时，有可能会看到这些分片的列表，显示其各自大小（以字节为单位）和所包含的信息类型。提供的信息大多由 Technical Support 使用以写出其分析报告的问题。

指定池名的示例输出

图: gstat -g ufr global 指定池名的命令输出

```
Memory usage for pool name global:
size       memid
1736       overhead
23544      mcbmsg
72         messages
33112      osenv
25432      rsam
88         shmblklist
5170664    net
```

指定会话 ID 的示例输出

以下示例显示会话 ID 6 的输出。

图: gstat -g ufr 指定会话 ID 的命令输出

```
Memory usage for pool name 6:
size       memid
3256       overhead
144        scb
2968       ostcb
18896      sqscb
3312       opentable
72         sql
808        filetable
352        fragman
```

```
552         hashfiletab
1584        gentcb
12096       log
2960        sqtcb
2928        osenv
720         keys
224         rdahead
16248       temprec
```

输出描述

size (decimal)

池分片大小（以字节为单位）

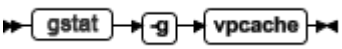
memid (string)

池分片的名称

gstat -g vpcache 命令：打印 CPU 虚拟处理器专用内存高速缓存的统计信息

可以使用 gstat -g vpcache 命令显示有关 CPU 虚拟处理器专用内存高速缓存的统计信息。

语法：



示例输出

每个 CPU 虚拟处理器的输出都有相同的格式。以下示例显示了一个 CPU 虚拟处理器的输出。

图: gstat -g vpcache 命令输出

```
CPU VP memory block cache statistics - 4096 byte blocks

Number of 4096 byte memory blocks requested for each CPU VP:262144
CPU VP memory block cache mode : Dynamic

vpid    pid         Blocks held  Hit percentage   Free cache
1       2557540     4667202       99.2 %           100.0 %

Current VP total allocations from cache: 59466799, Total frees: 60209953

  size  cur blks   tgt blks   alloc      miss     free      drain  draintime
  1     1662023    9661       49167485   0        49816526  0      Thu Apr 11 09:43:35 2013
  2     130        52428      7609556    297043   7609612   0      Thu Jan  1 00:00:00 1970
  3     329160     9          905094     0        943256    0      Thu Apr 11 09:43:36 2013
  4     424        9          306637     16192    306506    0      Thu Apr 11 09:43:33 2013
  5     10         9          119313     122607   119315    0      Thu Apr 11 09:43:36 2013
  6     20790      9          55305      0        57700     0      Thu Apr 11 09:43:23 2013
  7     9877       9          31164      0        31942     0      Thu Apr 11 09:43:14 2013
  8     2816       5242       6500       0        6537      0      Thu Jan  1 00:00:00 1970
  9     234        9          606575     8323     605525    0      Thu Apr 11 09:43:36 2013
  10    1130       9          5597       0        5679      0      Thu Apr 11 09:43:18 2013
  11    231        5242       1808       0        1753      0      Thu Jan  1 00:00:00 1970
  12    1068       9          5667       0        5666      0      Thu Apr 11 09:43:28 2013
  13    65         5242       7114       175      7110      0      Thu Jan  1 00:00:00 1970
  14    28         5242       26200      172      26185     0      Thu Jan  1 00:00:00 1970
  15    30         5242       13562      553      13547     0      Thu Jan  1 00:00:00 1970
  16    2627136    34         349124     0        408425    0      Thu Apr 11 09:43:35 2013
  17    1309       9          59         0        107       0      Thu Apr 11 09:27:33 2013
  18    198        5242       7          0        6         0      Thu Jan  1 00:00:00 1970
  19    190        5242       5          0        1         0      Thu Jan  1 00:00:00 1970
  20    60         5242       30         19       19        0      Thu Jan  1 00:00:00 1970
  21    462        5242       38         0        43        0      Thu Jan  1 00:00:00 1970
  22    22         5242       3          0        1         0      Thu Jan  1 00:00:00 1970
  23    69         5242       141        15       135       0      Thu Jan  1 00:00:00 1970
  24    4944       35         189509     2078     185347    0      Thu Apr 11 09:43:35 2013
  25    75         5242       1          0        1         0      Thu Jan  1 00:00:00 1970
  26    0          9          364        220      361       0      Thu Apr 11 09:39:17 2013
  27    27         5242       1          0        2         0      Thu Jan  1 00:00:00 1970
  28    56         5242       415        33       410       0      Thu Jan  1 00:00:00 1970
  29    319        5242       7101       735      7088      0      Thu Jan  1 00:00:00 1970
  30    3240       5242       174        0        223       0      Thu Jan  1 00:00:00 1970
  31    279        11         51994      2515     50682     0      Thu Apr 11 09:43:36 2013
  32    800        5242       256        0        243       0      Thu Jan  1 00:00:00 1970
```

输出描述

vpid

CPU 虚拟处理器的 ID

pid

操作系统分配的 CPU 虚拟处理器的进程 ID

Blocks held

4096 字节 block 数是专用内存高速缓存中可用的数量

Hit percentage

发生请求时 block 可用的时间百分比

Free cache

Block 被释放以供重复使用而不被排出的时间百分比

Current VP total allocations from cache

从高速缓存获取一个 block 或一组 block 的次数

Total frees

一个 block 或一组 block 被添加至高速缓存的次数

size

内存 block 的大小，以 4096 字节 block 为单位

cur blks

当前已分配的 4096 字节 block 的数量（是 size 的倍数）

tgt blks

高速缓存被排出之前该高速缓存条目 block 的目标数

alloc

请求者接收到该大小 block 的次数

miss

请求内存 block 时没有可用 block 的次数

free

内存 block 放入高速缓存中的次数

drain

将到期的 block 强制调出，为其他 block 让出空间的次数

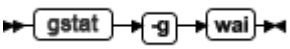
draintime

上一次内存 block bin 被排出的时间

gstat -g wai 命令：打印等待队列线程队列

可以使用 gstat -g wai 命令显示当前在等待队列中并没有得到执行的系统上的线程列表。输出按线程 ID 排序。

语法：



示例输出

图: gstat -g wai 命令输出

```
Waiting threads:
tid     tcb             rstcb           prty status               vp-      name
2       46b1ea40         0               1    IO Idle                5lio        lio vp 0
3       46b3dc58         0               1    IO Idle                6pio        pio vp 0
4       46b5dc58         0               1    IO Idle                7aio        aio vp 0
5       46b7cc58         0               1    IO Idle                8msc        msc vp 0
6       46b1ed10         460f5028        1    sleeping secs: 1       3cpu        main_loop()
9       46d0d6e0         0               1    sleeping forever       1cpu        soctcplst
10      46d70b48         0               1    sleeping forever       3cpu        sm_listen
11      46e5d9a0         0               1    sleeping secs: 1       3cpu        sm_discon
12      46e5dc70         460f5820        1    sleeping secs: 1       3cpu        flush_sub(0)
13      46e8a5a8         460f6018        1    sleeping secs: 1       3cpu        aslogflush
14      46fe8148         460f6810        1    sleeping secs: 41      3cpu        btscanner_0
15      46fe84a8         0               1    IO Idle               10aio        aio vp 1
16      46fe8778         460f7008        1    sleeping secs: 1       1cpu        onmode_mon
36      47531960         460f7ff8        1    sleeping secs: 253     3cpu        dbScheduler
37      47531c30         460f87f0        1    sleeping forever       4cpu        dbWorker1
38      47491028         460f7800        1    sleeping forever       4cpu        dbWorker2
```

输出描述

tid (decimal)

线程 ID

tcb (hex)

线程控制 block 的内存地址

rstcb (hex)

RSAM 线程控制 block 的内存地址

prty (decimal)

线程优先级。较大的数字代表较高的优先级

status (string)

线程的当前状态

vp- (decimal and string)

上次运行线程的 VP 的虚拟处理器整数 ID 与运行线程的 VP 类的名称连接

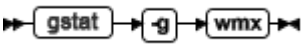
name (string)

线程名称

gstat -g wmx 命令：使用等待者打印所有互斥

可以使用 gstat -g wmx 命令显示所有有等待对象的互斥。

语法：



示例输出

图: gstat -g wmx 命令输出

```
Mutexes with waiters:
            mid      addr             name              holder   lkcnt  waiter   waittime
            134825   7000002043a9148  free_lock         11009    0      200      22921
            11010    22918
```

输出描述

mid

内部互斥标识

addr

已锁定的互斥的地址

name

互斥名

holder

持有该互斥的线程的线程 ID

0 = 在共享方式下持有读/写互斥

lkcnt

对于读/写互斥，是当前在共享方式下正在锁定的线程数量。对于重新锁定互斥，是持有该互斥的线程锁定或重新锁定该互斥的次数

waiter

正在等待该互斥的线程的 ID 列表

waittime

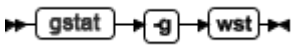
以秒表示线程正在等待的时间量

gstat -g wst 命令：打印线程的等待统计信息

可以使用 gstat -g wst 命令显示系统中线程的等待统计信息。

WSTATS 配置参数必须设置为 1 来启用等待信息收集。有关更多信息，请参阅 WSTATS 配置参数 。

语法：



示例输出

```
Version 8.8  -- On-Line -- Up 18:52:59 -- 78856 Kbytes
        name   tid    state            n         avg(us)    max(us)
        msc vp 0 5    ready            6               9        17
        msc vp 0 5    run              6            1107      2215
        msc vp 0 5    IO Idle          5          2985.9s   1496.1s

        main_loo 7    IO Wait          55           6496     16725
        main_loo 7    yield time       44929         1.2s    343.1s
        main_loo 7    ready            44998      206085     343.1s
        main_loo 7    run              44985           5       436

        ...

        sqlexec  63   IO Wait          2            1118      2165
        sqlexec  63   other cond       6           34237    204142
        sqlexec  63   ready            9               7        16
        sqlexec  63   run              7             1.1s      7.7s
```

输出描述

name (string)

线程名称

tid (decimal)

线程 ID

state (string)

说明在此输出行内等待的线程。如果单个线程在多个不同的状态中等待，那么它可能有多行输出。 state 字段能包含的值有：

chkpt cond: 线程等待 checkpoint 条件

cp mutex: 线程等待 checkpoint 互斥可用

deadlock mutex: 线程等待死锁互斥可用

empty Q: 线程等待队列中的空缓冲区

fork: 线程等待子线程运行

full Q: 线程等待队列上的一个完整的缓冲区

IO Idle: I/O 线程空闲

IO Wait: 线程产生，同时等待 I/O 完成

join wait: 线程等待另一个线程退出

lock mutex: 线程等待锁定互斥可用

lockfree mutex: 线程等待锁释放互斥可用

logflush: 发生逻辑日志清空

log mutex: 线程等待逻辑日志互斥可用

logcopy cond: 线程等待逻辑日志复制互斥可用

logio cond: 线程等待逻辑日志条件

lrus mutex: 线程等待缓冲区 LRU 互斥可用

misc: 现场等待杂项理由

other cond: 线程等待内部条件

other mutex: 线程等待内部系统互斥可用

other yield: 线程等待内部原因

OS read: 线程等待操作系统读请求完成

OS write: 线程等待操作系统写请求完成

ready: 线程已准备好运行

run: 线程已运行

sort io: 线程等待排序 I/O 完成

vp mem sync: 线程等待虚拟处理器同步

yield bufwait: 线程产生，同时等待缓冲区可用

yield 0: 线程产生，但立即超时

yield time: 线程产生超时

yield forever: 线程产生并保持这种方式，直到它被唤醒

n (decimal)

在此状态下等待的线程次数

avg(us) (floating point)

每次等待发生的时候，线程在此状态下等待的平均用户时间。时间的单位为微秒，值后的 s 表示以秒为单位计算用户时间。

max(us) (floating point)

等待发生的时候，线程在此状态下等待的最大用户时间。时间的单位为微秒，值后的 s 表示以秒为单位计算用户时间。

3.16.18 gstat -G 命令：打印 TP/XA 事务信息

可以使用 gstat -G 命令显示关于通过 TP/XA 库生成的全局事务的信息。

语法：

示例输出

图: gstat -G 命令输出

```
Global Transaction Identifiers
          address    flags  isol    timeout  fID       gtl  bql  data      dbpartnum
          45cb0318   -LH-G  COMMIT  0        4478019   2    2    30323032  100163
```

对于紧耦合事务，所有的分支将共用在地址列中显示的相同事务地址。

输出描述

address

事务地址

flags

位置 1 的标志代码（当前事务状态）：

A

用户线程已连接到事务

S

TP/XA 已暂挂事务

C

TP/XA 正在等待回滚

位置 2 的标志代码（事务方式）：

T

紧耦合方式（MTS）

L

松耦合方式（缺省方式）

位置 3 的标志代码（事务阶段）：

B

开始工作

P

准备好用于提交的分布式查询

X

TP/XA 已准备好提交

C

正在提交或已提交

R

正在回滚或已回滚

H

正在尝试回滚或已回滚

位置 4 的标志代码：

X

XA 数据源全局事务

位置 5 的标志代码（事务类型）：

G

全局事务

C

分布式查询协调者

S

分布式查询从属者

B

分布式查询协调者和从属者

M

重定向的全局事务

isol

事务的隔离级别

timeout

事务锁定超时

fID

格式 ID

gtl

全局事务 ID 长度

bql

分支限定符长度

data

指定事务的数据

dbpartnum

启动事务的数据库 ID

3.16.19 gstat -h 命令：打印缓冲区头哈希链信息

可以使用 gstat -h 命令显示有关缓冲区头哈希链（有时称为“哈希桶”）的信息，用于访问每个信息缓冲池中的页。

语法：

示例输出

输出中显示的信息以链长度的数字柱状图加每个缓冲池的摘要信息显示。输出中的所有数值均为十进制。 哈希链越短，服务器越快能找到请求的缓冲区，因为一般来说，在目标链上查找目标缓冲区所需检查的缓冲区头较少。

在每个缓冲池输出中，最先显示缓冲池页的大小（以字节为单位）。接着显示该缓冲池的柱状图和摘要信息。

图: gstat -h 命令输出

```
Buffer pool page size: 2048

buffer hash chain length histogram
# of chains         of len
3423              0
4546              1
223              2
8192   total chains
4992   hashed buffs
5000    total buffs

Buffer pool page size: 4096

buffer hash chain length histogram
# of chains         of len
707              0
315              1
2              2
1024   total chains
319   hashed buffs
1000    total buffs
```

输出描述

Histogram Information on Hash Chains

柱状图信息中有一行用于显示系统中当前存在的每个缓冲区哈希的长度。每一行有两列：

# of chains

给定长度的哈希链数

of len

这些链的长度

Summary Information Per Buffer Pool

total chains

该缓冲池内存在的哈希链数

hashed buffs

当前哈希到本缓冲池的哈希链中缓冲区头数

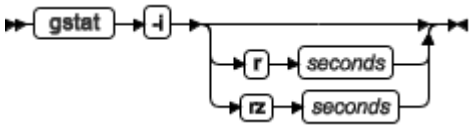
total buffs

该缓冲池中的缓冲区总数

3.16.20 gstat -i 命令： 开始® 交互方式

可以使用 gstat -i 命令将 gstat 实用程序置于交互方式。

语法：



在交互方式中，可以为每个会话输入多个 gstat 选项，但一次只能输入一个。gstat 提示显示并允许您输出选项。

重要： 在交互方式中，不要在该选项前加连字符。

其他选项

两个其他选项 gstat r seconds 和 gstat rz seconds 可用于交互方式。gstat r seconds 选项类似于当前的 gstat -r seconds 选项，它重复生成显示。如果管理员在交互方式提示处执行 gstat r seconds ，那么提示更改以反映指定的时间间隔（秒）并重新出现，等待下一命令。在以下示例中，由下一条命令生成的显示每 3 秒重复一次：

```
gstat> r 3
gstat[3]>
```

gstat rz seconds 选项使您可以如指定的那样重复下一条命令并在每个执行之间将所有概要文件计数器设置为 0 。

终止交互方式或重复顺序

要终止交互方式，请按 CTRL-d。

要终止重复顺序，请按 CTRL-c。

3.16.21 gstat -k 命令：打印活动的锁信息

可以使用 gstat -k 命令打印活动锁的信息，包括锁表中的该锁的地址。

语法：

示例输出

可用锁的最大数量由 onconfig 文件中的 LOCKS 配置参数进行指定。

图: gstat -k 命令输出

```
Locks
address  wtlist   owner    lklist   type     tblsnum  rowid    key#/bsiz
a095f78  0        a4d9e68  0        HDR+S    100002   203         0
1 active, 2000 total, 2048 hash buckets, 0 lock table overflows
```

在以下输出中，最后一行的数字 2 显示的是一个 Enterprise Replication 伪锁：

```
Locks
         address  wtlist    owner    lklist   type    tblsnum  rowid    key#/bsiz
         a1993e8  0         5c2f03d0 a19be30     S          2   1c05a            0
```

输出描述

address

锁表中锁的地址

如果用户线程正在等待该锁，那么锁的地址出现在 gstat -u （用户）输出的 wait 字段中。

wtlist

是正在等待锁的用户线程（如果有）列表中的第一项

owner

是正持有锁的线程的共享内存地址

此地址对应于 gstat -u （用户）输出的 address 字段中的地址。当 owner 值显示在括号中时，它代表事务结构的共享内存地址。只有锁是为全局事务而分配时，才会出现这种情况。

该地址对应于 gstat -G 的输出的地址字段

lklist

是刚才列出的所有者所持有的链接列表中的下一个锁

type

使用以下代码指定锁的类型：

HDR

头

B

字节

S

共享

X

互斥

I

意向

U

更新

IX

意向—互斥

IS

意向—共享

SIX

共享，意向—互斥

tblsnum

是锁定资源的 tblspace 编号。如果数值小于 10000，那么它表示 Enterprise Replication 伪锁

rowid

是行标识号

Rowid 提供以下锁的信息：

- 如果 rowid 等于 0 那么该锁为表锁
- 如果 rowid 以两个 0 结束，那么该锁为页锁
- 如果 rowid 为 6 个数字或更少且不以 0 结束，那么该锁很可能是行锁
- 如果 rowid 多于 6 个数字，那么该锁很可能是索引键值锁

key#/bsiz

是索引键号或对于 VARCHAR 锁的已锁定字节数

如果该字段包含 'K-' ，后跟值，那么是键锁。值标识哪个索引正在被锁定。例如：K-1 表示对表所定义的第一个索引上的锁。

3.16.22 gstat -l 命令：打印物理和逻辑日志信息

可以使用 gstat -l 命令显示有关物理日志、逻辑日志和临时逻辑日志的信息。

语法：

示例输出

图: gstat -l 命令输出

```
Physical Logging
Buffer bufused  bufsize  numpages numwrits pages/io
P-1  0        16       716      55       13.02
phybegin         physize  phypos   phyused  %used
1:263            500      270      0        0.00

Logical Logging
Buffer bufused  bufsize  numrecs  numpages numwrits recs/pages pages/io
L-3  0        16       42169    2872     1043     14.7       2.8
Subsystem    numrecs  Log Space used
OLDRSAM      42169    4436496

address  number   flags    uniqid   begin               size     used    %used
a517f70  1        U-B----  1        1:763                500      500   100.00
a517fb0  2        U-B----  2        1:1263               500      500   100.00
a40daf0  3        U-B----  3        1:1763               500      500   100.00
a40db30  4        U-B----  4        1:2263               500      500   100.00
a40db70  5        U-B----  5        1:2763               500      500   100.00
a40dbb0  6        U---C-L  6        1:3263               500      372    74.40
a40dbf0  7        A------  0        1:3763               500        0     0.00
a40dc30  8        A------  0        1:4263               500        0     0.00
8 active, 8 total
```

物理日志文件的输出描述

显示的第一部部分描述了物理日志配置：

buffer

是物理日志缓冲区的数量

bufused

是已使用的物理日志缓冲区页数

bufsize

是每个物理日志缓冲区的大小（以页为单位）

numpages

是写入物理日志的页数

numwrits

是对磁盘的写入数

pages/io

计算方法是 numpages/numwrits

该值指示正在缓存的物理日志写入的效率

phybegin

是日志开始处的物理页号

physize

是物理日志的大小（以页为单位）

phypos

是日志中发生下一个日志记录写入的当前数量

phyused

是日志中已使用页的数量

%used

是已使用页的百分比

gstat -l 命令输出的第二部分描述逻辑日志配置：

buffer

是逻辑日志缓冲区数

bufused

是逻辑日志缓冲区呢已使用的页数

bufsize

每个逻辑日志缓冲区的大小（以页为单位）

numrecs

是已写入记录的数量

numpages

是已写入页的数量

numwrits

是对逻辑日志的写入数

recs/pages

计算方法是 numrecs/numpages

您不能影响该值。不同类型的操作生成不同类型（和大小）的记录

pages/io

计算方法是 numpages/numwrits

可以通过更改逻辑日志缓冲区大小（由 ONCONFIG 文件中的 LOGBUFF 指定）或通过更改数据库的日志记录方式（从已缓冲到未缓冲，反之亦然）来影响此值

以下字段将对每个逻辑日志文件重复：

address

是日志文件描述符的地址

number

是逻辑日志文件的日志标识号

日志标识号可能是无序的，因为数据库服务器或管理员都可以直接插入日志文件

flags

提供每个日志的状态，如下所示：

A

新添加的（可以使用）

B

已备份

C

当前的逻辑日志文件

D

标记为已删除

要删除日志文件并释放其空间以再利用，那么必须对所有存储空间执行 0 级备份

F

可用的，可以使用

L

最新的 checkpoint 记录

U

已使用的

uniqid

是日志的唯一 ID 号

begin

是日志文件的起始页

size

是日志的大小（以页为单位）

used

是已使用页数量

%used

是已使用页的百分比

active

是活动逻辑日志的数量

total

是逻辑日志的总数

临时逻辑日志文件的输出描述

数据库服务器在热恢复过程中使用 temporary logical logs ，因为永久日志在那时是不可用的。以下字段将对每个临时逻辑日志文件重复：

address

是日志文件描述符的地址

number

是逻辑日志文件的日志标识号

flags

提供每个日志的状态，如下所示：

B

已备份

C

当前的逻辑日志文件

F

可用的，可以使用

U

已使用的

uniqid

是日志的唯一 ID 号

begin

是日志文件的起始页

size

是日志的大小（以页为单位）

used

是已使用页的数量

%used

是已使用页的百分比

active

是活动临时逻辑日志的数量

3.16.23 gstat -L 命令：打印可用锁的数量

可以使用 gstat -L 命令打印在锁可用列表上的可用锁的数量。

语法：

示例输出

图: gstat -L 输出

```
num   list head     available locks
0     10a143b70     19996
1     101010101     200
3     020202020     300
```

输出描述

num

列表编号

list head

列表的起始地址

available locks

该列表上的锁的数量

3.16.24 gstat -m 命令：打印最近的系统消息日志信息

可以使用 gstat -m 命令显示系统消息日志中 20 个最新行。

可以在数据库服务器处于任何方式（包括脱机）时使用 gstat -m 命令选项。

用法：

示例输出

此选项的输出列出消息日志文件和完整路径名的 20 个文件条目。一个日期和时间头分隔每天的条目。时间戳记放在每天中单个条目的开始处。消息日志由 ONCONFIG 文件中的 MSGPATH 进行指定。

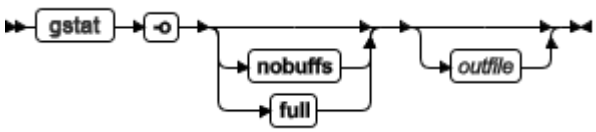
图: gstat -m 命令输出

```
Message Log File: /work/11.50/dbspaces/star3.log
11:26:33  Checkpoint Completed:  duration was 0 seconds.
11:26:33  Checkpoint loguniq 1, logpos 0x23c408, timestamp: 0x2cc2 Interval: 9
```

3.16.25 gstat -o 命令：输出共享内存内容

可以使用 gstat -o 命令将共享内存的内容复制到特定的文件，以便今后分析。如果没有指定输出文件，那么使用当前目录的缺省 gstat.out 文件。

语法：



使用 nobuffs 选项排除来自输出文件中的共享内存常驻段中的缓冲池。这会产生较小的输出文件。

使用 full 选项创建同 GBase 8s 实例的共享内存段大小相同的输出文件。您必须在文件系统上留出足够的空间来处理该输出。

如果您没有指定 nobuffs 或 full 选项，那么输出将会由数据库服务器中的 DUMPSHMEM 配置参数的设置控制：

- 如果 DUMPSHMEM 设置为 0 或 1 ，那么 gstat -o 命令写入一个完整共享内存转储文件。
- 如果 DUMPSHMEM 设置为 2 ，那么 gstat -o 命令写入一个 nobuffs 共享内存转储文件（在驻留段中没有缓冲池）。

通过对此文件运行其他 gstat 命令，您可以从先前保存的共享内存转储中收集信息。使用 gstat -o 命令创建的 outfile 可用作运行其他 gstat 命令的源文件。有关更多信息，请参阅在共享内存转储文件中运行 gstat 命令。

3.16.26 gstat -p 命令：打印概要文件计数

可以使用 gstat -p 命令显示自启动数据库服务器以来或自运行 gstat -z 命令以来的概要文件计数的信息。

语法：

示例输出

图: gstat -p 命令输出

```
Profile

            dskreads  pagreads   bufreads   %cached dskwrits   pagwrits   bufwrits   %cached
            16934     47321      203600361  99.99   103113     158697     950932     89.16

            isamtot   open     start    read     write   rewrite  delete  commit  rollbk
            139214865 9195777  12257208 94191268 362691  55696    38134   128294  24

            gp_read   gp_write   gp_rewrt   gp_del     gp_alloc   gp_free   gp_curs
            39        2          27         51         0          0         16

            ovlock    ovuserthread   ovbuff   usercpu  syscpu   numckpts   flushes
            0         0              0        1551.59  144.82   1822       1822

            bufwaits  lokwaits  lockreqs   deadlks  dltouts  ckpwaits  compress  seqscans
            176       1         195872383  0        0        1         39331     1259170

            ixda-RA   idx-RA     da-RA    logrec-RA  RA-pgsused lchwaits
            0         7594       2124     0          2002       18848
```

输出描述

输出的第一部分描述读取和写入。

读和写分成三类：从磁盘、从缓冲区以及页数（读取或写入）

第一个 %cached 字段衡量从缓冲区的读取数与从磁盘的读取数之比。第二个 %cached 字段衡量写入缓冲区的数量与写入磁盘的数量之比。

数据库服务器对信息执行缓冲并将其写入磁盘（以页为单位）。出于这个原因，以 dskwrits 显示的磁盘写入数通常小于个别用户执行的写入数：

dskreads

实际的磁盘读取数

pagreads

页读取数

bufreads

共享内存读取数

%cached

缓冲池中已高速缓存的读取数百分比

如果 bufreads 超过最大 integer （或 long）值，那么其内部表示变成负数，但值显示为 0.0

dskwrits

对磁盘的物理写入的实际数量

该数字包含对 gstat -l 中所报告的物理和逻辑日志的写入数

pagwrits

已写入页的数量

bufwrits

共享内存写入数

%cached

缓冲池中已高速缓存的写入数百分比

-p 的下一部分显示已执行不同类 ISAM 调用的次数的制表。这些调用发生在操作的最底层，并且不必与 SQL 语句一对一对应执行。单个查询可能产生多个 ISAM 调用。这些统计信息通过数据库服务器产生并且不能用来监视单个数据库上的活动，除非只有一个数据库是活动的或只存在一个数据库

isamtot

调用总数

open

当 tblspace 打开时增加

start

增加索引中的指针

read

当调用读取函数时增加

write

当每次写调用时增加

rewrite

当发生更新时增加

delete

当删除行时增加

commit

每次执行 iscommit() 调用时增加

该值与已执行的显式 COMMIT WORK 语句之间不存在一对一的对应关系

rollbk

当事务回滚时增加

gstat -p 命令输出的下一部分显示有关一般页的信息。一般页管理提供一个 API 让 GBase 8s 来管理数据库服务器缓冲池中的非标准页。下表描述了 gstat -p 命令输出中的 Generic Page Manager 。

gp_read

一般页读取数

gp_write

一般页写入数

gp_rewrt

一般页更新数

gp_del

一般页删除数

gp_alloc

一般页分配数

gp_free

已释放并返回给 tablspace 的一般页数

gp_curs

用于一般页的游标数

gstat -p 命令输出的下一部分显示了请求资源时没有可用资源的次数：

ovlock

会话尝试超过锁最大数量的次数

有关更多信息，请参阅第 1-56 页的 “LOCKS”

ovuserthread

用户尝试超过用户线程最大数量的次数

ovbuff

数据库服务器无法找到可用共享内存缓冲区的次数

当没有可用缓冲区时，数据库服务器将 dirty 缓冲区写入磁盘，然后设法找到可用缓冲区

usercpu

所有用户线程使用的用户 CPU 时间（以秒为单位）

该条目每 15 秒更新一次

syscpu

所有用户线程使用的全部系统 CPU 时间（以秒为单位）

该条目每 15 秒更新一次

numckpts

自引导时间以来的 checkpoint 数

flushes

缓冲池已清仓到磁盘的次数

gstat -p 命令输出的下一部分包含杂项信息，如下：

bufwaits

每次用户线程必须等待缓冲区时增加

lokwaits

每次用户线程必须等待锁时增加

lockreqs

每次请求锁时增加

deadlks

每次检测到潜在死锁并阻止时增阿基

dltouts

每次用户线程等待锁时已超过分布式死锁超时时增加

ckpwaits

Checkpoint 等待数

compress

每次压缩数据页时增加

seqscans

对每个顺序扫描增加

gstat -p 命令输出的最后一部分包含以下信息：

ixda-RA

索引页到数据页的预先读取计数

idx-RA

遍历索引页的预先读取计数

da-RA

仅数据路径扫描的计数

logrec-RA

数据库服务器预先读取的日志记录

RA-pgsused

数据库服务器预先读取所使用的页数

lchwaits

存储线程需要等待共享存储锁存器的次数

大的锁存器等待数通常是由大量处理活动引起，数据库服务器正在此活动中记录大多数事务

3.16.27 gstat -P 命令：打印分区信息

可以使用 gstat -P 命令显示所有分区的分区号和属于分区的缓冲池中的页。

语法：

有关在没有创建缓冲池的转储文件中上运行 gstat -P 命令的信息，请参阅 在共享内存转储文件中运行 gstat 命令.

示例输出

图: gstat -P 命令输出

```
Buffer pool page size: 2048
partnum  total    btree    data     other    dirty
0        36       1        8        27       0
1048577  2        0        0        2        0
1048578  4        1        1        2        0
1048579  23       10       12       1        0
1048580  68       31       36       1        0
4194309  3        0        1        2        0

Totals:  3000     786      1779     435      0
Percentages:
Data  59.30
Btree 26.20
Other 14.50

Buffer pool page size: 8192
partnum  total    btree    data     other    dirty
0        999      0        0        999      0
5242881  1        0        0        1        0

Totals:  1000     0        0        1000     0
Percentages:
Data  0.00
Btree 0.00
Other 100.00
```

输出描述

Buffer pool page size

以字节表示的缓冲池页面大小

partnum

分区号

total

分区总数

btree

分区中 B-tree 页数

data

分区中数据页数

other

分区中其他页数

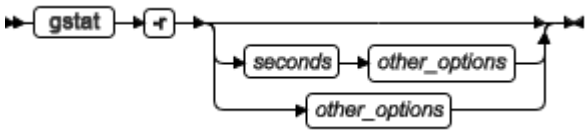
dirty

分区中 dirty 页数

3.16.28 gstat -r 命令：重复打印选择的统计信息

可以使用 gstat -r 命令反复在指定的时间间隔打印指定选项的统计信息。

语法：



使用 gstat -r seconds other_options 命令来指定重复其他选项的时间间隔。

使用 gstat -r other_options 命令来让此选项每隔五秒重复一次，这将使其他选项能与 -r 选项连接。如下所示： gstat -rFh 。

gstat -r 命令可以在命令方式和交互方式下使用，并可能对将命令重复输出到受监视系统的资源利用有用。

每隔五秒执行一次 gstat -r 命令的示例输出

图: 命令输出

```
gstat -r
GBase 8s Database Server Version 8.7.F       -- On-Line -- Up 20:05:25 -- 1067288 Kbytes
\GBase 8s Database Server Version 8.7.F       -- On-Line -- Up 20:05:30 -- 1067288 Kbytes
GBase 8s Database Server Version 8.7.F       -- On-Line -- Up 20:05:35 -- 1067288 Kbytes
```

每个十秒执行 gstat -r 命令的示例输出

图: 命令输出

```
gstat -r 10
GBase 8s Database Server Version 8.8       -- On-Line -- Up 20:06:58 -- 1067288 Kbytes
GBase 8s Database Server Version 8.8       -- On-Line -- Up 20:07:08 -- 1067288 Kbytes
GBase 8s Database Server Version 8.8      -- On-Line -- Up 20:07:18 -- 1067288 Kbytes
```

每隔一秒执行带有 -h 选项的 gstat -r 命令的示例输出

图: gstat -r 1 -h 命令输出

```
gstat -r 1 -h


           Buffer pool page size: 2048
           buffer hash chain length histogram
           # of chains        of len
           3841               0
```

```
            3767              1
            522               2
            62                3
            8192   total chains
            4351   hashed buffs
            5000    total buffs

            Buffer pool page size: 2048

            buffer hash chain length histogram
            # of chains       of len
            4020              0
            3392              1
            735               2
            43                3
            2                 4
            8192   total chains
            4172   hashed buffs
            5000    total buffs
```

每隔五秒执行带有 -Fh 选项的 gstat -r 命令的示例输出

图: gstat -rFh 命令输出

```
gstat -rFh


       Fg Writes     LRU Writes    Chunk Writes
       0             0             21

       address           flusher  state    data     # LRU    Chunk    Wakeups  Idle Tim
       460e6820         0        I        0        0        2        5        9.820
       states: Exit Idle Chunk Lru

       Buffer pool page size: 2048

       buffer hash chain length histogram
       # of chains         of len
       6342              0
       1850              1
       8192   total chains
       1850   hashed buffs
       5000    total buffs


       Fg Writes     LRU Writes    Chunk Writes
       0             0             21
```

```
address           flusher  state    data     # LRU    Chunk    Wakeups  Idle Tim
460e6820          0        I        0        0        2        10       22.755
states: Exit Idle Chunk Lru

Buffer pool page size: 2048

buffer hash chain length histogram
# of chains         of len
4396              0
3796              1
8192   total chains
3796   hashed buffs
5000    total buffs
```

3.16.29 gstat -R 命令：打印 LRU 、FLRU 和 MLRU 队列信息

可以使用 gstat -R 命令显示有关 LRU 队列、FLRU 队列和 MLRU 队列的详细信息。对于每个队列， gstat -R 命令显示队列中的缓冲区数和已修改缓冲区的数量和百分比。

关于三种队列的深入讨论，请参阅 GBase 8s 管理员指南 的共享内存一章中的 LRU 队列。

语法：

示例输出

图: gstat -R 命令输出

```
Buffer pool page size: 2048
8 buffer LRU queue pairs             priority levels
# f/m   pair total     % of    length       LOW      HIGH
0 f         375      100.0%      375         375         0
1 m                    0.0%        0           0         0
2 f         375      100.0%      375         375         0
3 m                    0.0%        0           0         0
4 f         375      100.0%      375         375         0
5 m                    0.0%        0           0         0
6 F         375      100.0%      375         375         0
7 m                    0.0%        0           0         0
8 f         375      100.0%      375         375         0
9 m                    0.0%        0           0         0
10 f        375      100.0%      375         375         0
11 m                   0.0%        0           0         0
12 f        375      100.0%      375         375         0
13 m                   0.0%        0           0         0
14 f        375      100.0%      375         375         0
```

```
15 m                 0.0%        0          0          0
0 dirty, 3000 queued, 3000 total, 4096 hash buckets, 2048 buffer size
start clean at  60.000% (of pair total) dirty, or 226 buffs dirty, stop at
50.000%
Buffer pool page size: 8192
4 buffer LRU queue pairs              priority levels
# f/m   pair total     % of    length       LOW      HIGH
0 F         250      100.0%      250         250         0
1 m                    0.0%        0           0         0
2 f         250      100.0%      250         250         0
3 m                    0.0%        0           0         0
4 f         250      100.0%      250         250         0
5 m                    0.0%        0           0         0
6 f         250      100.0%      250         250         0
7 m                    0.0%        0           0         0
0 dirty, 1000 queued, 1000 total, 1024 hash buckets, 8192 buffer size
start clean at  60.000% (of pair total) dirty, or 150 buffs dirty, stop at
50.000%
```

输出描述

Buffer pool page size

是以字节表示的缓冲池页面大小

#

显示队列编号

每个 LRU 队列由两个子队列构成：一个 FLRU 队列和一个 MLRU 队列。（有关 FLRU 和 MLRU 队列的定义，请参阅GBase 8s 管理员指南 的共享内存一章中的 LRU 队列 。）这样，队列 0 和 1 属于第一个 LRU 队列，队列 2 和 3 属于第二个 LRU 队列，依此类推。

f/m

标识队列类型

该字段有 4 种可能值：

f

可用 LRU 队列

在这种上下文中，可用意味着未修改过。尽管 LRU 队列中的几乎所有缓冲区都可用但数据库服务器尝试使用 FLRU 队列而不是 MLRU 队列。（ 在数据库服务器可以使用缓冲区之前，已修改缓冲区必须已写入磁盘。）

F

具有最少元素的可用 LRU

数据库服务器使用该估计确定接下来将未修改（可用）缓冲区置于何处

m

MLRU 队列

M

清仓程序正在清除的 MLRU 队列

length

跟踪受测量队列的长度（以缓冲区为单位）

% of

显示此子队列构成 LRU 队列的百分比

例如：假设一个 LRU 队列具有 50 个缓冲区，其中 30 个缓冲区正在 MLRU 队列中，20 个在 FLRU 队列中。则 % of 列将分别列出 60.00 和 40.00 的百分率

pair total

提供此 LRU 队列中缓冲区的总数

priority levels

显示优先级级别： LOW 、MED_LOW 、MED_HIGH 、HIGH

gstat -R 命令也列出优先级级别。

摘要信息在单独的 LRU 队列信息后面。可如下解释摘要信息：

dirty

是所有 LRU 队列中已修改缓冲区的总数

queued

是 LRU 队列中缓冲区的总数

total

是缓冲区的总数

hash buckets

是哈希桶数

buffer size

是每个缓冲区的大小

start clean

是指定的 BUFFERPOOL 配置参数的 lru_max_dirty 字段值

stop at

是指定的 BUFFERPOOL 配置参数的 lru_min_dirty 字段值

priority downgrades

是已降级为较低优先级的 LRU 队列的数量

priority upgrades

是已升级为较高优先级的 LRU 队列的数量

3.16.30 gstat -s 命令：打印锁存器信息

可以使用 gstat -s 命令显示一般锁存器信息，包含锁存器控制的资源。

语法：

示例输出

图: gstat -s 命令输出

```
Latches with lock or userthread set
name     address  lock wait userthread
```

输出描述

name

使用以下缩写标识锁存器所控制的资源：

archive

存储空间备份

bf

缓冲区

bh

哈希缓冲区

chunks

Chunk 表

ckpt

检查点

dbspace

Dbspace 表

flushctl

页清除程序控制

flushr

页清除程序

locks

Lock 表

loglog

逻辑日志

LRU

LRU 队列

physb1

第一个物理日志缓冲区

physb2

第二个物理日志缓冲区

physlog

物理日志

pt

Tblspace tblspace

tblsps

Tblspace 表

users

用户表

address

是锁存器的地址

如果线程正在等待锁存器，那么该地址显示在 gstat -u （用户）命令输出的 wait 字段

lock

标示锁存器是否已锁定并设置

标示锁状态的代码（1 或 0）与计算机有关

wait

标示是否有任何用户线程正在等待锁存器

userthread

是正在等待锁存器的任何用户线程的共享内存地址

此线程包含线程控制块地址，所有线程都有这些地址。您可以比较该地址与 gstat -u 输出中的用户地址以获得用户进程标识号

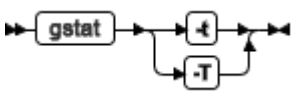
要从 tcb 地址中获得 rstcb 地址，请检查 gstat -g ath 命令的输出，它列出了每个用户线程的这两个地址。

3.16.31 gstat -t 和 gstat -T 命令：打印 tblspace 信息

可以使用 gstat -t 命令显示活动 tblspace 的 tblspace 信息。使用 gstat -T 命令显示所有 tblspace 的 tblspace 信息。

gstat -t 命令页列出了活动 tblspace 的数量和 tblspace 的总数。

语法：



示例输出

图: gstat -t 命令输出

```
Tblspaces
n address  flgs ucnt tblnum   physaddr         npages nused  npdata nrows  nextns
62 a40dc70  0    1    100001   1:14              250    250    0      0      1
195 ac843e0  0    1    1000df   1:236             16     9      4      53     2
2 active, 221 total
```

输出描述

n

是打开的 tblspace 的计数器

address

是共享内存 tblspace 表中的 tblspace 地址

flgs

使用以下标志位描述标志：

0x00000001

正在初始化分区结构

0x00000002

分区已修改。已修改页未清仓到磁盘中

0x00000004

正在删除分区

0x00000008

分区用于伪表

0x00000010

正在 ADD INDEX 或 DROP INDEX 操作中改变分区

0x00000020

正在 ALTER TABLE 操作中更改分区

0x00000080

当 dbspace 关闭时正在删除分区

0x00000100

当删除表时未删除 blobspace 中的简单大对象

0x00000200

分区更改页计数已更新

0x00000400

页已更改为最新的数据库模式

0x00000800

系统临时表

0x00001000

用户临时表

0x00004000

索引操作在恢复过程中推迟

0x00008000

正在截断分区

0x00010000

分区被部分截断

ucnt

使用计数，它指示当前正在访问 tblspace 的用户线程数

tblnum

是以十六进制值表示的 tblspace 编号

等价的整数值显示为 systables 系统目录表中的 partnum 值

physaddr

是 tblspace 的物理地址（在磁盘上）

npages

是分配给 tblspace 的页数

nused

是 tblspace 中已使用页的数量

npdata

是已使用数据页的数量

nrows

是已使用数据行的数量

nextns

是已分配非连续 extent 的数量

该数与已分配下一个 extent 的次数不相等

3.16.32 gstat -u 命令：打印用户活动概要文件

可以使用 gstat -u 命令显示用户活动的概要文件。

语法：

示例输出

图: gstat -u 命令输出

```
Userthreads
    address  flags   sessid   user     tty      wait     tout locks nreads   nwrites
     a4d8018  ---P--D 1        gbasedbt -        0        0    0     58       4595
     a4d8628  ---P--F 0        gbasedbt -        0        0    0     0        2734
     a4d8c38  ---P--- 5        gbasedbt -        0        0    0     0        1
     a4d9248  ---P--B 6        gbasedbt -        0        0    0     40       0
```

```
a4d9858  ---P--D 7        gbasedbt -        0        0    0    0        0
a4d9e68  Y--P--- 21       niraj    -        a65e5a8  0    1    0        0
 6 active, 128 total, 7 maximum concurrent
```

输出描述

-u 选项对每个用户线程提供以下输出：

address

在用户表中用户线程的共享内存地址

比较该地址与 -s 选项（锁存器）输出、-b 、-B 和 -X 选项（缓冲区）以及 -k 选项（锁）中显示的地址以了解该线程正在持有或等待什么资源

flags

提供会话状态。

位置 1 的标志代码：

B

正在等待缓冲区

C

正在等待 checkpoint

G

正在等待对逻辑日志缓冲区的写入

L

正在等待锁

S

正在等待互斥

T

正在等待事务

Y

正在等待条件

X

正在等待事务清除（回滚）

DEFUNCT

该线程已引起严重的断言失败，并已暂挂以允许其他线程继续其工作

位置 2 的标志代码：

*

I/O 故障过程中的事务是活动的

位置 3 的标志代码：

A

Dbspace 备份线程

有关此处显示的其他值，请参阅 -x 选项位置 3 的标志代码。

位置 4 的标志代码：

P

会话主线程

位置 5 的标志代码：

R

正在读取

X

临界段中的线程

位置 6 的标志代码：

R

恢复过程中使用的线程（例如：物理或逻辑恢复）

-

恢复过程中未使用的线程

位置 7 的标记代码：

B

B-tree 清除程序线程

C

已终止正在等待清除的用户线程

D

守护程序线程

F

页清除程序线程

sessid

会话标识符编号

在操作（例如并行排序和并行索引构建）过程中，会话可能有许多与其相关联的用户线程。出于这一原因，会话标识用来标识每个唯一的会话

user

用户登录名（从操作系统派生）

tty

用户正在使用的标准错误（stderr）文件名称（从操作系统派生）

该字段在 Windows™ 上是空白的

wait

如果用户线程正在等待特定锁存器、锁、互斥锁或条件，该字段显示该资源的地址。使用该地址映射到 -s （锁存器）或 -k （锁）选项输出中提供的信息。如果等待是用于持久条件的，那么对 gstat -a 输出中的地址运行 grep

tout

当前等待中的剩余秒数

如果值是 0，那么用户线程不再等待锁存器或锁。如果值是 -1 ，那么用户线程处于不定等待中

locks

用户线程正持有的锁数

-k 输出应包含持有的每个锁的列表

nreads

用户线程已执行的磁盘读取数

nwrites

是用户线程已执行的写调用数

所有写调用都写入共享内存缓冲区高速缓存

gstat -u 命令输出的最后一行显示自初始化数据库服务器以来已分配的并发用户线程的最大数量。例如：样本 gstat -u 命令输出的最后一行如下：

```
4 active, 128 total, 17 maximum concurrent
```

该行的最后部分（ 17 maximum concurrent）指示自初始化数据库服务器以来并发运行的用户线程的最大数为 17 。

该输出还指示了活动用户的数量和允许用户的最大数。

3.16.33 gstat -x 命令：打印数据库服务器事务信息

可以使用 gstat -x 命令显示在数据库服务器上的事务信息。

语法：

只有在以下情况下事务信息才是必需的：

- X/Open 环境
- 参与分布式查询的数据库服务器
- 数据库服务使用 Microsoft™ Transaction Server （MTS）事务管理器

示例输出

图: gstat -x 命令输出

```
Transactions

address   flags userthread  locks  begin    current     isol   est.     retrys coord
logpos   logpos             rb_time
a6d8028   A---- a695028     0      -         -          COMMIT  -         0
a6d8348   A---- a695878     0      -         -          COMMIT  -         0
a6d8668   A---- a6960c8     0      -         -          COMMIT  -         0
a6d8988   A---- a696918     0      -         -          COMMIT  -         0
a6d8fc8   A---- a698208     0      -         -          COMMIT  -         0
a6d92e8   A---- a6979b8     0      -         -          COMMIT  -         0
a6d9608   A---- a698a58     0      -         -          COMMIT  -         0
a6d9928   A---- a6992a8     1      -         -          DIRTY   -         0
a6d9c48   A---- a6992a8     0      -         -          NOTRANS -         0
a6d9f68   A---- a69a348     0      -         -          COMMIT  -         0
a6da288   A---- a69ab98     0      -         -          COMMIT  -         0
a6da5a8   A---- a69b3e8     0      -         -          COMMIT  -         0
a6da8c8   A---- a69bc38     0      -         -          COMMIT  -         0
a6dabe8   A---- a69c488     0      -         -          COMMIT  -         0
a6daf08   A---- a699af8     0      -         -          COMMIT  -         0
a6db228   A---- a6992a8     0      -         -          COMMIT  -         0
a6db548   A---- a69ccd8     1      -         -          DIRTY   -         0
a6db868   A---- a69d528     1      -         -          DIRTY   -         0
a6dbb88   A---- a69ccd8     0      -         -          COMMIT  -         0
a6dbea8   A---- a69dd78     0      -         -          COMMIT  -         0
a6dc1c8   A---- a69e5c8     0      -         -          COMMIT  -         0
a6dc4e8   A-B-- a69ee18   502    33:0x25018  34:0x486fc COMMIT  0:07      0
22 active, 128 total, 23 maximum concurrent
```

输出描述

可以如下解释 gstat -x 命令输出：

address

事务结构的共享内存地址

flags

位置 1 的标志代码（当前事务状态）：

A

用户线程已连接到事务

S

TP/XA 暂挂的事务

C

TP/XA 正在等待回滚

位置 2 的标志代码（事务方式）：

T

紧耦合方式（MTS）

L

松耦合方式（缺省方式）

位置 3 的标志代码（事务阶段）：

B

开始工作

P

准备好用于提交的分布式查询

X

准备好用于提交的 TP/XA

C

正在提交或已提交

R

正在回滚或已回滚

H

正在尝试回滚或已回滚

位置 4 的标志代码

X

XA 事务

位置 5 的标志代码 （事务的类型）：

G

全局事务

C

分布式查询协调者

S

分布式查询从属者

B

分布式查询协调者和从属者

M

重新定向的全局事务

userthread

拥有事务的线程（rstcb 地址）

locks

事务持有的锁数

begin logpos

BEGIN WORK 记录已记录到其中的日志

current logpos

事务最近写入的日志的当前日志位置 (当事务回滚时，当前日志位置会回退直到它到达起始日志位置。 当到达起始日志位置，回滚结束。）

isol

隔离级别

est. rb time

服务器回滚该事务所需的估计时间。随着事务前进，该时间增长。如果事务回滚，那么该时间随着事务的展开而减少

retrys

启动分布式查询的恢复线程的尝试次数

coord

从属者正在执行事务时事务协调者的名称

该字段告诉您哪个数据库服务器正在协调两阶段提交

gstat -x 命令输出的最后一行指示 8 是自初始化数据库服务器以来并发事务的最大数。

```
8 active, 128 total, 8 maximum concurrent
```

确定逻辑日志记录的位置

可以使用 gstat -x 命令确定逻辑日志记录的位置。

curlog 和 logposit 字段提供了逻辑日志记录的确切位置。如果事务不在回滚，curlog 和 logposit 描述最新写入的日志记录的位置。当事务正在回滚时，这些字段描述最新“撤销”的日志记录的位置。随着事务的回滚，curlog 和 logposit 值下降。在长事务中，logposit 和 beginlg 值的聚集率可以帮助您估计回滚还将花费多少时间。

有关 gstat -x 命令示例，请参阅 GBase 8s 管理员指南 中的多阶段提交协议一章中的监视全局事务。

确定全局事务的方式

gstat -x 命令对确定全局事务是以松耦合方式还是以紧耦合方式执行很有用。

gstat -x 命令输出的 flag 列的第二个位置显示全局事务的标志。T 标志指示紧耦合方式，L 标志指示松耦合方式。

- 松耦合方式意味着不同的数据库服务器协调事务但不共享锁。全局事务中的每个分支都具有独立的事务 XID 。在逻辑日志中，所有分支的记录都显示为独立的事务。
- 紧耦合方式意味着不同数据库服务器协调事务并共享诸如锁和日志之类的资源。在全局事务中，访问同一数据库的所有分支共享同一事务 XID 。具有相同 XID 的分支的日志记录显示在同一会话 ID 下面。MTS 使用紧耦合方式。

3.16.34 gstat -X 命令：打印线程信息

可以使用 gstat -X 命令获取关于正在等待缓冲区的线程的确切信息。

对于每个使用中的缓冲区，gstat -X 命令显示一般的缓冲区信息，这些信息也可以使用 gstat -b 或 gstat -B 命令获得。有关更多信息，请参阅 gstat -b 命令：打印正在使用的缓冲区信息 中的 gstat -b 命令。

语法：

示例输出

图: gstat -X 命令输出

```
Buffers (Access)
address  owner    flags pagenum          memaddr  nslots pgflgs scount   waiter
Buffer pool page size: 2048
0 modified, 3000 total, 4096 hash buckets, 2048 buffer size
Buffer pool page size: 8192
0 modified, 1000 total, 1024 hash buckets, 8192 buffer size
```

输出描述

gstat -X 命令具有 waiter 字段，用以列出所有正在等待缓冲区的用户线程，而 gstat -b 和 gstat -B 命令包含 waitlist 字段，它显示正在等待缓冲区的第一个用户线程的地址。共享缓冲区的最大数量以 ONCONFIG 文件中 BUFFERPOOL 配置参数的 buffers 字段进行指定。

Buffer pool page size

以字节表示的缓冲池页面大小

address

缓冲区表中缓冲区头的地址

flags

指示缓存页当前状态的标志：

0x01

已修改数据

0x02

数据

0x04

LRU

0x08

错误

0x10

共享锁

0x20

正在进行 LRU AIO 写

0x40

正在进行 Chunk 写

0x10

互斥锁

0x100

清除程序已指定到 LRU

0x200

缓冲区应该避免 bf_check 调用

0x400

在写页面之前进行日志清空

0x800

缓冲区已进行缓冲区检查

0x8000

缓冲区已固定

pagenum

磁盘上的物理页数

memaddr

缓冲区内存地址

nslots

页中 solt 表的条目的数量

该字段指示存储在该页上的行（或行的一部分）的数量

pgflgs

使用以下值（单独或组合）来描述页类型：

1

数据页

2

Tblspace 页

4

可用列表页

8

Chunk 可用列表页

9

剩余数据页

b

分区常驻 blobpage

c

Blobspace 常驻 blobpage

d

Blob chunk 可用列表位页

e

Blob chunk blob 图页

10

B-tree 节点页

20

B-tree 根节点页

40

B-tree 分支节点页

80

B-tree 叶节点页

100

逻辑日志页

200

逻辑日志的最后一页

400

逻辑日志的同步页

800

物理日志

1000

保留根页

2000

不需要物理日志

8000

带有缺省标志的 B-tree 叶

scount

显示正在等待缓冲区的线程数

waiter

列出正在等待缓冲区的所有用户线程的地址

3.16.35 gstat -z 命令：清除统计信息

可以使用 gstat -z 命令清除数据库服务器的统计信息（包括与 Enterprise Replication 相关联的统计信息），并将概要文件计数设置为 0 。

如果使用 gstat -z 命令重置和监视某些字段的计数，那么应了解概要文件计数对于数据库服务器管理的任何数据库中发生的所有活动都是增加的。任何用户都可以重设概要文件计数从而对另一用户正在执行的监视进行干预。

语法：

3.16.36 退出 gstat 实用程序时的返回码

当您退出 gstat 实用程序时，该实用程序会显示一个返回码的设置。

示例

以下是当您退出 gstat 实用程序时，显示的返回码和消息的几行示例：

```
GLS failures: -1

Failed to attach shared memory: -1

Failed to attach shared memory when running 'gstat -': 255

All other errors detected by gstat: 1

No errors detected by gstat: 0

Administration mode: 7
```

返回码值

下表列出了与退出 gstat 实用程序时显示的返回码相关的服务器方式。

值	解释
-1	GLS 本地语言环境初始化失败或 GBase 8s 连接共享内存失败
0	初始化方式
1	静默方式
2	恢复方式
3	备份方式
4	关机方式
5	联机方式
6	中止方式
7	用户方式
255	脱机方式

4 SQL 管理 API 函数

这些主题描述 SQL 管理 API admin() 和 task() 函数。

4.1 SQL 管理 API 概述

使用 SQL 管理 API 来通过 SQL 语句远程地管理 GBase 8s 。

SQL 管理 API 由两个函数组成：admin() 和 task()。这些函数执行相同的操作，但返回不同格式的结果。这些函数带有一个或多个定义操作的参数。许多操作是您也可通过命令行实用程序完成的操作。使用 SQL 管理 API 函数的好处是，您可从其他数据库服务器远程地运行它们；反之，您必须直接地连接到运行命令行实用程序命令的那台数据库服务器上。

您可在 SQL 语句内调用 admin() 和 task() 函数，该语句可包括表达式，或您可使用 EXECUTE FUNCTION 语句来调用它们。在一个事务内运行 admin() 或 task() 函数，该事务不包括任何其他语句。

在 sysadmin 数据库中定义 SQL 管理 API 函数。您必须连接到 sysadmin 数据库，或直接地或远程地来运行这些函数。

仅下列用户可运行 SQL 管理 API 函数：

- 用户 gbasedbt
- root 用户，如果将 sysadmin 数据库上的 Connect 权限授予该用户
- DBSA 组成员，如果将 sysadmin 数据库上的 Connect 权限授予该角色
- 通过带有 grant admin 参数的 admin() 和 task() 函数授予 SQL 管理 API 命令权限的用户。

为了复制文件中存在的存储空间、chunk 和日志，您可生成 SQL 管理 API 命令。为此，请运行带有 -c 选项的 dbschema 实用程序。

4.1.1 admin() 和 task() 函数语法行为

admin() 和 task() 函数带有一个或多个参数，是以逗号分隔的引用字符串。

admin() 和 task() 函数的语法包括下列规则：

- 每一参数必须用一对单引号（’）或双引号（″）界定。
- 必须用逗号分隔参数。
- 参数的最大数目是 28。
- 大多数参数不区分大小写，只有下列例外：
  - 紧跟在初始 gadmin 参数后的参数区分大小写。

  例如：

EXECUTE FUNCTION task("gadmin","D","50");

- 包括 cdr 参数的参数区分大小写。

例如：

```
EXECUTE FUNCTION task("cdr define server","-c=g_amsterdam","--init
g_amsterdam");
```

- 如果您未直接地连接到 sysadmin 数据库，则必须包括 sysadmin 数据库名和服务器名，根据标准“数据库对象名称”语法。例如，如果服务器名是 ids_server，则您可运行下列语句：

```
EXECUTE FUNCTION sysadmin@ids_server:admin("add bufferpool","2",
                    "50000","8","60.0","50.0");
```

要了解更多有关“数据库对象名称”语法的信息，请参阅 《GBase 8s SQL 指南：语法》。

4.1.2 admin() 和 task() 参数大小规范

缺省情况下，在 admin() 和 task() 函数中指定参数大小的单位是 KB。您可指定其他单位。

您可使用下列 admin() 和 task() 函数参数大小的单位：

记号

相当的单位

B

字节

K

KB（缺省）

M

MB

G

GB

T

TB

P

PB

忽略这些字符的字母大小写。

忽略在同一参数中分隔大小规范和单位缩写的空格。例如，规范 "128M" 与 "128 m"都解释为 128 MB。

当省略大小参数时，或者根据配置参数的设置，如未设置参数则或者根据系统缺省大小，应用对象的缺省大小。例如，存储空间有 100 MB 的缺省大小。

4.1.3 admin() 和 task() 函数返回代码

admin() 和 task() 函数执行相同的任务，但产生不同的返回代码类型。如果您想要一个整数返回代码，则使用 admin() 函数，或者如果您想要一个文本的返回代码，则使用 task() 函数。

当您运行 admin() 或 task() 函数时，函数：

- 执行指定的操作。
- 返回一个表示函数成功还是失败的值。
- 在 sysadmin 数据库的 command_history 表内插入一行。

admin() 和 task() 函数的返回代码以不同的格式表明函数是成功了还是失败了：

- task() 函数返回一个文本的消息。该消息也插入到 task() 函数插入到 command_history 表内那个新行中的 cmd_ret_msg 列。
- admin() 函数返回一个整数。这个数也插入到 admin() 函数插入到 command_history 表内那个新行中的 cmd_ret_msg 列。
  - 如果这个值大于零，则函数成功，且在 command_history 表内插入一新行。
  - 如果这个值是零，则函数成功，但是 GBase 8s 可能不在 command_history 表内插入一新行。
  - 如果这个值小于零，则函数失败，但是在 command_history 表内插入一新行。

admin() 或 task() 函数指定的操作发生在一个单独的事务中，与在 command_history 表内插入新行的事务分开。如果成功地执行命令，但是在 command_history 表内插入失败，则命令生效，但是一个 online.log 错误条目指明该问题。

当调用这个函数时，如果 command_history.cmd_number 连续计数器是 200，且命令成功，则 GBase 8s 执行命令并返回整数 201。如果命令失败，则这个示例返回值 -201。

假定 task() 函数执行了同一命令：

```
EXECUTE FUNCTION task("check extents");
```

这个命令指导数据库服务器检查 extent，并返回指定命令成功或失败的消息。

当调用这个函数时，如果 command_history.cmd_number 连续计数器是 201，且命令失败，则返回值是 -202。假定 DBSA 调用的下一个 SQL 管理 API 函数是这个：

```
EXECUTE FUNCTION admin('create dbspace',
    'dbspace2', '/work/CHUNKS/dbspace2', "20M");
```

如果在这种情况下命令成功，则返回的值是 203。DBSA 可使用下列查询来检查这些调用 admin() 函数插入 command_history 表的两行：

```
SELECT * FROM command_history WHERE cmd_number IN (202,203);
```

这个查询返回两行：

```
cmd_number      202
cmd_exec_time   2009-04-17 16:26:14
cmd_user        informix
cmd_hostname    olympia
cmd_executed    create dbspace
cmd_ret_status  -1
cmd_ret_msg     Unable to create file /work/dbspace2

cmd_number      203
cmd_exec_time   2009-04-17 16:26:15
cmd_user        informix
cmd_hostname    olympia
cmd_executed    create dbspace
cmd_ret_status  0
cmd_ret_msg     created dbspace number 2 named dbspace2
```

4.2 SQL 管理 API 门户：按权限组划分参数

您可查看 admin() 和 task() 函数参数的列表，按照权限组排序，带有指向有关参数信息的链接。

权限组标识用户可运行哪些 SQL 管理 API 命令。有些函数参数在多个权限组中。将权限组授予用户，以便他们为了工作需要可运行这些命令。缺省情况下，仅用户 gbasedbt 或 DBSA 可运行 SQL 管理 API 命令。

使用 grant admin 参数来授予权限，以及 revoke admin 参数来撤销权限。

- ADMIN：用户可运行所有 SQL 管理 API 函数。
- BAR 权限组：用户可运行备份和恢复函数。
- FILE 权限组：用户可管理消息日志并显示文件信息。
- GRANT 权限组：用户有权限授予和撤销权限。
- HA 权限组：用户可运行高可用性函数。
- MISC 权限组：用户可管理数据库服务器。
- MONITOR 权限组：用户可运行仅显示信息的所有 SQL 管理 API 函数。
- OPERATOR：用户可运行所有 SQL 管理 API 函数，除了 GRANT 权限组中的函数。
- REPLICATION 权限组：用户可运行 Enterprise Replication cdr 实用程序函数。
- SQL 权限组：用户可运行与管理数据库的 SQL 语句有关的函数。
- SQLTRACE 权限组：用户可运行 SQL 跟踪函数。
- STORAGE 权限组：用户可运行与空间相关的函数。
- TENANT 权限组：用户可运行 tenant 数据库功能。

BAR 权限组

BAR 权限组包括备份数据库的 SQL 管理 API 函数参数。

表 1. 备份和恢复admin() 和 task() 函数参数

参数
archive fake 参数：执行无记录的备份（SQL 管理 API）
gtape archive 参数：备份数据库上的数据（SQL 管理 API）
gbackuprestore 参数：备份存储空间（SQL 管理 API）
onsmsync 参数：与存储管理器目录同步（SQL 管理 API）

FILE 权限组

FILE 权限组包括管理消息日志和显示文件信息的 SQL 管理 API 函数参数。

表 2. 消息日志命令的 admin() 和 task() 函数参数

参数
file status 参数：显示消息日志文件的状态（SQL 管理 API）
message log rotate 参数：轮换消息日志文件（SQL 管理 API）
message log delete 参数：删除消息日志文件（SQL 管理 API）
message log truncate 参数：删除消息日志文件的内容（SQL 管理 API）
print file info 参数：显示目录或文件信息（SQL 管理 API）

GRANT 权限组

GRANT 权限组包括将运行 SQL 管理 API 命令的权限授予其他用户或撤销的 SQL 管理 API 函数参数。

表 3. 授予和撤销权限的 admin() 和 task() 函数参数

参数
grant admin 参数：授予运行 SQL 管理 API 命令的权限
revoke admin 参数：撤销运行 SQL 管理 API 命令的权限

HA 权限组

HA 权限组包括管理高可用性集群的 SQL 管理 API 函数参数。

表 4. 高可用性集群命令的 admin() 和 task() 函数参数

参数
ha make primary 参数：更改辅助服务器的模式（SQL 管理 API）
ha rss 参数：创建 RS 辅助服务器（SQL 管理 API）
ha rss add 参数：将 RS 辅助服务器添加到主服务器（SQL 管理 API）
ha rss change 参数：更改 RS 辅助服务器的口令（SQL 管理 API）
ha rss delete 参数：删除 RS 辅助服务器（SQL 管理 API）
ha sds clear 参数：停止共享磁盘复制（SQL 管理 API）
ha sds primary 参数：将 SD 辅助服务器转换到主服务器（SQL 管理 API）
ha sds set 参数：创建共享磁盘主服务器（SQL 管理 API）
ha set idxauto 参数：复制索引到辅助服务器（SQL 管理 API）
ha set ipl 参数：在主服务器上构建日志索引（SQL 管理 API）
ha set primary 参数：定义 HDR 主服务器（SQL 管理 API）
ha set secondary 参数：定义 HDR 辅助服务器（SQL 管理 API）
ha set standard 参数：将 HDR 服务器转换成为标准服务器（SQL 管理 API）
ha set timeout 参数：更改 SD 辅助服务器超时（SQL 管理 API）
gadmin 和 d 参数：设置数据复制类型（SQL 管理 API）

MISC 权限组

MISC 权限组包括管理数据库服务器的 SQL 管理函数参数：

- gstat
- 配置参数
- 数据、分区和 extent
- 监听线程
- 消息日志
- 内存
- PDQ
- 服务器模式
- SQL 语句高速缓存

- 其他管理任务

gstat

通过运行 gstat 命令监视数据库服务器的 SQL 管理 API 函数参数。

表 5. gstat 命令的 admin() 和 task() 函数参数

参数
gstat 参数：监视数据库服务器（SQL 管理 API）

配置参数

更新配置参数的 SQL 管理 API 函数参数。

表 6. 配置参数命令的 admin() 和 task() 函数参数

参数
export config 参数：导出配置参数值（SQL 管理 API）
import config 参数：导入配置参数值（SQL 管理 API）
modify config 参数：更改配置参数（SQL 管理 API）
gadmin 和 wf 参数：永久地更新配置参数（SQL 管理 API）
gadmin 和 wm 参数：临时地更新配置参数（SQL 管理 API）
gadmin、wm 和 AUTO_LRU_TUNING 参数：更改 LRU 调整状态（SQL 管理 API）
reset config 参数：恢复配置参数值（SQL 管理 API）
reset config all 参数：恢复所有动态地可更新的配置参数值（SQL 管理 API）
set onconfig memory 参数：临时地更改配置参数（SQL 管理 API）
set onconfig permanent 参数：永久地更改配置参数（SQL 管理 API）

数据、分区和 extent

管理数据、分区和 extent 的 SQL 管理 API 函数参数。

表 7. 数据、分区和 extent 命令的 admin() 和 task() 函数参数

参数
check data 参数：检查数据一致性（SQL 管理 API）

参数
check extents 参数：检查 extent 一致性（SQL 管理 API）
check partition 参数：检查分区一致性（SQL 管理 API）
checkpoint 参数：强制一检查点（SQL 管理 API）
create dbaccessdemo 参数：创建演示数据库（SQL 管理 API）
gadmin 和 C 参数：控制 B-tree 扫描程序（SQL 管理 API）
gadmin 和 c 参数：强制检查点（SQL 管理 API）
print partition 参数：打印分区信息（SQL 管理 API）
set dataskip 参数：启动或停止跳过 dbspace（SQL 管理 API）
set index compression 参数：更改索引页压缩（SQL 管理 API）

监听线程

不中断现有的连接而控制 SOCTCP 或 TLITCP 网络连接协议监听线程的 SQL 管理 API 函数参数。

表 8. 监听线程命令的 admin() 和 task() 函数参数

参数
restart listen 参数：动态地停止并启动监听线程（SQL 管理 API）
start listen 参数：动态地启动监听线程（SQL 管理 API）
stop listen 参数：动态地停止监听线程（SQL 管理 API）

消息日志

管理消息日志的 SQL 管理 API 函数参数。

表 9. 消息日志命令的 admin() 和 task() 函数参数

参数
file status 参数：显示消息日志文件的状态（SQL 管理 API）
message log rotate 参数：轮换消息日志文件（SQL 管理 API）

参数
message log delete 参数：删除消息日志文件（SQL 管理 API）
message log truncate 参数：删除消息日志文件的内容（SQL 管理 API）

内存

管理内存的 SQL 管理 API 函数参数。

表 10. 内存命令的 admin() 和 task() 函数参数

参数
add bufferpool 参数：添加缓冲池（SQL 管理 API）
add memory 参数：增加共享内存（SQL 管理 API）
gadmin 和 a 参数：添加共享内存段（SQL 管理 API）
gadmin 和 F 参数：释放不用的内存段（SQL 管理 API）
gadmin 和 n 参数：解锁驻留内存（SQL 管理 API）
gadmin 和 r 参数：强制共享内存的驻留（SQL 管理 API）
scheduler lmm enable 参数：指定自动的低内存管理设置（SQL 管理 API）
scheduler lmm disable 参数：停止自动的低内存管理（SQL 管理 API）

PDQ

管理 PDQ 的 SQL 管理 API 函数参数。

表 11. PDQ 命令的 admin() 和 task() 函数参数

参数
gadmin 和 D 参数：设置 PDQ 优先级（SQL 管理 API）
gadmin 和 M 参数：临时地更改决策支持内存（SQL 管理 API）
gadmin 和 Q 参数：设置决策支持查询的最大数目（SQL 管理 API）
gadmin 和 S 参数：设置决策支持扫描的最大数目（SQL 管理 API）

服务器模式

更改服务器模式的 SQL 管理 API 函数参数。

表 12. 服务器模式命令的 admin() 和 task() 函数参数

参数
gadmin 和 j 参数：切换数据库服务器到管理模式（SQL 管理 API）
gadmin 和 m 参数：切换到多用户模式（SQL 管理 API）

SQL 语句高速缓存

管理 SQL 语句高速缓存的 SQL 管理 API 函数参数。

表 13. SQL 语句高速缓存命令的 admin() 和 task() 函数参数

参数
gadmin 和 e 参数：更改 SQL 语句高速缓存的用法（SQL 管理 API）
gadmin 和 W 参数：重置语句高速缓存属性（SQL 管理 API）

其他管理任务

管理其他管理任务的 SQL 管理 API 函数参数。

表 14. 其他管理任务命令的 admin() 和 task() 函数参数

参数
alter logmode 参数：更改数据库日志记录模式（SQL 管理 API）
create dbaccessdemo 参数：创建演示数据库（SQL 管理 API）
gadmin 和 e 参数：更改 SQL 语句高速缓存的用法（SQL 管理 API）
gadmin 和 l 参数：切换到下一个逻辑日志（SQL 管理 API）
gadmin 和 p 参数：添加或移除虚拟处理器（SQL 管理 API）
gadmin 和 Y 参数：更改会话的查询计划度量（SQL 管理 API）
gadmin 和 z 参数：终止用户会话（SQL 管理 API）
gadmin 和 Z 参数：终止分布式事务（SQL 管理 API）
print error 参数：打印错误消息（SQL 管理 API）
reset sysadmin 参数：移动 sysadmin 数据库（SQL 管理 API）

参数
scheduler 参数：停止或启动调度程序（SQL 管理 API）

MONITOR 权限组

MONITOR 权限组包括监视消息日志、Enterprise Replication 和压缩估算的 SQL 管理函数参数。

表 15. 监视消息日志、Enterprise Replication 或压缩估算的 admin() 和 task() 函数参数

参数
cdr error、cdr finderr、cdr list repair、cdr list replicate、cdr list replicateset、cdr list server、cdr list template、cdr stats recv 和 cdr stats rqm 参数 cdr 参数：管理 Enterprise Replication（SQL 管理 API）
file status 参数：显示消息日志文件的状态（SQL 管理 API）
index estimate_compression 参数：估计索引压缩（SQL 管理 API）
print error 参数：打印错误消息（SQL 管理 API）
gstat 参数：监视数据库服务器（SQL 管理 API）
table estimate_compression 和 fragment estimate_compression 参数 table 或 fragment 参数：压缩数据和优化存储（SQL 管理 API）

REPLICATION 权限组

REPLICATION 权限组包括管理 Enterprise Replication 的 SQL 管理 API 函数参数。

表 16. Enterprise Replication 命令的 admin() 和 task() 函数参数

参数
cdr 参数：管理 Enterprise Replication（SQL 管理 API）

SQL 权限组

SQL 权限组包括创建和删除数据库以及查看错误消息的 SQL 管理 API 函数参数。

表 17. 数据库和错误消息的 admin() 和 task() 函数参数

参数
create database 参数：创建数据库（SQL 管理 API）
create dbaccessdemo 参数：创建演示数据库（SQL 管理 API）
drop database 参数：删除数据库（SQL 管理 API）
print error 参数：打印错误消息（SQL 管理 API）

SQLTRACE 权限组

SQLTRACE 权限组包括管理 SQL 跟踪的 SQL 管理 API 函数参数。

表 18. SQL 跟踪命令的 admin() 和 task() 函数参数

参数
set sql tracing 参数：设置全局 SQL 跟踪（SQL 管理 API）
set sql tracing database 参数：更改数据库跟踪（SQL 管理 API）
set sql tracing session 参数：控制对会话的跟踪（SQL 管理 API）
set sql tracing user 参数：控制对用户的跟踪（SQL 管理 API）
set sql user tracing 参数：设置对用户会话的全局 SQL 跟踪（SQL 管理 API）

STORAGE 权限组

STORAGE 权限组包括管理存储的下列方面的 SQL 管理 API 函数参数：

- 自动表存储位置参数
- 压缩
- 逻辑和物理日志
- 镜像
- 存储空间
- 存储供应

自动表存储位置参数

管理存储自动分配分片的 dbspace 列表的 SQL 管理 API 函数参数。

表 19. 表存储命令的 admin() 和 task() 函数参数

参数

参数
autolocate database add 参数：添加 dbspace 到 dbspace 列表（SQL 管理 API）
autolocate database anywhere 参数：添加所有 dbspace 到 dbspace 列表（SQL 管理 API）
autolocate database 参数：指定自动定位和分片的 dbspace（SQL 管理 API）
autolocate database off 参数：禁用数据库的自动分片（SQL 管理 API）
autolocate database remove 参数：从 dbspace 列表中移除 dbspace（SQL 管理 API）

压缩

管理数据压缩并优化存储的 SQL 管理 API 函数参数。

表 20. 压缩命令的 admin() 和 task() 函数参数

参数
index compress repack shrink 参数：优化 B-tree 索引的存储（SQL 管理 API）
index estimate_compression 参数：估计索引压缩（SQL 管理 API）
table 或 fragment 参数：压缩数据和优化存储（SQL 管理 API）
清除压缩字典参数：移除压缩字典（SQL 管理 API）

要了解压缩和存储优化命令的概况，请参阅表和分片压缩和解压缩操作（SQL 管理 API）。

逻辑和物理日志

管理逻辑和物理日志的 SQL 管理 API 函数参数。

表 21. 日志命令的 admin() 和 task() 函数参数

参数
add log 参数：添加新逻辑日志（SQL 管理 API）
alter logmode 参数：更改数据库日志记录模式（SQL 管理 API）
alter plog 参数：更改物理日志（SQL 管理 API）
drop log 参数：删除逻辑日志（SQL 管理 API）

镜像

管理镜像的 SQL 管理 API 函数参数。

表 22. 镜像命令的 admin() 和 task() 函数参数

参数
add mirror 参数：添加镜像 chunk（SQL 管理 API）
start mirroring 参数：启动存储空间镜像（SQL 管理 API）
stop mirroring 参数：停止存储空间镜像（SQL 管理 API）

存储空间

管理 chunk、blobspace、dbspace 和 sbspace 的 SQL 管理 API 函数参数。

表 23. 空间命令的 admin() 和 task() 函数参数

参数
add chunk 参数：添加新 chunk（SQL 管理 API）
alter chunk 参数：更改 chunk 状态为 online 或 offline（SQL 管理 API）
clean sbspace 参数：释放未引用的智能大对象（SQL 管理 API）
create blobspace 参数：创建 blobspace（SQL 管理 API）
create chunk 参数：创建 chunk（SQL 管理 API）
create dbaccessdemo 参数：创建演示数据库（SQL 管理 API）
create dbspace 参数：创建 dbspace（SQL 管理 API）
create sbspace 参数：创建 sbspace（SQL 管理 API）
create sbspace with accesstime 参数：创建跟踪访问时间的 sbspace（SQL 管理 API）
create sbspace with log 参数：创建带有事务日志记录的 sbspace（SQL 管理 API）
create tempdbspace 参数：创建临时 dbspace（SQL 管理 API）
create tempsbspace 参数：创建临时 sbspace（SQL 管理 API）
drop blobspace 参数：删除 blobspace（SQL 管理 API）
drop chunk 参数：删除 chunk（SQL 管理 API）

参数
drop dbspace 参数：删除 dbspace（SQL 管理 API）
drop sbspace 参数：删除 sbspace（SQL 管理 API）
drop tempdbspace 参数：删除临时 dbspace（SQL 管理 API）
gadmin 和 0 参数：标记禁用的 dbspace 为 down（SQL 管理 API）
print error 参数：打印错误消息（SQL 管理 API）
rename space 参数：重命名存储空间（SQL 管理 API）
set chunk 参数：更改 chunk 的状态（SQL 管理 API）
set sbspace accesstime 参数：控制访问次数跟踪（SQL 管理 API）
set sbspace avg_lo_size 参数：设置智能大对象的平均大小（SQL 管理 API）
set sbspace logging 参数：更改 sbspace 的日志记录（SQL 管理 API）

存储供应

管理来自存储池的 chunk、blobspace、dbspace 和 sbspace 的 SQL 管理 API 函数参数。

表 24. 存储供应空间命令的 admin() 和 task() 函数参数

参数
create blobspace from storagepool 参数：从存储池创建 blobspace（SQL 管理 API）
create chunk from storagepool 参数：从存储池创建 chunk（SQL 管理 API）
create dbspace from storagepool 参数：从存储池创建 dbspace（SQL 管理 API）
create plogspace 参数：创建 plogspace（SQL 管理 API）
create sbspace from storagepool 参数：从存储池创建 sbspace（SQL 管理 API）
create tempdbspace 参数：创建临时 dbspace（SQL 管理 API）
create tempdbspace from storagepool 参数：从存储池创建临时 dbspace（SQL 管理 API）
drop blobspace to storagepool 参数：从空 blobspace 归还空间到存储池（SQL 管理 API）

参数
drop chunk to storagepool 参数：从空 chunk 归还空间到存储池（SQL 管理 API）
drop dbspace to storagepool 参数：从空 dbspace 归还空间到存储池（SQL 管理 API）
drop plogspace 参数：删除 plogspace（SQL 管理 API）
drop sbspace to storagepool 参数：从空 sbspace 归还空间到存储池（SQL 管理 API）
drop tempsbspace to storagepool 参数：从空的临时 sbspace 归还空间到存储池（SQL 管理 API）
modify chunk extend 参数：扩展 chunk 的大小（SQL 管理 API）
modify chunk extendable off 参数：标记 chunk 为不可扩展的（SQL 管理 API）
modify chunk extendable 参数：标记 chunk 为可扩展的（SQL 管理 API）
modify space expand 参数：扩大空间的大小（SQL 管理 API）
modify space sp_sizes 参数：更改可扩展的存储空间的大小（SQL 管理 API）
storagepool add 参数：添加存储池条目（SQL 管理 API）
storagepool modify 参数：更改存储池条目（SQL 管理 API）
storagepool delete 参数：删除一个存储池条目（SQL 管理 API）
storagepool purge 参数：删除存储池条目（SQL 管理 API）

TENANT 权限组

TENANT 权限组包括管理 tenant 数据库的 SQL 管理 API 函数参数。

表 25. tenant 数据库命令的 admin() 和 task() 函数参数

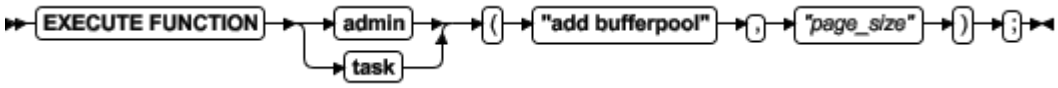
参数
tenant create 参数：创建 tenant 数据库（SQL 管理 API）
tenant drop 参数：删除 tenant 数据库（SQL 管理 API）
tenant update 参数：更改 tenant 数据库属性（SQL 管理 API）

4.3 add bufferpool 参数：添加缓冲池（SQL 管理

API）

随同 admin() 或 task() 函数，使用 add bufferpool 参数来创建缓冲池。

语法



元素	描述	关键考虑
page_size	以 KB 为单位的页大小。	页大小必须是一个缺省页大小的整数倍，且不可大于 16 KB。在 Windows™ 上，页大小总是 4 KB。

用法

使用 add bufferpool 参数来为已经没有缓冲池的页大小创建一个缓冲池。您创建的缓冲池的所有其他特性都设置为 BUFFERPOOL 配置参数缺省行中那些域的值。

这个函数等同于 glogadmin -b -g 命令和 BUFFERPOOL 配置参数。

示例

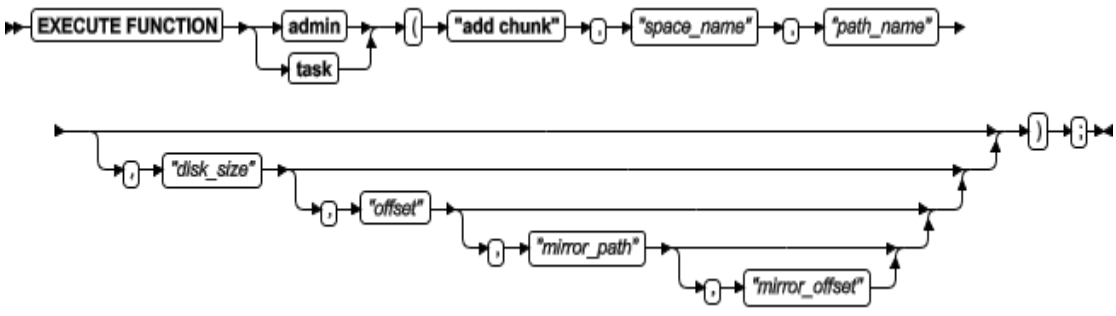
下列示例添加一个页大小为 8 KB 的缓冲池：

```
EXECUTE FUNCTION task("add bufferpool","8");
```

4.4 add chunk 参数：添加新 chunk（SQL 管理 API）

随同 admin() 或 task() 函数，使用 add chunk 参数来添加 chunk 到 dbspace 或 blobspace。

语法



元素	描述	关键考虑
disk_size	要添加的以 KB 为单位的磁盘空间量	请参阅 admin() 和 task() 参数大小规范。

元素	描述	关键考虑
mirror_offset	镜像 chunk 的位置。	
mirror_path	到镜像 chunk 的路径。	如果您正在添加 chunk 到镜像的存储空间，则还必须添加镜像 chunk。
offset	新 chunk 的位置。	
path_name	添加的磁盘空间的路径。	
space_name	您正在添加磁盘空间到其上的 dbspace、blobspace 或 sbspace 的名称。	

用法

chunk 的大小必须等于或大于 1000 KB 且是页大小的倍数。起始偏移量加上 chunk 大小不可超过最大 chunk 大小。最大偏移量是 4 TB。

这个函数等同于 gspaces -a 命令。

示例

下列示例在 5200 KB 偏移量处，添加 5 MB 裸磁盘空间的 chunk 到一名为 dbspc3 的 dbspace：

```
EXECUTE FUNCTION task("add chunk", "dbspc3","\\.\e:","5120","5200");
```

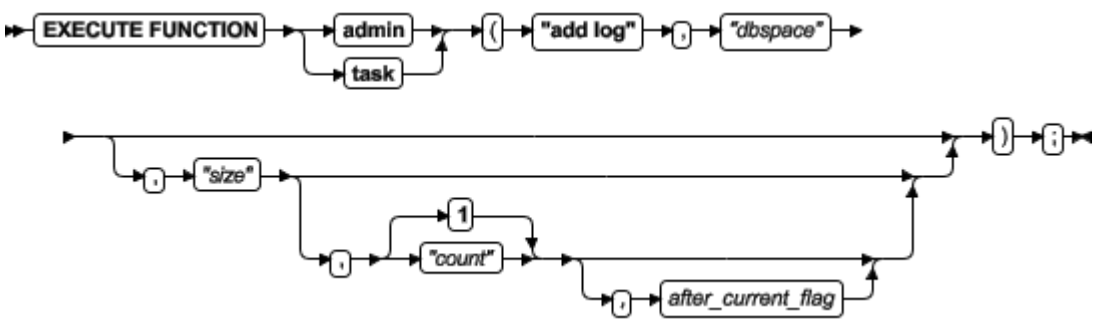
下列示例添加 10 MB 镜像 chunk 到名为 blobsp3 的 blobspace，主 chunk 和镜像 chunk 的偏移量都是 200 KB：

```
EXECUTE FUNCTION task("add chunk","blobsp3","/dev/raw_dev1","10240",
        "200","/dev/raw_dev2","200");
```

4.5 add log 参数：添加新逻辑日志（SQL 管理 API）

随同 admin() 或 task() 函数，使用 add log 参数来添加逻辑日志到 dbspace。

语法



元素	描述	关键考虑
after_current_flag	是在当前日志之后还是在最后的逻辑日志之后（缺省）添加新日志。	可能的值是： • 1 = 在当期日志之后添加新日志。 • 0 = 在最后的日志之后添加新日志。
count	要创建的日志文件数。缺省是 1。	该数目必须不导致逻辑日志文件的总数目超过 32,767。
dbspace	要将逻辑日志文件插入其中的那个 dbspace 的名称。	仅当数据库服务器有足够的连续空间时，您才可添加日志文件到 dbspace。 您可在备份期间添加日志文件。 您不可添加日志文件到 blobspace 或 sbspace。
size	以 KB 为单位的新逻辑日志文件的大小。缺省是由 LOGSIZE 配置参数指定的大小。	这个值必须是一个大于或等于 200 KB 的无符号整数。 还请参阅 admin() 和 task() 参数大小规范。

用法

新添加的日志文件有状态 A 且立即可用。使用 gstat -l 来查看逻辑日志文件的状态。运行这个函数之后，请尽快对包含该日志文件的 root dbspace 和 dbspace 进行 0 级备份。缺省情况下，新日志文件被添加在最后的逻辑日志之后。请包括 1 作为第五个参数来将逻辑日志文件添加在当前日志文件之后。

这个函数类似于 glogadmin -a -d 命令，可添加单个的逻辑日志文件。然而，您可调用这个函数一次，添加多个逻辑日志文件到指定的 dbspace。

示例

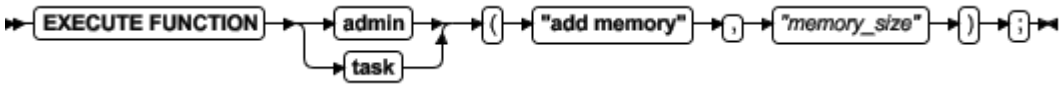
下列示例中的命令在当前日志之后添加三个逻辑日志，每一个的大小都是 5 MB：

```
EXECUTE FUNCTION task ("add log","logdbs","5M",3,1);
```

4.6 add memory 参数：增加共享内存（SQL 管理 API）

随同 admin() 或 task() 函数，使用 add memory 参数来添加共享内存的虚拟部分。

语法



元素	描述	关键考虑
memory_size	以 KB 为单位的新虚拟共享内存段的大小。	这个值必须不超过操作系统对共享内存段大小的限制

用法

这个大小预置 SHMADD 配置参数。

这个函数等同于 gadmin -a 命令。

示例

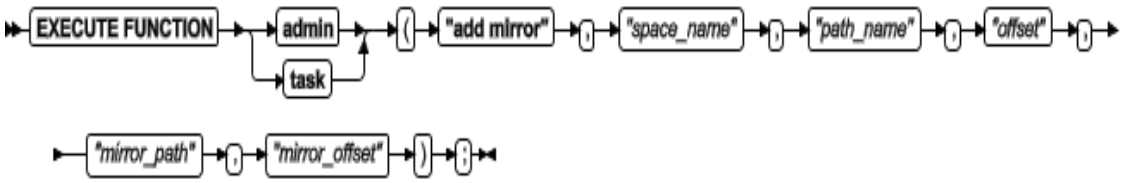
下列示例添加 500 KB 虚拟共享内存：

```
EXECUTE FUNCTION task("add memory","500");
```

4.7 add mirror 参数：添加镜像 chunk（SQL 管理 API）

随同 admin() 或 task() 函数，使用 add mirror 参数来添加镜像 chunk 到 dbspace。

语法



元素	描述	关键考虑
mirror_path	执行镜像的 dbspace、blobspace 或 sbspace 的初始 chunk 的磁盘分区或无缓冲的设备。	
mirror_offset	达到新镜像 dbspace、blobspace 或 sbspace 的镜像 chunk 的偏移量。	请参阅 admin() 和 task() 参数大小规范。
offset	磁盘分区内或无缓冲的设备内，以 KB 为单位的达到新镜像 dbspace、blobspace 或 sbspace 的初始 chunk 的偏移量。	请参阅admin() 和 task() 参数大小规范。
path_name	您想要镜像的 dbspace、blobspace 或 sbspace	

元素	描述	关键考虑
	的初始 chunk 的磁盘分区或无缓冲的设备。	
space_name	要镜像的 dbspace、blobspace 或 sbspace 的名称。	

用法

这个函数等同于 gspaces -m 命令。

示例

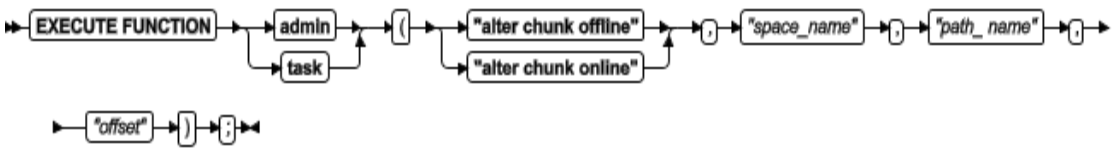
下列示例添加镜像 chunk 到名为 blobsp3 的 blobspace：

```
EXECUTE FUNCTION task("add mirror","blobsp3","/dev/raw_dev1",
"10240","/dev/raw_dev2","200");
```

4.8 alter chunk 参数：更改 chunk 状态为 online 或 offline（SQL 管理 API）

随同 admin() 或 task() 函数，使用 alter chunk 参数在 dbspace、blobspace 或 sbspace 中使 chunk 为 online 或使 chunk 为 offline。

语法



元素	描述	关键考虑
space_name	blobspace、dbspace 或 sbspace 的名称。	
path_name	chunk 的磁盘分区或无缓冲设备。	
offset	磁盘分区或无缓冲设备内达到 chunk 的偏移量（以 KB 为单位）。缺省是 0。	请参阅 admin() 和 task() 参数大小规范。

用法

chunk 必须在镜像对中，或非关键 dbspace 内的非主 chunk。

使用 alter chunk online 参数来更改 chunk 状态为 online。

使用 alter chunk offline 参数来更改 chunk 状态为 offline。

这个函数等同于 gspaces -s 命令。

示例

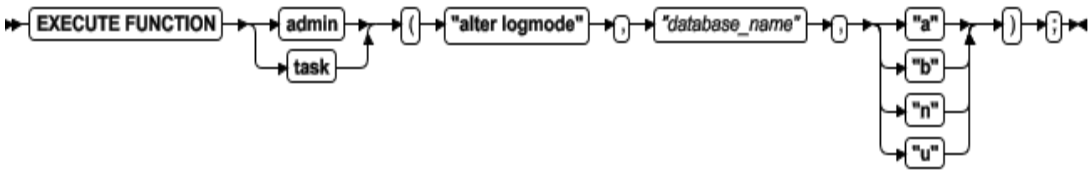
下列示例使名为 dbspace4 的 chunk 为 online：

```
EXECUTE FUNCTION task("alter chunk online","dbspace4","/dev/raw_dev1","0");
```

4.9 alter logmode 参数：更改数据库日志记录模式（SQL 管理 API）

随同 admin() 或 task() 函数，使用 alter logmode 参数来更改数据库日志记录模式为 ANSI、缓冲的、无日志记录或无缓冲的。

语法



元素	描述	关键考虑
database_name	您想要改变其日志记录模式的数据库的名称。	

用法

与通过 gdblogmode 或 gtape 实用程序更改数据库日志记录模式时不同，当您使用这个函数时，数据库保持可访问，通常不需要进行 0 级备份。在运行这个函数或函数失败之前，请确保没有其他会话是活动的。

使用 "a" 参数来更改数据库日志记录为符合 ANSI。创建或转换数据库为 ANSI 模式之后，您不可将它更改回其他日志记录模式的任何一种。

使用 "b" 参数来更改数据库日志记录为缓冲的，以便在事务信息写到逻辑日志之前，写到缓冲区。

使用 "n" 参数来更改数据库日志记录为无日志记录，以便不记录数据库事务日志。您必须在使用这个参数之前执行 0 级备份。

使用 "u" 参数来更改数据库日志记录为无缓冲的，以便在数据写到逻辑日志之前，不写到缓冲区。

示例

系列示例更改名为 employee 的数据库的日志记录模式为无缓冲的日志记录：

```
EXECUTE FUNCTION task("alter logmode","employee","u");
```

4.10 alter plog 参数：更改物理日志（SQL 管理

API）

随同 admin() 或 task() 函数，使用 alter plog 参数来更改物理日志的位置和大小。

语法

元素	描述	关键考虑
dbspace	物理日志的位置。	为物理日志分配的空间必须是连续的。
phys_log_size	物理日志的大小，以 KB 指定。	请参阅 admin() 和 task() 参数大小规范。

用法

要仅更改大小，请指定物理日志的当前 dbspace。

这个函数等同于 glogadmin -p 命令。

示例

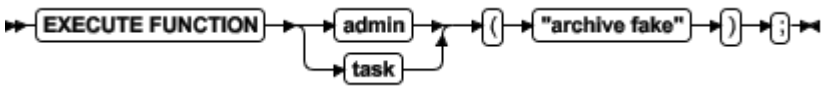
下列示例将物理日志移动到名为 phsdbs 的 dbspace：

```
EXECUTE FUNCTION task ("alter plog","physdbs","49 M");
```

4.11 archive fake 参数：执行无记录的备份（SQL 管理 API）

随同 admin() 或 task() 函数，使用 archive fake 参数来执行备份操作，克隆服务器中的数据而不创建一个可用于执行恢复的持续备份。

语法



用法

使用这个函数来填入“高可用性数据复制”对中的辅助服务器。

这个函数等同于运行带有 -F 选项的 gtape 命令。

示例

下列示例启动无记录的备份：

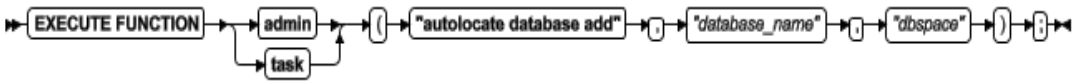
```
EXECUTE FUNCTION task("archive fake");
```

4.12 autolocate database add 参数：添加

dbspace 到 dbspace 列表（SQL 管理 API）

随同 admin() 或 task() 函数，使用 autolocate database add 参数来添加 dbspace 到可用的 dbspace 列表，用于指定数据库的表的自动定位和分片。

语法



元素	描述	关键考虑
database_name	数据库的名称	
dbspace	要添加到 dbspace 名称列表的 dbspace 的名称，在这些 dbsapce 中数据库服务器可自动地创建分片。	dbspace 必须存在。

用法

AUTOLOCATE 配置参数或会话环境变量必须设置为一个正整数。

可用 dbspace 的列表存储在 sysautolocate 系统目录表中。

示例

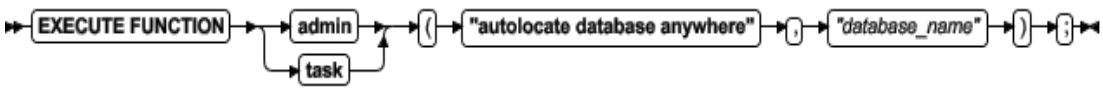
下列命令添加 dbspace dbspace9 到可用 dbspace 的列表，用于在 customer 数据库中的表的自动定位和分片。

```
EXECUTE FUNCTION task("autolocate database add", "customer", "dbspace9");
```

4.13 autolocate database anywhere 参数：添加所有 dbspace 到 dbspace 列表（SQL 管理 API）

随同 admin() 或 task() 函数，使用 autolocate database anywhere 参数来指定数据库服务器可使用任何非关键 dbspace，用于指定数据库的表的自动定位和分片。

语法



元素	描述	关键考虑
database_name	数据库的名称	不可是 tenant 数据库的名称。

用法

这个命令以一个所有可用 dbspace 的列表取代任何之前的 dbspace 列表。用于 tenant 数据库的 dbspace 不可用。可用 dbspace 的列表存储在 sysautolocate 系统目录表中。AUTOLOCATE 配置参数或会话环境变量必须设置为一正整数。

示例

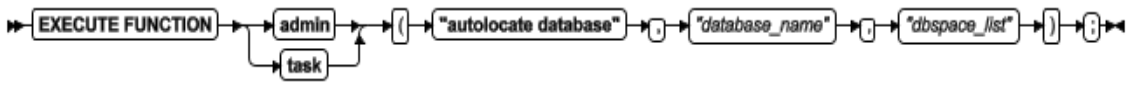
下列命令将所有非关键 dbspace 添加到可用 dbspace 的列表，用于 potential_cust 数据库中表的自动定位和分片：

```
EXECUTE FUNCTION task("autolocate database anywhere", "potential_cust");
```

4.14 autolocate database 参数：指定自动定位和分片的 dbspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 autolocate database 参数来为指定数据库的表的自动定位和分片指定可用 dbspace 的列表。

语法



元素	描述	关键考虑
database_name	数据库的名称	不可是 tenant 数据库的名称。
dbspace_list	以逗号分隔的 dbspace 名称的列表，数据库服务器可在这些 dbspace 中自动地创建分片。	这些 dbspace 必须存在。这些 dbspace 不可专用于 tenant 数据库。

用法

AUTOLOCATE 配置参数或会话环境变量必须设置为正整数。

缺省情况下，所有 dbspace 都可用。可用 dbspace 列表存储在 sysautolocate 系统目录表中。

示例

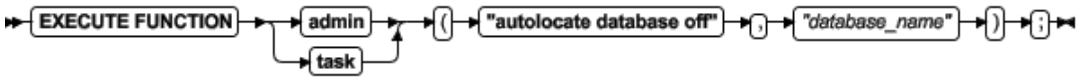
下列命令限定 customer 数据库中表的自动定位和分片的可用 dbspace 列表：

```
EXECUTE FUNCTION task("autolocate database", "customer",
 "dbspace1,dbspace2,dbspace4,dbspace8");
```

4.15 autolocate database off 参数：禁用数据库的自动分片（SQL 管理 API）

随同 admin() 或 task() 函数，使用 autolocate database off 参数来禁用指定数据库的表的自动定位和分片。

语法



元素	描述	关键考虑
database_name	数据库的名称	

用法

您在指定的数据库中创建的新表存储在与数据库相同的 dbspace 中，且不分片。随着表的增长，现有的自动地分片的表不分配新的分片。

示例

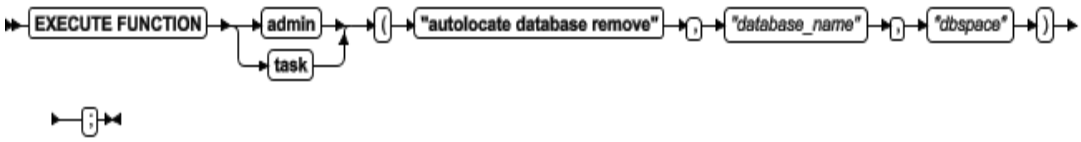
下列命令禁用在 customer_old 数据库中的表的自动定位和分片：

```
EXECUTE FUNCTION task("autolocate database off", "customer_old");
```

4.16 autolocate database remove 参数：从 dbspace 列表中移除 dbspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 autolocate database remove 参数来从可用 dbspace 列表移除 dbspace， 指定的数据库在这些可用 dbspace 内可自动地定位和分片表。

语法



元素	描述	关键考虑
database_name	数据库的名称	
dbspace	要从 dbspace 名称列表移除的 dbspace 的名称，数据库服务器在这些 dbspace 中可自动地创建分片。	该 dbspace 必须存在。

用法

AUTOLOCATE 配置参数或会话环境变量必须设置为正整数。

可用 dbspace 的列表存储在 sysautolocate 系统目录表中。

示例

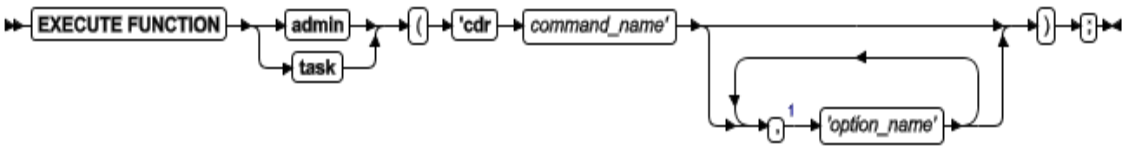
下列命令从 customer 的可用 dbspace 列表移除 dbspace1。

```
EXECUTE FUNCTION task("autolocate database remove", "customer", "dbspace1");
```

4.17 cdr 参数：管理 Enterprise Replication（SQL 管理 API）

随同 admin() 或 task() 函数，使用 cdr 参数来管理 Enterprise Replication。

语法



元素	描述	关键考虑
command_name	**cdr** 命令名。	您不可包括任何连字符、标志或包含 **cdr** 命令行实用程序要求的 *command_name* 的其他选项。
option_name	*command_name* 的 **cdr** 命令行选项的一个或多个元素。	这些元素必须以引号分隔。还包括（以正确的顺序）任何连字符、标志或 *command_name* 要求的 **cdr** 命令行选项的其他元素。您可使用缩写。

用法

使用这些函数产生的管理 Enterprise Replication 的效果与 cdr 命令行实用程序相同。该 SQL 管理 API 支持 cdr 命令用于管理 Enterprise Replication。不支持下列监视 Enterprise Replication 的命令：

- cdr list grid
- cdr list replicate
- cdr list replicateset
- cdr list server
- cdr list template
- cdr stats recv
- cdr stats rqm
- cdr -V
- cdr view

第一个参数必须仅包括 cdr 命令名，与 GBase 8s Enterprise Replication 指南 中 cdr 实用程序附录指定的名称完全一致，比如 cdr define server。命令名区分大小写且不支持缩写（比如 cdr sto replset 而不是 cdr stop replicateset）。在将这些参数传到 cdr 实用程序之前，该 SQL 管理 API 不执行任何验证。

第二个及任何后续参数包括命令选项。可在一个或最多六个参数中指定这些选项。

下列示例说明使用 SQL 管理 API 来定义 Enterprise Replication 服务器：

```
EXECUTE FUNCTION task ( 'cdr define server', '--connect=g_amsterdam
        --ats=/local0/er/ats --ris=/local0/er/ris --init g_amsterdam' );
```

下列示例展示这些选项可如何分为几个参数；上面的语句还可写为：

```
EXECUTE FUNCTION task( 'cdr define server',
        '--connect=g_amsterdam',
        '--ats=/local0/er/ats',
        '--ris=/local0/er/ris',
        '--init g_amsterdam' );
```

下列示例显示一个参数内的双引号字符串：

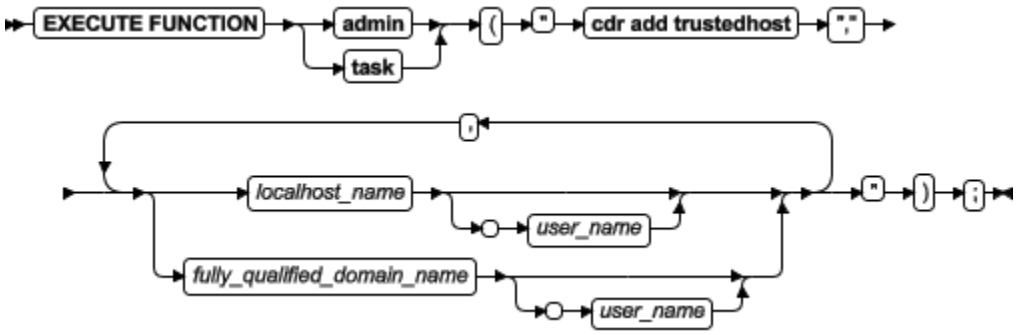
```
EXECUTE FUNCTION task('cdr change replicate',
'-d repl_1 -"db1@server1:antonio.table1" "db2@server2:carlo.table2"');
```

[1] 六项参数的最大值。

4.18 cdr add trustedhost 参数：添加可信任主机（SQL 管理 API）

随同 admin() 或 task() 函数，使用 cdr add trustedhost 参数来在高可用性集群或 Enterprise Replication 域中添加数据库服务器的可信任主机关系。对于参与高可用性集群或 Enterprise Replication 域的数据库，其主机必须罗列在其他高可用性或复制服务器的可信任主机文件中。

语法



元素	描述	关键考虑

元素	描述	关键考虑
localhost_name	数据库服务器的 localhost 名称。	
fully_qualified_domain_name	数据库服务器的完全域名称。	
user_name	在指定主机有数据库服务器实例授权的用户账户。	

用法

随同 admin() 或 task() 函数的 cdr add trustedhost 参数添加值到数据库服务器的 REMOTE_SERVER_CFG 配置参数指定的文件。如果数据库服务器是高可用性集群的一部分，则可信任主机信息还传播到其他集群服务器的可信任主机文件。可信任主机值指定 GBase_8s shard 集群中其他数据库服务器的 localhost 名或完全限定的域名称。为了添加的安全性，您可指定与特定主机相关的用户名。

如果未设置 REMOTE_SERVER_CFG 配置参数，且您运行带有 cdr add trustedhost 参数的 SQL 管理 API task() 或 admin() 函数，则数据库服务器执行下列活动：

- REMOTE_SERVER_CFG 配置参数设置为 authfile.DBSERVER。
- 在 $GBASEDBTDIR/etc 中创建 authfile.DBSERVER 文件。
- 将指定的可信任主机信息添加到 $GBASEDBTDIR/etc/authfile.DBSERVER。

如果您在高可用性集群中的服务器上运行带有 cdr add trustedhost 参数的 admin() 或 task() 函数，则将可信任主机信息添加到集群中所有数据库服务器的可信任主机文件。

您必须是“数据库服务器管理员”(DBSA)来运行带有 cdr add trustedhost 参数的 admin() 或 task() 函数。

要参阅可信任主机文件中的条目，请运行带有 cdr list trustedhost 参数的 admin() 或 task() 函数。

示例 1：添加可信任主机值到可信任主机文件

下列命令添加六个可信任主机值到数据库服务器 REMOTE_SERVER_CFG 配置参数指定的文件：

```
EXECUTE FUNCTION task("cdr add trustedhost","myhost1, myhost1.gbase.com,
        myhost2, myhost2.gbase.com, myhost3, myhost3.gbase.com");
```

该任务为三个数据库服务器指定 localhost 名称和完全限定的域名称。

示例 2：添加可信任主机和可信任用户值到可信任主机文件

下列命令添加四个可信任主机与用户的组合到数据库服务器 REMOTE_SERVER_CFG 配置参数指定的文件：

```
EXECUTE FUNCTION task("cdr add trustedhost", "myhost1 gbasedbt,
```

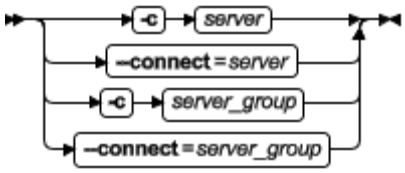
```
myhost1.gbase.com gbasedbt, myhost2 user_1, myhost2.gbase.com
user_1");
```

该任务为两个数据库服务器指定 localhost 名称、完全限定的域名称和用户名称。

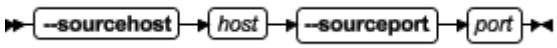
4.19 cdr autoconfig serv 参数：自动配置连接性和复制（SQL 管理 API）

随同 cdr autoconfig serv 参数的 admin() 或 task() 函数可自动配置高可用性集群中服务器的连接性，且可自动配置复制。

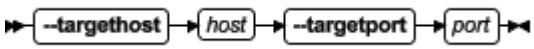
语法



Source options



Target options



元素	目的	约束
host	数据库服务器主机的名称	
port	用于通信的端口号	
server	要连接到的数据库服务器名	该名称必须是数据库服务器或服务器连接的名称。
server_group	包括要连接的数据库服务器的数据库服务器组名	该名称必须是现有的数据库服务器组名。

下表描述 cdr autoconfig serv 的选项。

长形	短形	含意
--sourcehost	-H	正在发送自动配置信息的数据库服务器的主机。如果未指定 --sourcehost 和 --sourceport，则将在其上运行函数的数据库服务器视为源数据库服务器。

长形	短形	含意
--sourceport	-P	正在发送自动配置信息的数据库服务器使用的端口。
--targethost	-h	正在接收自动配置信息的数据库服务器主机。
--targetport	-p	正在接收自动配置信息的数据库服务器使用的端口。

用法

运行带有 cdr autoconfig serv 参数的 admin() 或 task() 函数可自动配置高可用性集群或 Enterprise Replication 域中服务器的连接性，如果您正在添加数据库服务器到 Enterprise Replication 域，则还可自动配置复制。在您可运行带有 cdr autoconfig serv 参数的 admin() 或 task() 函数之前，CDR_AUTO_DISCOVER 配置参数必须在参与 Enterprise Replication 域的所有数据库上设置为 1。通过 cdr autoconfig serv 添加到 Enterprise Replication 域的新安装的数据库服务器必须有一个配置的存储池。

如果已为 Enterprise Replication 配置了源服务器，则函数执行下列活动：

1. 源服务器将其可信任主机文件传到目的服务器。
2. 目的服务器为自身和所有其他复制服务器添加条目到其 sqlhosts 文件。
3. 源服务器以目的服务器的条目更新其 sqlhost 文件。
4. 每一复制服务器以目的服务器的条目更新其 sqlhost 文件和可信任主机文件。
5. 目的数据库设置其 CDR_DBSPACE 配置参数并创建 Enterprise Replication 要求的 dbspace。
6. 目的服务器设置其 CDR_QDATA_SBSPACE 配置参数并创建 Enterprise Replication 要求的 sbspace。
7. 在目的服务器上创建中止事务 spooling（ATS）文件目录 $GBASEDBTDIR/tmp/ats_dbservername。
8. 在目的服务器上创建行信息 spooling（RIS）文件目录 $GBASEDBTDIR/tmp/ris_dbservername。
9. 启动到目的服务器的复制。

如果未为 Enterprise Replication 配置源服务器，则函数执行附加的活动：

1. 源服务器为自身添加条目到其 sqlhosts 文件。
2. 源服务器设置其 CDR_DBSPACE 配置参数并创建 Enterprise Replication 要求的 dbspace。

3. 源服务器设置其 CDR_QDATA_SBSPACE 配置参数并创建 Enterprise Replication 要求的 sbspace。
4. 在源服务器上创建中止事务 spooling（ATS）文件目录 $GBASEDBTDIR/tmp/ats_dbservername。
5. 在源服务器上创建行信息 spooling（RIS）文件目录 $GBASEDBTDIR/tmp/ris_dbservername。
6. 在开始目的服务器上的复制之前，开始源服务器上的复制。

对带有 cdr autoconfig serv 参数的 admin() 或 task() 函数使用下列约束：

- 所有复制服务器必须是活动的，否则函数失败。
- 如果您已经手工地配置了可信任主机信息，而不是通过运行带有 cdr add trustedhost 参数的 admin() 或 task() 函数配置的，则请不要运行带有 cdr autoconfig serv 参数的 admin() 或 task() 函数。
- 如果您的复制服务器有配置的安全端口，则请不要运行带有 cdr autoconfig serv 参数的 admin() 或 task() 函数。
- 带有 cdr autoconfig serv 参数的 admin() or task() 函数不复制 hosts.equiv 信息到可信任主机文件，该文件由 REMOTE_SERVER_CFG 配置参数设置。如果您必须从 hosts.equiv 文件添加信息到 REMOTE_SERVER_CFG 配置参数设置的可信任主机文件，请运行带有 cdr add trustedhost 参数的 admin() 或 task() 函数。

串行地配置数据库服务器。不支持并行配置。

您可如 cdr utility 命令那样运行这个函数。

示例 1：在本地服务器上配置 Enterprise Replication

对于这个示例，您有一个未为 Enterprise Replication 配置的本地数据库服务器：

在本地服务器上运行下列任务函数：

```
EXECUTE FUNCTION task('cdr autoconfig server');
```

该任务函数在本地服务器上配置 Enterprise Replication。

示例 2：通过使用源语法在两独立服务器之间配置连接性和 ER

对于这个示例，您有两数据库服务器：

为 Enterprise Replication 配置 host_1 上的 server_1

未为 Enterprise Replication 配置 host_2 上的 server_2

```
EXECUTE FUNCTION task('cdr autoconfig server', '--connect server_2
        --sourcehost host_1 --sourceport 9020');
```

该任务函数执行下列活动：

该命令连接到 server_2。

在 server_2 上定义 Enterprise Replication。

server_1 复制其数据到 server_2

示例 3：使用目的语法配置两独立服务器之间的连接性和 ER

下列命令

```
EXECUTE FUNCTION task('cdr autoconfig server', '--connect server_1
        --targethost host_2 --targetport 9030');
```

该任务函数执行下列活动：

1. 该命令连接到 server_1。
2. 在 server_2 上定义 Enterprise Replication。
3. server_1 复制其数据到 server_2

4.20 cdr list trustedhost 参数：罗列可信任主机（SQL 管理 API）

随同 admin() 或 task() 函数，使用 cdr list trustedhost 参数来罗列来自于数据库服务器的 REMOTE_SERVER_CFG 配置参数指定文件的可信任主机信息。

语法

用法

您必须是“数据库服务器管理员”（DBSA）才能运行这个函数。

示例

下列命令罗列来自于数据库服务器的可信任主机文件的可信任主机条目：

```
EXECUTE FUNCTION task("cdr list trustedhost");
```

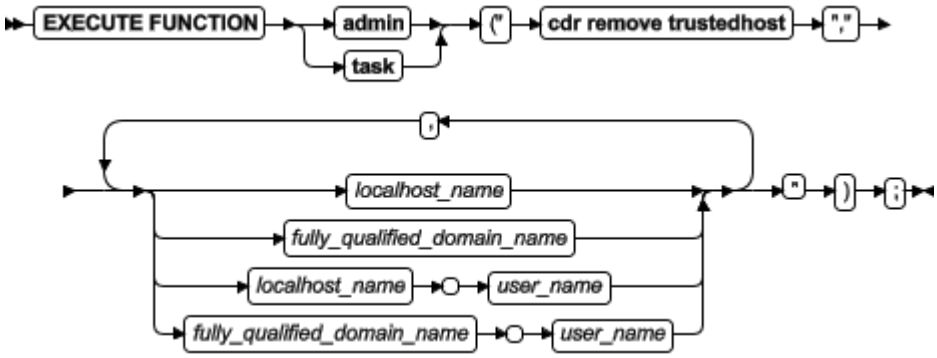
下列示例输出显示使用 cdr list trustedhost 参数可能的结果。

```
myhost1 user_1
myhost1.example.com user_1
myhost2 user_2
myhost2.example.com user_2
```

4.21 cdr remove trustedhost 参数：移除可信任主机（SQL 管理 API）

随同 admin() 或 task() 函数，使用 cdr remove trustedhost 参数来从数据库服务器的可信任主机文件移除条目。

语法



元素	描述	关键考虑
localhost_name	数据库服务器的 localhost 名称。	如果您在该命令中未指定 *user_name*，则移除包括指定主机名的所有条目。
fully_qualified_domain_name	数据库服务器的完全域名称。	如果您在该命令中未指定 *user_name*，则移除包括指定的完全限定的域名称的所有条目。
user_name	在指定主机有数据库服务器实例授权的用户账户。	

用法

cdr remove trustedhost 参数从数据库服务器 REMOTE_SERVER_CFG 配置参数指定的可信任主机文件移除可信任主机条目。对于参与高可用性集群或 Enterprise Replication 域的数据库，其主机必须罗列在其他高可用性或复制服务器的可信任主机文件中。当您在高可用性集群中的服务器上运行带有 cdr remove trustedhost 参数的 admin() 或 task() 函数时，从所有集群服务器的可信任主机文件移除可信任主机条目。

要参阅可信任主机文件中的条目，请运行带有 cdr list trustedhost 参数的 admin() 或 task() 函数。

您必须是“数据库服务器管理员”（DBSA）才能运行带有 cdr remove trustedhost 参数的 admin() 或 task() 函数。

示例 1：从可信任主机文件移除主机条目

下列命令从数据库服务器 REMOTE_SERVER_CFG 配置参数指定的可信任主机文件移除 localhost 名称值和完全限定的域名称值：

```
EXECUTE FUNCTION task("cdr remove trustedhost","myhost1, myhost1.gbase.com");
```

从数据库服务器的可信任主机文件移除 myhost1 和 myhost1.gbase.com 条目。

示例 2：从可信任主机文件移除主机和用户条目

下列命令从数据库服务器 REMOTE_SERVER_CFG 配置参数指定的可信任主机文件移除 localhost 名称值、完全限定的域名称值和用户值：

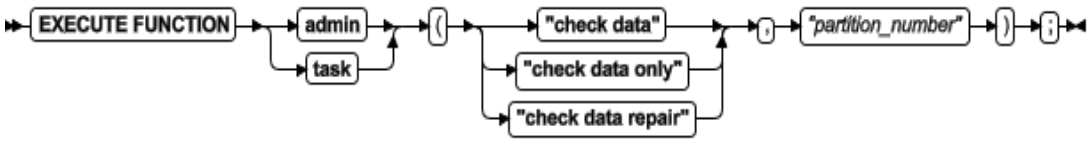
```
EXECUTE FUNCTION task("cdr remove trustedhost", "myhost2 john,myhost2.
gbase.cn john,myhost3 gbasedbt,myhost3.gbase.cn gbasedbt");
```

从数据库服务器的可信任主机文件移除用户 john 的 myhost2、用户 john 的 myhost2.gbase.cn、用户 gbasedbt 的 myhost3 和用户 gbasedbt 的 myhost3.gbase.cn。

4.22 check data 参数：检查数据一致性（SQL 管理 API）

随同 admin() 或 task() 函数，使用 check data 参数来检查或修理指定分区中所有页的一致性。

语法



元素	描述	关键考虑
partition_number	在其中检查数据的分区号。	找到 systables 系统目录表的 partnum 列中的分区号。

用法

使用 check data 参数来读除了 sbpage 之外的所有页，并检查每一页的一致性。这个参数等同于 gcheck -cD 命令。

使用 check data only 参数来读除了 blobpage 和 sbpage 之外的所有页，并检查每一页的一致性。这个参数等同于 gcheck -cd 命令。

使用 check data repair 参数来修理不一致的页。这个参数等同于 gcheck -cD -y 命令。

示例

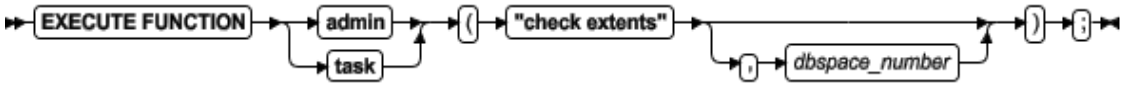
下列示例检查分区 1048611 中所有页的一致性：

```
EXECUTE FUNCTION task("check data","1048611");
```

4.23 check extents 参数：检查 extent 一致性（SQL 管理 API）

随同 admin() 或 task() 函数，使用 check extents 参数来验证磁盘上的 extent 与当前的控制信息是否相同。

语法



元素	描述	关键考虑
dbspace_number	要检查的 dbspace 号。	

用法

运行这个函数来检查每一空闲 chunk 列表和相应的空闲空间以及每一 tblspace extent。如果您未指定 dbspace 号，则检查所有 dbspace。该功能检查 dbspace、blobspace、智能大对象 extent 和用户数据以及 sbspace chunk 中的元数据信息。

这个函数等同于 gcheck -ce 命令。

示例

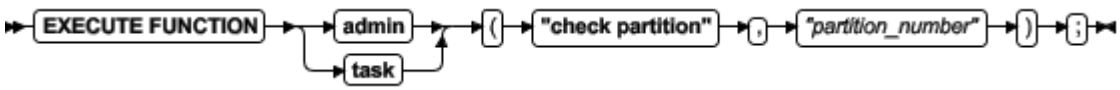
下列示例检查 2 号 dbspace 中的 extent：

```
EXECUTE FUNCTION task("check extents",2);
```

4.24 check partition 参数：检查分区一致性（SQL 管理 API）

随同 admin() 或 task() 函数，使用 check partition 参数来打印表或分片的 tblspace 信息。

语法



元素	描述	关键考虑
partition_number	您想要检查其一致性的分区号。	找到 **systables** 系统目录表的 **partnum** 列中的分区号。

用法

带有 check partition 参数的 task() 函数返回的信息等同于 gcheck -pt 命令的输出。输出包含通用信息，诸如最大行大小、键号、extent 号及大小、 分配的页及每 extent 使用的页、当前的序列值和创建表的日期。

admin() 函数返回一个整数，可用来找到 sysadmin 数据库中 command_history 表中的信息。

示例

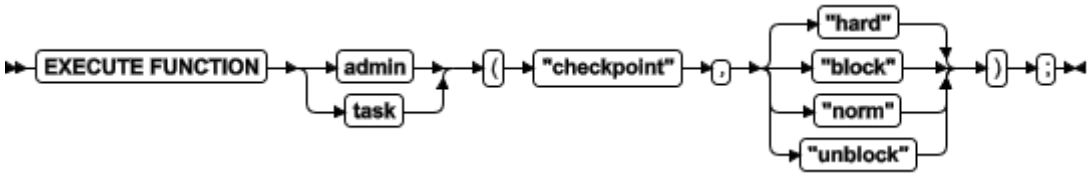
下列示例打印分区 1048611 的信息：

```
EXECUTE FUNCTION task("check partition","1048611");
```

4.25 checkpoint 参数：强制一检查点（SQL 管理 API）

随同 admin() 或 task() 函数，使用 checkpoint 参数来强制一检查点。

语法



用法

这个函数强制一检查点，经缓冲区清空到磁盘。如果逻辑日志中最近的检查点记录阻止释放逻辑日志文件（状态 U-B-L），则可使用这个函数来强制一检查点。

使用 block 参数来防止数据库服务器处理任何事物。在 GBase 8s 上使用这个选项执行外部备份。在数据库服务器阻塞时，用户不可访问它，除非在只读模式中。直到解除数据库服务器阻塞，事务才可完成。

使用 hard 参数来强制一阻塞检查点。这是缺省情况。

使用 norm 参数来强制一非阻塞检查点。

使用 unblock 参数来解除数据库服务器阻塞。当解除数据库服务器阻塞时，可恢复数据事务和正常的数据库服务器操作。请在 GBase 8s 上完成外部备份之后使用这个选项。

这个函数等同于 gadmin -c 命令。

示例

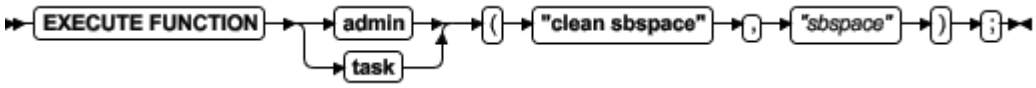
下列示例启用一阻塞检查点：

```
EXECUTE FUNCTION task("checkpoint","block");
```

4.26 clean sbspace 参数：释放未引用的智能大对象（SQL 管理 API）

随同 admin() 或 task() 函数，使用 clean sbspace 参数来从 sbspace 释放任何未引用的 BLOB 或 CLOB 对象。

语法



元素	描述	关键考虑
sbspace	要清理的 sbspace 的名称。	

用法

这个函数等同于 gspaces -cl 命令。

示例

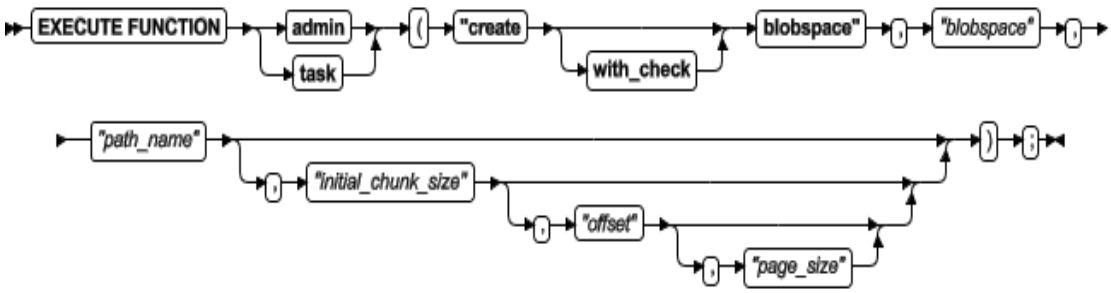
下列示例清理名为 sbsp1 的 sbspace：

```
EXECUTE FUNCTION task("clean sbspace","sbsp1");
```

4.27 create blobspace 参数：创建 blobspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 create blobspace 参数来创建 blobspace。

语法



元素	描述	关键考虑
blobspace	要创建的 blobspace 的名称。	
initial_chunk_size	新 blobspace 的初始 chunk 的大小，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。。
offset	达到新 blobspace 的初始 chunk 的磁盘分区内或设备内的偏移量，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。。
page_size	blobspace blobpage 大小。	在多个缺省 GBase 8s 页大小中，为操作系统指定 blobpage 的大小。

元素	描述	关键考虑
		要了解更多信息，请参阅 *GBase 8s 性能指南* 中对 blobpage 大小的考虑。
path_name	正在创建的 blobspace 的初始 chunk 的磁盘分区或设备。	

用法

使用 create with_check blobspace 参数来检查指定的路径名，如果路径不存在，则返回错误。

这个函数等同于 gspaces -c -b 命令。

示例

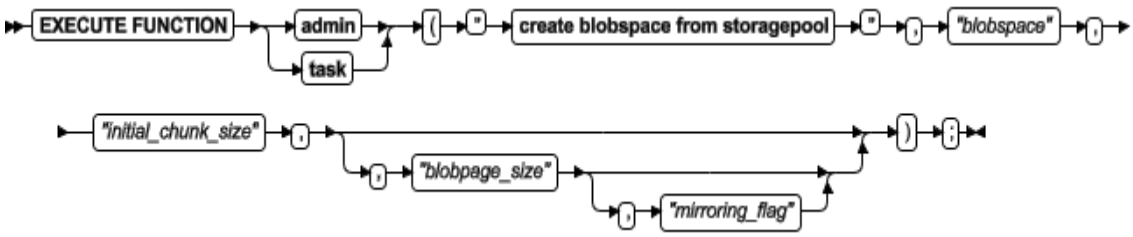
下列示例创建一 blobspace，大小为 20 MB，偏移量为 0 且 page_size 为 2。在 Windows™（4K 基本页大小）上，blob 页是 2 * 基本页大小 = 8 K。

```
EXECUTE FUNCTION task ("create with_check blobspace","blobs3",
"$GBASEDBTDIR/WORK/blobs3","20 M","0","2");
```

4. 28 create blobspace from storagepool 参数：从存储池创建 blobspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 create blobspace from storagepool 参数来从存储池中一个条目创建一 blobspace。

语法



元素	描述	关键考虑
blobspace	blobspace 的名称。	blobspace 名称必须是唯一的且不可超过 128 字节。名称以一个字母或下划线开头，且必须仅包含字母、数、下划线或 $ 字符。

元素	描述	关键考虑
blobpage_size	blobpage 大小，根据 *page_unit* 指定，每 blobpage 的磁盘页数。	页大小是可选的。然而，如果您为 mirroring 指定 1，则还必须指定页大小。
initial_chunk_size	新 blobspace 的初始 chunk 的大小，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。
mirroring_flag	二者之一： • 1 = 镜像 • 0 = 无镜像	镜像标志是可选的。

示例

下列命令创建名为 blobspace1 的镜像 blobspace。新 blobspace 的大小为 100 GB，100 页的 blobpage 大小。

```
EXECUTE FUNCTION task("create blobspace from storagepool", "blobspace1", "100 GB", "100", "1");
```

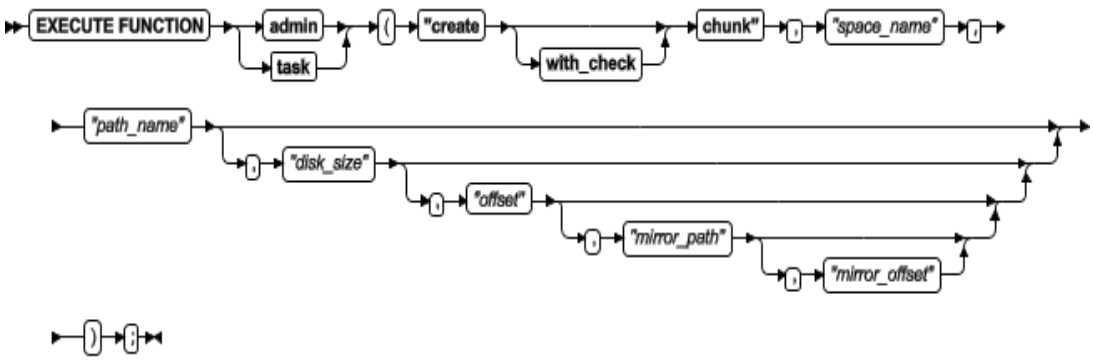
下列命令创建一名为 blobspace2 的无镜像 blobspace，采用缺省 blobpage 大小，因此未指定 blobpage 大小：

```
EXECUTE FUNCTION task("create blobspace from storagepool", "blobspace2", "5000");
```

4.29 create chunk 参数：创建 chunk（SQL 管理 API）

随同 admin() 或 task() 函数,使用 create chunk 参数来在 dbspace 中或在 blobspace 中创建 chunk。

语法



元素	描述	关键考虑
disk_size	要添加的磁盘空间的数量，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。
mirror_offset	镜像 chunk 的位置。	
mirror_path	到镜像 chunk 的路径。如果您正在添加 chunk 到镜像的存储空间，则还必须添加镜像 chunk。	
offset	新 chunk 的位置。	
path_name	添加的磁盘空间的路径。	
space_name	您正在添加磁盘空间的 dbspace、blobspace 或 sbspace 的名称。	

用法

使用 create with_check chunk 参数来检查指定的路径名，如果路径不存在，则返回错误。

这个函数等同于 gspaces -a 命令。

示例

下列示例添加 5 MB 裸磁盘空间的 chunk 到名为 dbspc3 的 dbspace ，偏移量为 5200 KB：

```
EXECUTE FUNCTION task("create chunk", "dbspc3","\\.\e:","5120","5200");
```

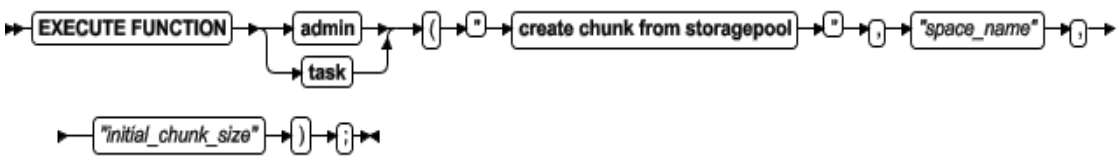
下列示例添加 10 MB 镜像 chunk 到名为 blobsp3 的 blobspace，主 chunk 和镜像 chunk 的偏移量都是 200 KB：

```
EXECUTE FUNCTION task("create with_check
chunk","blobsp3","/dev/raw_dev1","10240",
        "200","/dev/raw_dev2","200");
```

4.30 create chunk from storagepool 参数：从存储池创建 chunk（SQL 管理 API）

随同 admin() 或 task() 函数，使用 create chunk from storagepool 参数来手工地从存储池中的条目创建 chunk。

语法



元素	描述	关键考虑

元素	描述	关键考虑
space_name	您正在添加 chunk 到其上的存储空间的名称。	
initial_chunk_size	初始 chunk 的大小，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。

用法

您还可使用带有 modify space expand 参数的 SQL 管理 API 命令来手工地从存储池创建 chunk，并将该 chunk 添加到指定的存储空间。然而，如果该空间有可扩展的 chunk，则 GBase 8s 可能扩展一 chunk 而不创建一个新的。与 modify space expand 参数不一样，create chunk from storagepool 参数强制 GBase 8s 添加 chunk。

示例

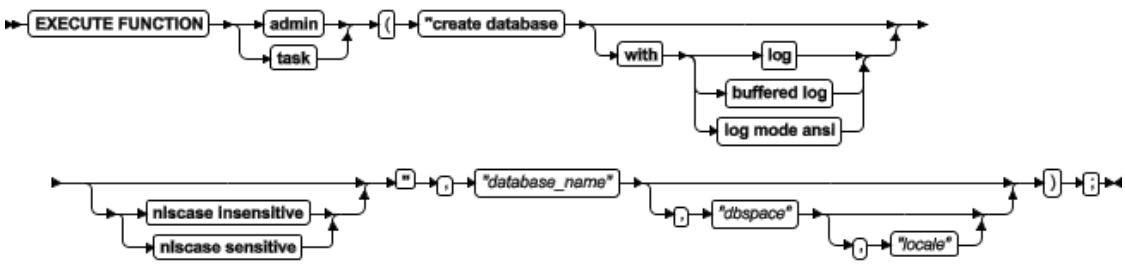
下列命令添加 chunk 到名为 logdbs 的 dbspace。新 chunk 的大小为 200 MB。

```
EXECUTE FUNCTION task("create chunk from storagepool", "logdbs", "200 MB");
```

4.31 create database 参数：创建数据库（SQL 管理 API）

随同 admin() 或 task() 函数，使用 create database 参数来创建数据库。

语法



元素	描述	关键考虑
database_name	数据库的名称。	
dbspace	为这个数据库存储数据的 dbspace 的名称。缺省是 root dbspace。	该 dbspace 必须在数据库服务器上已存在。
locale	与数据库相关的语言环境。	*locale* 的值与 DB_LOCALE 环境变量的值相同。 如果您忽略这个特性， 则以 DB_LOCALE 环境变量的值设置该语言环境。缺省语言环

元素	描述	关键考虑
		境为 US English。

用法

这个函数等同于 CREATE DATABASE 语句。

您不可使用这个函数来创建 tenant 数据库。请您以 tenant create 参数创建 tenant 数据库。

示例

下列示例创建名为 demodbs 的数据库，带有无缓存的日志记录：

```
EXECUTE FUNCTION task("create database with log","demodbs");
```

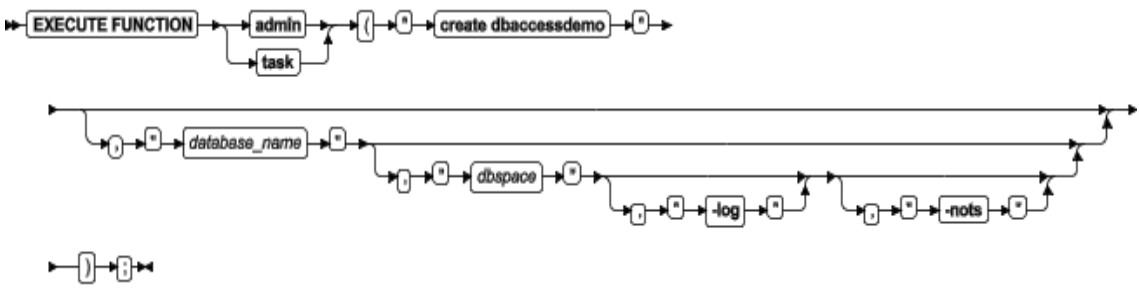
下列示例在名为 dataspace1 的 dbspace 中创建与大小写无关的名为 demodbs2 的数据库，带有符合 ANSI 的日志记录：

```
EXECUTE FUNCTION task("create database with log mode ansi nlscase insensitive",
                      "demodbs2","dataspace1");
```

4.32 create dbaccessdemo 参数：创建演示数据库（SQL 管理 API）

随同 admin() 或 task() 函数，使用 create dbaccessdemo 参数来创建 stores_demo 演示数据库。

语法



元素	描述	关键考虑
database_name	要创建的数据库名称。	缺省数据库名是 stores_demo。
dbspace	要在其中创建数据库的 dbspace 名称。	缺省 dbspace 是 root dbspace。

用法

运行这个函数来创建 stores_demo 数据库。

使用 -log 选项来启用演示数据库的事务日志记录。

使用 -nots 选项来防止在演示数据库中创建带时间序列数据的表。

示例

下列命令在 root dbspace 中创建 stores_demo 数据库：

```
EXECUTE FUNCTION task("create dbaccessdemo");
```

下列命令在名为 dbs1 的 dbspace 中创建名为 demo2 的演示数据库。：

```
EXECUTE FUNCTION task("create dbaccessdemo","demo2","dbs1");
```

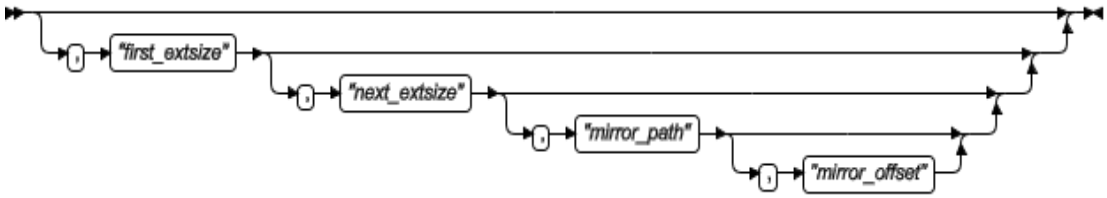
下列命令在名为 dbs1 的 dbspace 中创建带有事务日志记录的 stores_demo 数据库：

```
EXECUTE FUNCTION task("create dbaccessdemo","stores_demo","dbs1","-log");
```

4. 33 create dbspace 参数：创建 dbspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 create dbspace 参数来创建 dbspace。

语法



元素	描述	关键考虑
dbspace	要创建的 dbspace 名。	
first_extsize	tblspace **tblspace** 的第一个 extent 的大小，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。
initial_chunk_size	新 dbspace 的初始 chunk 的大小，以 KB 为单位。该大小取整到页大小的倍数。	请参阅 admin() 和 task() 参数大小规范。
mirror_offset	镜像 chunk 的偏移量，以 KB 为单位。	
mirror_path	对 dbspace 的初始 chunk 镜像的那个 chunk 的路径名。	
next_extsize	tblspace **tblspace** 中下一个 extent 的大小，以 KB 为	请参阅 admin() 和 task() 参数大小规范。

元素	描述	关键考虑
	单位。	
offset	磁盘分区内或设备内达到新 dbspace 的初始 chunk 的偏移量，以 KB 为单位。	
page_size	新 dbspace 的非缺省页大小，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。
path_name	正在创建的 dbspace 的初始 chunk 的磁盘分区或设备。	有效的页大小依赖于计算机的缺省页大小： ● 2 KB 缺省页大小：2、4、6、8、10、12 或 16 KB ● 4 KB 缺省页大小：4、8、12 或 16 KB

用法

使用 create with_check dbspace 参数来检查指定的路径名，如果路径不存在，则返回错误。

这个函数等同于 gspaces -c -d 命令。

示例

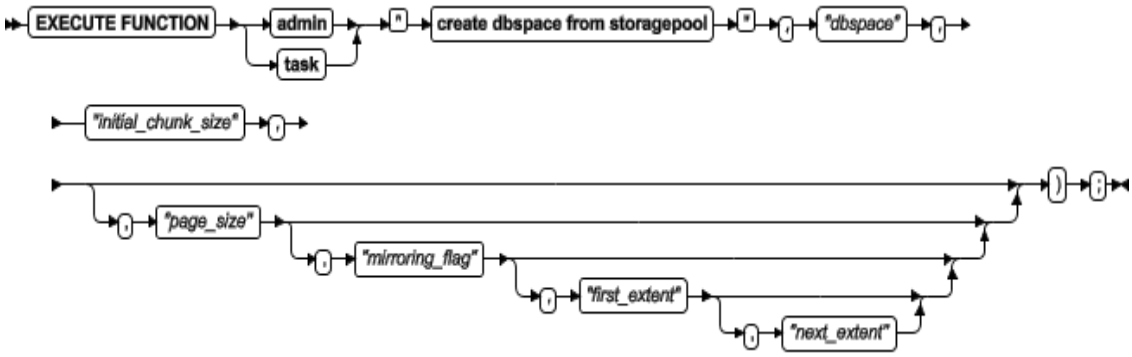
下列示例创建大小为 20 MB、偏移量为 0 的 dbspace。

```
EXECUTE FUNCTION task ("create dbspace", "dbspace3",
"$GBASEDBTDIR/WORK/dbspace3", "20 M", "0");
```

4.34 create dbspace from storagepool 参数：从存储池创建 dbspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 create dbspace from storagepool 参数来从存储池中的一个条目创建永久 dbspace。

语法



元素	描述	关键考虑
dbspace	要创建的 dbspace 名。	在 dbspace 名称中必须是唯一的，且不可超过 128 字节。名称必须以一个字母或下划线开头，且仅可包括字母、数字、下环线（_）符或 $ 字符。
first_extent	tblspace **tblspace** 的第一个 extent 的大小，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。
initial_chunk_size	新 dbspace 的初始 chunk 大小。	请参阅 admin() 和 task() 参数大小规范。
mirroring_flag	二者之一： • 1 = 镜像 • 0 = 无镜像	镜像标志是可选的。如果未指定，则缺省是无镜像的 dbspace。
next_extent	tblspace **tblspace** 中下一个 extent 的大小，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。
page_size	新 dbspace 的非缺省页大小，以 KB 为单位。	页大小是可选的。然而，如果您指定 mirroring 为 1，则还必须指定页大小。有效的页大小依赖于计算机的缺省页大小： ● 2 KB 缺省页大小：2、4、6、8、10、12 或 16 KB ● 4 KB 缺省页大小：4、8、12 或 16 KB

要了解创建临时 dbspace 的 admin() 或 task() 语法，请参阅 create tempdbspace 参数：创建临时 dbspace（SQL 管理 API）。

示例

下列命令创建名为 dbspace3 的镜像的 dbspace。新 dbspace 的大小为 1 GB，页大小为 6 KB，tblspace 第一个 extent 大小为 200 KB，且下一个 extent 大小为 400 KB。

```
EXECUTE FUNCTION task("create dbspace from storagepool",
 "dbspace3",  "1 GB", "6", "1", "200", "400");
```

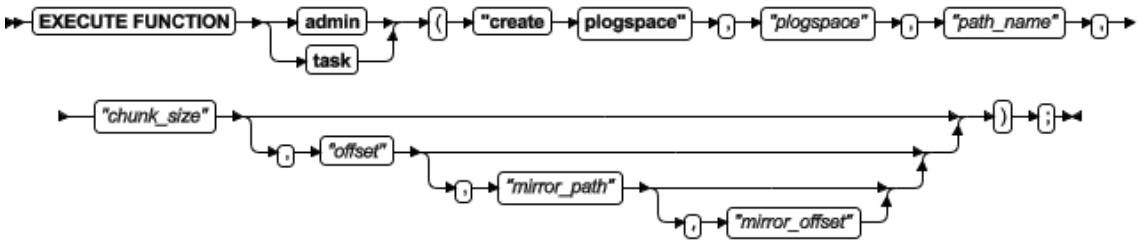
下列命令创建名为 dbspace8 的无镜像的 dbspace。新 dbspace 的大小为 50 MB。因为未指定页大小，所以新 dbspace 为缺省页大小。

```
EXECUTE FUNCTION task("create dbspace from storagepool",
 "dbspace8", "50000");
```

4.35 create plogspace 参数：创建 plogspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 create plogspace 参数来创建在其中存储物理日志的 plogspace。

语法



元素	描述	关键考虑
chunk_size	新 plogspace 的 chunk 大小，以 KB 为单位。该大小取整到页大小的倍数。	请参阅 admin() 和 task() 参数大小规范。
mirror_offset	镜像 chunk 的偏移量，以 KB 为单位。	无符号整数。该大小必须等于或大于 1000 KB 且是页大小的倍数。起始偏移量加上 chunk 大小不可超过最大 chunk 大小。 最大的 chunk 大小为 2 或 4 TB，依赖于平台。
mirror_path	到镜像 plogspace 的 chunk 的那个 chunk 的路径名。	如果镜像 plogspace，则 plogspace chunk 不可为可扩展的。
offset	磁盘分区内或设备内，达到新	无符号整数。该大小必须等于或大于 1000 KB 且是页大小的倍数。起始偏移量加上 chunk

元素	描述	关键考虑
	plogspace 的 chunk 的偏移量，以 KB 为单位。	大小不可超过最大的 chunk 大小。 最大的 chunk 大小为 2 或 4 TB，依赖于平台。
path_name	正在创建的 plogspace 的 chunk 磁盘分区或设备。	该 chunk 必须是现存的无缓冲的设备或缓冲的文件。当您指定路径名时，您可使用完全路径名或相对路径名。然而，如果您使用相对路径名，则该路径名必须与您初始化数据库服务器时的当前目录有关联。 UNIX™ 示例（无缓冲的设备）： /dev/rdsk/c0t3d0s4 UNIX 示例（缓冲的设备）： /ix/ifmx/db1chunk Windows™ 示例： c:\Ifmxdata\ol_icecream\mychunk1.dat
plogspace	要创建的 plogspace 名。	plogspace 名称必须是唯一的且不可超过 128 字节。名称必须以一个字母或下划线开头且必须仅包含字母、数、下划线或 $ 字符。 语法必须符合 Identifier 段。要了解更多信息，请参阅《GBase 8s SQL 指南：语法》。

用法

这个函数等同于 gspaces -c -P 命令。

一个示例可仅有一个 plogspace。如果存在 plogspace，则当您创建新 plogspace时，将物理日志移到新的空间，且删除旧的 plogspace。

物理日志必须存储在单个 chunk 上。缺省情况下，该 chunk 标记为可扩展，以便如果有必要扩展物理内存，则数据库服务器可扩展该 plogspace。如果您镜像该 plogspace，则不可扩展该空间，因为镜像 chunk 不可为可扩展的。

示例

下列示例创建大小为 30000 KB、偏移量为 0 的 plogspace。

```
EXECUTE FUNCTION task ("create plogspace", "plogdbs",
"/dev/chk1", 30000, 0);
```

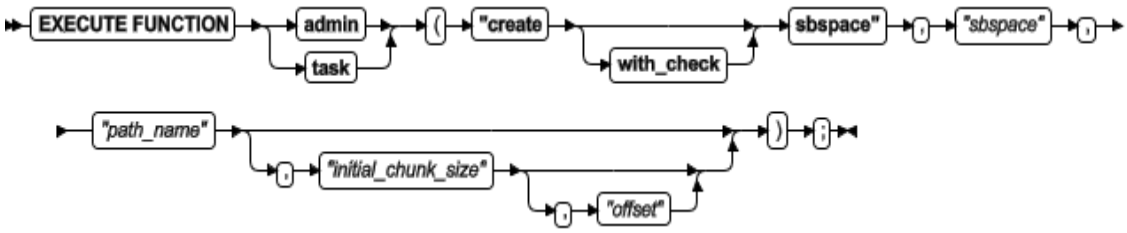
下列示例创建大小为 30000 KB、偏移量为 0 的镜像 plogspace。

```
EXECUTE FUNCTION task ("create plogspace", "plogdbs",
"/dev/chk1", 30000, 0, "/dev/mchk1", 0);
```

4.36 create sbspace 参数：创建 sbspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 create sbspace 参数来创建 sbspace。

语法



元素	描述	关键考虑
initial_chunk_size	新 sbspace 的初始 chunk 的大小，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。
offset	磁盘分区内或设备内达到新 sbspace 的初始 chunk 的偏移量，以 KB 为单位。	
path_name	该 sbspace 的初始 chunk 的磁盘分区或无缓冲的设备。	
sbspace	要创建的 sbspace 名。	

用法

使用 create with_check sbspace 参数来检查指定的路径名，如果路径不存在，则返回错误。

这个函数等同于 gspaces -c -S 命令。

示例

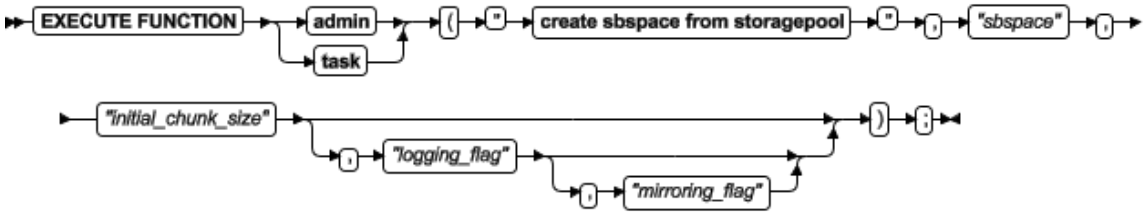
下列示例创建大小为 20 MB、偏移量为 0 的新 sbspace：

```
EXECUTE FUNCTION task ("create sbspace","sbspace2",
"$GBASEDBTDIR/WORK/sbspace2","20 M","0");
```

4.37 create sbspace from storagepool 参数：从存储池创建 sbspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 create sbspace from storagepool 参数来从存储池的条目中创建 sbspace。

语法



元素	描述	关键考虑
sbspace	sbspace 名。	sbspace 名称必须是唯一的且不可超过 128 字节。名称必须以一个字母或下划线开头，且必须仅包含字母、数、下划线或 $ 字符。
initial_chunk_size	新 sbspace 的初始 chunk 的大小，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。
logging_flag	二者之一： • 1 = 日志记录 • 0 = 无日志记录	日志记录标志是可选的。然而，如果您指定 mirroring 为 1，则还必须指定日志记录标志。
mirroring_flag	二者之一： • 1 = 镜像 • 0 = 无镜像	镜像标志是可选的。

示例

下列命令创建镜像的且记录日志的名为 sbspace1 的 sbspace。新 sbspace 的大小为 240 MB。

```
EXECUTE FUNCTION task("create sbspace from storagepool", "sbspace1",
 "240 MB", "1", "1");
```

下列命令创建无镜像的且不记录日志的名为 sbspace2 的 sbspace。这个 sbspace 的大小为 5 GB。

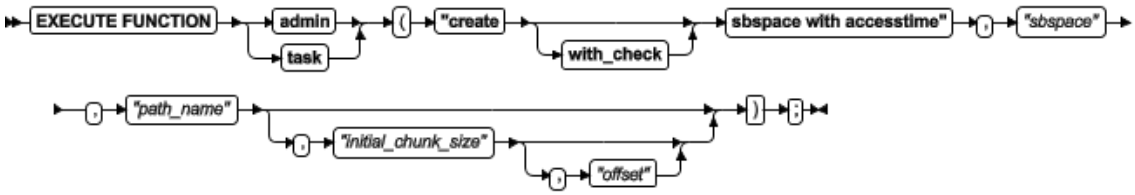
```
EXECUTE FUNCTION task("create sbspace from storagepool", "sbspace2", "5 GB");
```

4. 38 create sbspace with accesstime 参数：创建

跟踪访问时间的 sbspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 create sbspace with accesstime 参数来创建 sbspace，对存储在该 sbspace 中的所有智能大对象跟踪访问的时间。

语法



元素	描述	关键考虑
initial_chunk_size	新 sbspace 的初始 chunk 的大小，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。
offset	磁盘分区内或设备内达到新 sbspace 的初始 chunk 的偏移量，以 KB 为单位。	
path_name	该 sbspace 的初始 chunk 的磁盘分区或无缓冲的设备。	
sbspace	要创建的 sbspace 名。	

用法

使用 create with_check sbspace 参数来检查指定的路径名，如果路径不存在，则返回错误。

这个函数等同于 gspaces -c -S 命令创建 sbspace，且随同 admin() 或 task() 函数，使用 set sbspace accesstime 参数来启动对存储在该 sbspace 中的所有智能大对象跟踪访问的时间。

示例

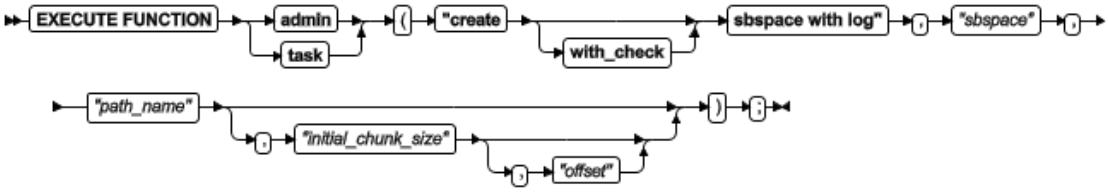
下列示例创建跟踪访问时间的新 sbspace。这个 sbspace 的大小为 20 MB，偏移量为 0：

```
EXECUTE FUNCTION task ("create sbspace with accesstime","sbspace4",
"$GBASEDBTDIR/WORK/sbspace4","20 M","0");
```

4.39 create sbspace with log 参数：创建带有事务日志记录的 sbspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 create sbspace with log 参数来创建启用事务日志记录的 sbspace。

语法



元素	描述	关键考虑
initial_chunk_size	新 sbspace 的初始 chunk 的大小，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。
offset	磁盘分区内或设备内达到新 sbspace 的初始 chunk 的偏移量，以 KB 为单位。	
path_name	该 sbspace 的初始 chunk 的磁盘分区或无缓冲的设备。	
sbspace	要创建的 sbspace 名。	

用法

使用 create with_check sbspace 参数来检查指定的路径名，如果路径不存在，则返回错误。

这个函数等同于 gspaces -c -S 命令来创建开启日志记录的 sbspace。

示例

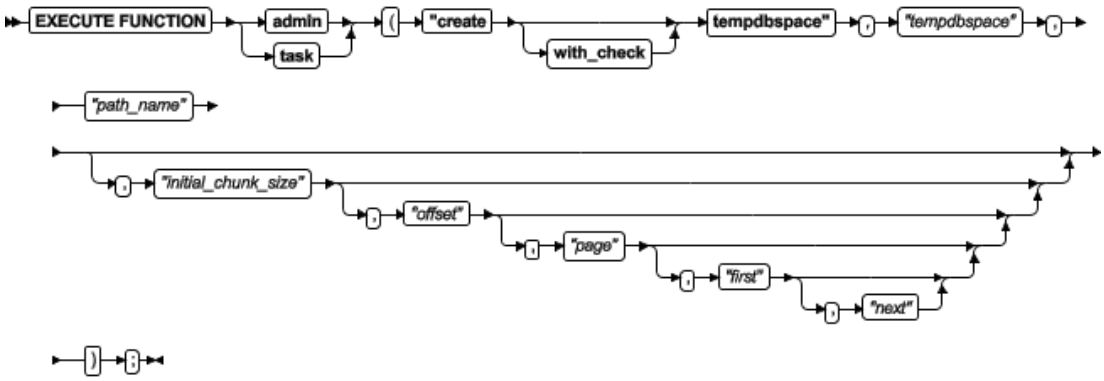
下列示例创建开启事务日志记录的新 sbspace。该 sbspace 的大小为 20 MB，偏移量为 0：

```
EXECUTE FUNCTION task ("create sbspace with log","sbspace2",
"$GBASEDBTDIR/WORK/sbspace2","20 M","0");
```

4. 40 create tempdbspace 参数：创建临时 dbspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 create tempdbspace 参数来创建临时 dbspace。

语法



元素	描述	关键考虑
first	tblspace tblspace 的第一个 extent 的大小，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。
initial_chunk_size	新的临时 dbspace 的初始 chunk 的大小，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。
next	tblspace tblspace 的下一个 extent 的大小，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。
offset	磁盘分区内或设备内，达到新的临时 dbspace 的初始 chunk 的偏移量，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。
page	新的临时 dbspace 的非缺省页大小，以 KB 为单位。	有效的页大小依赖于计算机的缺省页大小： ● 2 KB 缺省页大小：2、4、6、8、10、12 或 16 KB ● 4 KB 缺省页大小：4、8、12 或 16 KB
path_name	到您正在创建的临时 dbspace 的初始 chunk 的磁盘分区或设备的路径。	
tempdbspace	要创建的临时 dbspace 名。	不可超过 128 字节。名称必须以一个字母或下划线开头，且仅可包括字母、数字、下划线（_）符号或 $ 字符。

用法

使用 create with_check tempdbspace 参数来检查指定的路径名，如果路径不存在，则返回错误。

这个函数等同于 gspaces -c -d -t 命令。

示例

下列示例创建大小为 20 MB、偏移量为 0 的临时 dbspace：

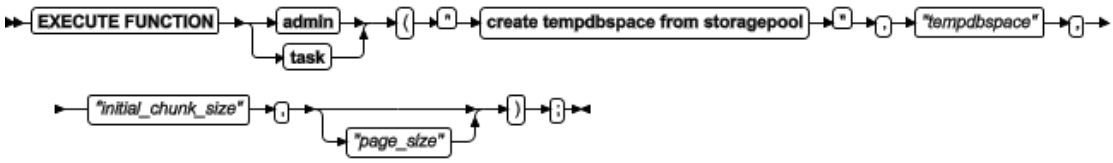
```
EXECUTE FUNCTION task("create tempdbspace","tempdbspace3",
"$GBASEDBTDIR/WORK/tempdbspace3","20 M","0");
```

要了解 admin() 或 task() 语法来从存储池创建永久的 dbspace，请参阅 create dbspace from storagepool 参数：从存储池创建 dbspace（SQL 管理 API）。

4.41 create tempdbspace from storagepool 参数：从存储池创建临时 dbspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 create tempdbspace from storagepool 参数来从存储池中一条目创建临时 dbspace。

语法



元素	描述	关键考虑
initial_chunk_size	新的临时 dbspace 的初始 chunk 大小，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。
page_size	新的临时 dbspace 的非缺省页大小，以 KB 为单位。	页大小是可选的。
tempdbspace	临时 dbspace 名。	

示例

下列命令创建名为 tempdbspace1 的临时 dbspace。该新 dbspace 的大小为 1 GB，页大小为 12 KB。

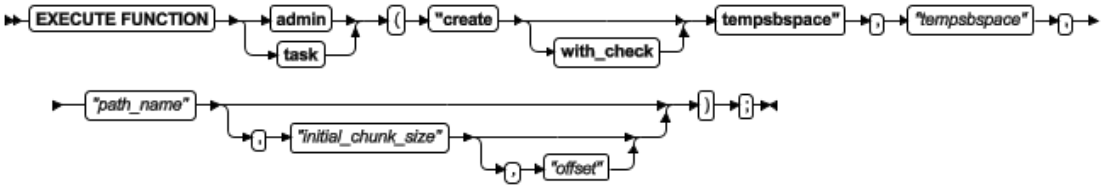
```
EXECUTE FUNCTION task("create tempdbspace from storagepool", "tempdbspace1",
 "1 GB", "12");
```

4.42 create tempsbspace 参数：创建临时

sbspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 create sbspace 参数来创建 sbspace。

语法



元素	描述	关键考虑
initial_chunk_size	新的临时 sbspace 的初始 chunk 大小，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。
offset	磁盘分区内或设备内，达到新的临时 sbspace 的初始 chunk 的偏移量，以 KB 为单位。	
path_name	临时 sbspace 的初始 chunk 的磁盘分区或无缓冲的设备。	
tempsbspace	要创建的临时 sbspace 名。	

用法

使用 create with_check sbspace 参数来检查指定的路径名，如果路径不存在，则返回错误。

这个函数等同于带有 -t 选项的 gspaces -c -S 命令，创建临时 sbspace。

示例

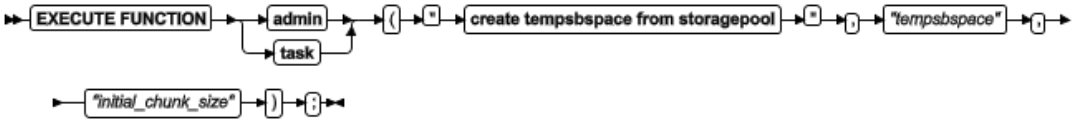
下列示例创建大小为 20 MB、偏移量为 0 的临时 sbspace：

```
EXECUTE FUNCTION task ("create tempsbspace","tempsbspace3",
"$ GBASEDBTDIR/WORK/tempsbspace3","20 M","0");
```

4. 43 create tempsbspace from storagepool 参数：从存储池创建临时 sbspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 create tempsbspace from storagepool 参数来从存储池中的条目创建临时 sbspace。

语法



元素	描述	关键考虑
initial_chunk_size	新 sbspace 的初始 chunk 大小，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。
tempsbspace	临时 sbspace 名。	临时 sbspace 名称必须是唯一的且不可超过 128 字节。名称必须以一个字母或下划线开头且必须仅包含字母、数、下划线或 $ 字符。

示例

下列命令创建名为 tempsbspace5 的临时 sbspace。临时 sbspace 的大小为 240 MB。

```
EXECUTE FUNCTION task("create tempsbspace from storagepool",
 "tempsbspace5", "240 MB");
```

4.44 defragment 参数：动态地对分区 extent 取消分片（SQL 管理 API）

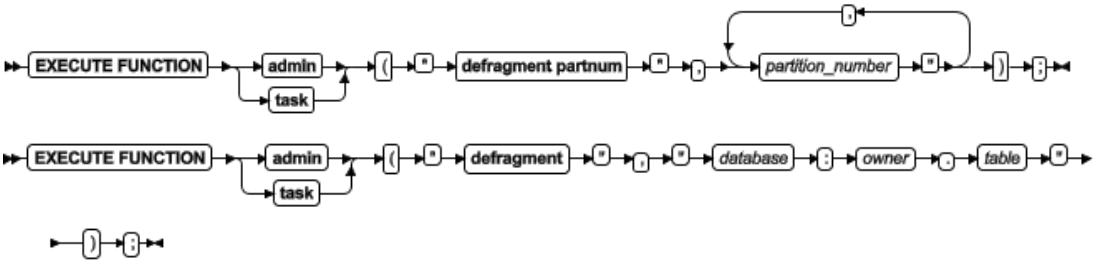
随同 admin() 或 task() 函数，使用 defragment 参数来对表或索引取消分片以合并不相邻的 extent。

对表取消分片使得数据行离得更近，以避免分区标题页移除问题，并可提高性能。

在对分区取消分片之前，您应回顾对分区取消分段。

语法

使用下列语法，您或可指定 defragment 参数，或可指定 defragment partnum 参数：



元素	描述	关键考虑
database	包括您想要取消分片的表或索引的数据库名。	

元素	描述	关键考虑
owner	表所有者的用户 ID。	
table	要取消分片的表名。	
partition_number	要取消分片的一个或多个分区号。	要指定多于一个分区，请使用逗号分隔的分区号列表。

用法

使用 defragment 参数来对特定的表取消分片。使用 defragment partnum 参数来对一个或多个磁盘分区取消分片。

关于取消分片的信息存储在共享内存中。使用 gcheck -pt 和 -pT：显示表或分片的 tblspaces 命令来显示关于特定的表或分片的 extent 数目信息。使用 gstat -g defragment 命令：打印磁盘碎片整理的分区 extent 。

如果取消分片需要减少 extent 的数目，减少最少 1 extent，则需要返回 0（成功），即使在分区中有许多 extent。

如果分区有单个 extent，则取消分片需要返回 0 来表明需要成功，即使无 extent 合并。

示例

要对 stores_demo 数据库中的 customer 表取消分片，请使用下列函数之一：

```
EXECUTE FUNCTION task("defragment","stores_demo:gbasedbt.customer");
EXECUTE FUNCTION admin("defragment","stores_demo:gbasedbt.customer");
```

要对索引取消分片，您必须指定该索引的分区号，如这两个函数示例所示：

```
EXECUTE FUNCTION task("defragment partnum","2097154");
EXECUTE FUNCTION admin("defragment partnum","2097154");
```

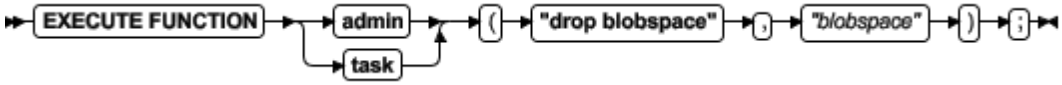
要对分区列表取消分片，请使用下列函数之一：

```
EXECUTE FUNCTION task("defragment partnum", "16777217,28477346");
EXECUTE FUNCTION admin("defragment partnum", "16777217,28477346");
```

4.45 drop blobspace 参数：删除 blobspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 drop blobspace 参数来删除指定的 blobspace。

语法



元素	描述	关键考虑

元素	描述	关键考虑
blobspace	要删除的 blobspace 名。	必须是现有的 blobspace。 在您删除 blobspace 之前，请删除包括引用该 blobspace 的 TEXT 或 BYTE 列的所有表。

用法

这个函数等同于 gspaces -d 命令。

示例

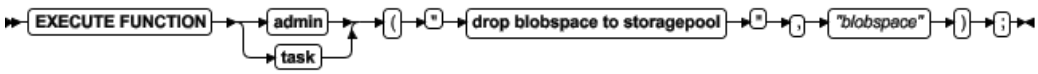
下列示例删除名为 blobspace3 的 blobspace：

```
EXECUTE FUNCTION task("drop blobspace","blobspace3");
```

4.46 drop blobspace to storagepool 参数：从空 blobspace 归还空间到存储池（SQL 管理 API）

随同 admin() 或 task() 函数，使用 drop blobspace to storagepool 参数来从空 blobspace 归还空间到存储池。

语法



元素	描述	关键考虑
blobspace	空 blobspace 的名称。	

示例

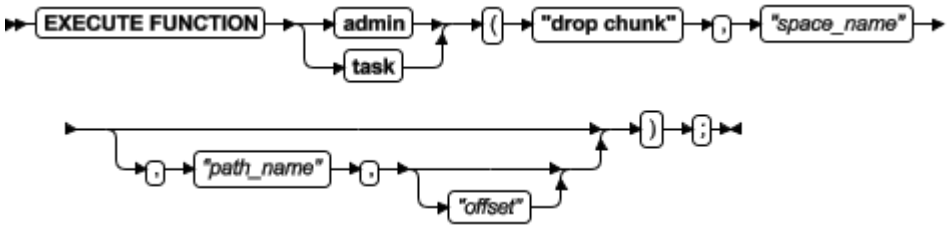
下列命令删除名为 blob2 的空 blobspace 并将所有释放的空间添加到存储池。

```
EXECUTE FUNCTION task("drop blobspace to storagepool", "blob2");
```

4.47 drop chunk 参数：删除 chunk（SQL 管理 API）

随同 admin() 或 task() 函数，使用 drop chunk 参数来从 dbspace、blobspace 或 sbspace 删除指定的 chunk。

语法



元素	描述	关键考虑
offset	磁盘分区内或无缓冲的设备内，达到您正在删除的 dbspace、blobspace 或 sbspace 的初始 chunk 的偏移量，以 KB 为单位。	无符号整数的初始偏移量必须等于或大于 0。起始偏移量加上 chunk 大小不可超过最大的 chunk 大小。最大的偏移量为 4 TB。 还请参阅 admin() 和 task() 参数大小规范。
path_name	您正在删除的 dbspace、blobspace 或 sbspace 的初始 chunk 的磁盘分区或无缓冲的设备。	该 chunk 必须是现有的无缓冲的设备或缓冲的文件。当您指定路径名时，你可使用完全路径名或相对路径名。然而，如果您使用相对路径名，则其必须为相对于当您初始化数据库服务器时的当前目录。
space_name	要从中删除 chunk 的 dbspace、sbspace 或 blobspace 名。	当数据库服务器为 online 或 quiescent 时，您可从 dbspace、临时 dbspace 或 sbspace 删除 chunk。 仅当数据库服务器处于 quiescent 模式中时，您可从 blobspace 删除 chunk。

用法

这个函数等同于 gspaces -d 命令。

示例

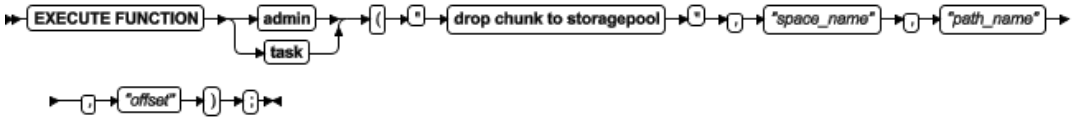
下列示例从名为 dbspc3 的 dbspace 在偏移量 5200 KB 处删除 chunk：

```
EXECUTE FUNCTION task("drop chunk", "dbspc3","\\.\e:","5200");
```

4. 48 drop chunk to storagepool 参数：从空 chunk 归还空间到存储池（SQL 管理 API）

随同 admin() 或 task() 函数，使用 drop chunk to storagepool 参数来从空 chunk 归还空间到存储池。

语法



元素	描述	关键考虑
space_name	chunk 所在其中的存储空间的名称。	
path_name	chunk 的路径。	
offset	chunk 的偏移量，以 KB 为单位。	

示例

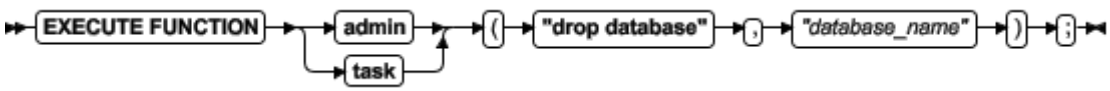
下列命令在名为 bigdbs 的 dbspace 中删除空 chunk，并将所有释放的空间添加到存储池。

```
EXECUTE FUNCTION task("drop chunk to storagepool", "bigdbs", "/dev/rawdisk23",
 "100 KB");
```

4.49 drop database 参数：删除数据库（SQL 管理 API）

随同 admin() 或 task() 函数，使用 drop database 参数来删除数据库。

语法



元素	描述	关键考虑
database_name	数据库名。	

用法

这个函数等同于 DROP DATABASE 语句。这个函数删除整个数据库，包括所有系统目录表、对象和数据。

示例

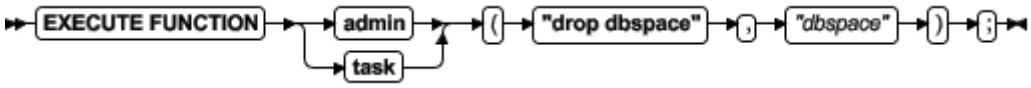
下列示例删除名为 demodbs 的数据库：

```
EXECUTE FUNCTION task("drop database","demodbs");
```

4.50 drop dbspace 参数：删除 dbspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 drop dbspace 参数来删除指定的 dbspace。

语法



元素	描述	关键考虑
dbspace	要删除的 dbspace 名。	该 dbspace 必须存在。 在您删除 dbspace之前，请删除所有您以前在该 dbspace 中创建的数据库和表。

用法

这个函数等同于 gspaces -d 命令。

示例

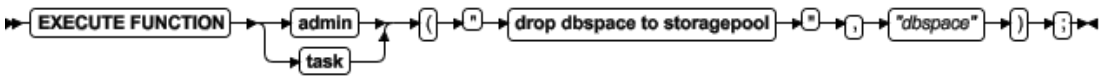
下列示例删除名为 dbspace4 的 dbspace：

```
EXECUTE FUNCTION task("drop dbspace","dbspace4");
```

4.51 drop dbspace to storagepool 参数：从空 dbspace 归还空间到存储池（SQL 管理 API）

随同 admin() 或 task() 函数，使用 drop dbspace to storagepool 参数来从空 dbspace 归还空间到存储池。

语法



元素	描述	关键考虑
dbspace	空 dbspace 的名称。	

示例

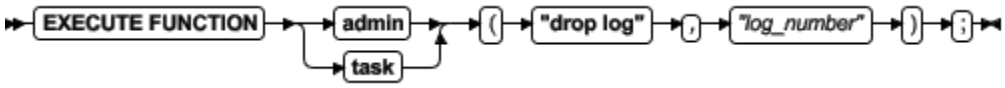
下列命令删除名为 dbs5 的空 dbspace，并将所有释放的空间添加到存储池。

```
EXECUTE FUNCTION task("drop dbspace to storagepool", "dbs5");
```

4.52 drop log 参数：删除逻辑日志（SQL 管理 API）

随同 admin() 或 task() 函数，使用 drop log 参数删除指定的逻辑日志。

语法



元素	描述	关键考虑
log_number	逻辑日志文件编号。	该编号必须是一个大于或等于 0 的无符号整数。

用法

使用这个函数来删除单个逻辑日志文件。

数据库服务器随时都要求最少三个逻辑日志文件。如果数据库服务器仅有三个逻辑日志文件，则您不可删除日志文件。

重要： 在您可删除前三个逻辑日志文件的任何一个之前，必须添加新的逻辑日志文件并运行逻辑日志文件的备份。必须使用 gtape -a 命令或 gtape -c 命令运行备份。在您添加新的逻辑日志文件并运行备份之后，您可使用 glogadmin -d -llognum 来删除前三个逻辑日志文件。

日志文件的状态决定能否删除日志文件，还决定当删除日志文件时数据库服务器采取的行动：

- 如果您删除一个还没有写过的日志文件，状态为新“添加的”（A），则数据库服务器删除日志文件并立即释放空间。
- 如果您删除一个状态为“用户”（U）或“释放”（F）的已使用过的日志文件，则数据库服务器将日志文件标记为“删除的”(D)。在您对包含日志文件的 dbspace 和 root dbspace 进行 0 级备份之后，数据库服务器删除该日志文件并释放空间。
- 您不可删除当前在用（C）或包含最后检查点记录（L）的日志文件。

您可从 gstat -l 命令的编号域获取日志编号。日志编号可能是无序的。

这个函数等同于 glogadmin -d -l lognum 命令。

示例

下列示例删除文件编号为 2 的逻辑日志:

```
EXECUTE FUNCTION task("drop log","2");
```

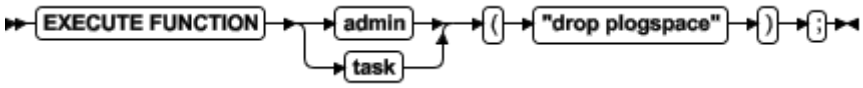
下列示例通过根据 chunk 编号查找日志编号来删除特定 chunk 的日志：

```
SELECT task("drop log", number) FROM sysmaster:syslogfil WHERE chunk = 1;
```

4.53 drop plogspace 参数：删除 plogspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 drop plogspace 参数来删除 plogspace。

语法



用法

要删除的 plogspace 必须为空。例如，如果您将物理日志移出 plogspace 并通过运行 glogadmin -p 命令移入 dbspace，则可删除该 plogspace。要不，您可通过创建新的 plogspace 将 plogspace 移到不同的 chunk。自动地移除旧的 plogspace。

这个函数等同于 gspaces -d 命令。

示例

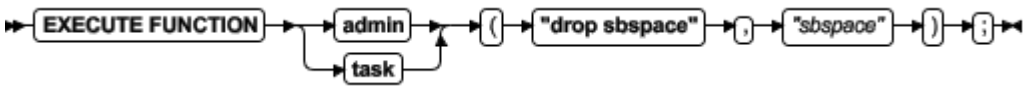
下列示例删除 plogspace：

```
EXECUTE FUNCTION task("drop plogspace");
```

4.54 drop sbspace 参数：删除 sbspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 drop sbspace 参数来删除指定的 sbspace。

语法



元素	描述	关键考虑
sbspace	要删除的 sbspace 名。	该 sbspace 必须存在。 在您删除 sbspace 之前，请删除所有包括引用该 sbspace 的 BLOB 或 CLOB 列的表。

用法

这个函数等同于 gspaces -d 命令。

示例

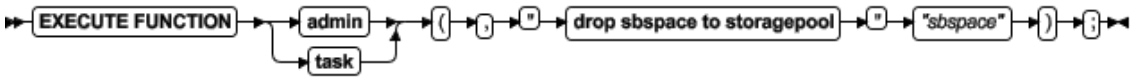
下列示例删除名为 sbspace3 的 sbspace：

```
EXECUTE FUNCTION task("drop dbspace","sbspace3");
```

4.55 drop sbspace to storagepool 参数：从空 sbspace 归还空间到存储池（SQL 管理 API）

随同 admin() 或 task() 函数，使用 drop sbspace to storagepool 参数来从空 sbspace 归还空间到存储池。

语法



元素	描述	关键考虑
sbspace	空 sbspace 名。	

示例

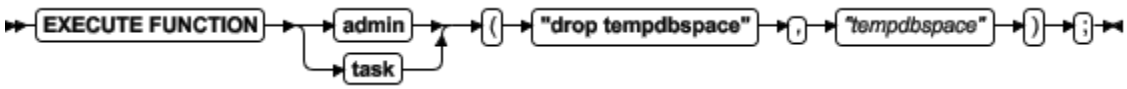
下列命令删除名为 sbspace8 的空 sbspace，并将所有释放的空间添加到存储池。

```
EXECUTE FUNCTION task("drop sbspace to storagepool", "sbspace8");
```

4.56 drop tempdbspace 参数：删除临时 dbspace（SQL 管理 API）

随同 admin() 或 task() 函数,使用 drop tempdbspace 参数来删除指定的临时 dbspace。

语法



元素	描述	关键考虑
tempdbspace	要删除的临时 dbspace 名称。	该临时 dbspace 必须存在。 在您删除临时 dbspace 之前，请删除您之前在该临时 dbspace 中创建的所有数据库和表。

用法

这个函数等同于 gspaces -d 命令。

示例

下列示例删除名为 tdbspace2 的临时 temporary：

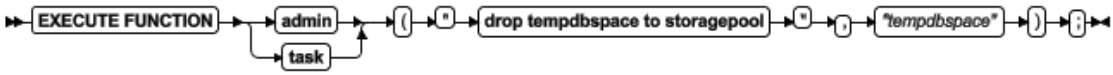
```
EXECUTE FUNCTION task("drop tempdbspace","tdbspace2");
```

4.57 drop tempdbspace to storagepool 参数：从空的临时 dbspace 归还空间到存储池（SQL 管

理 API）

随同 admin() 或 task() 函数，使用 drop tempdbspace to storagepool 参数来从空的临时 dbspace 归还空间到存储池。

语法



元素	描述	关键考虑
tempdbspace	空的临时 dbspace 名称。	

示例

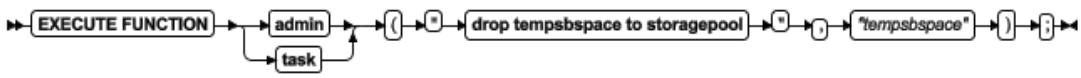
下列命令删除名为 tempdbs1 的临时 dbspace，并将所有释放的空间添加到存储池。

```
EXECUTE FUNCTION task("drop tempdbspace to storagepool", "tempdbs1");
```

4.58 drop tempsbspace to storagepool 参数：从空的临时 sbspace 归还空间到存储池（SQL 管理 API）

随同 admin() 或 task() 函数，使用 drop tempsbspace to storagepool 参数来从空的临时 sbspace 归还空间到存储池。

语法



元素	描述	关键考虑
tempsbspace	空的临时 sbspace 名称。	

示例

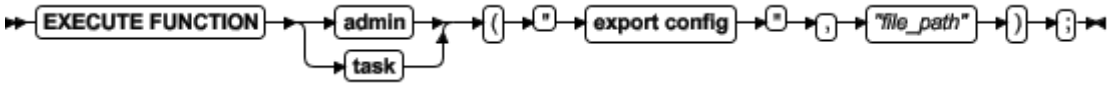
下列命令删除名为 tempsbspace3 的空的临时 sbspace，并将所有释放的空间添加到存储池。

```
EXECUTE FUNCTION task("drop tempsbspace to storagepool", "tempsbspace3");
```

4.59 export config 参数：导出配置参数值（SQL 管理 API）

随同 admin() 或 task() 函数，使用 export config 参数来导出包含所有配置参数及其当前值的文件。

语法



元素	描述	关键考虑
file_path	文件的完全路径名	请不要添加扩展名。

用法

该 SQL 管理 API 导出命令自动地创建一 ASCII 文件，赋予它您在命令中指定的名字。该文件的格式与 onconfig.std 文件的格式相同。

您必须指定完全路径名。不可指定相对路径。

这个命令等同于 gadmin -we 命令。

示例

下列命令将所有配置参数及其当前值导出到 /tmp 目录中的名为 cfg_12 的文件：

```
EXECUTE FUNCTION task("export config", "/tmp/cfg_12");
```

4.60 file status 参数：显示消息日志文件的状态（SQL 管理 API）

随同 admin() 或 task() 函数，使用 file status 参数来指定一 online、ON-Bar 活动或 ON-Bar 调试消息日志文件的状态。

语法

元素	描述	关键考虑
file_path	online、ON-Bar 活动或 ON-Bar 调试消息日志文件的完全路径名。	

示例

下列示例展示您可用来显示 /usr/gbasedbt/online.log 文件状态的参数：

```
execute function task("file status", "/usr/gbasedbt/online.log");
```

然后，服务器显示如下信息：

```
(expression)  File name          = /tmp/x
              Is File            = 1
              Is Directory       = 0
              Is Raw Device      = 0
              Is Block Device    = 0
              Is Pipe            = 0
              File Size          = 554
              Last Access Time   = 11/29/2010 21:55:02
              Last Modified Time = 11/29/2010 21:51:45
              Status Change Time = 11/29/2010 21:51:45
              User Id            = 200
              Group id           = 102
              File Flags         = 33206
```

4.61 grant admin 参数：授予运行 SQL 管理 API 命令的权限

随同 admin() 或 task() 函数，使用 grant admin 参数来授予运行 SQL 管理 API 命令的权限。

语法

元素	描述	关键考虑
user_name	授予权限的用户名。	
privilege_group	权限组名。	请参阅 SQL 管理 API 门户：按权限组划分参数 查看权限组列表。

用法

可授予个别用户权限来通过运行 SQL 管理 API 命令管理数据库。拥有这些权限的用户可以他们的用户名连接到数据库服务器并运行 SQL 管理 API 命令，或通过使用 GBase OpenAdmin Tool (OAT) for GBase 8s 或直接地通过连接。

仅用户 gbasedbt，或拥有 ADMIN 或 SQL 管理 API 命令的 GRANT 权限的用户可使用 grant admin 参数。

示例

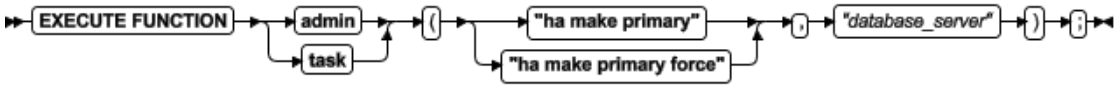
下列命令授予运行备份和恢复 SQL 管理命令的权限给用户 Bob：

```
EXECUTE FUNCTION task("grant admin", "Bob", "BAR");
```

4.62 ha make primary 参数：更改辅助服务器的模式（SQL 管理 API）

随同 admin() 或 task() 函数，使用 ha make primary 参数来将指定的辅助服务器更改为主服务器或标准服务器。

语法



元素	描述	关键考虑
database_server	数据库服务器名。	该名称必须在 sqlhosts 文件中的 dbservername 条目中定义，或作为 Enterprise Replication 组名。

用法

依赖于辅助服务器的类型，这个函数会有不同的结果：

- HDR 辅助服务器：关闭当前主服务器，且 HDR 辅助服务器成为主服务器。
- RS 辅助服务器：RS 辅助服务器更改为标准服务器。
- SD 辅助服务器：SD 辅助服务器成为新的主服务器。

使用 ha make primary 参数来将一不活动的辅助服务器更改为主服务器，当他们之间有活动的连接时。

使用 ha make primary force 参数来将一不活动的辅助服务器更改为主服务器，无论辅助服务器是否连接到它。如果连接是活动的，则函数成功，然而，如果您以 force 参数 在 SD 辅助服务器上运行该函数，则共享磁盘子系统可遭损坏。

这个函数等同于 gadmin -d make primary 命令。

示例

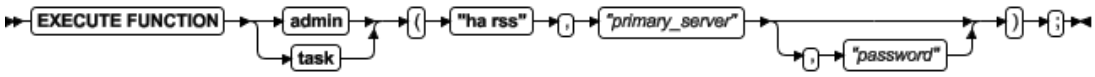
下列示例将名为 ids_stores2 的 HDR 辅助服务器转换成为主服务器：

```
EXECUTE FUNCTION task("ha make primary","ids_stores2");
```

4.63 ha rss 参数：创建 RS 辅助服务器（SQL 管理 API）

随同 admin() 或 task() 函数，使用 ha rss 参数来创建远程独立（RS）辅助服务器。

语法



元素	描述	关键考虑
password	要设置或更改的口令。	仅在第一次连接尝试期间使用此口令。主服务器和辅助服务器已经连接之后，不可更改该口令。
primary_server	主数据库服务器的名称。	必须在 DBSERVERNAME 或 DBSERVERALIASES 配置参数中定义该名称，或作为 Enterprise Replication 组名。

用法

在标准服务器或 quiescent HDR 辅助服务器上运行这个函数来将其转换为 RS 辅助服务器。这个函数等同于 gadmin -d RSS 命令。

示例

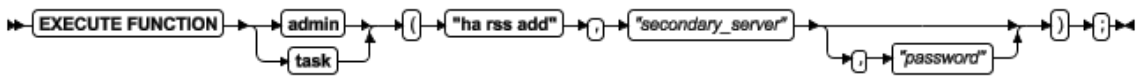
下列示例将标准服务器转换成为 RS 辅助服务器，主服务器名为 ids_stores：

```
EXECUTE FUNCTION task("ha rss","ids_stores");
```

4.64 ha rss add 参数：将 RS 辅助服务器添加到主服务器（SQL 管理 API）

随同 admin() 或 task() 函数，使用 ha rss add 参数来将主服务器与远程独立（RS）辅助服务器关联。

语法



元素	描述	关键考虑
password	要设置或更改的口令。	仅在第一次连接尝试期间使用该口令。主服务器和辅助服务器已经连接之后，不可更改该口令。
secondary_server	要转换到 RS 辅助服务器的数据库服务器的名称。	必须在 sqlhosts 文件中的 dbservername 条目中定义该名称，或作为 Enterprise

元素	描述	关键考虑
		Replication 组名。

用法

从已建立的主服务器运行这个函数来创建 RS 辅助服务器并在 sysha 数据库中注册该 RS 辅助服务器名。

这个函数等同于 gadmin -d add RSS 命令。

示例

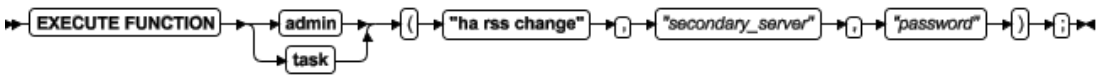
下列示例将名为 ids_stores2 的服务器作为 RS 辅助服务器与主服务器关联：

```
EXECUTE FUNCTION task("ha rss add","ids_stores2");
```

4.65 ha rss change 参数：更改 RS 辅助服务器的口令（SQL 管理 API）

随同 admin() 或 task() 函数，使用 ha rss change 参数来更改指定的 RS 辅助服务器的连接口令。

语法



元素	描述	关键考虑
password	要设置或更改的口令。	仅在第一次连接尝试期间使用该口令。主服务器与辅助服务器已经连接之后，不可更改该口令。
secondary_server	要转换到 RS 辅助服务器的数据库服务器名。	必须在 sqlhosts 文件中的 dbservername 条目中定义该名称，或作为 Enterprise Replication 组名。

用法

在已建立的主服务器上运行这个函数来更改在主服务器与辅助服务器之间连接的口令。

这个函数等同于 gadmin -d change RSS 命令。

示例

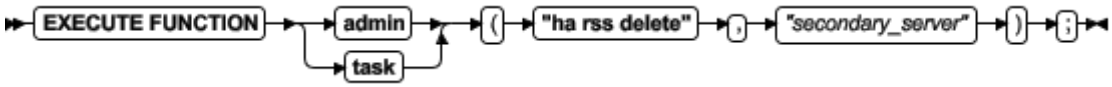
下列示例更改 RS 辅助服务器的口令为 secure：

```
EXECUTE FUNCTION task("ha rss change","ids_stores2","secure");
```

4.66 ha rss delete 参数：删除 RS 辅助服务器（SQL 管理 API）

随同 admin() 或 task() 函数，使用 ha rss delete 参数来停止复制并删除 RS 辅助服务器。

语法



元素	描述	关键考虑
secondary_server	要转换到 RS 辅助服务器的数据库服务器名。	必须在 sqlhosts 文件中的 dbservername 条目中定义该名称，或作为 Enterprise Replication 组名。

用法

从已建立的主服务器运行这个函数来停止复制，删除 RS 辅助服务器，并将 RS 辅助服务器转换到标准服务器。

这个函数等同于 gadmin -d delete RSS 命令。

示例

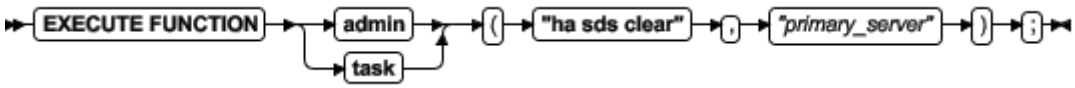
下列示例删除名为 ids_stores2 的 RS 辅助服务器：

```
EXECUTE FUNCTION task("ha rss delete","ids_stores2");
```

4.67 ha sds clear 参数：停止共享磁盘复制（SQL 管理 API）

随同 admin() 或 task() 函数，使用 ha sds clear 参数来停止复制到共享磁盘（SD）辅助服务器并将主服务器转换为标准服务器。

语法



元素	描述	关键考虑
primary_server	要转换到标准服务器的主服务器名。	必须在 sqlhosts 文件中的 dbservername 条目中定义该名称，或作为 Enterprise

元素	描述	关键考虑
		Replication 组名。

用法

在已建立的主服务器上运行这个函数来停止复制到 SD 辅助服务器。

这个函数等同于 gadmin -d clear SDS primary 命令。

示例

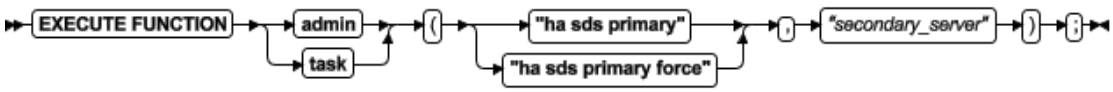
下列示例停止从名为 ids_stores 的主服务器复制到 SD 辅助服务器：

```
EXECUTE FUNCTION task("ha sds clear","ids_stores");
```

4.68 ha sds primary 参数：将 SD 辅助服务器转换到主服务器（SQL 管理 API）

随同 admin() 或 task() 函数，使用 ha sds primary 参数来将共享磁盘（SD）辅助服务器更改为主服务器。

语法



元素	描述	关键考虑
secondary_server	要设置作为主服务器的 SD 辅助服务器的名称。	必须在 sqlhosts 文件中的 dbservername 条目中定义该名称，或作为 Enterprise Replication 组名。

用法

在已建立的 SD 辅助服务器上运行这个函数来将其转换到主服务器。

使用 ha sds primary 参数来将不活动的 SD 辅助服务器转换到主服务器，如果该 SD 辅助服务器连接到它。

使用 ha sds primary force 参数来将不活动的 SD 辅助服务器转换为主服务器，无论是否有任何 SD 辅助服务器连接到它。如果会话是活动的，则调用成功，但共享磁盘子系统可被损坏。

这个函数等同于 gadmin -d make primary 命令。

示例

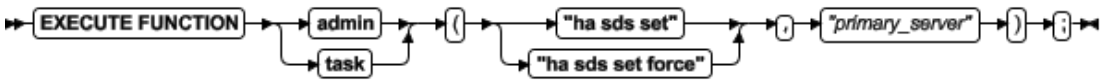
下列示例将名为 ids_stores3 的 SD 辅助服务器转换到主服务器：

```
EXECUTE FUNCTION task("ha sds primary","ids_stores3");
```

4.69 ha sds set 参数：创建共享磁盘主服务器（SQL 管理 API）

随同 admin() 或 task() 函数，使用 ha sds set 参数来定义主服务器复制到共享磁盘（SD）辅助服务器。

语法



元素	描述	关键考虑
primary_server	要设置作为主服务器的数据库服务器名。	必须在 sqlhosts 文件中的 dbservername 条目中定义该名称，或作为 Enterprise Replication 组名。

用法

在标准服务器上运行这个函数来定义其作为 SD 辅助服务器的主服务器。

使用 ha sds set 参数来定义不活动的标准服务器作为主服务器，如果 SD 辅助服务器连接到它。

使用 ha sds set force 参数来定义不活动的标准服务器作为主服务器，无论是否有任何 SD 辅助服务器连接到它。如果会话是活动的，则调用成功，但共享磁盘子系统可被损坏。

这个函数等同于 gadmin -d set SDS primary 命令。

示例

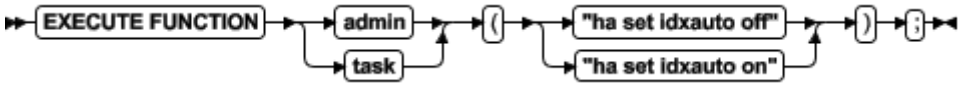
下列示例将名为 ids_stores 的标准服务器转换到主服务器：

```
EXECUTE FUNCTION task("ha sds set","ids_stores");
```

4.70 ha set idxauto 参数：复制索引到辅助服务器（SQL 管理 API）

随同 admin() 或 task() 函数，使用 ha set idxauto 参数来控制是否自动地将索引复制到辅助服务器。

语法



用法

在已建立的主服务器上运行这个函数来启用或禁用向辅助服务器的自动索引复制。

您可在任何类型的主服务器上运行这个函数。

这个函数等同于 gadmin -d idxauto 命令。

示例

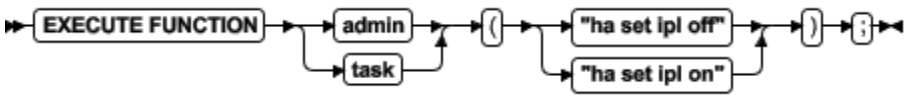
下列示例启用自动的索引复制：

```
EXECUTE FUNCTION task("ha set idxauto on");
```

4.71 ha set ipl 参数：在主服务器上构建日志索引（SQL 管理 API）

随同 admin() 或 task() 参数，使用 ha set ipl 参数来控制是否在主服务器上构建日志索引。

语法



用法

在已建立的主服务器上运行这个函数来启用或禁用索引构建的日志记录。这个函数重置 ONCONFIG 文件中 LOG_INDEX_BUILDS 配置参数的值。

您可在任何类型的主服务器上运行这个函数。

这个函数等同于 gadmin -wf LOG_INDEX_BUILDS 命令。

示例

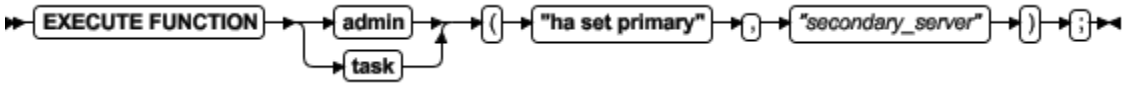
下列示例启用索引构建的日志记录：

```
EXECUTE FUNCTION task("ha set ipl on");
```

4.72 ha set primary 参数：定义 HDR 主服务器（SQL 管理 API）

随同 admin() 或 task() 函数，使用 ha set primary 参数来定义“高可用性数据复制”（HDR）主服务器并指定辅助服务器。

语法



元素	描述	关键考虑

元素	描述	关键考虑
secondary_server	要连接到的 HDR 辅助服务器名。	必须在 sqlhosts 文件中的 dbservername 条目中定义该名称，或作为 Enterprise Replication 组名。

用法

在标准服务器上运行这个函数来将其转换到 HDR 主服务器并连接到指定的 HDR 辅助服务器。如果连接成功，则开始复制。

这个函数等同于 gadmin -d primary 命令。

示例

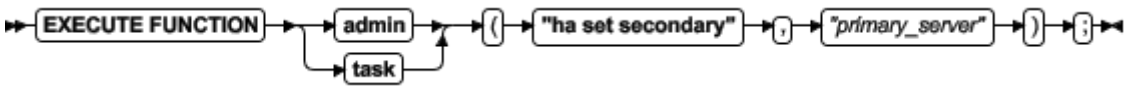
下列示例将名为 ids_stores 的标准服务器转换到 HDR 主服务器：

```
EXECUTE FUNCTION task("ha set primary","ids_stores");
```

4.73 ha set secondary 参数：定义 HDR 辅助服务器（SQL 管理 API）

随同 admin() 或 task() 函数，使用 ha set secondary 参数来定义“高可用性数据复制”（HDR）辅助服务器并指定主服务器。

语法



元素	描述	关键考虑
primary_server	要连接到的 HDR 主服务器名。	必须在 sqlhosts 文件中的 dbservername 条目中定义该名称，或作为 Enterprise Replication 组名。

用法

在标准数据库服务器上运行这个函数来将其转换到 HDR 辅助服务器，并连接到指定的主服务器。如果连接成功，则开始复制。

这个函数等同于 gadmin -d secondary 命令。

示例

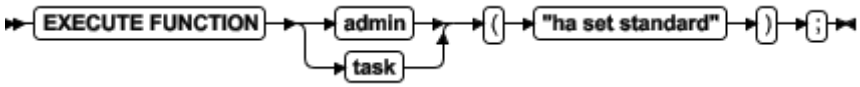
下列示例将标准服务器转换到 HDR 辅助服务器，主服务器名为 ids_stores：

```
EXECUTE FUNCTION task("ha set secondary","ids_stores");
```

4.74 ha set standard 参数：将 HDR 服务器转换成为标准服务器（SQL 管理 API）

随同 admin() 或 task() 函数，使用 ha set standard 参数来将“高可用性数据复制”（HDR）主服务器或辅助服务器转换到标准服务器。

语法



用法

在 HDR 主服务器或辅助服务器上运行这个函数来将其转换到标准服务器。删除主服务器与辅助服务器之间的连接，且终止复制。不更改 HDR 对中其他服务器的模式。

这个函数等同于 gadmin -d standard 命令。

示例

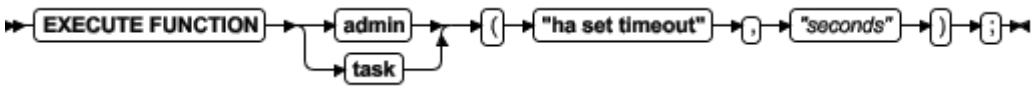
下列示例将 HDR 辅助服务器转换到标准服务器：

```
EXECUTE FUNCTION task("ha set standard");
```

4.75 ha set timeout 参数：更改 SD 辅助服务器超时（SQL 管理 API）

随同 admin() 或 task() 函数，使用 ha set timeout 参数来更改主服务器等待来自共享磁盘（SD）辅助服务器的响应的时间秒数。

语法



元素	描述	关键考虑
seconds	在断开 SD 辅助服务器连接之前，主服务器等待的秒数。	该值必须是下列范围内的正整数： • 从 2 • 至 2 147 483 647

用法

在已建立的共享磁盘主服务器上运行这个函数来指定主服务器等待从 SD 辅助服务器发送日志位置响应的时间秒数。如果在指定的时间量内未收到来自 SD 辅助服务器的日志位置响应， 则主服务器从 SD 辅助服务器断开并继续。如果在等待 SD 辅助服务器时页清空已经超时，则等待指定秒数之后，主服务器启动移除 SD 辅助服务器。

这个函数重置 ONCONFIG 文件中 SDS_TIMEOUT 配置参数的值。

这个函数等同于 gadmin -wf SDS_TIMEOUT 命令。

示例

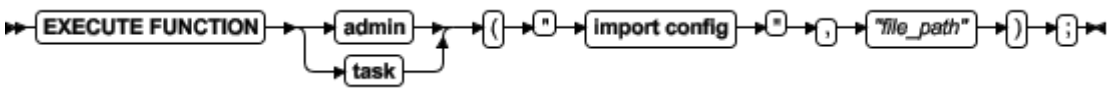
下列示例设置超时期间为 5 秒：

```
EXECUTE FUNCTION task("ha set timeout","5");
```

4.76 import config 参数：导入配置参数值（SQL 管理 API）

随同 admin() 或 task() 函数，使用 import config 参数来导入包含一个或多个可动态地更新的配置参数的文件并应用这些新值。

语法



元素	描述	关键考虑
file_path	包含一个或多个可动态地更新的配置参数名称和值的以前导出的文件的完全路径名。	

用法

可动态地更改的配置参数是那些您可以 gadmin -wf 或 gadmin -wm 命令为会话更改的参数。

您必须指定完全路径名。你不可指定相对路径。

这个命令等同于 gadmin -wi 命令。

示例

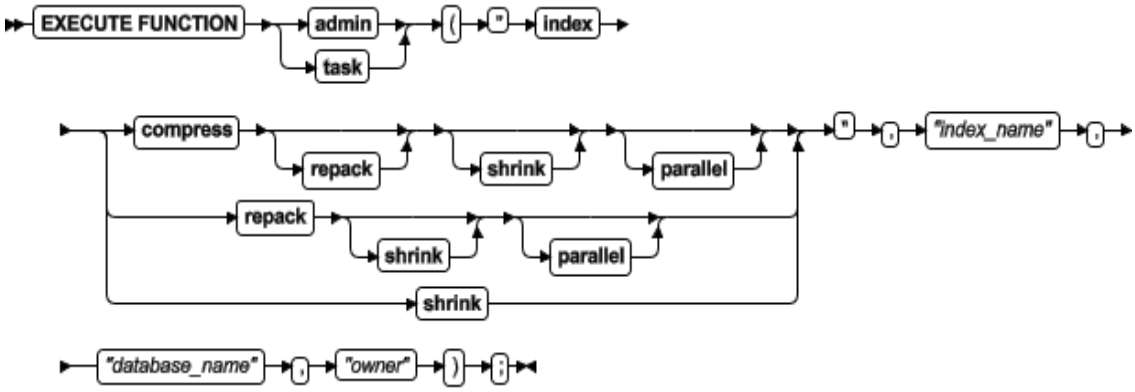
下列命令导入 /tmp 目录中名为 cfg_12 的文件：

```
EXECUTE FUNCTION task("import config", "/tmp/cfg_12");
```

4.77 index compress repack shrink 参数：优化 B-tree 索引的存储（SQL 管理 API）

随同 admin() 或 task() 函数，使用 index compress repack shrink 参数来压缩分离的 B-tree 索引，合并空闲空间（重新打包）并归还分区中的空闲空间（收缩）。

语法：索引压缩命令参数



命令参数

下表包含每一参数的简要说明。

表 1. 索引压缩操作的参数

参数	描述
compress	压缩索引。
parallel	并行地运行压缩或重新打包操作。为每一表的分片或分片列表启动一个线程，且跨越这些分片并行地运行该操作。
repack	通过将数据移到索引的前部来合并空闲空间。
shrink	将索引末端的空闲空间归还到 dbspace，从而减少索引的总大小。

命令元素

下表显示您可用来压缩、重新打包和收缩索引的元素。

表 2. 索引压缩命令元素

元素	描述	关键考虑
index_name	您想要压缩的索引名称。	必需的。 您必须使用系统目录表中相同的大写或小写字母。
database_name	包含您想要压缩的索引的数据库名。	可选的。 如果您未指定数据库，则数据库服务器使用当前的数据库。

元素	描述	关键考虑
		如果您输入数据库名，则必须使用与系统目录表中相同的大写或小写字母。
owner	包含您想要压缩的索引的数据库的所有者名。	可选的。 如果您未指定所有者，则数据库服务器使用当前的所有者。 如果您输入所有者名，则必须使用与系统目录表中相同的大写或小写字母。

用法

您可压缩一个在分片的或未分片的表上分离的 B-tree 索引。

要被压缩，索引必须有至少 2000 个键。如果索引内的分片没有至少 2000 个键，则当创建索引时数据库服务器不压缩索引或分片。即使有新的键添加到索引，索引也保持不压缩。

如果您想要压缩该索引，请运行另一个带有 index compress 参数的 SQL 管理 API task() 或 admin() 函数。

要确定索引是否包含最小键数，请运行 gcheck -pT 命令并查看 Number of keys 域中的信息。

通常情况下，请您在压缩操作之后执行重新打包操作，并在重新操作之后执行收缩。

压缩操作仅压缩索引的叶子（底层）。

你可取消操作，例如在 DB-Access 中按下 CTRL-C。

你不可解压缩索引。如果您想要解压缩索引，则可删除压缩的索引并重新创建它。

示例

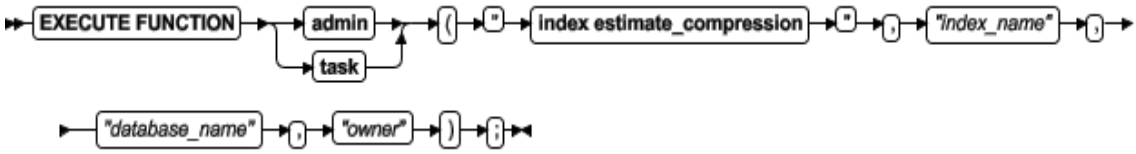
下列命令并行地压缩、重新打包并收缩索引。

```
EXECUTE FUNCTION task("index compress repack shrink parallel",
"ind5", "customer", "jayson");
```

4.78 index estimate_compression 参数：估计索引压缩（SQL 管理 API）

随同 admin() 或 task() 函数，使用 index estimate_compression 参数来估计您是否可通过压缩 B-tree 索引节省磁盘空间。

语法：index estimate_compression 命令参数



命令元素

下表显示您可用来估计索引压缩的元素。

元素	描述	关键考虑
index_name	您想要估算压缩成效的那个索引的名称。	要求的。 您必须使用与系统目录表中相同的大写或小写字母。
database_name	包含该索引的数据库名。	可选的。 如果您未指定数据库，则数据库服务器使用当前的数据库。 如果您输入数据库名称，则必须使用与系统目录表中相同的大写或小写字母。
owner	包含该索引的数据库的所有者名称。	对于索引可选的。 如果您未指定所有者，则数据库服务器使用当前的所有者。 如果您输入所有者名称，则必须使用与系统目录表中相同的大写或小写字母。

用法

您可仅对分片的或未分片的表上分离的 B-tree 索引估计压缩。

估计压缩操作显示索引名、可被归档的估计的压缩比率、当前的压缩比率和百分率损益的估计。如果索引未压缩，则当前的比率为 0.0。

示例

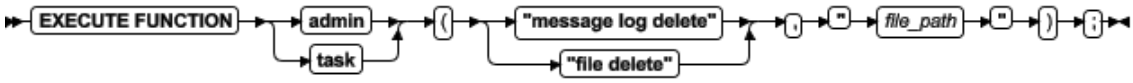
下列命令估计 customer 数据库中名为 ind4 的索引压缩成效，数据库所有者是 anjul。

```
EXECUTE FUNCTION task("index estimate_compression","ind4","customer","anjul");
```

4.79 message log delete 参数：删除消息日志文件（SQL 管理 API）

随同 admin() 或 task() 函数，使用 message log delete 参数或 file delete 参数来指定要删除的特定 online、ON-Bar 活动或 ON-Bar 调试消息日志。

语法



元素	描述	关键考虑
file_path	特定的 online、ON-Bar 活动或 ON-Bar 调试消息日志文件的完全路径名。	

示例

下列示例显示您可用来删除 /usr/gbasedbt/online.log 文件的参数：

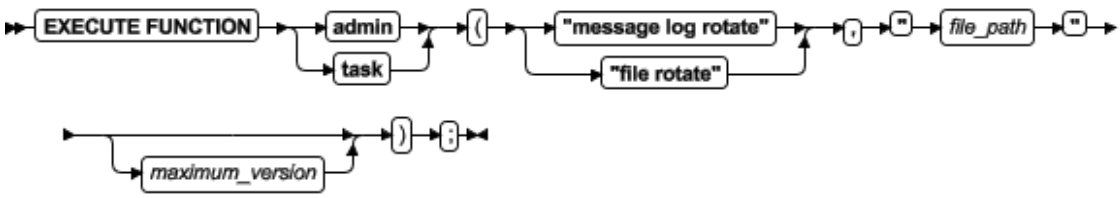
```
execute function task("message log delete", "/usr/gbasedbt/online.log");
execute function task("file delete", "/usr/gbasedbt/online.log");
```

4.80 message log rotate 参数：轮换消息日志文件（SQL 管理 API）

随同 admin() 或 task() 函数，使用 message log rotate 参数或 file rotate 参数来指定要轮换的特定 online、ON-Bar 活动或 ON-Bar 调试消息日志文件，并指明要轮换的消息日志的最大数目。

当消息日志文件轮换时，数据库服务器切换到新的 online 消息日志文件，且将先前的日志文件 ID 号增加 1。当达到日志文件的最大数目时，删除最高 ID 的日志文件。

语法



元素	描述	关键考虑
file_path	服务器将轮换的 online、ON-Bar 活动或 ON-Bar 调试日志文件的全路径名，例如，/usr/gbasedbt/online.log。	
maximum_version	最高 ID 的日志文件。这个是服务器将轮换的最大消息日志版本。	

示例

下列示例显示您可用来轮换最大 52 的 /usr/gbasedbt/online.log 文件的参数：

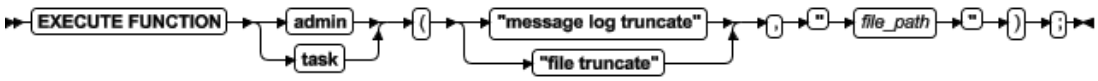
```
execute function task("message log rotate", "/usr/gbasedbt/online.log",52);
execute function task("file rotate", "/usr/gbasedbt/online.log",52);
```

当数据库服务器轮换这些文件时，服务器删除该文件的版本 52。版本 51 成为版本，版本 50 成为版本 51，依此类推。新的 online 日志成为版本 1。

4.81 message log truncate 参数：删除消息日志文件的内容（SQL 管理 API）

随同 admin() 或 task() 函数，使用 message log truncate 参数或 file truncate 参数来指定要删节的特定 online、ON-Bar 活动或 ON-Bar 调试消息日志文件。当数据库服务器删节消息日志文件时，服务器删除日志文件中的消息，但保留日志文件。

语法



元素	描述	关键考虑
file_path	online、ON-Bar 活动或 ON-Bar 调试消息日志文件的完全路径名。	

示例

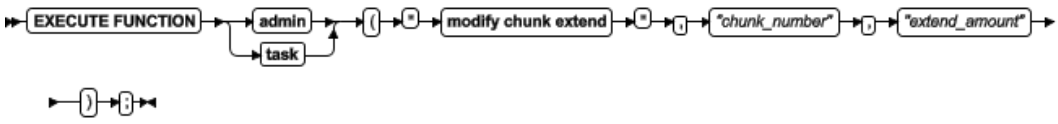
下列示例显示您可用来删节 /usr/gbasedbt/online.log 文件的参数：

```
execute function task("message log truncate", "/usr/gbasedbt/online.log");
execute function task("file truncate", "/usr/gbasedbt/online.log");
```

4.82 modify chunk extend 参数：扩展 chunk 的大小（SQL 管理 API）

随同 admin() 或 task() 函数，使用 modify chunk extend 参数来以指定的最小数量扩展 chunk 的大小。该 chunk 必须标记为可扩展的。

语法



元素	描述	关键考虑
chunk_number	The number of the chunk.	chunk 号。
extend_amount	要添加到 chunk 的空间的最小数量，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。

用法

在该 chunk 可被扩展之前，你必须或手工地或自动地标记 chunk 为可扩展的。随同 admin() 或 task() 函数，使用 modify chunk extendable 参数来标记 chunk 为可扩展的。

modify chunk extend SQL 管理 API 命令是 adm_add_storage 任务的替代方式。当包含该 chunk 运行的空间空闲页不足或用尽时，服务器可运行这个任务来自动地扩展 chunk 的大小。

您不可在镜像的空间中扩展 chunk，且如果您在运行 modify chunk extend SQL 管理 API 命令时提供了镜像 chunk 号，则您将收到错误提示。

要识别镜像的空间中的主 chunk 和镜像 chunk，请在 gstat -d 命令输出中 flags 域的位置 1 中查找 P （主）或 M（镜像）。

服务器可能截取需要的大小，依赖于页大小和该空间配置的创建大小和扩展大小。

示例

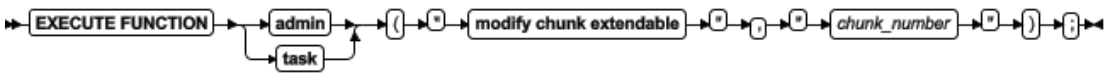
假定您的 gstat -d 命令输出显示 3 号 chunk 是镜像 chunk，4 号 chunk 不是镜像 chunk。您不可扩展 3 号 chunk 的大小。然而，您可更改 4 号 chunk。下列命令扩展 4 号 chunk 的大小 10000 KB：

```
EXECUTE FUNCTION task("modify chunk extend", "4", "10000");
```

4.83 modify chunk extendable 参数：标记 chunk 为可扩展的（SQL 管理 API）

随同 admin() 或 task() 函数，使用 modify chunk extendable 参数来指定未镜像的 dbspace 或临时 dbspace 中的特定 chunk 可扩展。

语法



元素	描述	关键考虑
chunk_number	chunk 号。	

用法

如果 chunk 标记为可扩展的，则二者之一：

当包含该 chunk 的未镜像的 dbspace 或临时 dbspace 运行空闲页太少或用尽时，服务器可自动地扩展 chunk。

您可随同 admin() 或 task() 函数，使用 modify chunk extend 参数来扩展 chunk 的大小。

然而，如果 dbspace 或临时 dbspace 的扩展大小设置为 0，则服务器不可在那个空间中自动地扩展可扩展的 chunk。在这种情况下，您仍可手动地扩展 chunk。

服务器将自动地标记那些从可扩展的存储池条目分配的 chunk 为可扩展的。因此，您不需要标记这些 chunk 为可扩展的。要了解关于可扩展的存储池条目的信息，请参阅 storagepool add 参数：添加存储池条目（SQL 管理 API）。

不可扩展镜像的空间中的 chunk。如果您试图使镜像 chunk 为可扩展的，则您会收到错误提示。

要识别镜像的空间中的主 chunk 和镜像 chunk，请在 gstat -d 命令输出中 flags 域的位置 1 中查找 P（主）或 M（镜像）。

示例

下列 gstat -d 输出的片段显示 3 号 chunk 是镜像 chunk：

```
Chunks
address      chunk/dbs  offset   size     free    bpages   flags pathname
451191c8     1      1   0        225000   101572           PO-B-- /reg1/rootchunk
451197d0     2      2   0        1250     1149             PO-B-- /reg1/dbs1
451199d0     3      3   0        1250     1149             PO-B-- /reg1/dbs2
46a36638     3      3   0        1250     0                MO-B-- /reg1/chunk2
45119bd0     4      4   0        1250     1149             PO-B-- /reg1/dbs3
```

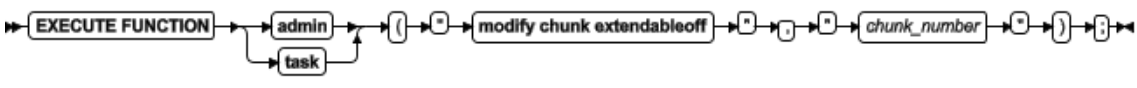
如此，您不可扩展 3 号 chunk 的大小。然而，您可指定 4 号 chunk 为可扩展的，如下所示：

```
EXECUTE FUNCTION sysadmin:task("modify chunk extendable", "4");
```

4.84 modify chunk extendable off 参数：标记 chunk 为不可扩展的（SQL 管理 API）

随同 admin() 或 task() 函数，使用 modify chunk extendable off 参数来指定特定的 chunk 不可被扩展。

语法



元素	描述	关键考虑
chunk_number	chunk 号。	

用法

chunk 的缺省状态是不可扩展的。如果您先前标记了 chunk 为可扩展的，则您可更改状态为不可扩展的。

如果 chunk 标记为不可扩展的，则：

当包含该 chunk 的空间运行空闲页太低或用尽时，服务器不可自动地扩展该 chunk。

您不可手工地扩展该 chunk 的大小。

如果存储池包含条目，则服务器可通过添加另一 chunk 到该存储空间来扩展存储空间。

示例

下列示例指定您或服务器不可扩展 chunk 9：

```
EXECUTE FUNCTION task("modify chunk extendable off", "9");
```

4.85 modify config 参数：更改配置参数（SQL 管理 API）

随同 admin() 或 task() 函数，使用 modify config 参数来更改内存中配置参数的值，直到您重启数据库服务器。使用 modify config persistent 参数来更改内存中配置参数的值并在您重启服务器之后在 onconfig 文件中保存该值。

语法

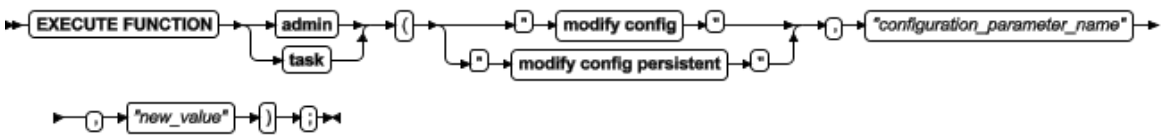


表 1. modify config 命令元素

元素	描述	关键考虑
configuration_parameter_name	您想要更改的配置参数的名称。	
new_value	配置参数的新值。	要了解关于配置参数的有效值信息，请参阅 数据库配置参数。

用法

这个 SQL 管理 API 命令等用于使用 gadmin -wm 或 -wf 命令来更改配置参数的值。

示例

下列命令为当前使用将内存中的 DYNAMIC_LOGS 配置参数的值更改为 2：

```
EXECUTE FUNCTION task("modify config","DYNAMIC_LOGS","2");
```

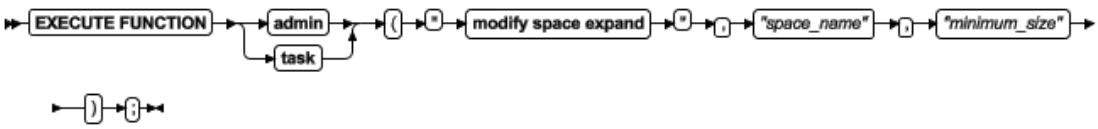
下列命令为当前使用更改 DYNAMIC_LOGS 配置参数的值。在您重启服务器之后，更改的值保留在 onconfig 文件中。

```
EXECUTE FUNCTION task("modify config persistent","DYNAMIC_LOGS","2");
```

4.86 modify space expand 参数：扩大空间的大小（SQL 管理 API）

随同 admin() 或 task() 函数，使用 modify space expand 参数来立即扩大空间的大小，当您不想等待 GBase 8s 来自动地扩大空间时。

语法



元素	描述	关键考虑
space_name	存储空间的名称。	
minimum_size	您想要扩大空间大小的最小值。	请参阅 admin() 和 task() 参数大小规范。

用法

modify space expand SQL 管理 API 命令立即扩大存储空间，或通过扩展该空间中的可扩展 chunk，或者通过添加新 chunk。空间的创建大小和扩展大小设置不影响这个操作。

添加到空间的实际 KB 数可能超过您需要的大小，这依赖于一些因素，比如空间的页大小，以及存储池中可用条目的 chunk 大小设置。

存储池必须包含服务器可用来扩展该空间的条目（比如裸设备、熟文件或目录）。

您运行 modify space expand SQL 管理 API 命令之后， GBase 8s 通过扩展该空间中可扩展 chunk 来首次尝试扩大该空间。如果该空间不包含任何可扩展的 chunk，则服务器使用存储池中的条目来扩展该空间。

您不可扩大镜像的存储空间。

示例

下列命令扩大 dbspace5 10 MB：

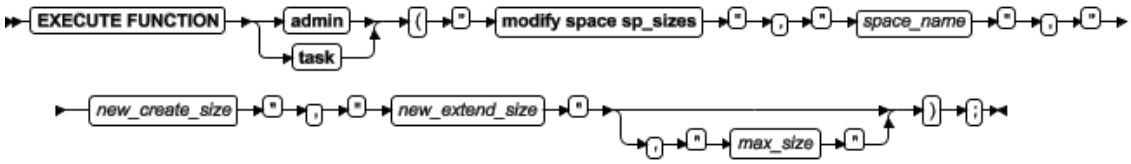
```
EXECUTE FUNCTION task("modify space expand", "dbspace5", "10 MB");
```

4.87 modify space sp_sizes 参数：更改可扩展的

存储空间的大小（SQL 管理 API）

随同 admin() 或 task() 函数，使用 modify space sp_sizes 参数来更改与扩大的存储空间相关的创建大小、扩展大小和大小的最大值。更改大小来控制 GBase 8s 如何使用特定存储空间的存储池条目。

语法



元素	描述	关键考虑
space_name	存储空间的名称。	
max_size	存储空间大小的最大值，以 KB 为单位。	大小的缺省最大值为 0，表明大小不受限制。
new_create_size	当使用存储池动态地扩大这个空间时，服务器可创建的新 chunk 大小的最小值。您可定义该大小为 KB 数，或为全部空间的百分率。	缺省创建大小设置为该空间全部大小的 10%。 指定的大小会影响服务器自动地创建的 chunk。该大小不影响您可能为相关的空间创建的任何手工 chunk。
new_extend_size	当在无镜像的 dbspace 或临时 dbspace 中自动地扩展 chunk 时，服务器可使用该大小的最小值。该大小可指定为 KB 数，或全部空间的百分率。	缺省扩展大小为 10 MB。 指定的大小会影响服务器自动地扩展的 chunk。该大小不影响您对相关的空间进行的任何手工 chunk 扩展。

用法

如果创建大小或扩展大小值为 100 或更低的值，则 GBase 8s 将该值解释为百分率（例如，10 = 10%，2.84 = 2.84%）。 如果该值为 1000 或更高，则服务器将该值解释为特定的 KB 数。值 100 - 1000 无效。

如果您设置创建大小和扩展大小为 0，则即使当空间变满时 GBase 8s 也不自动地扩大该空间。此外，如果您设置扩展大小为 0，则您还从那个空间中的所有 chunk 移除“Extendable”标志。这是使用一个操作将空间中所有 chunk 标记为不可扩展的一种简便方法。

创建大小和扩展大小值是大小的最小值。空间扩大的实际大小可能更大，这依赖于服务器正在使用的存储池的 chunk 大小，或服务器在那个特定时间需要的空间量。

例如，假定您在必要时创建了一个存储池条目来扩大存储空间。然后假定名为 logdbs 的 dbspace 用尽空闲页并要求额外的 500 MB 做新日志。如果 logdbs 中没有 chunk 可扩展，则 GBase 8s 添加一个 chunk，其大小的最小值是由 logdbs dbspace 的创建大小值指定的。如果 logdbs dbspace 的创建大小小于或等于 500 MB，则服务器尝试去找到一个最小 500 MB 的空间。如果 logdbs 的创建大小为 1 GB，则服务器忽略所需要的大小并添加一个 1 GB chunk。

如果服务器不能找到所要求的最小量空间，则服务器返回空间不足错误提示，且日志创建失败。

如果您将存储空间大小的最大值设置为非 0 值，则存储空间不可超过该大小的最大值，不管新的扩展大小。当在大小最大值之前留出的扩大空间量小于新的扩展大小时，截断该扩展大小并将空间扩展到大小的最大值。当空间达到大小的最大值时，触发事件报警 86001。当留出的扩大空间量小于存储池的 chunk 大小的最小值时，不扩大该空间并返回错误提示。

示例

下列命令将最小的创建大小设置为 60 MB，最小的扩展大小为 10 MB，且名为 dbspace3 的 dbspace 大小的最小值为 100 MB：

```
EXECUTE FUNCTION task("modify space sp_sizes", "dbspace3", "60000",
      "10000", "100000");
```

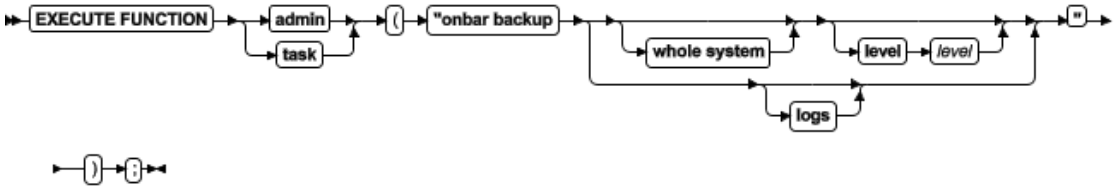
下列命令将最小的创建大小设置为 20%，且名为 dbspace8 的 dbspace 的最小扩展大小为 1.5%：

```
EXECUTE FUNCTION task("modify space sp_sizes", "dbspace8", "20", "1.5");
```

4.88 gbackuprestore 参数：备份存储空间（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gbackuprestore 参数来备份存储空间。

语法



元素	描述	关键考虑
gbackuprestore backup	执行存储空间的完全备份	如果您未指定级别，则执行 0 级备份。
whole system	执行全系统备份	这个等同于从命令行发出带有 -w 选项的 gbackuprestore 命令。如果您未指定级别，则执行 0 级备份。
level *level*	指定在存储空间上执行的备份级别： • 0 代表完全备份。这是缺省值。 • 1 表示自从最后 0 级备份以来的更改 • 2 表示自从最后 1 级备份以来的更改	如果您需要增量备份且尚未执行特定存储空间的级别备份，则这个功能在先前的级别备份存储空间。 例如，如果您需要 1 级备份，且该函数发现无 0 级备份，则改为进行 0 级备份。 这等同于从命令行发出带有 -L *level* 选项的 gbackuprestore 命令。
logs	执行逻辑日志文件的备份	这等同于从命令行发出带有 -l 选项的 gbackuprestore 命令。

用法

这个函数等同于调用 gbackuprestore 命令的特定选项来创建存储空间和逻辑日志文件的备份。

示例

下列示例创建存储空间的 0 级备份：

```
EXECUTE FUNCTION task("gbackuprestore backup");
```

下列示例创建存储空间的 1 级备份：

```
EXECUTE FUNCTION task("gbackuprestore backup level 1");
```

下列示例创建逻辑日志文件的 1 级备份：

```
EXECUTE FUNCTION task("gbackuprestore backup logs");
```

下列示例创建存储空间的全系统 0 级备份：

```
EXECUTE FUNCTION task("gbackuprestore backup whole system");
```

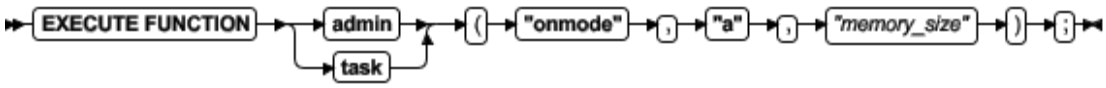
下列示例创建存储空间的全系统 2 级备份：

```
EXECUTE FUNCTION task("gbackuprestore backup whole system level 2");
```

4.89 gadmin 和 a 参数：添加共享内存段（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 a 参数来添加共享内存段。

语法



元素	描述	关键考虑
memory_size	新的虚拟共享内存段的大小，以 KB 为单位。	*size* 的值必须是正整数，不超过操作系统对共享内存段大小的限制。

用法

因为数据库服务器自动地添加所需要的段，所以您不需要添加段到共享内存的虚拟部分。然而，随着添加段，在得到需要的内存之前，数据库服务器可能达到操作系统对段的最大数目限制。当 SHMADD 配置参数设置较小以至于数据库服务器在得到某操作需要的内存之前就用尽了可用段的数目时，通常发生这种情况。

您可使用这个函数来添加段，这个段比 SHMADD 配置参数指定的大小更大。通过使用这个参数来添加段，您可遵守操作系统对段的限制，同时满足数据库服务器对更多内存的需要。

这个函数等同于 gadmin -a 命令。

示例

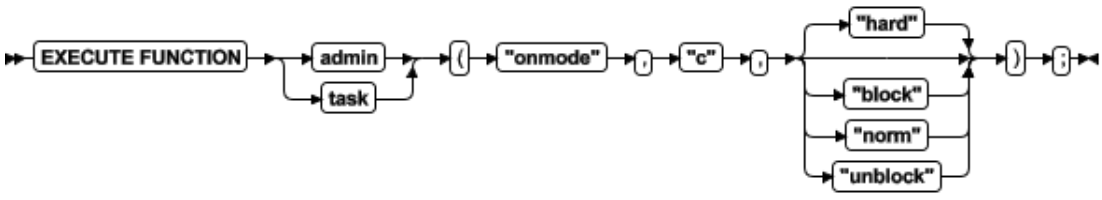
下列示例添加 500 KB 虚拟共享内存：

```
EXECUTE FUNCTION task("gadmin","a","500");
```

4.90 gadmin 和 c 参数：强制检查点（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 c 参数来强制检查点。

语法



用法

这个函数强制检查点，清空缓冲区到磁盘。如果逻辑日志中的最近检查点阻止释放逻辑日志文件（状态 U-B-L），则您可使用 c 选项来强制检查点。

使用 block 参数来阻止数据库服务器处理任何事务。使用这个选项在 GBase 8s 上执行外部备份。 当数据库服务器被阻塞时，用户不可访问它，除了处于 read-only 模式。直到解除数据库服务器阻塞，事务才可完成。

使用 hard 参数来强制阻塞的检查点。这是缺省值。

使用 norm 参数来强制未阻塞的检查点。

使用 unblock 参数来解除数据库服务器阻塞。当解除数据库服务器阻塞时，数据事务和正常的数据库服务器操作可重新开始。请您在 GBase 8s 上完成外部备份之后使用这个选项。

这个函数等同于 gadmin -c 命令。

示例

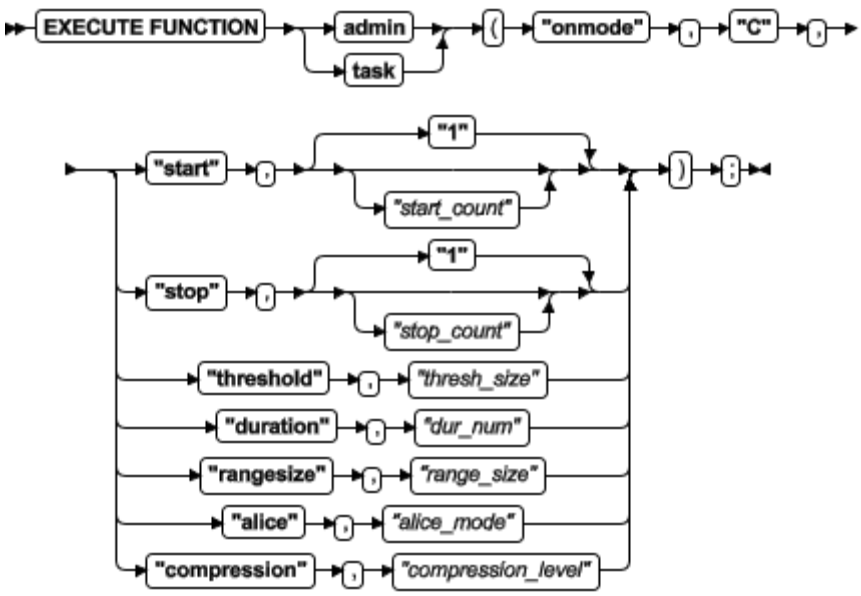
下列示例启动阻塞的检查点：

```
EXECUTE FUNCTION task("gadmin","c","hard");
```

4.91 gadmin 和 C 参数：控制 B-tree 扫描程序（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 C 参数来控制 B-tree 扫描程序清除已删除条目的索引。

语法



元素	描述	关键考虑
alice_mode	系统的 **alice** 模式。	从 0（OFF）至 12 范围内的有效整数值。

元素	描述	关键考虑
compression_level	对于数据库服务器实例，两个部分地已使用索引页的合并级别。如果那些页上的数据总数达到设置的级别，则合并页。	级别的有效值为 low、med（中级）、high 和 default。系统缺省值为 med。
dur_num	热列表有效的秒数。	这个秒数期满后，即使列表上有未处理的条目，也由下一个可用的 B-tree 扫描程序线程重建热列表。正在处理请求的扫描程序不中断。
range_size	在启用索引范围清除之前索引的大小。	可用大小 -1 来禁用范围扫描。
start_count	要启动的 B-tree 扫描程序线程数。	如未指定 *start_count*，则再启动一个线程。同时最多可启动 32 个线程。但是，同时运行的扫描程序线程数没有限制。
stop_count	要停止的 B-tree 扫描程序线程数。	如未指定 *stop_count*，则停止单个线程。停止所有索引扫描程序防止所有索引清除。 如果您指定一个比正在运行的线程数更大的 *stop_count* 值，则不发出错误提示，但是停止所有扫描程序线程。
thresh_size	在索引置于热列表上之前，索引必须遇到的删除条目的最小数。	在阈值之上的所有索引都已清除，且没有 B-tree 扫描程序要做的其他工作之后，阈值之下的索引被添加到热列表。

用法

B-tree 扫描程序有统计信息，追踪索引效率以及索引在服务器上增添了多少额外工作。根据因为提交的删除索引条目索引已经完成的额外工作量，B-tree 扫描程序开发一个导致服务器做额外工作的索引的排序列表，称为热列表。 首先清除索引导致的额外工作的最高量，并以递减的顺序清除剩余的索引。DBA 可动态地分配清除线程来配置工作负载。

这个函数等同于 gadmin -C 命令。

示例

下列命令启动 60 个 B-tree 扫描程序线程：

```
EXECUTE FUNCTION admin("gadmin","C","start","30");
EXECUTE FUNCTION admin("gadmin","C","start","30");
```

下列命令停止所有这些线程：

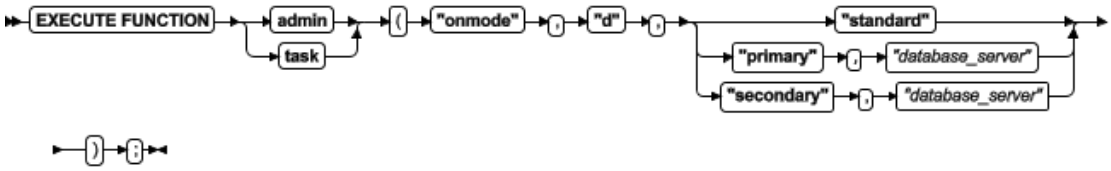
```
EXECUTE FUNCTION admin("gadmin","C","stop","30000");
```

当 stop_count 值大于正在运行的线程数时，不发出错误提示。

4.92 gadmin 和 d 参数：设置数据复制类型（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 d 参数来更改服务器参与高可用性数据复制（HDR）的模式。

语法



元素	描述	关键考虑
database_server	主或辅助数据库服务器的名称。	*dbserver* 名称必须对应于预期的辅助数据库服务器在 sqlhosts 文件中的 dbservername 条目中指定的值。

用法

使用这个函数来设置“高可用性数据复制”类型为 standard、primary 或 secondary。当数据库服务器处于 quiescent、online 或 read-only 模式时，您可使用 standard 参数。在共享内存重新初始化之后，保存数据复制对中其他数据库服务器的 dbserver 参数以及数据库服务器的类型（standard、primary 或 secondary）。

standard 参数删除数据复制对（如果存在的话）中数据库服务器之间的连接，并设置当前数据库服务器的数据库服务器类型为 standard。这个选项不更改数据复制对中其他数据库服务器的模式或类型。

primary 和 dbserver 参数设置数据库服务器类型为 primary，并试图与 dbserver 指定的数据库服务器连接。如果连接成功，则开启数据复制。主数据库服务器进入 online 模式，且辅助数据库服务器进入 read-only 模式。如果连接不成功，则数据库服务器进入 online 模式，但不开启数据复制。

secondary 和 dbserver 参数设置数据库服务器类型为 secondary，并试图与 dbserver 指定的数据库服务器连接。如果连接成功，则开启数据复制。主数据库服务器成为 online，且辅助数据库服务器进入 read-only 模式。如果连接不成功，则数据库服务器进入 read-only 模式，但不开启数据复制。

这个函数等同于 gadmin -d 命令。

示例

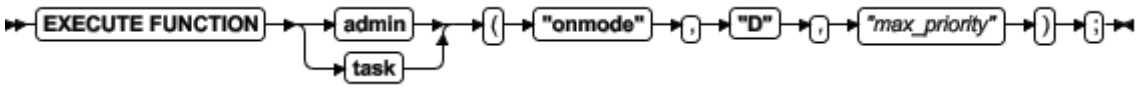
下列示例设置名为 ids_stores 的服务器为 HDR 主服务器：

```
EXECUTE FUNCTION task("gadmin","d","primary","ids_stores");
```

4.93 gadmin 和 D 参数：设置 PDQ 优先级（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 D 参数来临时地重置数据库服务器可分配给任何一个决策支持查询的 PDQ 资源。

语法



元素	描述	关键考虑
max_priority	实际分配给查询的用户请求的 PDQ 资源的百分率。	该值必须是从 0 至 100 的无符号整数。

用法

当数据库服务器处于 online 时，使用这个函数来覆盖 MAX_PDQPRIORITY 配置参数设置的限制。新的值仅影响数据库服务器的当前实例；这些值不记录在 onconfig 文件中。 如果您关闭并重启数据库服务器，这些参数的值恢复为 onconfig 文件中的值。

这个函数等同于 gadmin -D 命令。

示例

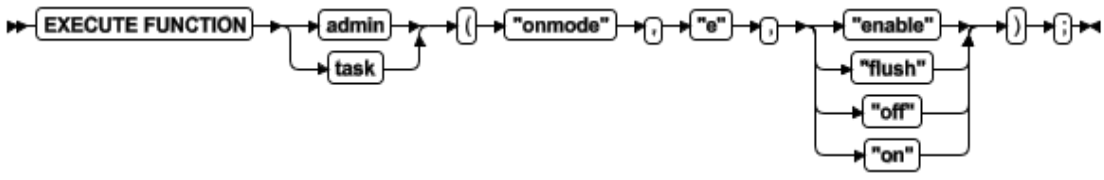
下列示例设置可分配给查询的 PDQ 资源的百分率为 50%：

```
EXECUTE FUNCTION task("gadmin","D","50");
```

4.94 gadmin 和 e 参数：更改 SQL 语句高速缓存的用法（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 e 参数来临时地更改 SQL 语句高速缓存的模式。

语法



用法

如果禁用了 SQL 语句高速缓存，则使用 enable 参数来启用它。仅在执行下列活动之后，单个用户会话可使用该语句高速缓存：

- 设置环境变量 STMT_CACHE 为 1。
- 执行 SQL 语句 SET STATEMENT CACHE ON。

使用 flush 参数来从 SQL 语句高速缓存清空不使用的语句，SQL 语句高速缓存保持启用。清空高速缓存之后，gstat -g ssc ref_cnt 域显示为 0。

使用 off 参数来关闭 SQL 语句高速缓存，以便不高速缓存语句。

使用 on 参数来高速缓存所有语句，除了那些用户通过下列活动之一关闭的之外：

- 使用这个命令来指定 OFF 模式。
- 设置环境变量 STMT_CACHE 为 0。
- 执行 SQL 语句 SET STATEMENT CACHE OFF。

这个函数不可更改 ONCONFIG 文件中 STMT_CACHE 配置参数的设置，但最后的参数覆盖那个设置（或如果未设置 STMT_CACHE，则为缺省值）。您以这个命令进行的任何语句高速缓存行为的变更都仅影响当前数据库服务器会话。 当您重启数据库服务器时，数据库服务器使用 ONCONFIG 文件中 STMT_CACHE 参数的设置。如果在 ONCONFIG 文件中未定义 STMT_CACHE 配置参数，则服务器不使用语句高速缓存。

这个函数等同于 gadmin -e 命令。

示例

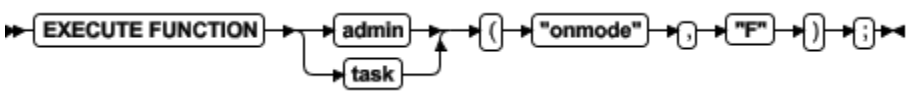
下列示例启用 SQL 语句高速缓存：

```
EXECUTE FUNCTION task("gadmin","e","enable");
```

4.95 gadmin 和 F 参数：释放不用的内存段（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 F 参数来释放不用的内存段。

语法



用法

当您执行这个函数时，内存管理器检测不用的内存的每一内存池。内存管理器立即释放定位的不用的内存块。内存管理器检查每一内存池之后，它开始检查内存段并释放数据库服务器不再需要的任何内存段。

当您执行该使用程序时，运行这个命令对任何用户都会导致重大的性能降级。虽然执行时间短暂（1 至 2 秒），但对单用户数据库服务器的降级可达 100%。有多个 CPU 虚拟处理器的系统感觉降级会成比例地减轻。

要确认不用的内存已释放，请检查消息日志。如果内存管理器释放一个或多个段，它显示消息表明释放了多少个段以及多少内存字节。

提示： 请从操作系统日程安排工具有规律地运行这个命令，并在数据库服务器执行任何创建更多的内存段的函数，包括构建大索引、排序或备份之后，运行这个命令。

这个函数等同于 gadmin -F 命令。

示例

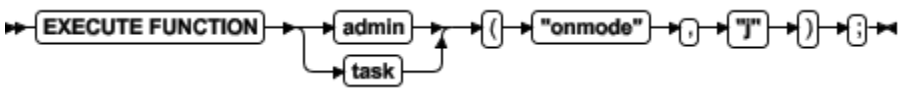
下列示例释放不用的内存块：

```
EXECUTE FUNCTION task("gadmin","F");
```

4.96 gadmin 和 j 参数：切换数据库服务器到管理模式（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 j 参数来更改数据库服务器为管理模式。

语法



用法

当服务器变更到管理模式时，除了下列用户的会话之外，所有会话失去到数据库服务器的连接：

- 用户 gbasedbt
- DBSA 组中的用户
- ADMIN_MODE_USERS 设置中标识的用户

这个函数等同于 gadmin -j 命令。

示例

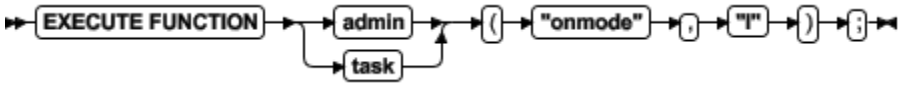
下列示例变更服务器到管理模式：

```
EXECUTE FUNCTION task("gadmin","j");
```

4.97 gadmin 和 l 参数：切换到下一个逻辑日志（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 l 参数来从当前逻辑日志文件切换到下一个逻辑日志文件。

语法



用法

这个函数等同于 gadmin -l 命令。

要了解关于切换到下一个逻辑日志文件的信息，请参阅 GBase 8s 管理员指南 中关于管理逻辑日志文件的章节。

示例

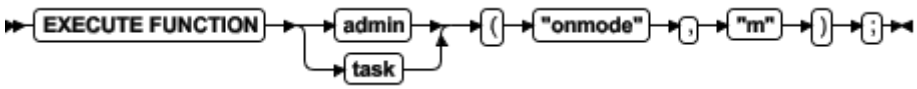
下列示例将逻辑日志移出 root chunk：

```
SELECT task("gadmin", "l") FROM sysmaster:syslogfil
        WHERE chunk = 1 AND sysmaster:bitval(flags,"0x02")>0;
```

4.98 gadmin 和 m 参数：切换到多用户模式（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 m 参数来更改数据库服务器到多用户模式。

语法



用法

使用这个函数来使数据库服务器从 quiescent 模式或从管理模式改为 online。

这个函数等同于 gadmin -m 命令。

示例

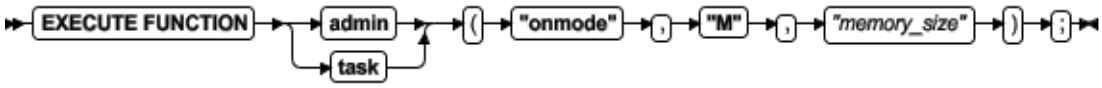
系列示例更改服务器到多用户模式：

```
EXECUTE FUNCTION task("gadmin","m");
```

4.99 gadmin 和 M 参数：临时地更改决策支持内存（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 M 参数来临时地更改并行查询的可用内存的大小。

语法



元素	描述	关键考虑
memory_size	并行查询可用的内存最大量的新大小限制（以 KB 为单位）。	32 位平台的最大值是 2 GB。 64 位平台的最大值是 4 GB。

用法

在数据库服务器处于 online 时，使用这个函数来覆盖 DS_TOTAL_MEMORY 配置参数设置的限制。新的值仅影响数据库服务器的当前实例；这些值不记录在 ONCONFIG 文件中。 如果您关闭并重启数据库服务器，则参数的值恢复为 ONCONFIG 文件中的值。

这个函数等同于 gadmin -M 命令。

示例

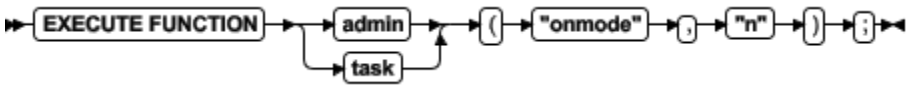
下列示例设置并行查询的大小限制为 50 MB：

```
EXECUTE FUNCTION task("gadmin","M","50000");
```

4.100 gadmin 和 n 参数：解锁驻留内存（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 n 参数来终结共享内存驻留部分的强制驻留。

语法



用法

在您可运行这个函数之前，RESIDENT 配置参数必须在 ONCONFIG 文件中设置为 1。

这个函数不影响 ONCONFIG 文件中 RESIDENT 配置参数、forced-residency 参数的值。

这个函数等同于 gadmin -n 命令。

示例

下列示例解锁驻留内存：

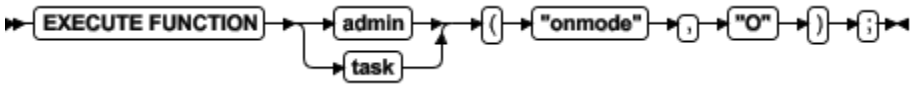
```
EXECUTE FUNCTION task("gadmin","n");
```

4.101 gadmin 和 O 参数：标记禁用的 dbspace

为 down（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 0 参数来标记禁用的 dbspace 为 down，以便禁用的 dbspace 正阻塞的检查点可继续且释放任何被阻塞的线程。

语法



用法

这个函数覆盖 ONDBSPACEDOWN 配置参数的 WAIT 模式。仅在下列情况下使用这个命令：

- ONDBSPACEDOWN 设置为 WAIT。
- 发生禁用的 I/O 错误，导致数据库服务器阻塞所有正在更新的线程。
- 你不可或不想纠正导致禁用的 I/O 错误的问题。
- 您想要数据库服务器标记禁用的 dbspace 为 down 并继续处理。

这个函数等同于 gadmin -O 命令。

示例

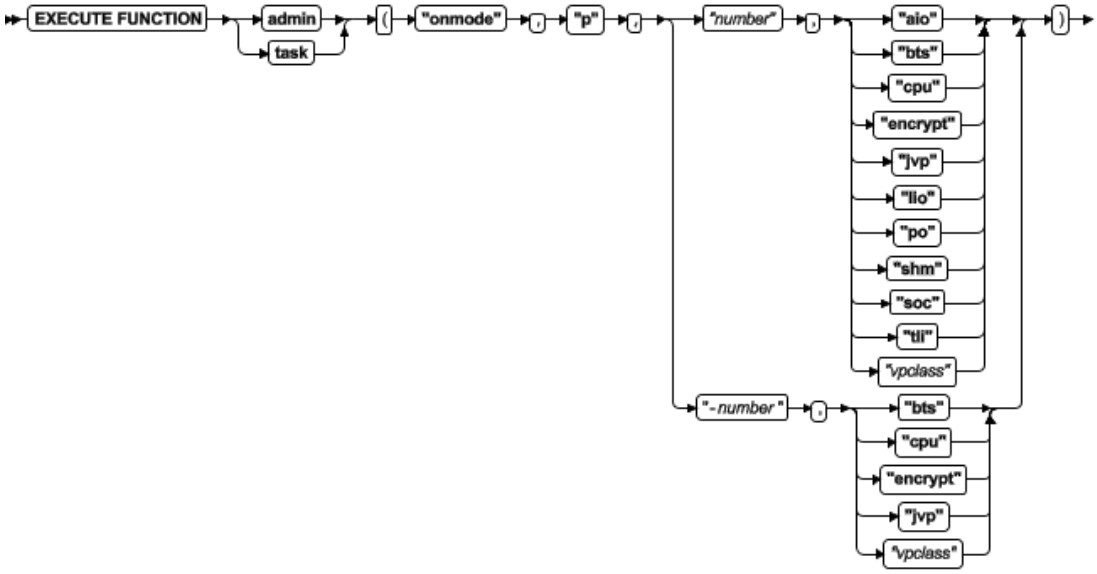
下列示例标记禁用的 dbspaces 为 down：

```
EXECUTE FUNCTION task("gadmin","O");
```

4.102 gadmin 和 p 参数：添加或移除虚拟处理器（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 p 参数来动态地为当前数据库服务器会话添加或移除虚拟处理器。这个函数不更新 onconfig 文件。

语法



元素	描述	关键考虑
number	要添加或移除的虚拟处理器的数目。	正的数目添加虚拟处理器。您可添加的虚拟处理器的最大数目依赖于操作系统。 UNIX™：负的数目移除虚拟处理器。要删除的虚拟处理器数不可超过指定类型的处理器的实际数目。
vpclass	用户定义的虚拟处理器类的名称。	Windows™：*number* 参数必须设置为 1，因为您仅可创建用户定义的虚拟处理器的一个实例。

用法

仅当数据库服务器处于 online 模式时，您才可使用这个函数。

CPU VP 的数目不应超过您的系统上物理处理器的数目，但如果超过了也不报错。数据库服务器使用 CPU VP 的数目来为并行数据库查询（PDQ）分配资源。 如果您删除 CPU VP，则您的查询可能运行得明显变慢。在您更改 CPU VP 的数目之后，从命令 gstat -g mgm 输出中的 Reinit 域显示有多少查询正在等待其他查询完成。

要了解更多关于 CPU VP 类的性能影响信息，请参阅 GBase 8s 性能指南。

要了解每一虚拟处理器类的描述，请参阅 GBase 8s 管理员指南。

这个函数等同于 gadmin -p 命令。

示例

下列示例添加一个 CPU 虚拟处理器：

```
EXECUTE FUNCTION task("gadmin","p","1","cpu");
```

下列示例移除一个 Java™ 虚拟处理器：

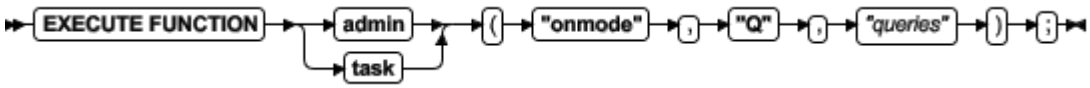
```
EXECUTE FUNCTION task("gadmin","p","-1","jvp");
```

[1] 仅限于 UNIX™

4.103 gadmin 和 Q 参数：设置决策支持查询的最大数目（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 Q 参数来更改当前正在执行的决策支持查询的最大数目。

语法



元素	描述	关键考虑
queries	当前正在执行的并行查询的最大数目。	该数目必须是从 1 至 8,388,608 的无符号整数。

用法

当数据库服务器处于 online 时，使用这个函数来覆盖 DS_MAX_QUERIES 配置参数设置的限制。新的值仅影响数据库服务器的当前实例；这些值不记录在 ONCONFIG 文件中。如果您关闭并重启数据库服务器，则该参数的值恢复为 ONCONFIG 文件中的值。

要了解用于控制 PDQ 参数的信息，请参阅 GBase 8s 性能指南。

这个函数等同于 gadmin -Q 命令。

示例

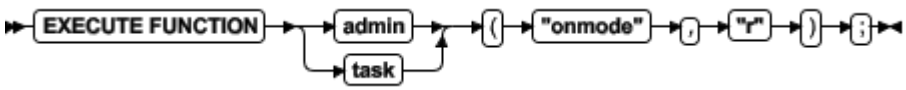
下列示例设置当前正在执行的并行查询的最大数目为 8：

```
EXECUTE FUNCTION task("gadmin","Q","8");
```

4.104 gadmin 和 r 参数：强制共享内存的驻留（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 r 参数来启动共享内存的驻留部分的强制驻留。

语法



用法

在您可运行这个函数之前，RESIDENT 配置参数必须在 ONCONFIG 文件中设置为 1。

这个函数不影响 ONCONFIG 文件中 RESIDENT 配置参数、forced-memory 参数的值。

这个函数等同于 gadmin -r 命令。

示例

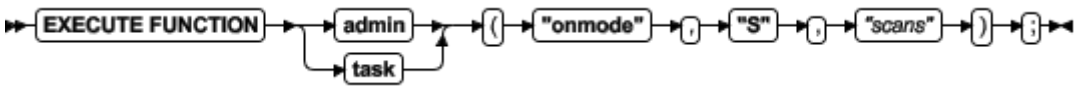
下列示例启动共享内存的强制驻留：

```
EXECUTE FUNCTION task("gadmin","r");
```

4.105 gadmin 和 S 参数：设置决策支持扫描的最大数目（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 S 参数来更改当前正在执行的当期会话决策支持扫描的最大数目。

语法



元素	描述	关键考虑
scans	当前正在执行的并行扫描的最大数目。	该数目必须是从 10 至 1 048 576 的无符号整数。

用法

在数据库服务器处于 online 时，使用这个函数来覆盖 DS_MAX_SCANS 配置参数设置的限制。新的值仅影响数据库服务器的当前实例；这些值不记录在 ONCONFIG 文件中。如果您关闭并重启数据库服务器，则该参数的值恢复为 ONCONFIG 文件中的值。

要了解用于控制 PDQ 参数的信息，请参阅 GBase 8s 性能指南。

这个函数等同于 gadmin -S 命令。

示例

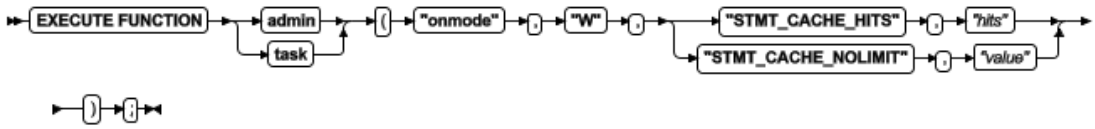
下列示例设置当前正在执行的并行扫描的最大数目为 2000：

```
EXECUTE FUNCTION task("gadmin","S","2000");
```

4.106 gadmin 和 W 参数：重置语句高速缓存属性（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 W 参数来更改语句是否以及何时可插入到 SQL 高速缓存。

语法



元素	描述	关键考虑
hits	在语句完全地插入到 SQL 语句高速缓冲之前，命中（引用）语句的数目。	可能的值为： ● 0 = 在高速缓存中插入所有满足条件的语句及其内存结构。 ● 1 或更大 = 排除 ad hoc 查询进入该高速缓存。
value	语句是否插入到 SQL 语句高速缓存中。	可能的值为： ● 0 = 数据库服务器不插入语句到高速缓存内。 ● 1 = 数据库服务器总是在高速缓存中插入语句。

用法

在数据库服务器处于 online 时，使用这个函数来重置 STMT_CACHE_HITS 或 STMT_CACHE_NOLIMIT 配置参数的值。新的值仅影响数据库服务器的当前实例；该值不记录在 ONCONFIG 文件中。如果您关闭并重启数据库服务器，则参数的值恢复为 ONCONFIG 文件中的值。

如果您设置 STMT_CACHE_HITS 的值等于 0，则数据库服务器在该高速缓存中插入所有满足条件的语句及其内存结构。如果该值大于 0 且 SQL 语句已被执行的次数小于 STMT_CACHE_HITS 的值，则数据库服务器在高速缓存中插入 key-only 条目。在语句已经发生指定的命中数之后，数据库服务器在高速缓存中插入满足条件的语句。新的 STMT_CACHE_HITS 值显示在 gstat -g ssc 输出的 #hits 域中。

如果未分享任何查询，则设置 STMT_CACHE_NOLIMIT 为 0 来阻止数据库服务器为语句高速缓存分配大量内存。

这个函数等同于 gadmin -W 命令。

示例

下列示例阻止 ad hoc 查询进入 SQL 语句高速缓存：

```
EXECUTE FUNCTION task("gadmin","W","STMT_CACHE_HITS","1");
```

4.107 gadmin 和 wf 参数：永久地更新配置参数

（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 wf 参数来动态地更新 onconfig 文件中配置参数的值。

语法

元素	描述	关键考虑
configuration_parameter_name	配置参数的名称。	该配置参数必须是您可动态地更新的。 您可动态地更新的配置参数列表与 **gadmin -wf** 命令的相同。
new_value	配置参数的一个或多个新值。	该值对于配置参数必须是有效的。 新值的格式必须严格地遵循那个配置参数的语法。

用法

使用这个函数来永久地更新配置参数的值。新的值立即生效并在服务器重新启动之后保存在 ONCONFIG 文件中。

这个函数等同于 gadmin -wf 命令。

示例

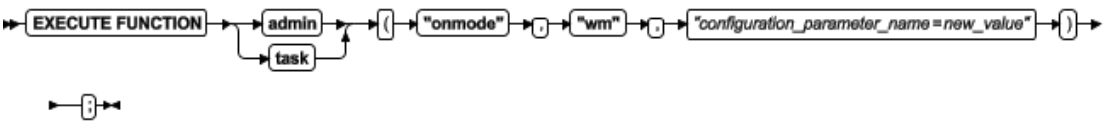
下列示例设置 onconfig 文件中 DYNAMIC_LOGS 配置参数的值为 2：

```
EXECUTE FUNCTION task("gadmin","wf","DYNAMIC_LOGS=2");
```

4.108 gadmin 和 wm 参数：临时地更新配置参数（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 wm 参数来动态地更新内存中配置参数的值。

语法



元素	描述	关键考虑
configuration_parameter_name	配置参数的名称。	您指定的配置参数必须是您可动态地更新的。 您可动态地更新的配置参数列表与 **gadmin -wf** 命令的相同。
new_value	配置参数的一个或多个新值。	该值对于配置参数必须是有效的。 新值的格式必须严格地遵循那个配置参数的语法。

用法

使用这个函数来临时地更新您可动态地更新的配置参数的值。新值立即生效。新的值不写到 ONCONFIG 文件且当数据库服务器重启时丢失。

这个函数等同于 gadmin -wm 命令。

示例

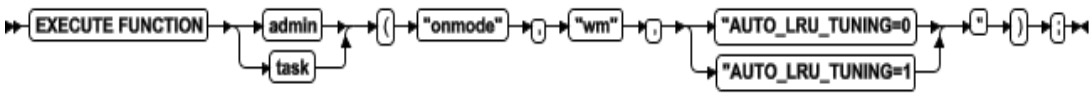
下列示例为当前会话设置 DYNAMIC_LOGS 配置参数的值为 2：

```
EXECUTE FUNCTION task("gadmin","wm","DYNAMIC_LOGS=2");
```

4.109 gadmin、wm 和 AUTO_LRU_TUNING 参数：更改 LRU 调整状态（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin、wm 和 AUTO_LRU_TUNING 参数来更改 LRU 调整状态而不更新 onconfig 文件。

语法



用法

使用 AUTO_LRU_TUNING=1 参数来启用自动的 LRU 调整。

使用 AUTO_LRU_TUNING=0 参数来禁用自动的 LRU 调整。

这个函数等同于 gadmin -wm AUTO_LRU_TUNING 命令。

示例

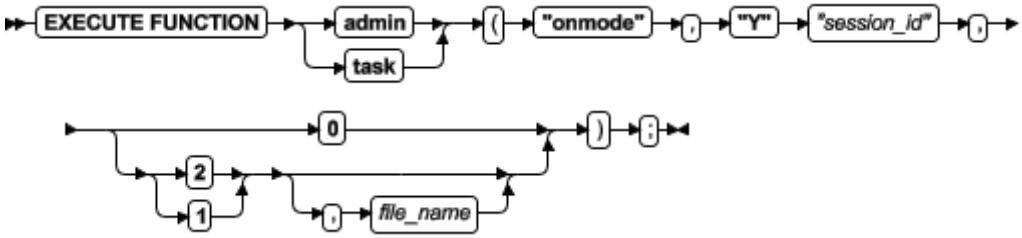
下列示例启用自动的 LRU 调整：

```
EXECUTE FUNCTION task("gadmin","wm","AUTO_LRU_TUNING=1");
```

4.110 gadmin 和 Y 参数：更改会话的查询计划度量（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 Y 参数来更改对个别会话查询计划度量的输出。

语法



元素	描述	关键考虑
file_name	说明输出文件名。	如果未包括该文件的绝对路径，则在缺省示例输出文件位置中创建示例输出文件。如果文件已经存在，说明输出附加到该文件。如果从 SET EXPLAIN 语句已经存在一个文件，则不再使用那个文件直到关闭动态的说明。
session_id	标识特定的会话。	无。
-Y	动态地更改 SET EXPLAIN 语句的值。	无。

用法

您可使用这个函数来与 SET EXPLAIN 语句竞争。

最后的参数确定是否记录查询度量，包括查询优化器的计划、对返回行数的估计以及查询的相对成本。

使用 2 参数来使得数据库服务器能发送查询计划到说明输出文件。

使用 1 参数来使得数据库服务器能发送查询计划和统计到说明输出文件。这个设置等同于特定会话的 SET EXPLAIN ON 语句。

使用 0 参数来使得查询度量不能输出到当前会话的说明输出文件。这个设置等同于 SET EXPLAIN OFF 语句。

这个函数等同于 gadmin -Y 命令。

示例

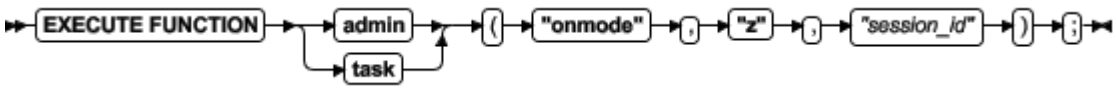
下列示例使得 ID 为 32 的用户会话不能输出查询度量：

```
EXECUTE FUNCTION task("gadmin","Y","32","0");
```

4.111 gadmin 和 z 参数：终止用户会话（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 z 参数来终止指定的用户会话。

语法



元素	描述	关键考虑
session_id	会话 ID。	该值必须为大于 0 的无符号整数，且必须为当前正在运行的会话的会话标识号。

用法

这个函数等同于 gadmin -z 命令。

示例

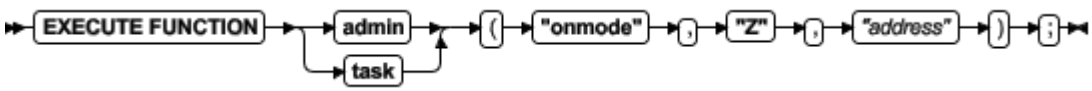
下列示例终止 ID 为 14 的用户会话：

```
EXECUTE FUNCTION task("gadmin","z","14");
```

4.112 gadmin 和 Z 参数：终止分布式事务（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gadmin 和 Z 参数来终止指定的分布式事务。仅当参与的数据库服务器之间已失去通信才使用这个函数。 如果应用正在执行分布式事务，则终止分布式事务之一可令客户端/服务器数据库系统处于不一致的状态。

语法



元素	描述	关键考虑
address	与分布式事务相关联的共享内存地址。	这必须是正在进行的分布式事务的地址，该事务已超过了 TXTIMEOUT 配置参数指定的时间量。 *address* 必须符合特定操作系统对共享内存寻址

元素	描述	关键考虑
		的规则。这个地址可从 gstat -x 输出获取。

用法

仅当该分布式事务已超过了 TXTIMEOUT 配置参数指定的时间量，这个函数才会成功。

这个函数等同于 gadmin -Z 命令。

示例

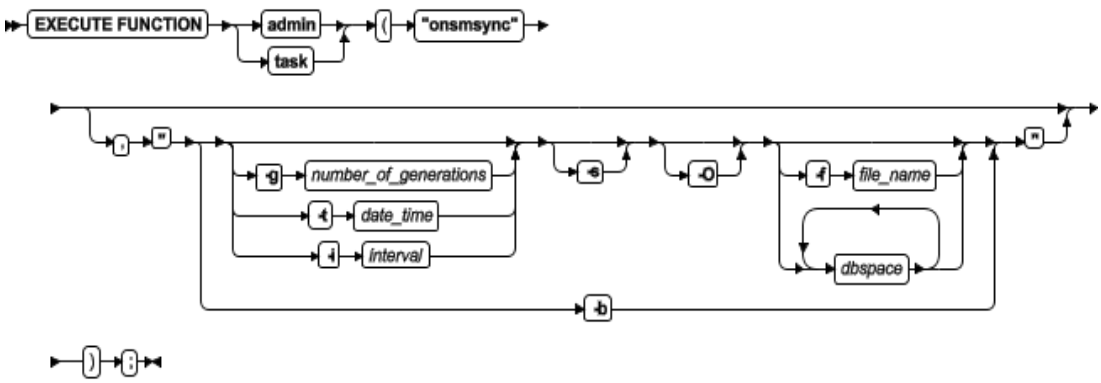
下列示例终止地址为 0xa509018 的分布式事务：

```
EXECUTE FUNCTION task("gadmin","Z","0xa509018");
```

4.113 onsmsync 参数：与存储管理器目录同步（SQL 管理 API）

随同 admin() 或 task() 函数，使用 onsmsync 参数来将 sysutils 数据库和紧急 boot 文件与存储管理器目录同步。

语法



元素	描述	关键考虑
无选项	将 sysutils 数据库和紧急 boot 文件与存储管理器目录同步。	无。
-b	从彼此重新生成紧急 boot 文件（ixbar.servernum）和 sysutils 数据库。	如果 ixbar 文件为空或不存在，则 onsmsync -b 重新创建 ixbar 文件并从 syutils 表填入它。 如果 ixbar 不为空且包含对象数据，则 onsmsync -b 更新 sysutils 数据库和 ixbar 文件，以便它们同步。

元素	描述	关键考虑
		如果 ixbar 文件有条目且 **sysutils** 数据库已重构，但因为它不包含文件而为空，则 onsmsync -b 选项重新从 ixbar 创建 **sysutils** 数据。 -b 元素不与其他 onsmsync 选项一起使用。另外，它不与存储管理器同步。
dbspace	指定一个或多个要终止的存储空间	如果您输入多于一个存储空间，则请使用空格来分隔名称。
-f *file_name*	指定包含要终止的存储空间列表的文件的路径名	使用这个选项来避免键入一个存储空间的长列表。文件名可为任何有效的 UNIX™ 或 Windows™ 文件名。
-g *number_of_generations*	保留每一 0 级备份的一定数目的版本	保留备份的最近一版，且终止所有的较早版本。
-i *interval*	终止早于某些时间期间的所有备份	保留晚于这个间隔的备份。如果需要从那个间隔之后的其他备份恢复，则不终止早于间隔的备份。使用 *interval* 的 ANSI 或 GLS 格式：YYYY-MM 或 DD HH:MM:SS
-s	跳过存储管理器已终止的备份	使用这个选项来跳过与那些从存储管理器已经终止的对象的同步。如果提供 -s 选项，则会根据其他的参数终止对象。
-O	严格地实施终止策略	如果随同 -t、-g 或 -i 选项一起使用，则终止所有级别的备份，即使其中的一些需要从终止日期之后发生的备份来恢复。-O 选项不影响逻辑日志终止。
-t *date_time*	终止特定时间和日期之前的所有备份	保留晚于这个 *datetime* 的备份。如需要从晚于那个 *datetime* 的其他备份来恢复，则不终止早于 *datetime* 的备份。请使用 *datetime* 的 ANSI 或 GLS_DATETIME 格式。

用法

这个函数调用 onsmsync 实用程序来将 sysutils 数据库和紧急 boot 文件与存储管理器目录同步。

示例

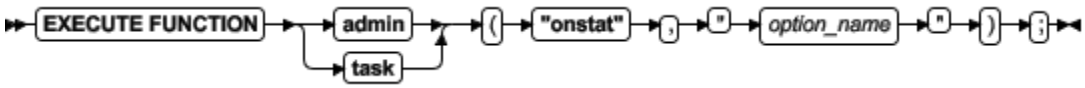
下列示例调用 onsmsync 实用程序并指定保留的备份数目为 1 且终止所有较早的备份版本：

```
EXECUTE FUNCTION task("onsmsync", "-g 1");
```

4.114 gstat 参数：监视数据库服务器（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gstat 参数来监视数据库服务器。

语法



元素	描述	关键考虑
option_name	gstat 命令选项。	该选项必须包括一连字符且符合 gstat 选项语法。要了解 gstat 选项，请参阅 gstat 实用程序。

用法

使用这些命令来显示与运行 gstat 实用程序命令相同的关于数据库服务器的信息。

示例

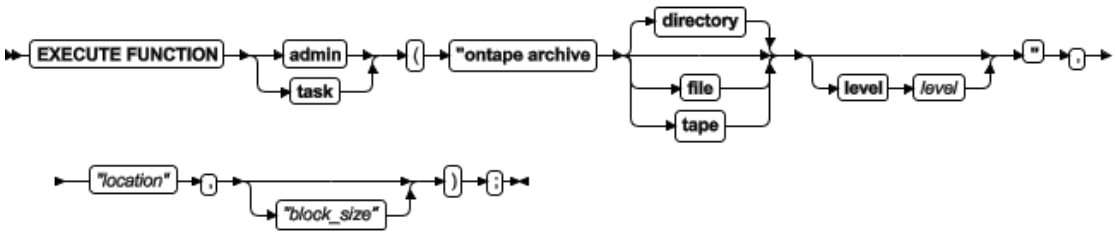
下列示例运行 gstat -g ses 命令：

```
EXECUTE FUNCTION task("gstat","-g ses");
```

4.115 gtape archive 参数：备份数据库上的数据（SQL 管理 API）

随同 admin() 或 task() 函数，使用 gtape archive 参数来创建数据库数据的备份。

语法



元素	描述	关键考虑
level	有效的级别为：0、1 或 2。缺省值为 0。	

元素	描述	关键考虑
location	到文件或目录或磁带设备的路径	
block_size	在存储空间备份期间，gtape 写到的设备的块大小	缺省块大小为 512 KB。

用法

这个函数调用 gtape 实用程序来创建备份。

从备份的位置您有三种设备可选：

目录或 dir

现有的目录路径。缺省的备份设备。

file

现有的文件。

tape

现有的磁带设备。

示例

这个函数在目录路径 /local/gbasedbt/backup/ 中创建 0 级归档：

```
EXECUTE FUNCTION task("gtape archive","/local/gbasedbt/backup/");
```

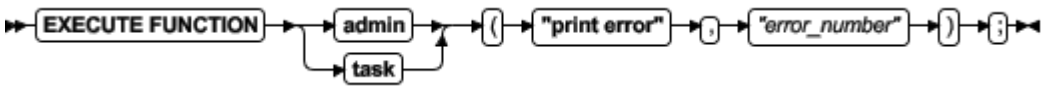
这个函数在目录路径 /local/gbasedbt/backup/ 中创建 0 级归档，块大小为 256 KB：

```
EXECUTE FUNCTION task("gtape archive directory level 0",
"/local/gbasedbt/backup/","256");
```

4.116 print error 参数：打印错误消息（SQL 管理 API）

随同 admin() 或 task() 函数，使用 print error 参数来打印与指定的错误号相关联的消息。

语法



元素	描述	关键考虑
error_number	错误号，无减号。	*error_number* 必须是现有的错误号。

用法

这个函数等同于 finderr 实用程序。

示例

下列示例打印错误号 -105 的消息文本：

```
EXECUTE FUNCTION task("print error","105");
 (expression)  ISAM error:  bad isam file format.
```

4.117 print file info 参数：显示目录或文件信息（SQL 管理 API）

随同 admin() 或 task() 函数，使用 print file info 参数来显示关于目录或文件的信息。

语法

元素	描述	关键考虑
file_path	到目录或文件的路径	

示例：文件信息

下列示例显示您要用来打印关于 /tmp 目录中 x 文件的信息的参数：

```
execute function task("print file info","/tmp/x");
```

返回下列信息：

```
(expression)  File name           = /tmp/x
              Is File             = 1
              Is Directory        = 0
              Is Raw Device       = 0
              Is Block Device     = 0
              Is Pipe             = 0
              File Size           = 554
              Last Access Time    = 11/29/2010 21:55:02
              Last Modified Time  = 11/29/2010 21:51:45
              Status Change Time  = 11/29/2010 21:51:45
              User Id             = 200
              Group id            = 102
              File Flags          = 33206
```

示例：目录信息

下列示例显示您要用来打印关于 /tmp 目录的信息的参数：

```
execute function task("print file info","/tmp");
```

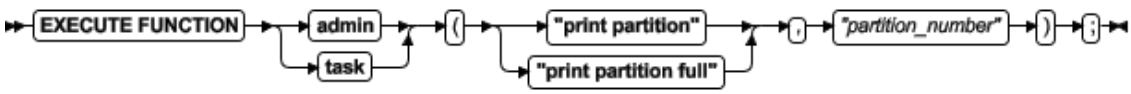
返回下列信息：

```
(expression)  File name          = /tmp
              Is File            = 0
              Is Directory       = 1
              Is Raw Device      = 0
              Is Block Device    = 0
              Is Pipe            = 0
              File Size          = 32768
              Last Access Time   = 12/06/2010 11:53:00
              Last Modified Time = 12/06/2010 12:05:53
              Status Change Time = 12/06/2010 12:05:53
              User Id            = 0
              Group id           = 0
              File Flags         = 17407
```

4. 118 print partition 参数：打印分区信息（SQL 管理 API）

随同 admin() 或 task() 函数，使用 print partition 参数来打印指定分区的头部。

语法



元素	描述	关键考虑
partition_number	分区号。	请在 systables 系统目录表的 partnum 列中找到分区号。

用法

使用这个函数来打印指定分区的 tblspace 报告。

运行带有 full 参数的这个函数来包括按 dbspace 的页类型排列的特定索引信息和页分配信息。

带有 print partition 参数的这个函数等同于 gcheck -pt 命令。

带有 print partition full 参数的这个函数等同于 gcheck -pT 命令。

示例

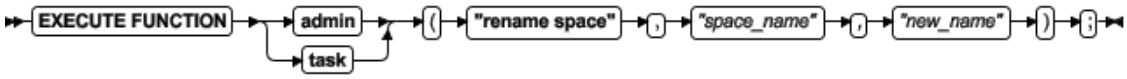
下列示例打印 1048611 号分区的头部：

```
EXECUTE FUNCTION task("print partition","1048611");
```

4. 119 rename space 参数：重命名存储空间（SQL 管理 API）

随同 admin() 或 task() 函数，使用 rename space 参数来重命名 dbspace、blobspace、sbspace 或 extspace。

语法



元素	描述	关键考虑
new_name	空间的新名称。	
space_name	您想要重重名的 dbspace、blobspace、sbspace 或 extspace 的名称。	

用法

这个函数等同于 gspaces -ren 命令。

示例

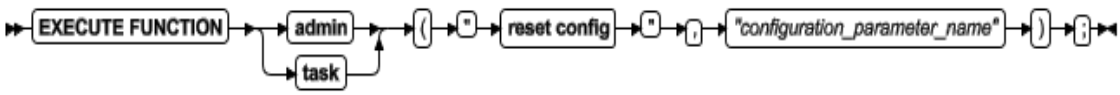
下列示例将名为 dbsp1 的 dbspace 重命名为 dbsp2：

```
EXECUTE FUNCTION task("rename space","dbsp1","dbsp2");
```

4.120 reset config 参数：恢复配置参数值（SQL 管理 API）

随同 admin() 或 task() 函数，使用 reset config 参数来将动态地可更新的配置参数值恢复为 onconfig 文件中的值。动态地可更新的配置参数是您可用 gadmin 或 SQL 管理 API 命令为会话变更的那些参数。

语法



元素	描述	关键考虑
configuration_parameter_name	您想要恢复其值的配置参数的名称。	

用法

reset config 参数将配置参数的值恢复为 onconfig 文件中最后保存的值，即使在数据库服务器启动之后更改了该值。

示例

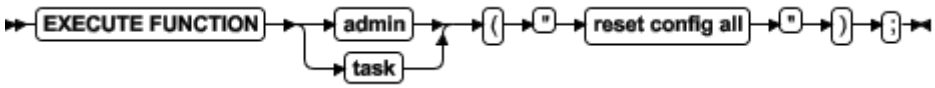
下列命令将 DYNAMIC_LOGS 配置参数的值恢复为 onconfig 文件中的值。

```
EXECUTE FUNCTION task("reset config","DYNAMIC_LOGS");
```

4.121 reset config all 参数：恢复所有动态地可更新的配置参数值（SQL 管理 API）

随同 admin() 或 task() 函数，使用 reset config all 参数来将所有动态地可更新的配置参数的值恢复到 onconfig 文件中的值。动态地可更新的配置参数是您可以 gadmin 或 SQL 管理 API 命令为会话更改的那些参数。

语法



用法

reset config all 参数将所有动态地可更新的配置参数的值恢复为 onconfig 文件中最后保存的值，即使在数据库服务器启动之后更改了这些值。

示例

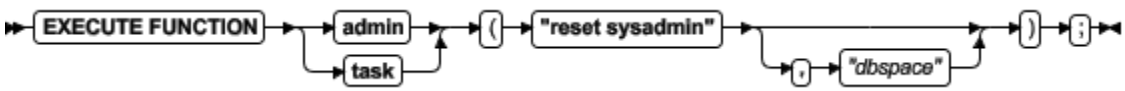
下列命令恢复所有动态地可调整的配置参数的值。

```
EXECUTE FUNCTION task("reset config all");
```

4.122 reset sysadmin 参数：移动 sysadmin 数据库（SQL 管理 API）

随同 admin() 或 task() 函数，使用 reset sysadmin 参数来将 sysadmin 数据库移到指定的 dbspace。移动 sysadmin 数据库将该数据库重置回到首次创建它时的原始状态；所有数据、command history 和结果表都会丢失。仅内建任务、传感器和阈值保留在 sysadmin 表中。

语法



元素	描述	关键考虑
dbspace	dbspace 的名称。	

用法

这个函数没有等同的实用程序命令。

如果您未指定 dbspace 作为最后的参数，则这个命令删除 sysadmin 数据库，然后在 rootdbs 中重新创建它。删除所有 ph_* 表和 command_history 行，并删除所有结果表。

示例

下列示例删除现有的 sysadmin 数据库并在名为 dbsp1 的 dbspace 内创建新的 sysadmin 数据库：

```
EXECUTE FUNCTION task("reset sysadmin","dbs1");
```

下一个示例删除 sysadmin 数据库，然后在 rootdbs 中重新创建它。

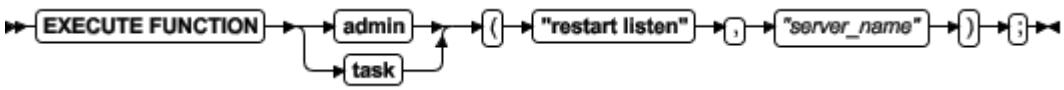
```
EXECUTE FUNCTION admin("reset sysadmin");
```

除了内建任务、传感器和阈值之外，从 ph_ 表中删除所有数据行，通过这个函数调用从 sysadmin 删除所有结果表。该函数执行完成之后，command_history 表为空。

4. 123 restart listen 参数：动态地停止并启动监听线程（SQL 管理 API）

随同 admin() 或 task() 函数，使用 restart listen 参数来停止然后启动现有的 SOCTCP 或 TLITCP 网络协议的监听线程，而不中断现有的连接。

语法



元素	描述	关键考虑
server_name	您想要停止并重启监听线程的数据库服务器的名称。	

用法

必须在 sqlhosts 文件中存在监听线程的定义。

如有必要，在您重启监听线程之前，请修订 sqlhosts 条目。例如，如果正在运行的监听线程一定为端口 7777，则您可更改 sqlhosts 文件中的端口，然后重启该线程。

这个函数等同于 gadmin -P restart server_name 命令。

这个函数不更新 sqlhosts 文件。

示例

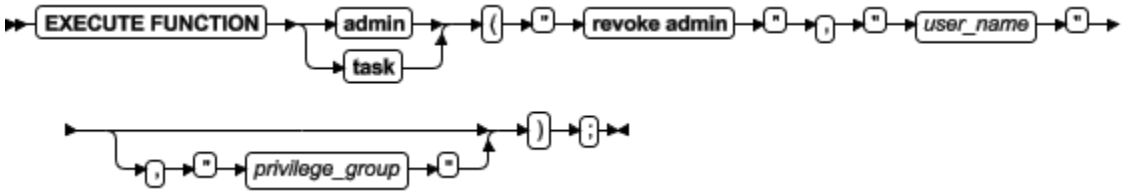
下列命令停止并再启动名为 ids_serv1 服务器的监听线程：

```
EXECUTE FUNCTION task("restart listen","ids_serv1");
```

4. 124 revoke admin 参数：撤销运行 SQL 管理 API 命令的权限

随同 admin() 或 task() 函数，使用 revoke admin 参数来撤销运行 SQL 管理 API 命令的权限。

语法



元素	描述	关键考虑
user_name	要取消其权限的用户名。	
privilege_group	权限组名。	请参阅 SQL 管理 API 门户：按权限组划分参数 查看权限组的列表。

用法

仅用户 gbasedbt 或者对于 SQL 管理 API 命令有 ADMIN 或 GRANT 权限的用户，可使用 revoke admin 参数。

使用 revoke admin 参数来撤销个别用户运行 SQL 管理 API 命令的权限。您可撤销特定权限组的权限，或撤销所有权限。

示例

下列命令撤销用户 Bob 运行备份和恢复 SQL 管理命令的权限：

```
EXECUTE FUNCTION task("revoke admin", "Bob", "BAR");
```

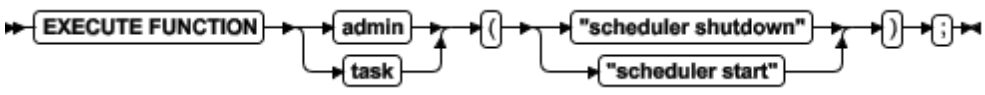
下列命令撤销用户 Bob 运行任何 SQL 管理命令的所有权限：

```
EXECUTE FUNCTION task("revoke admin", "Bob");
```

4.125 scheduler 参数：停止或启动调度程序（SQL 管理 API）

随同 admin() 或 task() 函数，使用 scheduler 参数来启动或停止调度程序。

语法



用法

使用 scheduler shutdown 参数来停止调度程序并重新分配它的资源。

使用 scheduler start 参数来启动调度程序。

这个函数没有等同的实用程序命令。

您可用 gstat -g dbc 命令查看调度线程的状态。

示例

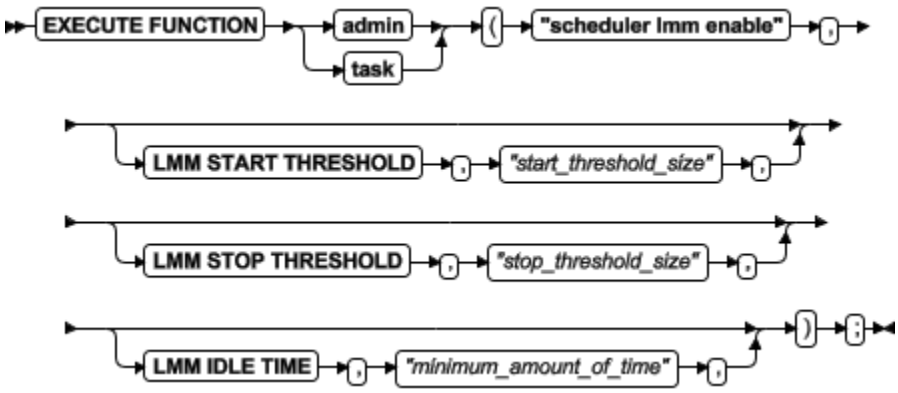
在调度程序已关闭之后，下列示例启动调度程序：

```
EXECUTE FUNCTION task("scheduler start");
```

4.126 scheduler lmm enable 参数：指定自动的低内存管理设置（SQL 管理 API）

随同 admin() 或 task() 函数，使用 scheduler lmm enable 参数来启动自动的低内存管理并更新低内存阈值设置。

语法



元素	描述	关键考虑
start_threshold_size	您想要数据库服务器保持的空闲内存量。如果内存量少于 *start_threshold_size*，则服务器自动地释放内存并终止应用。	该值可表示为 SHMTOTAL 配置参数值的百分率或特定数量。如果该值小于 50，则认为是百分率。输入参数的结果值必须大于 5 MB 且小于 95 MB。 缺省值为 5 MB。 在 LMM START THRESHOLD 与 LMM STOP THRESHOLD 之间必须至少差 5 MB
stop_threshold_size	在服务器停止自动释放内存和终止应用之前，您想要数据库服务器拥有的空闲内存量。	该值可表示为 SHMTOTAL 配置参数值的百分率或特定数量。如果该值小于 50，则认为是百分率。输入参数

元素	描述	关键考虑
		的结果值必须大于 10 MB 且小于 100 MB。该值还必须至少比 LMM START THRESHOLD 多 5 MB。 缺省值为 10 MB。
minimum_amount_of_time	定义会话为空闲的时间量，以秒为单位	该值必须在 1 与 86400 之间。 缺省值为 300 秒。

用法

随同 admin() 或 task() 函数，您使用 scheduler lmm disable 参数来在主数据库服务器或标准数据库服务器中停止当前和后续的低内存管理进程。 当触发低内存管理时，数据库服务器按下列顺序执行这些任务：

1. 数据库服务器终止会话，从会话有最大的空闲时间量开始一次启动一个，如有必要继续到会话有最小空闲时间量，时间量超过 LMM IDLE TIME 设置中指定的数量。当达到 LMM STOP THRESHOLD 时，服务器停止终止会话。
2. 数据库服务器终止会话，从使用最多内存的会话开始，如有必要继续到使用最小内存量的会话，直到达到 LMM STOP THRESHOLD。
3. 通过设置 VP_MEMORY_CACHE 配置参数为 0，数据库服务器执行内存重新配置，并运行 gadmin -F 命令来释放不用的共享内存段。

当低内存管理操作完成时，通过将 VP_MEMORY_CACHE 配置参数设置回其原始值，低内存管理器返回到监视模式并恢复数据库服务器的内存配置。

数据库服务器在 ph_threshold 表中存储自动低内存管理设置。

您可用 gstat -g lmm 命令查看低内存管理设置和近期的活动。

注意： 如果您启用自动的低内存管理并配置数据库服务器来使用 SHMTOTAL 配置参数指定值的百分率作为启动和停止阈值，则当更改 SHMTOTAL 配置参数值时请使用警告。更改 SHMTOTAL 配置参数值可导致自动低内存管理的配置变得无效，强制 GBase 8s 使用缺省设置。

设置低内存管理阈值设置的示例

下列示例指定当数据库服务器有 10 MB 或更少的空闲内存时，服务器会启动自动低内存管理停止应用并释放内存。该示例还指定如果会话尚未运行 300 秒则认为会话空闲，该示例指定当服务器有 20 MB 或更多空闲内存时会停止自动低内存管理。

```
EXECUTE FUNCTION task("scheduler lmm enable",
        "LMM START THRESHOLD", "10MB",
        "LMM STOP THRESHOLD", "20MB",
        "LMM IDLE TIME", "300");
```

SHMTOTAL 配置参数影响低内存管理阈值设置的示例

假定您设置 SHMTOTAL 配置参数为 1000000（1000 MB 或 1 GB），LMM START THRESHOLD 为 2，且 LMM STOP THRESHOLD 为 3。因为任何小于 50 的值都是 SHMTOTAL 值的百分率，所以实际的 LMM START THRESHOLD 为 20000（20 MB）且实际的 LMM STOP THRESHOLD 为 30000（30 MB）。

当剩余空闲内存为 20 MB 或更少时，数据库服务器开始管理低内存，且当空闲内存量为 30 MB 或更多时，停止管理内存。

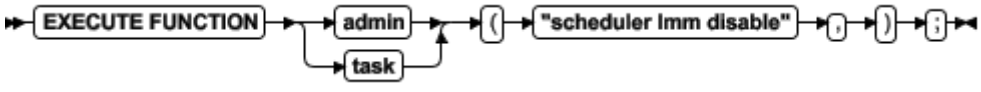
假定您决定更改 SHMTOTAL 配置参数的值，因为您知道现在不需要这么多内存，您想要让操作系统使用内存。请您设置 SHMTOTAL 的值为 250000（250 MB）。这更改实际的 LMM START THRESHOLD 为 5000（5 MB），实际的 LMM STOP THRESHOLD 为 7500（7.5 MB）。LMM STOP THRESHOLD 现在无效，因为 LMM START THRESHOLD 与 LMM STOP THRESHOLD 值之间必须至少差 5 MB。LMM STOP THRESHOLD 值还必须至少为 10 MB。

对于您的系统，你可能已经决定差 10 MB 是正确的数量。但是差 5 MB，数据库服务器可能会在低内存管理进程上花费太多的时间，这可能导致性能问题。

4.127 scheduler lmm disable 参数：停止自动的低内存管理（SQL 管理 API）

随同 admin() 或 task() 函数，使用 scheduler lmm disable 参数来停止当前和后续的自动低内存管理调用。

语法



用法

如果启用自动的低内存管理，则您可通过下列指定禁用它：

```
EXECUTE FUNCTION task("scheduler lmm disable");
```

请您随同 admin() 或 task() 函数，使用 scheduler lmm enable 参数来启动自动的低内存管理并更新阈值设置。

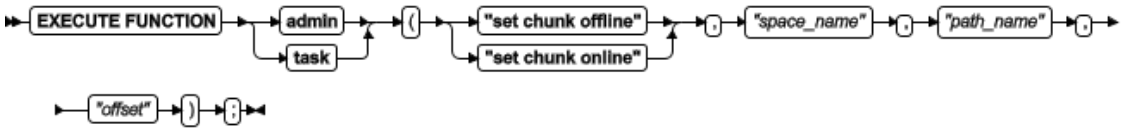
用 gstat -g lmm 命令，您可查看关于自动低内存管理设置的信息以及近期活动。

4.128 set chunk 参数：更改 chunk 的状态（SQL

管理 API）

随同 admin() 或 task() 函数，使用 set chunk 参数来更改 blobspace、dbspace 或 sbspace 的状态为 online 或 offline。

语法



元素	描述	关键考虑
space_name	blobspace、dbspace 或 sbspace 的名称。	
path_name	chunk 的磁盘分区或无缓冲的设备。	
offset	磁盘分区或无缓冲的设备内达到 chunk 的偏移量，以 KB 为单位。	请参阅 admin() 和 task() 参数大小规范。

用法

该 chunk 必须在镜像对中，或必须是非关键 dbspace 内的非主 chunk。

使用 set chunk offline 参数来更改 chunk 的状态为 offline。

使用 set chunk online 参数来更改 chunk 的状态为 online。

这个函数等同于 gspaces -s 命令。

示例

下列示例更改 chunk 的状态为 online：

```
EXECUTE FUNCTION task("set chunk online","dbs1","/dev/raw_dev1","0");
        Database selected.

        (expression)  Chunk status successfully changed.
        Chunk number 2 "/dev/raw_dev1" -- Online

        1 row(s) retrieved.
```

4. 129 set dataskip 参数：启动或停止跳过 dbspace（SQL 管理 API）

随同 admin() 或 task() 函数，使用 set dataskip 参数来指定数据库服务器在处理事务期间是否跳过不可用的 dbspace。

语法

元素	描述	关键考虑
dbspace	要开始或要停止跳过的 dbspace 名。	

用法

运行这个函数来更新 DATASKIP 配置参数的值，指定数据库服务器在处理事务的过程中是否跳过不可用的 dbspace（例如，由于介质故障）。

当指定的 dbspace 停止工作时，使用 set dataskip on 参数来开始跳过它。

使用 set dataskip off 参数来停止跳过指定的 dbspace。

这个函数等同于 gspaces -f 命令。

示例

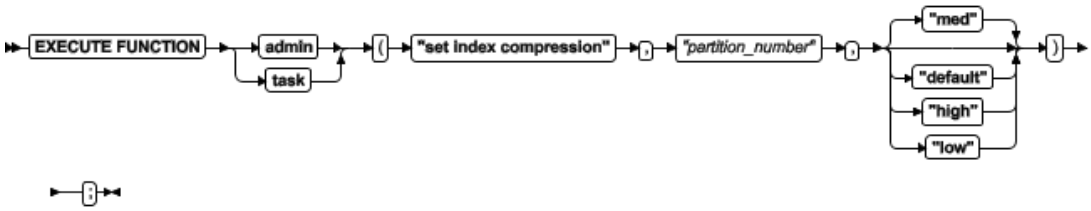
如果名为 dbsp1 的 dbspace 停止工作，则下列示例跳过它：

```
EXECUTE FUNCTION task("set dataskip on","dbsp1");
```

4. 130 set index compression 参数：更改索引页压缩（SQL 管理 API）

随同 admin() 或 task() 函数，使用 set index compression 参数来更改合并两个部分地使用的索引页的级别。

语法



元素	描述	关键考虑
partition_number	分区号。	请在 systables 系统目录表的 partnum 列中找到该分区号。

用法

请使用这个函数来调整索引页压缩。如果在那些页上的数据合计到设置的级别，则合并这些页。如果您的索引快速地增长，要优化空间和事务处理，则可降低压缩级别。如果您的索引有很少的删除和插入操作或如果执行批量更新，则可提高级别。

如果您预计索引会以频繁的分裂快速地增长，则请使用 low 参数。

如果索引的增长或更改适中，则请使用 med 或 default 参数。

如果索引 90% 或更多为 read-only，或者有许多更改，则请使用 high 参数。

这个函数等同于 gadmin -C 命令以及 BTSCANNER 配置参数的 compression 选项。

示例

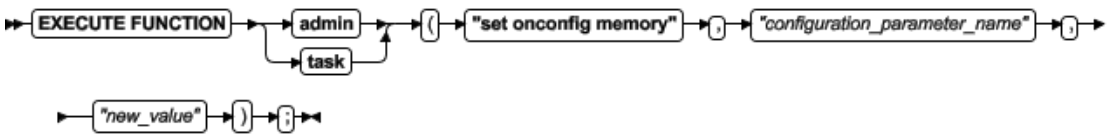
下列示例设置分区的索引压缩为 high：

```
EXECUTE FUNCTION task("set index compression","1048611","high");
```

4.131 set onconfig memory 参数：临时地更改配置参数（SQL 管理 API）

随同 admin() 或 task() 函数，使用 set onconfig memory 参数来动态地更新内存中配置参数的值。

语法



元素	描述	关键考虑
configuration_parameter_name	配置参数的名称。	该配置参数必须是您可动态地更新的。 您可动态地更新的配置参数列表与 gadmin -wf 命令的相同。
new_value	该配置参数的一个或多个新的值。	该配置参数的这个或这些新的值必须是有效的。 新值的格式必须严格地遵循那个配置参数的语法。

用法

使用这个函数来临时地更新可动态地更新的配置参数的值。新的值立即生效。新的值不写到 onconfig 文件，当数据库服务器重启时，新的值丢失。

这个函数等同于 gadmin -wm 命令。

示例

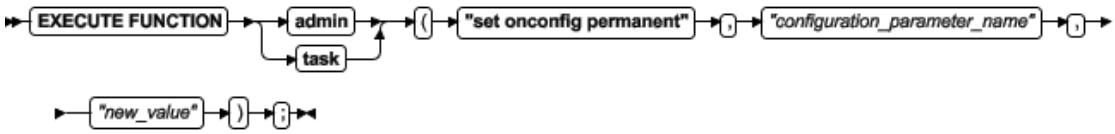
下列示例为当前会话设置 DYNAMIC_LOGS 配置参数的值为 2：

```
EXECUTE FUNCTION task("set onconfig memory","DYNAMIC_LOGS","2");
```

4.132 set onconfig permanent 参数：永久地更改配置参数（SQL 管理 API）

随同 admin() 或 task() 函数，使用 set onconfig permanent 参数来动态地更新 onconfig 文件中的配置参数值。

语法



元素	描述	关键考虑
configuration_parameter_name	配置参数的名称。	该配置参数必须是您可动态地更新的。 您可动态地更新的配置参数列表与 **gadmin -wf** 命令的相同。
new_value	配置参数的一个或多个新的值。	这个或这些新的配置参数值必须是有效的。 新值的格式必须严格地遵循那个配置参数的语法。

用法

使用这个函数来永久地更新配置参数的值。新的值立即生效并在服务器重启之后保存在 onconfig 文件中。

这个函数等同于 gadmin -wf 命令。

示例

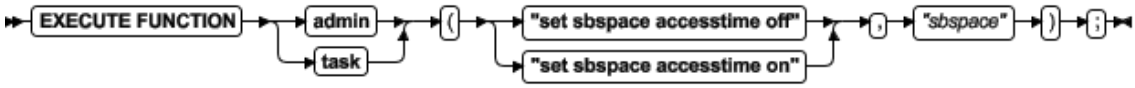
下列示例在 onconfig 文件中设置 DYNAMIC_LOGS 配置参数的值为 2：

```
EXECUTE FUNCTION task("set onconfig permanent","DYNAMIC_LOGS","2");
```

4.133 set sbspace accesstime 参数：控制访问次数跟踪（SQL 管理 API）

随同 admin() 或 task() 函数，使用 set sbspace accesstime 参数来启动或停止跟踪存储在该 sbspace 中所有智能大对象的访问次数。

语法



元素	描述	关键考虑
sbspace	sbspace 的名称。	

用法

使用 set sbspace accesstime off 参数来关闭访问次数的跟踪。

使用 set sbspace accesstime on 参数来开启跟踪在该 sbspace 中的所有智能大对象的访问次数。

这个函数等同于 gspaces -ch 命令。

示例

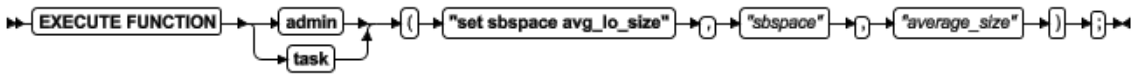
下列示例关闭跟踪对名为 sbsp1 的 sbspace 的访问次数：

```
EXECUTE FUNCTION task("set sbspace accesstime off","sbsp1");
```

4. 134 set sbspace avg_lo_size 参数：设置智能大对象的平均大小（SQL 管理 API）

随同 admin() 或 task() 函数，使用 set sbspace avg_lo_size 参数来指定在指定的 sbspace 中智能大对象的预期平均大小，以便数据库服务器可计算元数据区的大小。

语法



元素	描述	关键考虑
sbspace	sbspace 的名称。	
average_size	该 sbspace 中智能大对象的平均大小，以 KB 为单位。	Windows™：4 至 2**31 UNIX™：2 至 2**31

用法

这个函数等同于 gspaces -ch 命令。

示例

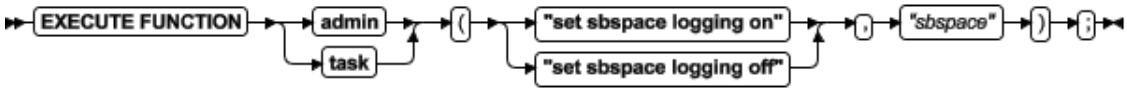
下列示例设置名为 sbsp1 的 sbspace 中智能大对象的预期平均大小为 8 KB：

```
EXECUTE FUNCTION task("set sbspace avg_lo_size","sbsp1","8");
```

4.135 set sbspace logging 参数：更改 sbspace 的日志记录（SQL 管理 API）

随同 admin() 或 task() 函数，使用 set sbspace logging 参数来指定是否将数据库服务器日志更改到 sbspace 的用户数据区域。

语法



元素	描述	关键考虑
sbspace	sbspace 的名称。	

用法

使用 set sbspace logging on 参数来将日志更改到 sbspace 的用户数据区域。

使用 set sbspace logging off 参数不将日志更改到 sbspace 的用户数据区域。

这个函数等同于 gspaces -ch 命令。

示例

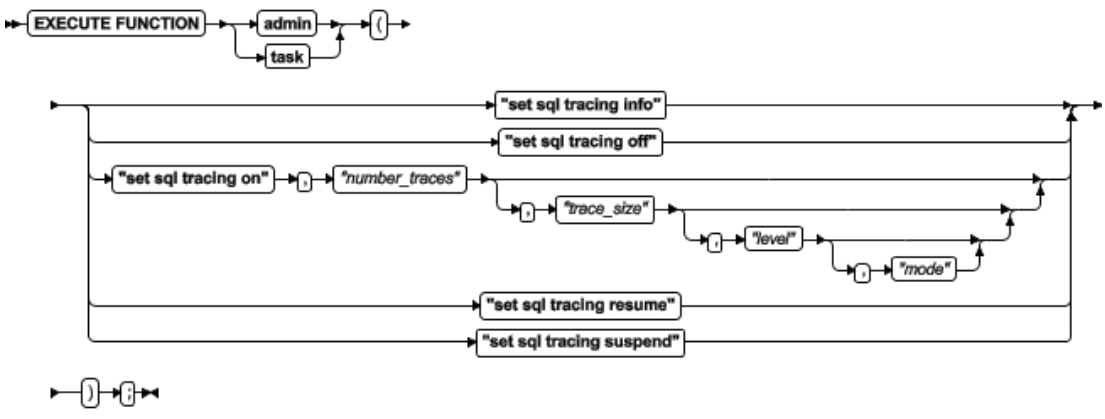
下列示例为名为 sbsp1 的 sbspace 启动 sbspace 日志记录：

```
EXECUTE FUNCTION task("set sbspace logging on","sbsp1");
```

4.136 set sql tracing 参数：设置全局 SQL 跟踪（SQL 管理 API）

随同 admin() 或 task() 函数，使用 set sql tracing 参数来设置全局 SQL 跟踪。

语法



元素	描述	关键考虑

元素	描述	关键考虑
level	跟踪级别。缺省为 low。	可能的值为： ● low ● med ● high
mode	跟踪所有用户还是选中的用户。	可能的值为： ● global ● user
number_traces	要跟踪的 SQL 语句的数目。缺省值为 1000。	
trace_size	跟踪缓冲区大小的 KB 数。如果超过这个缓冲区大小，则数据库服务器丢弃保存的数据。缺省大小为 2 KB。	

用法

使用这个函数来重置 SQLTRACE 配置参数的值。

请使用 set sql tracing info 参数来显示全局 SQL 跟踪的状态。

请使用 set sql tracing off 参数来关闭全局 SQL 跟踪。

请使用 set sql tracing on 参数来开启全局 SQL 跟踪。您可选择地指定跟踪级别和模式，或更改跟踪缓冲区的大小。

- 使用 low 参数来捕获语句统计、语句文本和语句迭代器。
- 使用 med 参数来捕获包括在低级跟踪的所有信息，加上表名、数据库名和存储过程堆栈。
- 使用 high 参数来捕获包括在中级跟踪的所有信息，加上主机变量。
- 使用 global 参数来启用对所有用户的跟踪。
- 使用 user 参数来启用跟踪那些由 set sql tracing user 参数启用跟踪的用户。

当暂停 SQL 跟踪时，使用 set sql tracing resume 参数来重启 SQL 跟踪。

使用 set sql tracing suspend 参数来暂停 SQL 跟踪，而不重新分配任何资源。

示例

下列示例对 1500 SQL 语句启动高级别全局跟踪到 4 KB 跟踪缓冲区内：

```
EXECUTE FUNCTION task("set sql tracing on","1500","4","high","global");
```

下列示例暂停 SQL 跟踪：

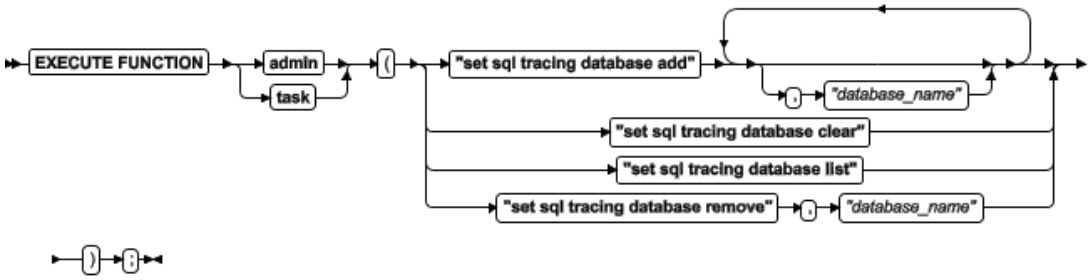
```
EXECUTE FUNCTION task("set sql tracing suspend");
```

4.137 set sql tracing database 参数：更改数据库

跟踪（SQL 管理 API）

随同 admin() 或 task() 函数，使用 set sql tracing database 参数来启动或停止对数据库的跟踪，或罗列哪些数据库正被跟踪。

语法



元素	描述	关键考虑
database_name	数据库名。	指定一个数据库名。

用法

使用 set sql tracing database add 参数来指定对一个或多个数据库的跟踪，而不是跟踪所有数据库。缺省为所有数据库。在单个 admin() 或 task() 函数中指定至多六个参数。数据库名的最大数目可设置为 16。

使用 set sql tracing database clear 参数来从正被跟踪的数据库列表清除所有数据库。跟踪返回到缺省的全部数据库。

使用 set sql tracing database list 参数来罗列正被跟踪的数据库。

使用 set sql tracing database remove 参数来从正被跟踪的数据库列表移除单个数据库。

当您使用 set sql tracing database 参数时，仅可指定一个数据库的名称。在您可有最多 16 个数据库名时，必须在分别的函数调用中指定每一附加的数据库。您每次调用该函数时，该函数添加另一数据库到列表，直到列表包含 16 个数据库。

示例

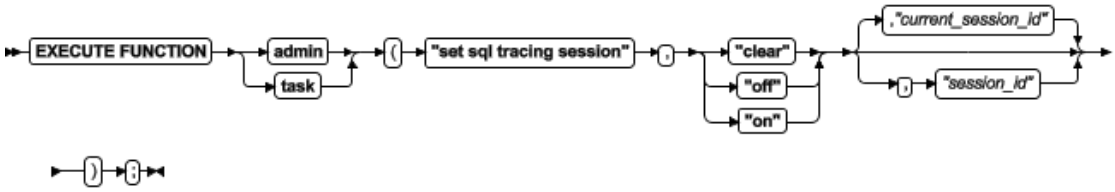
下列示例设置对名为 db1、db2 和 db3的三个数据库设置 SQL 跟踪：

```
EXECUTE FUNCTION task("set sql tracing database add","db1");
EXECUTE FUNCTION task("set sql tracing database add","db2");
EXECUTE FUNCTION task("set sql tracing database add","db3");
```

4. 138 set sql tracing session 参数：控制对会话的跟踪（SQL 管理 API）

随同 admin() 或 task() 函数，使用 set sql tracing session 参数来更改对当前会话的 SQL 跟踪。

语法



元素	描述	关键考虑
current_session_id	当前会话的 ID。这是缺省会话 ID。	
session_id	这个命令应用的会话的 ID。	

用法

使用 clear 参数来清除任何全局跟踪覆盖。该会话将符合全局跟踪策略。

使用 off 参数来关闭对该会话的跟踪，即使设置全局跟踪策略为启用跟踪。

使用 on 参数来开启对该会话的跟踪，即使设置全局跟踪策略为禁用跟踪。

示例

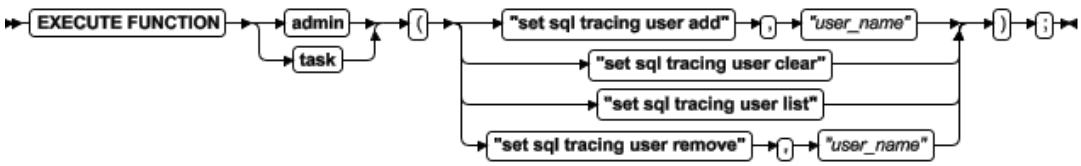
下列示例停止对当前会话的跟踪：

```
EXECUTE FUNCTION task("set sql tracing session","off");
```

4. 139 set sql tracing user 参数：控制对用户的跟踪（SQL 管理 API）

随同 admin() 或 task() 函数，使用 set sql tracing user 参数来更改对用户的 SQL 跟踪。

语法



元素	描述	关键考虑
user_name	用户名。	

用法

使用 set sql tracing user add 参数来指定对特定用户的跟踪。

使用 set sql tracing user clear 参数来从跟踪列表移除所有用户。

使用 set sql tracing user list 参数来罗列正被跟踪的用户。

使用 set sql tracing user remove 参数来从正被跟踪的用户列表移除单个用户。

示例

下列示例对名为 fred 的用户停止跟踪 SQL 语句：

```
EXECUTE FUNCTION task("set sql tracing user remove","fred");
```

4. 140 set sql user tracing 参数：设置对用户会话的全局 SQL 跟踪（SQL 管理 API）

随同 admin() 或 task() 函数，使用 set sql user tracing 参数来对指定的用户会话设置全局 SQL 跟踪的模式。

语法

元素	描述	关键考虑
session_id	会话的 ID。	

用法

使用 set sql user tracing clear 来对指定的用户会话清除用户跟踪标志，以便其遵守全局跟踪策略。

使用 set sql user tracing off 来禁用对用户会话的 SQL 跟踪，即使全局模式为 ON。

使用 set sql user tracing on 来启用对用户会话的用户 SQL 跟踪。即使全局跟踪模式为 OFF，也跟踪这个用户会话的 SQL 语句。

示例

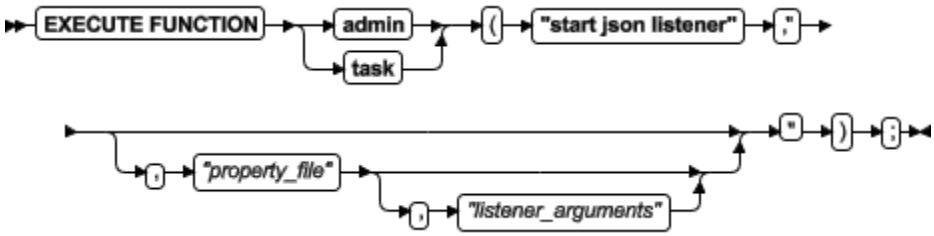
下列示例启动 对 ID 18 会话的跟踪：

```
EXECUTE FUNCTION task("set sql user tracing on","18");
```

4. 141 start json listener 参数：启动有线监听器

随同 admin() 或 task() 函数，使用 start json listener 参数来启动有线监听器。

语法



元素	描述	关键考虑
property_file	要使用的资产文件名，而不是缺省的。	*property_file* 是可选的。缺省资产文件在 $GBASEDBTDIR/etc/jsonListener.properties 中。
listener_arguments	传递给有线监听器的命令行参数。	

用法

start json listener 参数启动有线监听器。

示例

在这个示例中，通过使用 mycustom.properties 文件启动有线监听器，而不使用缺省 jsonListener.properties 文件：

```
EXECUTE FUNCTION task("start json listener", "mycustom.properties");
```

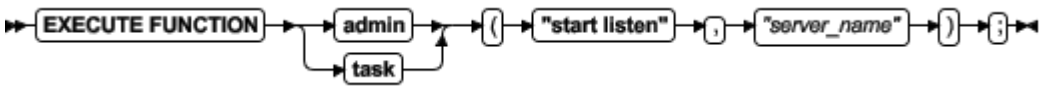
在这个示例中，通过使用 mycustom.properties 文件启动有线监听器，而不使用缺省 jsonListener.properties，并传递给有线监听器命令行参数：

```
EXECUTE FUNCTION task("start json listener", "mycustom.properties",
            "-port 27018 -loglevel debug");
```

4.142 start listen 参数：动态地启动监听线程（SQL 管理 API）

随同 admin() 或 task() 函数，使用 start listen 参数来为 SOCTCP 或 TLITCP 网络协议启动现有的监听线程，而不中断现有的连接。

语法



元素	描述	关键考虑
server_name	你想要为其启动监听线程的那个数据库服务器的名称。	

用法

在该服务器的 sqlhosts 文件中必须存在监听线程的定义。如果在 sqlhosts 文件中不存在监听线程的定义，则您必须在可动态地启动监听线程之前添加它。要了解有关添加监听线程的信息，请参阅 GBase 8s 管理员指南。

这个函数不更新 sqlhosts 文件。

这个函数等同于 gadmin -P start server_name 命令。

示例

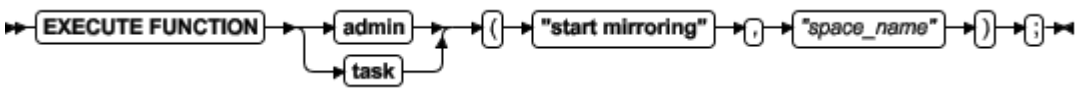
下列命令为名为 ids_serv2 的服务器启动新的监听线程：

```
EXECUTE FUNCTION task("start listen","ids_serv2");
```

4.143 start mirroring 参数：启动存储空间镜像（SQL 管理 API）

随同 admin() 或 task() 函数，使用 start mirroring 参数来启动对指定的 dbspace、blobspace 或 sbspace 的镜像。

语法



元素	描述	关键考虑
space_name	blobspace、dbspace 或 sbspace 的名称。	

用法

这个函数等同于 gspaces -m 命令。

示例

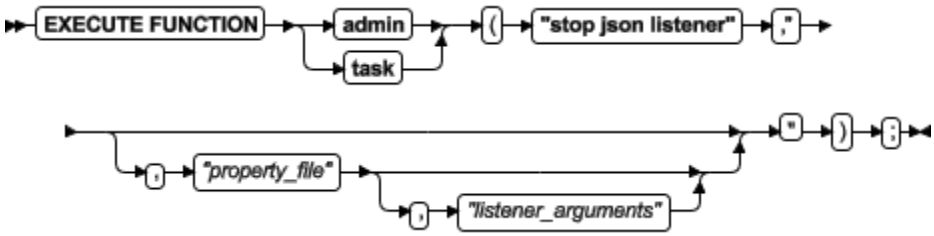
下列示例启动对名为 dbsp1 的 dbspace 的镜像：

```
EXECUTE FUNCTION task("start mirroring","dbsp1");
```

4.144 stop json listener 参数：停止有线监听器

随同 admin() 或 task() 函数，使用 stop json listener 参数来停止有线监听器。

语法



元素	描述	关键考虑
property_file	要使用的资产文件名，不是缺省的。	*property_file* 是可选的。缺省资产文件在 $GBASEDBTDIR/etc/jsonListener.properties 中。
listener_arguments	要传递到有线监听器的命令行参数。	

用法

stop json listener 参数停止有线监听器。

示例

在下列示例中，通过使用 mycustom.properties 文件停止有线监听器，而不通过缺省 jsonListener.properties 文件：

```
EXECUTE FUNCTION task("stop json listener", "mycustom.properties");
```

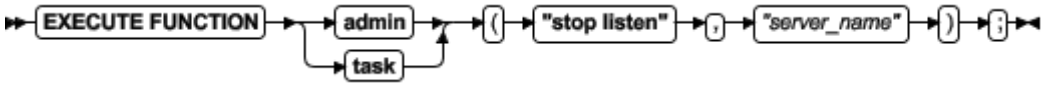
在这个示例中，通过使用 mycustom.properties 文件停止有线监听器，而不通过缺省 jsonListener.properties，且传递到有线监听器命令行参数：

```
EXECUTE FUNCTION task("stop json listener", "mycustom.properties", "-port 27018");
```

4.145 stop listen 参数：动态地停止监听线程（SQL 管理 API）

随同 admin() 或 task() 函数，使用 stop listen 参数来为 SOCTCP 或 TLITCP 网络协议停止现有的监听线程，而不中断现有的连接。

语法



元素	描述	关键考虑
server_name	你想要为其停止监听线程的那个数据库服务器的名称。	

用法

在 sqlhosts 文件中必须存在监听线程的定义。

这个函数不更新 sqlhosts 文件。

这个函数等同于 gadmin -P stop server_name 命令。

示例

下列命令为名为 ids_serv3 的服务器停止监听线程：

```
EXECUTE FUNCTION task("stop listen","ids_serv3");
```

4.146 stop mirroring 参数：停止存储空间镜像（SQL 管理 API）

随同 admin() 或 task() 函数，使用 stop mirroring 参数来为指定的 dbspace、blobspace 或 sbspace 停止镜像。

语法

元素	描述	关键考虑
space_name	blobspace、dbspace 或 sbspace 的名称。	

用法

这个函数等同于 gspaces -r 命令。

示例

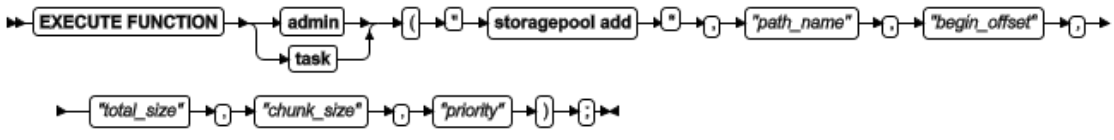
下列示例停止对名为 dbsp1 的 dbspace 的镜像：

```
EXECUTE FUNCTION task("stop mirroring","dbsp1");
```

4.147 storagepool add 参数：添加存储池条目（SQL 管理 API）

随同 admin() 或 task() 函数，使用 storagepool add 参数来添加条目到存储池（ GBase 8s 可用于自动地添加空间到现有存储空间的裸设备、熟文件或目录的集合）。

语法



元素	描述	关键考虑
path_name	当需要附加的存储空间时，服务器可使用的文件、目录或设备的路径。	您无需添加结尾斜杠（“/”）到目录名。 当运行 oninit 命令时，如果在您的环境中有环境变量，则您可使用该路径中的环境变量。
begin_offset	设备内 GBase 8s 可开始分配空间处的偏移量，以 KB 为单位。	如果您指定了到目录的路径，则必须指定 0 为偏移量。
total_size	在这个条目中 GBase 8s 可用的合计空间。服务器可从这个空间量分配多个 chunk。	请确保为目录的合计大小指定 0。如果您为目录指定非零的值，则 SQL 管理 API 命令返回错误提示。 如果您为文件或设备指定 0，则服务器会从该条目分配一个可扩展的 chunk。
chunk_size	从该设备、文件或目录可分配的 chunk 大小的最小值。	您可创建的最小 chunk 为 1000 K。因此，您可指定的最小 chunk 大小为 1000 K。 请参阅 admin() 和 task() 参数大小规范。
priority	当服务器通过存储池查找空间时，该目录、文件或设备的优先级。 • 1 = 高优先级 • 2 = 中优先级 • 3 = 低优先级	在服务器从较低优先级条目分配空间之前，会试图从高优先级条目分配空间。

用法

如果有必要添加新的 chunk 到存储空间，则服务器使用存储池中的条目。

当您添加条目到存储池时，可能先要对如何使用那个条目施加控制。例如，减少实例中的 chunk 数，您可能仅想从特定裸设备分配大型空间 chunk，且可能不想这些 chunk 是可扩展的。 在这种情况下，请配置那个存储池的 chunk 大小为大型的。

您可以添加下列类型的条目到存储池：

- 定长裸设备
- 定长熟文件
- 可扩展的裸设备（为了扩展 chunk 的大小）
- 可扩展的熟文件（为了扩展 chunk 的大小）
- 目录

目录存储池条目通常归类为可扩展的，因为它没有合计大小。如果在该目录中自动地创建新的 chunk，则服务器标记那些 chunk 为可扩展的。当您添加目录存储池条目时，您可能想要小的 chunk 大小，因为服务器可扩展该目录中的任何 chunk，且较少的 chunk 大小可减少实例中浪费的空间量。

如果存储池条目在“高可用性数据复制”（HDR）主服务器上，则在该 HDR 集群中的所有辅助服务器上，该条目中相同的路径必须是可用的。

大小和偏移量的缺省单位为 KB。然而，在您可以下列示例中显示的任何方式指定信息：

- "100000"
- "100000 K"
- "100 MB"
- "100 GB"
- "100 TB"

示例：为目录添加存储池条目

下列命令添加名为 /region2/dbspaces 的目录，起始偏移量为 0，合计大小为 0，初始 chunk 大小为 20 MB，且为高优先级：

```
DATABASE sysadmin;
EXECUTE FUNCTION task("storagepool add", "/region2/dbspaces", "0", "0","20000",
"1");
```

示例：为定长裸设备添加存储池条目

下列命令将路径名为 /dev/raw/device1 且合计 500 MB 空间的定长裸设备添加到存储池。该命令指定起始偏移量为 50 MB，合计大小为 10 GB，最小分配到 chunk 100 MB，且为低优先级.

```
EXECUTE FUNCTION task("storagepool add", "/dev/rawdevice1", "50 MB", "10 GB",
"100 MB", "3");
```

示例：为定长熟文件添加存储池条目

下列命令添加定长熟文件和 1 GB 空间到存储池。该命令指定起始偏移量为 0， 合计大小为 1000000 KB，最小分配到 chunk 50000 KB，且为中优先级：

```
EXECUTE FUNCTION task("storagepool add", "/ifmx_filesystem/storage/cooked7",
```

```
"0", "1000000", "50000", "2");
```

当添加这个条目时，服务器试图将 cooked7 文件的大小增加到 1 GB。如果由于文件系统已满，服务器不能增加大小，则服务器返回错误消息且不添加条目到存储池。

GBase 8s 最初使用熟文件的一部分，但随着空间填满，如有必要可使用更多设备。

示例：为可扩展的熟文件添加存储池条目

下列命令添加路径名为 /ifmx/CHUNKFILES/cooked2 的熟文件。如果服务器使用这个条目，则服务器创建一个初始大小为 1 GB 的 chunk，且服务器自动地标记该 chunk 为可扩展的。

```
EXECUTE FUNCTION task("storagepool add", "/ifmx/CHUNKFILES/cooked2",
 "0", "0", "1 GB", "2");
```

示例：用路径中的环境变量添加存储池条目

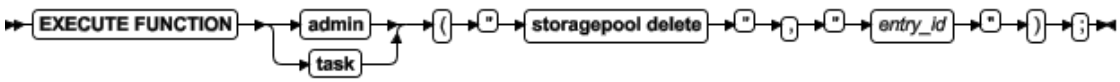
下列示例包括路径中的环境变量。当运行 oninit 命令时，该变量出现在服务器环境中。

```
EXECUTE FUNCTION task("storagepool add", "$DBSDIR/chunk1",
 "0", "100000", "20000", "2");
```

4.148 storagepool delete 参数：删除一个存储池条目（SQL 管理 API）

随同 admin() 或 task() 函数，使用 storagepool delete 参数来从存储池删除条目。

语法



元素	描述	关键考虑
entry_id	存储池条目的 ID。	sysadmin 数据库中的 storagepool 表包含一个显示存储池中每一条目 ID 的列。

用法

当扩充存储空间时，如果您不想服务器继续使用该条目，则删除存储池条目。

要删除所有存储池条目、有状态为 Full 的存储池条目，或有状态为 Error 的存储池条目，请使用 SQL 管理 API storagepool purge 命令。（sysadmin 数据库中的 storagepool 表中包含显示存储池中每一条目状态的列。）

示例

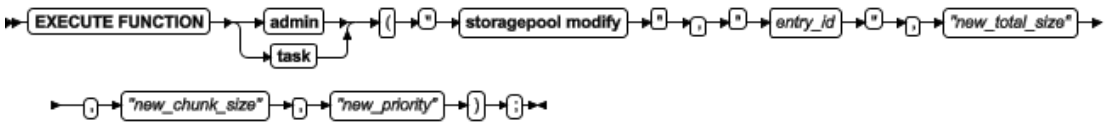
下列命令删除条目 ID 为 13 的存储池条目：

```
EXECUTE FUNCTION task("storagepool delete", "13");
```

4.149 storagepool modify 参数：更改存储池条目（SQL 管理 API）

随同 admin() 或 task() 函数，使用 storagepool modify 参数来更改当需要附加的存储空间时 GBase 8s 可使用的目录、熟文件或裸设备的条目。

语法



元素	描述	关键考虑
entry_id	存储池条目的 ID。	sysadmin 数据库中的 storagepool 表包含显示存储池中每一条目 ID 的列。
new_total_size	在这个条目中 GBase 8s 可用的合计空间的新数量。服务器可从这个空间量分配多个 chunk。	请确保指定目录的合计大小为 0。如果您为目录指定非零的值，则 SQL 管理 API 命令返回错误提示。 如果您为文件或设备指定 0，则服务器从该条目分配一个可扩展的 chunk。
new_chunk_size	可从设备、文件或目录分配的 chunk 大小的最小值。	您可创建的最小 chunk 为 1000 K。因此，您可指定的最小 chunk 大小为 1000 K。 请参阅 admin() 和 task() 参数大小规范。
new_priority	当服务器通过存储池查找空间时，目录、文件或设备的优先级。 • 1 = 高优先级 • 2 = 中优先级 • 3 = 低优先级	在服务器从较低优先级条目分配空间之前，它试图从高优先级条目分配空间。

用法

有时您可能想要更改存储池条目。例如，当存储池空间用尽时，您可能想要增加存储池的合计大小，或您可能想要更改 chunk 大小或优先级。即使您不想更改所有值，当您更改条目时，也包括合计大小、chunk 大小和优先级。

您不可更改存储池条目的路径或起始偏移量。如果您想要更改那些值中的任何一个，则必须删除该存储池条目并以新的路径或起始偏移量添加条目。

如果存储池条目在“高可用性数据复制”（HDR）主服务器上，则在 HDR 集群中所有辅助服务器上的该条目中相同的路径必须可用。

存储池大小和偏移量的缺省单位为 KB。然而，您可以下列示例中显示的任何方式指定信息：

- "100000"
- "100000 K"
- "100 MB"
- "100 GB"
- "100 TB"

示例

下列命令将条目 ID 为 4 的存储池条目的合计大小、chunk 大小和优先级更改为 10 GB、10 MB 和中优先级。

```
EXECUTE FUNCTION task("storagepool modify", "4", "10 GB", "10000", "2");
```

假定您添加条目到存储池，且条目路径为（/dev/IDS/chunk2），偏移量为 0，合计大小为 100 MB，最小 chunk 大小为 100 MB，优先级为 2。 在 GBase 8s 从这个条目分配任何空间之前，请您使用 gspaces 来手工地添加带有相同路径(/dev/IDS/chunk2)的 50 MB chunk，偏移量为 50 MB。当服务器尝试使用这个条目来自动地创建 chunk 时，服务器仅检测重叠。那时，服务器标记该条目为“Error”状态并尝试使用另一个条目来创建 chunk。

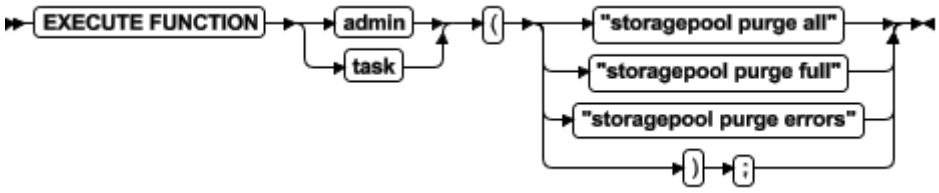
您可通过更改存储池条目的合计大小来修正该问题（例如，对于条目 2），更改为 50 MB 且通过更改该条目的最小 chunk 大小为 50 MB，如下：

```
EXECUTE FUNCTION task("storagepool modify", "2", "50 MB", "50 MB", "2");
```

4.150 storagepool purge 参数：删除存储池条目（SQL 管理 API）

随同 admin() 或 task() 函数，使用 storagepool purge 参数来删除所有存储池条目、有 Full 状态的存储池条目，或有 Error 状态的存储池条目。

语法



用法

使用 storagepool purge all 参数来删除存储池中的所有条目。

使用 storagepool purge full 参数来删除所有状态为 Full 的存储池条目。

使用 storagepool purge errors 参数来删除所有状态为 Error 的存储池条目。

sysadmin 数据库中的 storagepool 表包含显示存储池中每一条目的状态的列。

示例

下列命令删除所有状态为 Full 的存储池条目：

```
EXECUTE FUNCTION task("storagepool purge full");
```

4.151 表和分片压缩和解压缩操作（SQL 管理 API）

您可用 SQL 管理 API admin() 或 task() 函数和参数压缩和解压缩表中或表分片中的数据。压缩操作仅适用于数据行的内容和那些出现在逻辑日志记录中的数据行的映像。

在每一 GBase 8s 实例的 sysadmin 数据库中，定义内建 SQL 管理 API admin() 或 task() 函数。缺省情况下，仅用户 gbasedbt 可调用这些函数。如果将 sysadmin 数据库上的 Connect 权限授予用户 root 或 DBSA 组成员，则当他们直接地或远程地连接到 sysadmin 数据库时，他们也可调用 SQL 管理 API admin() 或 task() 函数。

您可在表和表分片中用于压缩和解压缩操作的 SQL 管理 API admin() 或 task() 命令参数为：

table 压缩参数

对于指定表的所有分片执行不同的操作。要了解更多信息，请参阅 table 或 fragment 参数：压缩数据和优化存储（SQL 管理 API）。

fragment 压缩参数

对于属于特定表的单个分片或指定的分片集合执行不同的压缩操作。要了解更多信息，请参阅 table 或 fragment 参数：压缩数据和优化存储（SQL 管理 API）。

compression purge_dictionary

删除所有不活动的压缩字典或在您指定日期之前创建的所有不活动的压缩字典。要了解更多信息，请参阅 清除压缩字典参数：移除压缩字典（SQL 管理 API）。

表和分片压缩操作包括创建压缩字典、估计压缩率、压缩表和表分片中的数据、合并空闲空间（重新打包）、归还空闲空间到 dbspace（收缩）、解压缩数据以及删除个别的表和分片压缩字典。

当您运行 SQL 管理 API 压缩和解压缩命令时，压缩和解压缩 dbspace 中的行数据和简单大对象。您还可指定是否仅压缩或解压缩行数据，或仅压缩或解压缩 dbspace 中的简单大对象。

admin() 命令返回一个整数；task() 命令返回一个字符串。

要了解关于您可压缩的数据类型、压缩率、压缩估计和压缩字典的信息，以及使用压缩命令参数的步骤， 请参阅 GBase 8s 管理员指南 中的行数据的压缩。要了解关于显示压缩信息的实用程序以及 sysmaster 表和视图的信息，请参阅 syscompdicts_full。

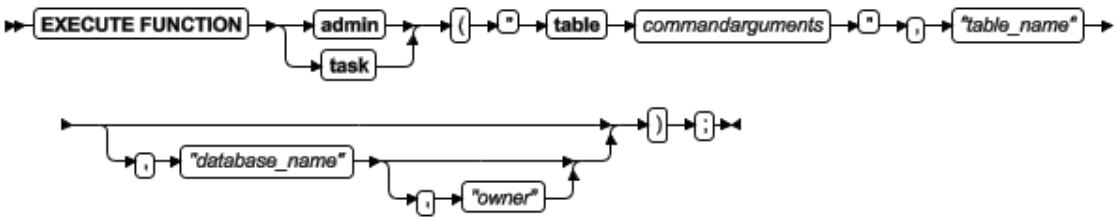
您还可压缩、优化存储和估计对 B-tree 的压缩收益。请参阅 index compress repack shrink 参数：优化 B-tree 索引的存储（SQL 管理 API）和 index estimate_compression 参数：估计索引压缩（SQL 管理 API）。

4.151.1 table 或 fragment 参数：压缩数据和优化存储（SQL 管理 API）

使用带有 table 或 fragment 参数的 SQL 管理 API 函数来创建压缩字典、来估计压缩率、来压缩表和表分片中的数据、来合并空闲空间（重新打包）、 来归还空闲空间到 dbspace（收缩）、来解压缩数据，以及来删除压缩字典。

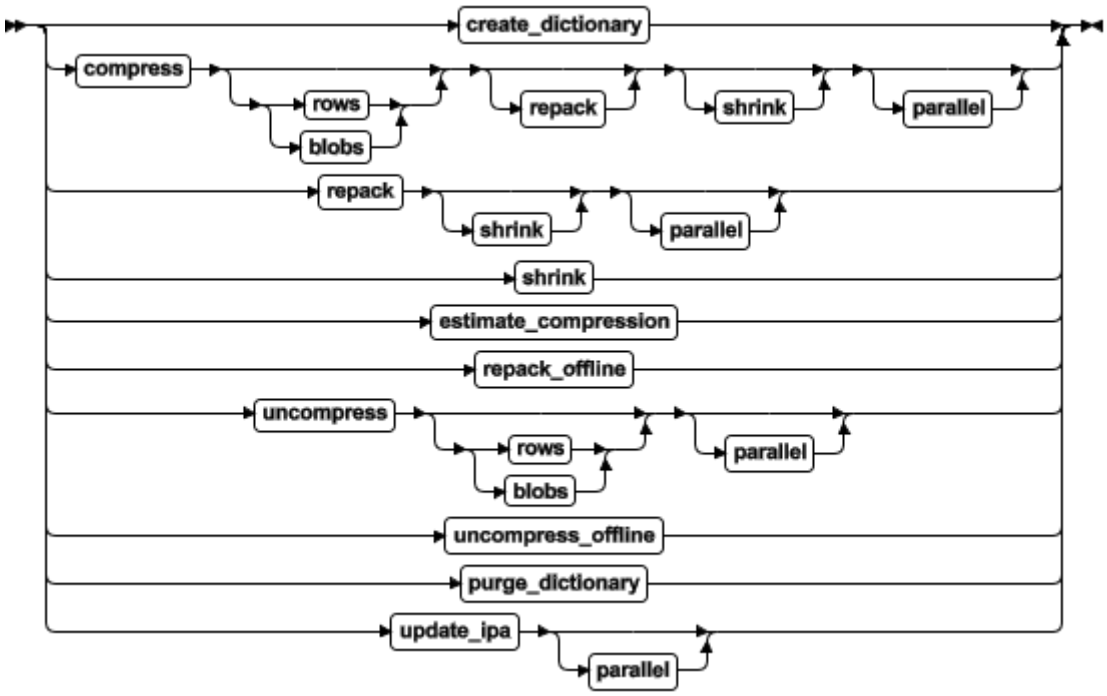
当您运行 SQL 管理 API 压缩和解压缩命令时，您压缩或解压缩 dbspace 中的行数据和简单大对象。您还可指定是否仅压缩或解压缩行数据，或仅压缩或解压缩 dbspace 中的简单大对象。

语法：表数据压缩命令参数：



语法：分片数据压缩命令参数

表和分片命令参数



命令参数

下表描述每一参数。

表 1. 压缩和解压缩操作的参数

参数	描述
blobs	指定您仅想要压缩或解压缩 dbspace 中的简单大对象，而不是行数据。
compress	就地压缩所有现有的行，而不移动它们（不重新打包表）。 这个选项自动地压缩 dbspace 中的行数据和简单大对象。要仅压缩行数据或仅压缩 dbspace 中的简单大对象，还要使用 rows 或 blobs 元素。 如果目标表或分片的压缩字典不存在，则压缩操作还创建该字典。
create_dictionary	构建压缩字典。压缩字典是频繁地发生的模式和符号库，在压缩的行中取代它们。 创建字典之后，如果新插入的或更新的行可以压缩，则会压缩它们。
estimate_compression	估计新的压缩率和当前的压缩率。如果该表未压缩，则当前压缩率为 0.0%。
parallel	并行地运行压缩、重新打包、update_ipa 或解压缩操作。

参数	描述
	为每个表的分片或分片列表启动一个线程，跨那些分片并行地运行该操作。
purge_dictionary	在您解压缩表或分片之后，删除不活动的压缩字典。
repack	通过将数据移至分片或表的前部合并空闲空间。 在分片处于 online 时，因为重新打包操作移动行，所以访问正在使用隔离级别低于 Repeatable Read 的分片的其他查询可能偶尔地找到同一行两次或找不到行。要避免这种可能性，请为并发查询使用 Repeatable Read 隔离级别；或者，不使用 repack 参数，而使用 repack_offline 参数。
repack_offline	在表或分片上持有排他锁时，通过将数据移至表或分片的前部，合并空闲空间。这个操作阻止所有其他对数据的访问，直到该操作完成。
rows	指定您仅想要压缩或解压缩行数据，而不是 dbspace 中的简单大对象。
shrink	将分片或表后部的空闲空间归还到 dbspace，从而减少该分片或表的合计大小。
uncompress	停止对新的 INSERT 和 UPDATE 操作压缩，解压缩所有压缩的行，并停用压缩字典。这个操作还为分片分配新的页，并将那些在原始页上不再适合的解压缩的行移至新页。 在分片处于 online 时，因为这个操作移动行，所以正在访问使用低于 Repeatable Read 隔离级别的分片的其他操作可能偶尔地找到同一行两次或找不到行。为了避免这种可能性，请为并发查询使用 Repeatable 隔离级别，或不使用 uncompress 参数，而使用 uncompress_offline 参数。 这个选项自动地解压缩 dbspace 中的行数据和简单大对象。要仅压缩行数据或仅压缩 dbspace 中的简单大对象，还请使用 rows 或 blobs 元素。
uncompress_offline	在持有分片上的排他锁时，停止对新的 INSERT 和 UPDATE 操作压缩，解压缩所有压缩的行，并停用压缩字典。这样阻止对分片数据的所有其他访问，直到该操作完成。 这个操作还为分片分配新的页，并将那些在其原始页上不再适合的解压缩的行移至新的页。
update_ipa	为指定的表或分片移除未完成的就地更改操作。

命令元素

下表显示您可在命令中使用的元素。

表 2. 表压缩和存储优化命令元素

元素	描述	关键考虑
database_name	包含指定的表的数据库名。	可选的。 如果您未指定 *database*，则 GBase 8s 使用当前数据库。 如果您输入数据库名，则必须使用与系统目录表中相同的大写或小写字母。
owner	包含指定的表的数据库所有者的授权标识符。	可选的。 如果您未指定 *owner*，则 GBase 8s 使用当前的所有者。 如果您输入所有者名，则必须使用与系统目录表中相同的大写或小写字母。
table_name	包含数据的表名。	您必须使用与系统目录表中相同的大写或小写字母。

表 3. 分片压缩和存储优化命令元素

元素	描述	关键考虑
partition_number	属于同一表的用空格分隔的分区号列表。	

用法

GBase 8s 使用压缩字典来压缩数据。

您在表或分片上运行 compress 命令之后，GBase 8s 自动地压缩您添加到表或分片的任何新行。当您运行 compress 时，如果表或分片包含的行数多于 2000，则构建压缩字典且压缩所有行。当您运行压缩命令时，如果表或分片包含的行数少于 2000，则启用该表或分片的自动压缩。插入 2000 行之后，创建压缩字典并压缩初始 2000 行之后的所有行。要压缩初始的 2000 行，请再次运行 compress 命令。

如果您显著地更改数据，则压缩字典可能无效。在这种情况下，请解压缩然后再压缩。

您可取消带有 compress 或 uncompress 参数的命令，例如，在 DB-Access 中键入 CTRL-C。在前一个中断的命令之后，您可重新发出带有 repack、 repack_offline、uncompress 和 uncompress_offline 参数的命令。

在表或分片上已发生了 compress、repack、repack_offline、shrink、uncompress 或 uncompress_offline 操作的任何一个时，您不可在表或分片上执行这些操作。

当您在单个命令中指定多个操作时，服务器按如下顺序执行这些操作：

- create_dictionary
- compress
- repack
- shrink

compress、repack、repack_offline、uncompress 和 uncompress_offline 操作可消耗大量日志文件。如果包括但不限于这些压缩操作，您预期运行的任何工作负载消耗日志文件快于每 30 秒一个，则请将日志配置大些。

日志记录 compress、repack 和 uncompress 操作，但运行在小部分中。

在执行压缩操作之后，如果您更改分片策略，则表失去其压缩状态，并将需要重新压缩。

在您完成repack_offline 或 uncompress_offline 操作之前删除或禁用索引，可减少数据库完成该操作所花费的时间量。随后，您可重新创建或重新启用这些索引，最好利用 PDQ。删除或禁用索引，然后再创建或启用它们，与不这么做比起来，可更快地完成 repack_offline 或 uncompress_offline 操作。

如果 dbspace 曾包含压缩的表，请别删除正在使用“更改数据捕获”（CDC）的 dbspace，因为这可能删除 CDC 仍需要的压缩字典。

重新打包

压缩操作通常在个别数据和剩余页上创建一些空闲空间，但该空间不合并在表或分片的后部。相反，该空间可用于保持新插入的行，表不再增大直到这个空间填满。

仅发生在 online 的压缩操作就地压缩表的行。重新打包操作移动这些行。您可 online 或 offline 执行重新打包操作。online 操作允许在表上发生并发活动。然而，这可导致 phantom rows。（幻象行是在后来回滚的事务期间初始地更改或插入的那些行。）

要避免幻象行，当您可经得起使得其他用户不能访问表或分片时，您可能想要重新打包 offline。例如，您可以在日间执行带有并发活动的压缩操作，然后当预计表上没有并发活动时，在晚间执行 repack_offline 操作。

您不可执行带有 online 操作的 offline 操作。例如，在您可执行压缩与重新打包相结合的操作时，您不可执行压缩与 repack_offline 相结合的操作。如果您想要重新打包 offline，则必须分两步做到：

1. 执行压缩操作。

2. 执行 repack_offline 操作。

类似地，您不可执行 repack_offline 与收缩操作。

在发生重新打包操作时，如果在表中发生轻量级追加（无缓冲区的、无日志记录的插入操作），则重新打包操作不在表或分片的后部完成空间合并。重新打包操作未完成是因为新的 extent 添加在重新打包操作已经发生的位置中，所以不能将空间归还到 dbspace。要完成重新打包进程，您必须在轻量级追加活动完成后运行第二次重新打包操作。这个第二次重新打包操作构建在第一次重新打包操作的成果之上。

收缩

通常在重新打包操作之后执行收缩操作。

不损害表的分配策略妥协，您可稳妥地收缩整个表。例如，如果您有一分片的表，一周的每天一个分片，并为未来使用而预先分配了许多分片，则您可收缩该表，而不损害这个分配策略。如果该表为空，则 GBase 8s 将表收缩到当该表创建时指定的初始 extent 大小。

当您初始化收缩操作时， GBase 8s 缩短 extent，如下：

- 除第一个 extent 之外，将所有 extent 缩短到尽可能小的大小。
- 如果该表完全在第一个 extent 之中（例如，因为该表是空表），则 GBase 8s 不将第一个 extent 收缩到比 extent 大小更小的大小。当以 CREATE TABLE 语句创建该表时指定了这个 extent 大小。

您可使用 ALTER TABLE 语句的 MODIFY EXTENT SIZE 子句来减小当前 extent 大小。您这么做之后，可重新运行收缩操作来将第一个 extent 收缩到新的 extent 大小。

解压缩

对任何未压缩的表或表的分片，解压缩操作无效。

解压缩表或分片之后，您可执行 purge_dictionary 操作来删除那个表或分片的字典。

清除

对表和分片执行 purge_dictionary 操作之前，您必须：

- 解压缩表和分片。

  当您解压缩表或分片时， GBase 8s 将该表或分片的字典标记为不活动的。

- 请确保 Enterprise Replication 函数不需要较早日志的压缩字典。
- 请归档包含带有压缩字典的表或分片的任何 dbspace，即使您已经解压缩了表或分片中的数据且字典不在是活动的。

您还可删除所有压缩字典或在指定日期之前以及那天创建的所有压缩字典。要了解相关信息，请参阅清除压缩字典参数：移除压缩字典（SQL 管理 API）。

示例

下列命令压缩、重新打包和收缩 insurance 数据库中名为 auto 的表中 dbspace 中的行数据和简单大对象。该数据库所有者为 tjones。

```
EXECUTE FUNCTION task("table compress repack shrink","auto",
"insurance","tjones");
```

下列命令仅并行地压缩名为 dental 的表中的行数据。

```
EXECUTE FUNCTION task("table compress rows parallel","dental");
```

下列命令解压缩分区号为 14680071 的分片。

```
EXECUTE FUNCTION task("fragment uncompress","14680071");
```

下列命令仅并行地解压缩分区号为 14680071 的分片中的行数据。

```
EXECUTE FUNCTION task("fragment uncompress rows parallel","14680071");
```

下列命令估计 insurance 数据库中名为 home 的 表的压缩收益，该数据库的所有者为 fgomez。

```
EXECUTE FUNCTION task("table estimate_compression","home",
"insurance","fgomez");
```

下列命令并行地移除名为 auto 的表上挂起的就地更改操作。

```
EXECUTE FUNCTION task("table update_ipa parallel","auto");
```

在您运行该命令之后，数据库服务器显示您可取得的压缩率，以及当前取得的压缩率（ 如果存在的话）。要了解关于该命令输出的信息，请参阅估计压缩操作的输出（SQL 管理 API）。

4.151.2 估计压缩操作的输出（SQL 管理 API）

在您运行估计压缩率的命令之后，数据库服务器显示信息，显示可取得的压缩率的估计，以及当前取得的压缩率（如果存在的话）。

表 1. estimate_compression 命令显示的信息

列	显示的信息
est	这是对使用新的压缩字典可取得的压缩率的估计。该估计是与不压缩对比而节省的空间的百分率。
curr	这是对当前取得的压缩率的估计。这个估计是与不压缩对比而节省的空间的百分率。对未压缩的分片或表，通常会出现 0.0%。
change	通过切换到新的压缩字典您可取得的压缩率中，这是赢得的百分点（或可能损失的百分点，虽然应该很罕见）。这就是 est 与 curr 之间 的差异。 如果表或分片未压缩，则您可用该压缩参数创建压缩字典。如果压缩分片，则在可压缩之前，您必须执行解压缩或 uncompress_offline 操作。
partnum	这是分片的分区号。
coloff	这个值定义这是对行数据的估计，还是对 dbspace 中的简单大对象的估计，如下：

列	显示的信息
	• -1 表明是对行数据的估计 • 正数值表明是对分区简单大对象的估计，由该值标识简单大对象的偏移量。该偏移量是表中的列偏移量，以字节为单位。
table	这是该分片所属的表的全名，格式为 *database:owner.tablename* 如果您正在估计对索引的压缩收益，则在这列中出现索引的全名。

示例

下列输出显示如果您重新压缩第一个分片，则可发生节省空间增加 4%。如果压缩未压缩的第二个分片，则可发生 75.7% 增长。 coloff 列中的值 -1 表明压缩的是行数据。

```
est      curr     change   partnum        coloff   table
--------------------------------------------------------------------------------
75.7%    75.3%     +0.4    0x00200003        -1    insurance:bwilson.auto
75.7%    0.0%     +75.7    0x00300002        -1    insurance:pchang.home
```

下列输出显示对行数据（在第一行中）的压缩估计，和对在偏移量 4 和 60 的简单大对象（在第二和第三行中）的压缩估计。：

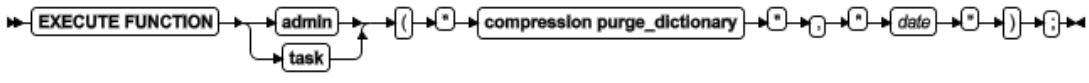
```
est      curr     change   partnum        coloff    table
--------------------------------------------------------------------------------
75.4%    71.5%    +3.9     0x00200002        -1     test:mah.table1
 5.0%    75.0%    +0.0     0x00200002         4     test:mah.table1
75.0%    75.0%    +0.0     0x00200002        60     test:mah.table1
```

对表与分片的压缩估计的输出看上去一样，除了表的数据通常显示表中的所有分片，而分片的输出仅显示指定分片的信息。

4.151.3 清除压缩字典参数：移除压缩字典（SQL 管理 API）

调用带有 compression purge_dictionary 初始命令的 admin() 或 task() 函数来删除所有不活动的压缩字典，或删除在指定的日期之前为压缩的表或分片创建的所有不活动的压缩字典。在您删除为该表和分区创建的任何压缩字典之前，必须解压缩表和分片。

语法：



用法

在对表和分片执行 purge_dictionary 操作之前，您必须：

- 解压缩表和分片。

当您解压缩表或分片时， GBase 8s 标记该表或分片的字典为不活动的。

- 请确保 Enterprise Replication 函数不需要该压缩字典。
- 归档包含带有压缩字典的表或分片的任何 dbspace，即使您已经解压缩该表或分片中的数据，且该字典不再是活动的。

compression purge_dictionary 命令删除所有压缩字典。

带有日期作为第二个参数的 compression purge_dictionary 命令删除在指定的日期之前以及那一天创建的所有压缩字典。您可根据您的语言环境和环境使用可转换到 DATE 日期格式的任何格式的日期。例如，您可指定 03/29/2009、03/29/09 或 Mar 29, 2009。

您还可通过调用以 table 或 fragment 为初始命令，且以 purge_dictionary 为下一个参数的 admin() 或 task() 命令，删除特定的压缩字典。

您不可删除为索引创建的压缩字典。当删除这些索引时，数据库服务器移除这些压缩字典。

下列命令告诉 GBase 8s 移除在 2009 年 7 月 8 日之前和那一天创建的所有字典：

```
EXECUTE FUNCTION task("compression purge_dictionary", "07/08/2009");
```

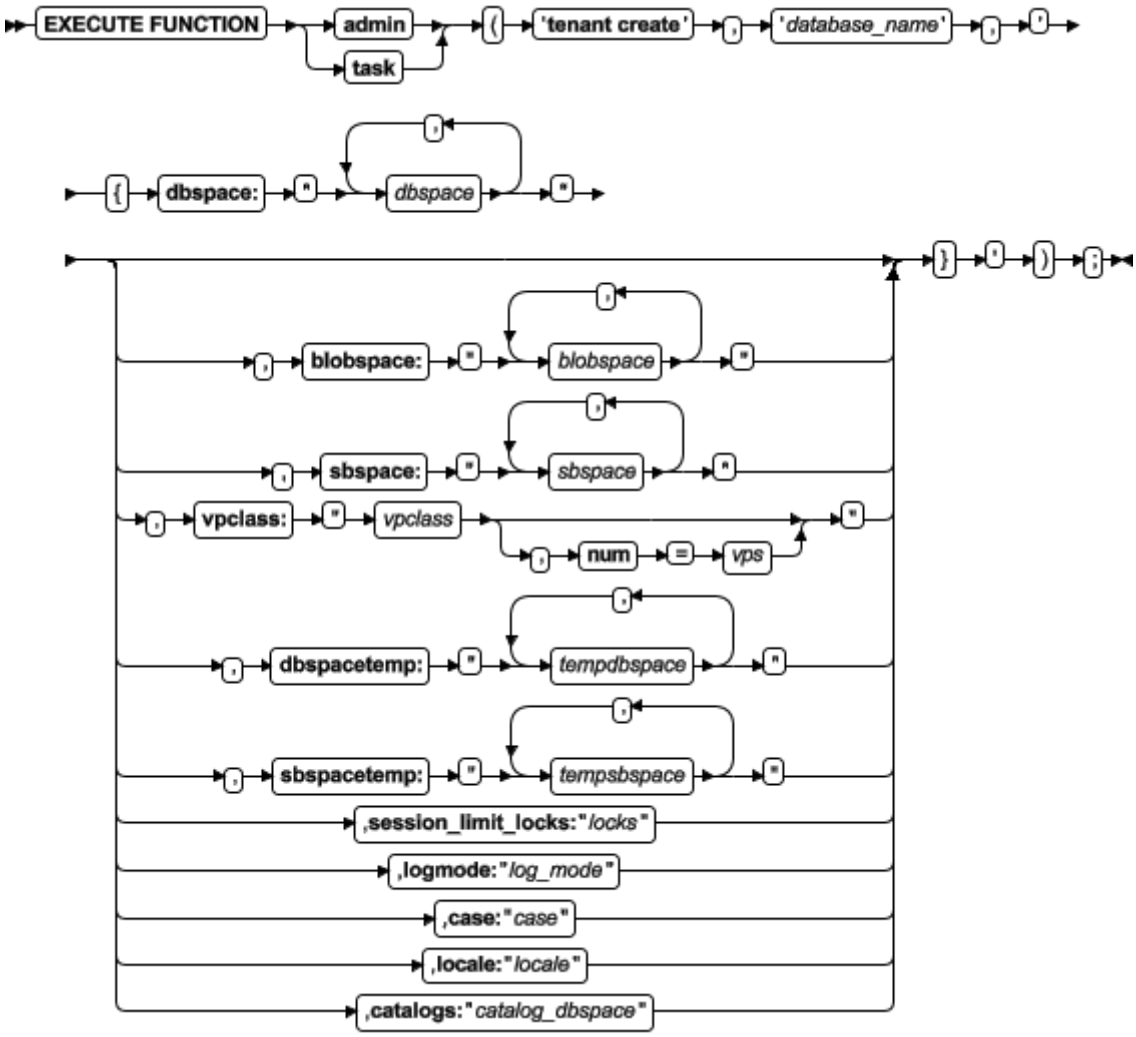
下列命令告诉 GBase 8s 移除 insurance 数据库中名为 auto 的表的不活动字典，该数据库的所有者为 tjones。

```
EXECUTE FUNCTION task("table purge_dictionary", "auto", "insurance", "tjones");
```

4.152 tenant create 参数：创建 tenant 数据库（SQL 管理 API）

随同 admin() 或 task() 函数，使用 tenant create 参数来创建 tenant 数据库。

语法



元素	描述	关键考虑
database_name	tenant 数据库名。	在数据库服务器的数据库名中必须是唯一的。该数据库标记为 tenant 数据库。
dbspace	dbspace 的名称。	罗列要在其中存储永久用户数据的一个或多个 dbspace。用逗号分隔 dbspace 名。每一 dbspace 必须存在且为空。 tenant 数据库必须存储在一个或多个专用的 dbspace 中。仅在 tenant 数据库中创建的对象可存储在该专用的 dbspace 中。
blobspace	blobspace 的名称。	如果 tenant 数据库会包含简单大对象，则罗列一个或多个 blobspace。用逗号分隔 blobspace 名称。每一 blobspace 必须存在且为空。 仅存储在 tenant 数据库表中的简单大对象

元素	描述	关键考虑
		可存储在该专用的 blobspace 中。
sbspace	sbspace 的名称。	如果 tenant 数据库会包含智能大对象，则罗列一个或多个 sbspace。用逗号分隔 sbspace。每一 sbspace 必须存在且为空。 智能大对象可包括 BLOB 或 CLOB 数据，以及那些过大以至于不适合在行中的表统计信息。 有些 GBase 8s 特性，诸如 Enterprise Replication 和基本文本搜索，需要 sbspace。 仅存储在 tenant 数据库表中的智能大对象可存储在专用的 sbspace 中。
vpclass	虚拟处理器类的名称。	在其中为 tenant 数据库运行会话线程的 tenant 虚拟处理器类。如果您省略这个属性，则在 CPU 虚拟处理器上运行会话线程。 *vpclass_name* 限定为 8 字符。最多可创建 200 个 tenant 虚拟处理器。 如果 *vpclass_name* 是唯一的，则您可创建新的 tenant 虚拟处理器类。如果存在 *vpclass_name*，则该 tenant 数据库与其他 tenant 数据库共享该类。
vps	要运行的虚拟处理器数。	如果您未包括 num=*vps* 属性，则启动一个虚拟处理器。
tempdbspace	临时 dbspace 的名称。	罗列要在其中存储临时用户数据的一个或多个临时 dbspace。用逗号分隔临时 dbspace 名称。如果省略这个属性，则临时表存储在由 DBSPACETEMP 配置参数或环境变量指定的临时 dbspace 中。 您可通过设置 **DBSPACETEMP** 环境变量取代会话的 dbspacetemp 属性。
tempsbspace	临时 sbspace 的名称。	罗列在其中存储临时智能大对象的一个或多个临时 sbspace。用逗号分隔临时 sbspace 名称。如果您省略这个属性，则临时智能大对象存储在由 SBSPACETEMP 配置参数指定

元素	描述	关键考虑
		的临时 sbspaces 中。
locks	对于没有 DBA 权限的用户，会话的锁的最大数。	*locks* 的值必须为 500 - 2147483647。如果您省略这个属性，则由 SESSION_LIMIT_LOCKS 配置参数设置锁的数目。如果 SESSION_LIMIT_LOCKS 配置参数未设置，则会话的锁的最大数目为 2147483647。 您可通过设置 IFX_SESSION_LIMIT_LOCKS 环境选项取代会话的 session_limit_locks 属性。
log_mode	日志模式定义： **UNBUFFERED** 无缓冲区的数据库日志记录。这是缺省值。 **ANSI** 符合 ANSI 的数据库日志记录。 **BUFFERED** 有缓冲区的数据库日志记录。 **NONE** 无数据库日志记录。	如果您省略这个属性，则日志记录模式为无缓冲区的。
case	大小写定义： **INSENSITIVE** 不区分大小写。这是缺省值。 **SENSITIVE** 区分大小写。	如果您省略这个属性，则数据库不缺分大小写。
locale	数据库的语言环境。	*locale* 的值与 **DB_LOCALE** 环境变量的值相同。 如果您省略这个属性。则由 **DB_LOCALE** 环境变量的值设置语言环境。缺省语言环境为 US English。
catalog_dbspace	存储数据库目录的 dbspace 的名称。必须是由 dbspace 属性罗	如果您省略这个属性，则 dbspace 属性列表中的第一个 dbspace 包含数据库目录。

元素	描述	关键考虑
	列的 dbspace 的名称。	

用法

您必须拥有 DBA 权限或被授予 TENANT 权限来运行这个命令。

随同 admin() 或 task() 函数，运行 tenant create 参数来创建 tenant 数据库。创建该数据库的用户被授予 DBA 权限。您可在 sysadmin 数据库中的 tenant 中查看 tenant 数据库属性。

下列语句创建名为 companyA 的 tenant 数据库：

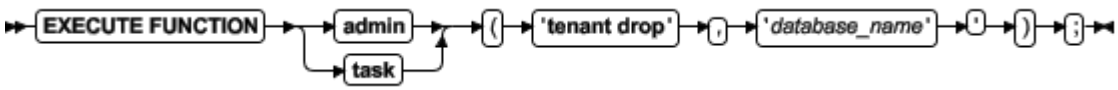
```
EXECUTE FUNCTION task('tenant create', 'companyA',
        '{dbspace:"companyA_dbs1,companyA_dbs2,companyA_dbs3",
        sbspace:"companyA_sbs",
        vpclass:"tvp_A,num=6",
        dbspacetemp:"companyA_tdbs",
        session_limit_locks:"1000",
        logmode:"ansi"}'
        );
```

该 tenant 数据库有三个专用的 dbspace、一个专用的 sbspace、六个 tenant 虚拟处理器、一个专用的临时 dbspace，每一会话限定 1000 个锁，且 logmode 为 ANSI。该 tenant 数据库没有 blobspace，在由 SBSPACETEMP 配置参数指定的 sbspace 中存储临时智能大对象，且区分大小写。

4.153 tenant drop 参数：删除 tenant 数据库（SQL 管理 API）

随同 admin() 或 task() 函数，使用 tenant drop 参数来删除 tenant 数据库。

语法



元素	描述	关键考虑
database_name	tenant 数据库的名称。	必须是现有的 tenant 数据库。

用法

您必须拥有 DBA 权限或已被授予 TENANT 权限来运行这个命令。不可打开其他到该数据库的连接。

删除该数据库中的表和数据。释放 tenant 数据库专用的存储空间。从 sysadmin 数据库中的 tenant 表中移除数据库 tenant 属性。如果不与其他任何 tenant 数据库相关，则删除相关的 tenant 虚拟处理器类。

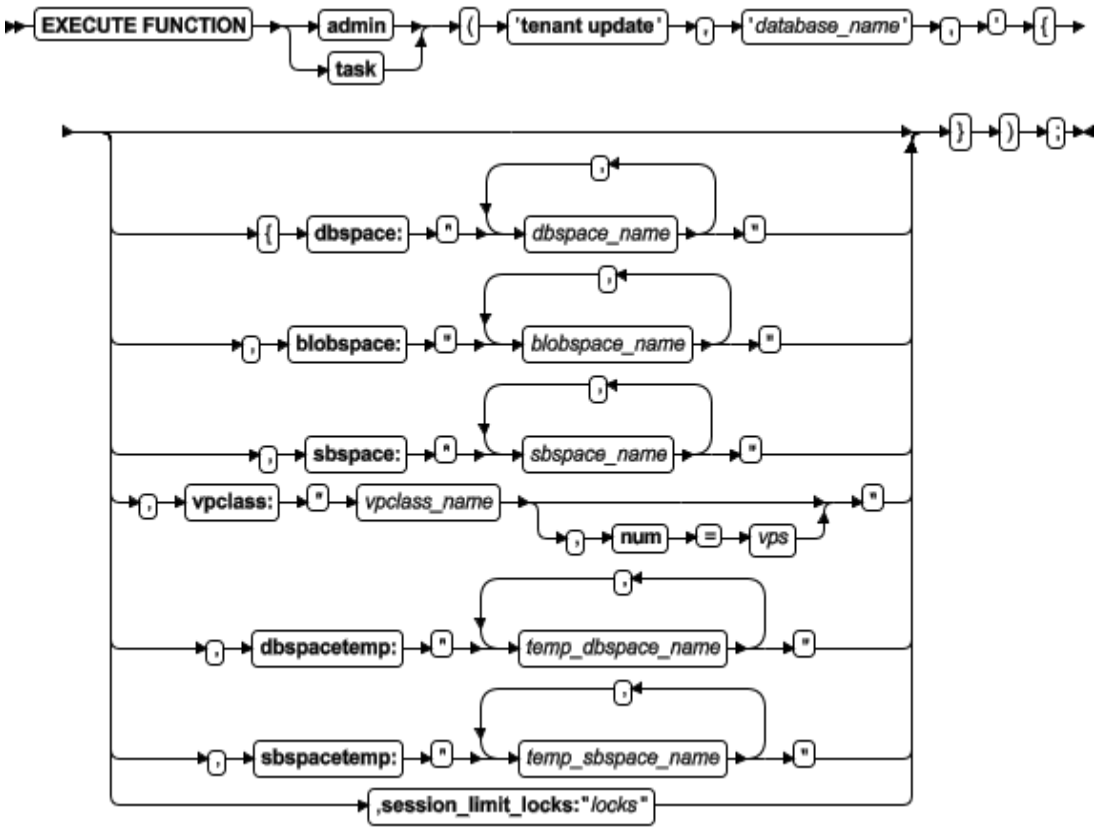
下列语句删除 companyA tenant 数据库：

```
EXECUTE FUNCTION task('tenant drop', 'companyA');
```

4.154 tenant update 参数：更改 tenant 数据库属性（SQL 管理 API）

随同 admin() 或 task() 函数，使用 tenant update 参数来更改 tenant 数据库的属性。

语法



元素	描述	关键考虑
database_name	tenant 数据库的名称。	必须是现有的 tenant 数据库。
dbspace_name	dbspace 的名称。	罗列在其中存储永久的用户数据的一个或多个 dbspace。用逗号分隔 dbspace 名称。每一 dbspace 必须存在且为空。

元素	描述	关键考虑
		将指定的 dbspace 追加到 dbspace 属性的现有的 dbspace 列表。
blobspace_name	blobspace 的名称。	罗列在其中存储简单大对象的一个或多个 blobspace。用逗号分隔 blobspace 名称。每一 blobspace 必须存在且为空。 如果 blobspace 属性有现有的值，则将指定的 blobspace 追加到现有的 blobspace 列表。
sbspace_name	sbspace 的名称。	罗列在其中存储智能大对象的一个或多个 sbspace。用逗号分隔 sbspace 名称。每一 sbspace 必须存在且为空。 如果 sbspace 属性有现有的值，则将指定的 sbspace 追加到现有的 sbspace 列表。
vpclass_name	虚拟处理器类的名称。	*vpclass_name* 限定为 8 个字符。最多可创建 200 个 tenant 虚拟处理器类。 如果 *vpclass_name* 是唯一的，则请您创建新的 tenant 虚拟处理器类。如果 *vpclass_name* 存在，则该 tenant 数据库与其他 tenant 数据库分享该类。 如果 vpclass 属性有现有的值，则请您为 tenant 数据库更改虚拟处理器类。 如果您未包括 num=*vps* 属性，则启动一个虚拟处理器。
tempdbspace_name	临时 dbspace 的名称。	罗列在其中存储临时用户数据的一个或多个临时 dbspace。用逗号分隔临时 dbspace 名称。 如果 dbspacetemp 属性有现有的值，则替换现有的值。
tempsbspace_name	临时 sbspace 的名称。	罗列在其中存储临时智能大对象的一个或多个临时 sbspace。用逗号分隔临时 sbspace 名称。

元素	描述	关键考虑
		如果 sbspacetemp 属性有现有的值，则替换现有的值。
locks	对于没有 DBA 权限的用户，会话的锁的最大数目。	如果 session_limit_locks 属性有现有的值，则替换现有的值。 *locks* 的值必须为 500 - 2147483647。

用法

您必须拥有 DBA 权限或已被授予了 TENANT 权限来运行这个命令。

数据库属性的变更对新的会话生效。

下列语句更新名为 companyA 的 tenant 数据库的属性：

```
EXECUTE FUNCTION task('tenant update', 'companyA',
        '{dbspace:"companyA_dbs4,companyA_dbs5",
        sbspace:"companyA_sbs3",
        vpclass:"tvp_B"}'
        );
```

该 tenant 数据库获得两个 dbspace 和一个 sbspace。更改虚拟处理器类。

附录

数据库服务器文件

数据库服务器文件创建在缺省目录下或配置参数指定的相关目录下。数据库管理员可能需要编辑或检查被数据库服务器所使用的文件的内容。

- 表 1 列出了您可能需要查看、复制、编辑、移动或删除的数据库服务器文件（除了标记过的）。
- 表 2 列出了只供内部使用的数据库服务器文件。您不能编辑、移动或删除这些文件。

表 1. 您可以使用的数据库服务器文件. 该表列出了当配置或使用数据库服务器时，您可能要参考或使用的文件。

文件名	目录	用途	创建者
af.*xxx* *xxx* 标识指定的断言失败	$GBASEDBTDIR/tmp(UNIX™) 由 DUMPDIR 配置参数指定	断言失败信息	数据库服务器
ac_msg.log	$GBASEDBTDIR/tmp(UNIX™)	archecker 实用程序的消息日志	数据库服务器
ac_config.std	$GBASEDBTDIR/etc（UNIX）	archecker 参数值的模板	数据库服务器
bar_act.log	/tmp （UNIX） 由 BAR_ACT_LOG 配置参数指定	ON-Bar 活动日志	ON-Bar
bar_dbug.log	/usr/gbasedbt/ （UNIX） 由 BAR_DEBUG_LOG 配置参数指定	ON-Bar debug 日志	ON-Bar
bldutil.*process_id* （UNIX）	/tmp （UNIX）	有关创建 **sysutils** 数据库的错误消息	数据库服务器
buildsmi.out （UNIX）	/tmp （UNIX）	有关创建 **sysmaster** 数据库的错误消息	数据库服务器
concdr.sh	$GBASEDBTDIR /etc/conv	在升级时转换	数据库服务器

文件名	目录	用途	创建者
	（UNIX）	syscdr 数据库格式	
core （UNIX）	数据库服务器启动时的目录	核心转储	数据库服务器
gcore.*xxx* （UNIX）	$GBASEDBTDIR/tmp（UNIX）	断言失败信息	数据库服务器
.gbasedbt （UNIX）	用户主目录	设置个人环境变量	用户
gbasedbt.rc （UNIX）	$GBASEDBTDIR/etc	设置所有用户的缺省环境变量	数据库管理员
InstallServer.log （ Windows ）	C:\temp	数据库服务器安装日志	数据库服务器
ixbar.*servernum*	$GBASEDBTDIR/etc（UNIX）	冷恢复中使用的紧急启动文件	ON-Bar
jvp.log	/urs/gbasedbt 由 JVPLOGFILE 配置参数指定	来自 Java™ 虚拟处理器的消息	数据库服务器
.jvpprops	urs/gbasedbt/extend/krakatoa 由 JVPPROPFILE 配置参数指定	Java VP 属性的样本	安装过程中
oncfg_ *servername*.*servernum*	$GBASEDBTDIR/etc（UNIX）	通过 ON-Bar 全系统恢复的配置信息	数据库服务器
online.log	$GBASEDBTDIR/tmp（UNIX） 由 MSGPATH 配置参数指定	数据库服务器消息日志，包含错误消息和状态信息	数据库服务器
onconfig	$GBASEDBTDIR/conf/（UNIX）	配置信息	数据库管理员或数据库服务器管理员
psm_act.log	/tmp （UNIX） 由 PSM_ACT_LOG 配置参数指定	GBase 8s 主存储管理器的日志文件	ON-Bar
pua.map	$GBASEDBTDIR/gls/etc（UNIX）	用于在 Unicode 私人使用区域（PUA）范围内显示字符映射文件	用户

文件名	目录	用途	创建者
revcdr.sh （UNIX）	$GBASEDBTDIR/etc/conv（UNIX）	将 syscdr 数据库还原至较早的格式	数据库服务器
shmem.*xxx*	$GBASEDBTDIR/tmp（UNIX）DUMPDIR 配置参数指定	断言错误消息	数据库服务器
sm_versions.std	$GBASEDBTDIR/etc（UNIX）	标识使用中的存储管理器	安装过程中
sqlhosts.std	$GBASEDBTDIR/etc（UNIX）	连接信息的样本	安装过程中

表 2. 仅供内部使用的数据库服务器文件. 该表列出了数据库服务器需要的文件。

重要： 不要移动、修改或删除这些文件，除非 GBase 软件支持指示您这么做。

文件名	目录	用途	创建者
illlsrra.xx	$GBASEDBTDIR/lib（UNIX）	数据库服务器和一些实用程序的共享库	安装程序
GBASEDBTTMP	/GBASEDBTTMP(UNIX)	内部文件的临时目录	数据库服务器
.inf.servicename	/GBASEDBTTMP(UNIX)	连接信息	数据库服务器
.infos.dbservername	$GBASEDBTDIR/etc（UNIX）	连接信息	数据库服务器
JVM_vpid	由 JVPLOG 配置参数指定	Java 虚拟机产生的消息	Java 虚拟机
servicename.exp	/GBASEDBTTMP(UNIX) drive:\GBASEDBTTMP（ Windows ）	连接信息	数据库服务器

文件名	目录	用途	创建者
servicename.str	/GBASEDBTTMP(UNIX) drive:\GBASEDBTTMP (Windows)	连接信息	数据库服务器
VP.servername.nnx	/GBASEDBTTMP(UNIX) drive:\GBASEDBTTMP (Windows)	连接信息	数据库服务器

捕获错误

偶然地，一系列事件导致数据库服务器返回预料之外的错误代码。

您可以使用以下诊断工具获取故障排除错误的信息：

- gadmin -I
- 跟踪点
- ifxcollect 工具

使用 gadmin -I 收集诊断信息

要帮助收集附加诊断信息，可以使用 gadmin -I 指示数据库服务器执行 GBase 8s 管理员指南 所描述的诊断信息收集过程。要在遇到错误编号时使用 gadmin -I ，请提供 iserrno 和可选的会话 ID。有关 gadmin 的更多信息，请参阅 gadmin 实用程序。

创建跟踪点

跟踪点在调试用 C 编写的用户定义例程时很有用。您可以创建用户定义的跟踪点以发送有关用户定义例程的当前执行状态的特殊信息。

每个跟踪点具有以下部分：

- trace 类将相关的跟踪点归在一起，以便可同时打开或关闭它们。

  您可以使用名为 _myErrors 的内置跟踪类或创建自己的跟踪类。要创建自己的跟踪类，请向 systracees 系统目录表插入行。
- trace message 是数据库服务器发送给跟踪输出文件的文本。

  您可以将国际化的跟踪消息存储在 systracemsgs 系统目录表中。
- tracepoint threshold 确定何时执行跟踪点。

缺省情况下，数据库服务器将所有跟踪选项置于 tmp 目录中具有以下文件的跟踪输出文件：

session_num.trc

有关跟踪用户定义的例程的更多信息，请参阅 GBase 8s DataBlade API 程序员指南。

使用 ifxcollect 工具收集数据

如果有必要解决指定的问题，可以使用 ifxcollect 工具收集诊断数据。例如：断言失败。您也可以通过文件传输协议（FTP）指定传输收集的数据的选项。

ifxcollect 工具在 $GBASEDBTDIR/bin 目录中。ifxcollect 目录产生的输出文件在 $GBASEDBTDIR/isa/data 目录中。

每个目录和子目录收集的数据类型都会在 $GBASEDBTDIR/isa/ 目录中的 XML 文件中进行预定义。这些 XML 文件可以被修改以添加或移除指定的命令。

重要： 这些 XML 文件可以包含重写指定数据收集选项的命令。例如：一个 XML 文件可能包含带有较短秒数重写选项的休眠命令；或者一个 XML 文件可能包含对 gstat -z 的调用。

语法

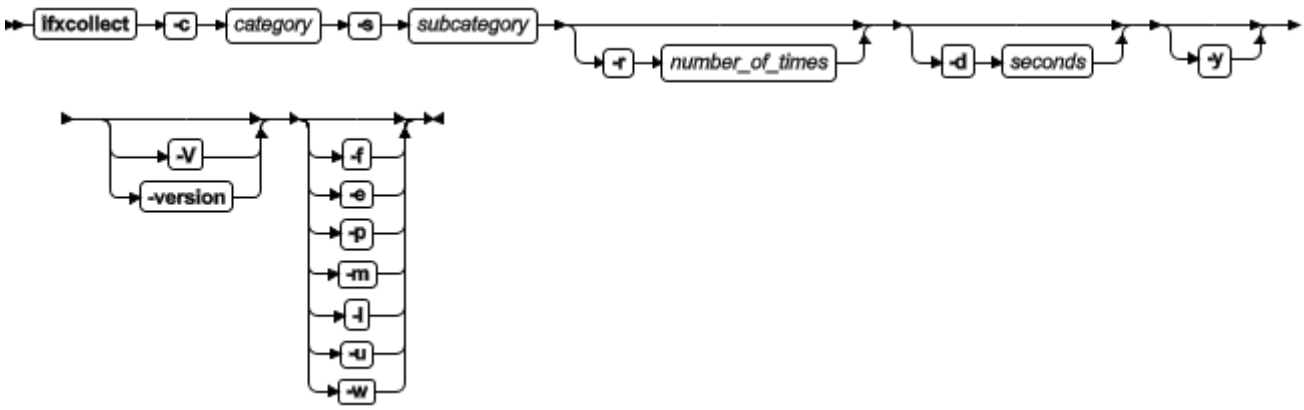


表 1. 数据收集的选项

元素	描述	关键注意事项
-c *category*	告诉服务器收集指定目录中的数据	必须指定要收集的数据的目录
-s *category*	告诉服务器收集指定子目录中的数据	必须指定要收集的数据的子目录
-r *number of times*	指定重复收集数据的次数	可选。缺省值为 1
-d *number of seconds*	指定在收集操作中停止的次数	可选。缺省值为 0
-y	导致数据库服务器对所有提示自动回应是	可选.

元素	描述	关键注意事项
-V	显示软件版本号和序列号	可选 请参阅 获取实用程序的版本信息
-version	显示构件版本、主机、操作系统、数量、日期和 GLS 版本	可选。 请参阅 获取实用程序的版本信息

表 2. FTP 选项（如果传输数据）

元素	描述	关键注意事项
-f	FTP 项收集	数据传输需要
-e *email address*	电子邮件地址	数据传输需要
-p *the PMR number*	PMR 编号	数据传输需要
-m *machine name*	要连接的机器	数据传输需要
-l *directory*	包含数据的目录	数据传输需要
-u *user name*	FTP 的用户名	数据传输需要
-w *password*	FTP 密码	数据传输需要

用法

下表显示了您的目录中可以使用的目录和子目录的组合。

表 3. 目录和子目录组合

目录和子目录	解释
-c ids -s general	收集与所有 GBase 8s 产品有关问题的一般数据
-c af -s general	收集断言失败的一般数据
-c er -s general	收集 Enterprise Replication 的一般数据

目录和子目录	解释
-c er -s init	收集 Enterprise Replication 初始化问题的一般数据
-c performance -s general	收集性能说明数据
-c performance -s cpu	收集 CPU 利用率问题的数据
-c gbackuprestore -s archive_failure	收集 gbackuprestore 归档失败的数据
-c gbackuprestore -s restore_failure	收集 gbackuprestore 恢复失败的数据
-c gtape -s archive_failure	收集 gtape 归档失败的数据
-c gtape -s restore_failure	收集 gtape 恢复失败的数据
-c connection -s failure	收集连接失败的数据
-c connection -s hang	收集连接挂起的数据
-c cust -s prof	收集客户概要文件信息

要查看所有 ifxcollect 实用程序，在命令提示中输入 ifxcollect

示例

要收集一般断言失败的信息，请运行该命令：

ifxcollect －c af －s general

要收集与 CPU 利用率相关的性能问题的信息，请运行该命令：

ifxcollect －c performance －s cpu

要包含 FTP 信息的话，指定如下所示的附加信息：

-f -e user_name@company_name.org -p 9999.999.999

-f -m machine -l /tmp -u user_name -w password

事件警报

数据库服务器提供一个根据发生在数据库服务器环境中的事件自动触发管理操作的机制。该机制就是事件警报功能。

事件可能是参考性的（例如： 备份 已完成）或可以指示需要您注意的错误条件（例如：无法分配内存）。

使用 ALARMPROGRAM 捕获事件

使用 alarmprogram.sh（在 UNIX™ 上），或使用 alarmprogram.bat shell 脚本（在 Windows™ 上）处理事件警报和启动自动的日志备份。有关设置指示信息，请参阅 ALARMPROGRAM 配置参数。

要仅自动化逻辑日志备份，提供了 2 个现成脚本：log_full.[sh|bat] 和 no_log.[sh|bat] ，将 ALARMPROGRAM 设置为该脚本的完整路径名。有关信息，请参阅 ALARMPROGRAM 配置参数。

设置 ALRM_ALL_EVENTS

可以设置 ALRM_ALL_EVENTS 以指定 ALARMPROGRAM 是对记录到 MSGPATH 的所有事件还是仅对指定的值得通知的事件（事件严重性大于 1）运行。

编写自己的警报脚本

另一种方法是，您可以编写包含事件警报参数的自己的 shell 脚本、batch 文件或二进制程序。当事件发生时，数据库服务器调用该可执行文件并向它递事件警报参数（参阅 表 1）。例如：您的脚本可以使用 _id 和 _msg 参数在发生表失败时执行管理操作。请将 ALARMPROGRAM 设置为该可执行文件的完整路径名。

定制 ALARMPROGRAM 脚本

您可以根据您自己的环境定制 ALARMPROGRAM 脚本。

该邮件实用程序必须已存在。

请按照以下步骤定制 alarmprogram.[sh|bat] 脚本。您可以使用 alarmprogram.[sh|bat] 而非 log_full.[sh|bat] 自动化日志备份。

定制 ALARMPROGRAM 脚本：

1. 将 ADMINMAIL 值更改为数据库服务器管理员的邮件地址。
2. 将 PAGERMAIL 值更改为传呼服务电子邮件地址。
3. 设置 MAILUTILITY 参数的值。
   - UNIX™ ： /usr/bin/mail
   - Windows™ ： $GBASEDBTDIR/bin/ntmail.exe
   - Linux™ ： /usr/lib/sendmail -t
4. 要在逻辑日志已满时自动备份它们，请将 BACKUP 更改为 yes 。
5. 要停止自动化日志备份，请将 BACKUP 更改为 yes 除外的任何值。
6. 在 ONCONFIG 文件中，将 ALARMPROGRAM 设置为 alarmprogram.[sh|bat] 的网路径名。
7. 重新启动数据库服务器。

严重性为 1 或 2 的警报不将任何消息写入消息日志也不发送电子邮件。严重性为 3 或更高的警报向数据库管理员发送电子邮件。严重性为 4 和 5 还通过电子邮件通知传呼机。

警报脚本中的前台操作预防

要确保连续的服务器可用性，不要在警报脚本中运行某些前台操作。

当数据库调用一个警报脚本时，服务器有时在诉讼前等待脚本完成。例如：

- 当由于致命错误而调用警报时，服务器会等待该脚本完成向错误日志写信息操作。在某些情况下，警报事件 5 和 6 会在前台运行。
- 一些 Enterprise Replication 事件警报会在前台运行，例如：事件警报 31 、34 、37 和 39 。

因为服务器可能会等待警报程序运行完成，所以不要在该警报脚本中的前台执行以下操作：

- 强制用户与服务器断开连接的 gadmin 命令，例如：gadmin -u 或 gadmin -yuk 。这种类型的 gadmin 命令可能会在服务器和警报程序中导致死锁，因为服务器可能等待警报脚本完成而执行 gadmin 命令的警报脚本正等待用户会话关闭，并且这些会话中的任意会话正在运行警报脚本。
- 此操作结束可能要花费较长时间或它具有高可变的运行时间。要花费长时间的操作可能导致在操作运行时服务器出现类似没有响应的现象。

如果您需要在警报脚本中执行以上操作，请在使用以下其中之一的操作系统实用程序的后台下执行：

在 UNIX™ 上：使用 nohup 实用程序。例如：nohup gadmin -yuk & 指示 nohup 继续运行此命令，即使它的父命令终止并且有符号 & 在后台运行此命令，因此它不会阻塞警报程序脚本的执行。

在 Windows™ 上：使用带有 /B 标志的 start 实用程序。例如：start /B gadmin -yuk 。

解释事件警报消息

数据库服务器报告给消息日志的某些事件导致它调用警报程序。类消息指示数据库服务器所报告的事件。

数据库服务器在消息日志中报告非零退出代码。在警报程序中，将 EXIT_STATUS 变量设置为 0 表示成功完成，设置另一个数字表示失败。

例如：如果线程尝试获得锁，但使用中的锁的最大数量已达到，那么数据库服务器向消息日志写入以下消息：

```
10:37:22  Checkpoint Completed:  duration was 0 seconds.
10:51:08  Lock table overflow - user id 30032, rstcb 10132264
10:51:10  Lock table overflow - user id 30032, rstcb 10132264
10:51:12  Checkpoint Completed:  duration was 1 seconds.
```

当数据库服务器调用 alarmprogram.sh 或 alarmprogram.bat 程序或您自己的警报程序时，它生成一条描述事件严重性和类的消息。如果严重性大于 2 ，那么消息采用以下格式：

操作	消息

操作	消息
一个合理严重的服务器事件	Severity: 3 Class ID: 21 Class msg: Database server resource overflow: 'Locks'. Specific msg: Lock table overflow - user id 30032, rstcb 10132264 See Also: # optional message Event ID: 21005
该消息出现在每个 e-mailed 消息末尾	This e-mail was generated by the server ALARMPROGRAM script on *servername* because something untoward just happened to *eventname*.

ph_alert 表中的事件

所有生成的事件警报都将添加到 sysadmin 数据库的 ph_alert 表中。

您可以在本地或远程服务器上查询 ph_alert 表以查看该服务器最近的事件警报。您可以编写基于 ph_alert 表的 SQL 脚本以处理事件警报而不是使用由 ALARMPROGRAM 配置参数控制的脚本。

缺省情况下，警告在 ph_alert 表中保留 15 天，之后将会清除。

示例

以下示例显示了 ph_alert 表中的一个事件警报：

```
SELECT * FROM ph_alerts WHERE alert_object_type=ALARM;

        id                  34
        alert_task_id       18
        alert_task_seq      10
        alert_type          INFO
        alert_color         YELLOW
        alert_time          2010-03-08 12:05:48
        alert_state         NEW
        alert_state_chang+  2010-03-08 12:05:48
        alert_object_type   ALARM
        alert_object_name   23
        alert_message       Logical Log 12 Complete, timestamp: 0x8e6a1.
        alert_action_dbs    sysadmin
        alert_action
        alert_object_info   23001
```

事件警报参数

事件警报有描述每个事件的五个参数。

下表列出了事件警报的部分参数。

表 1. 事件警报参数

参数	描述	数据类型
severity	事件严重性	integer
class_id	分类已发生的事件的类型的数字标识	integer
class_msg	描述事件类型的简要消息	string
specific_msg	描述已发生的事件的特定消息	string
see_also	包含与事件有关的其他信息的参考文件	string
uniqueid	特定消息的唯一事件标识	bigint

事件严重性

事件严重性代码是一个事件严重性的数字指标。每个事件在消息日志中都包含一个严重性代码。传递给警报程序的第一个参数是事件严重性代码。在 ph_alert 表中，事件严重性通过警报的颜色和警报类型的组合来表示。下表列出了事件严重性代码。

表 2. 事件严重性代码

严重性	描述
1	**不值得通知的**。事件（例如：消息日志中的日期更改）不报告给警报程序，除非启用 ALRM_ALL_EVENTS 配置参数。 在 ph_alert 表中，该警报颜色是绿色，警报类型是 INFO
2	**参考性**。未发生任何错误，但某些例程事件成功完成了（例如：checkpoint 或日志备份已完成）。 在 ph_alert 表中，该警报颜色是黄色，警报类型是 INFO
3	**注意**。该事件不危及数据或阻止系统的使用；然而，它值得引起注意（例如：镜像对当中的一个 chunk 关闭）。向系统管理员发送电子邮件。 在 ph_alert 表中，该警报颜色是黄色，警报类型是 WARNING
4	**紧急**。发生了意外，可能会危及数据或数据访问。例如：断言失败或 gcheck 报告数据损坏。请立即执行操作，当此类事件严重性发生时，将通知系统管理员。

严重性	描述
	在 ph_alert 表中，该警报颜色是红色，警报类型是 ERROR
5	**致命**。发生了意外并已导致数据库服务器失败。当此类事件严重性发生时，将通知系统管理员。 在 ph_alert 表中，该警报颜色是红色，警报类型是 ERROR

事件类 ID

事件类 ID 是一个整数，它标识导致数据库服务器运行警报程序的事件。该事件 ID 是数据库服务器显示在您警报程序中的第二个参数。

该事件 ID 存储在 ph_alert 表的 alert_object_name 列中。

类消息

类消息是简要描述的文本消息或公告，该事件导致数据库服务器执行警报程序。该类消息是数据库服务器显示在您警报程序中的第三个参数。

特定消息

特定消息是详细描述导致数据库服务器运行警报程序的事件的文本消息。特定消息是数据库服务器显示的警报程序中的第四个参数。对于多个报警，该消息的文件与写入消息日志的该事件消息的文本相同。

特定消息存储在 ph_alert 表的 alert_message 列中。

另见途径

对于某些事件，数据库服务器在该事件发生时将其信息写入文件。该内容中的路径名是指数据库服务器写入其他信息的文件的路径名。

事件 ID

事件 ID 对于每个给定的消息是唯一的编号。可以在自定义警报处理脚本中使用事件 ID ，以创建对特定事件的响应。

事件 ID 存储在 ph_alert 表的 alert_object_info 列中。

事件警报 ID

事件警报的类 ID 标识事件的类型。事件 ID 指示特定的事件。

下表列出了事件警报 ID 和消息或从哪里可以获得更多的信息。许多警报都有其他解释和用户操作。许多触发事件警报的问题也在联机消息日志中显示。消息日志的位置通过 MSGPATH 配置参数指定。

ID	严重性	消息	解释
类 ID： 1 事件 ID： 1001	3	类消息： 表失败：'*dbname:"owner".tabname*' 特定消息： '*object*' 页分配发生错误	数据库服务器在表或索引页分配期间检测到不一致。 **联机日志：**断言失败或声明警告（附上问题的详细信息）。 **数据库状态：**取决于发现问题的性质。 **用户操作：**审查 online.log 文件以获得合适的操作。您需要在消息中标识的 '*dbname:"owner".tabname*' 中执行 gcheck 实用程序。偶然地，数据库服务器会自动解决该问题，并且将解决方法标示在 online.log 文件中
类 ID： 1 事件 ID： 1002	3	类消息： 表失败：'*dbname:"owner".tabname*' 特定消息： '*object*' 行分配失败	数据库服务器在表或索引行分配期间，检测到不一致。 **联机日志：**断言失败或声明警告（附上问题的详细信息）。 **数据库状态：**取决于发现问题的性质。 **用户操作：**审查 online.log 文件以获得合适的操作。您需要在消息中标识的 '*dbname:"owner".tabname*' 中执行 gcheck 实用程序。
类 ID： 1 事件 ID： 1003	3	类消息： 表失败：'*dbname:"owner".tabname*' 特定消息： '*object*' slot 分配发生错误	数据库服务器在行处理过程中检测到不一致。 **联机日志：**断言失败或声明警告（附上问题的详细信息）。 **数据库状态：**取决于发现问题的性质。

ID	严重性	消息	解释
			用户操作：审查 online.log 文件以获得合适的操作。您需要在消息中标识的 *'dbname:"owner".tabname'* 中执行 gcheck 实用程序。
类 ID： 1 事件 ID： 1004	3	类消息: 表失败：*'dbname:"owner".tabname'* 特定消息： 内部错误阻止数据库服务器在该 tblspace 中找到下一个可能数据页	数据库服务器在表或索引的行分配期间检测到位图页不一致。 **联机日志：**断言失败或声明警告（附上问题的详细信息）。 **数据库状态：**取决于发现问题的性质。 **用户操作：**审查 online.log 文件以获得合适的操作。您需要在消息中标识的 *'dbname:"owner".tabname'* 中执行 gcheck 实用程序。
类 ID： 1 事件 ID： 1005	3	类消息： 表失败： *'dbname:"owner".tabname'* 特定消息： 删除错误的 TBLSpace ，请求删除的 *tblspace_name* != 实际删除的 *tblspace_name*	尝试删除表时，数据库服务器检测到请求删除的表和现有表不匹配。没有表被删除。 **联机日志：**断言失败或声明警告（附上问题的详细信息）。 **数据库状态：**联机 **用户操作：**审查 online.log 文件以获得合适的操作。您需要在消息内标识的 *'dbname:"owner".tabname'* 中执行 gcheck 实用程序。
类 ID： 1 事件	3	类消息: 表失败： *'dbname:"owner".tabname'* 特定消息： 内部错误，由于数据损坏阻止了数据库服务	在表或索引更改位图页时，数据库服务器遇到可能的数据损坏错误。 **联机日志：**断言失败或声明警告（附上问题的详细信息）。

ID	严重性	消息	解释
ID： 1006		器更改此分区的位图页	**数据库状态：**联机 **用户操作：**审查 online.log 文件以获得合适的操作。您需要在消息中标识的 *'dbname:"owner".tabname'* 中执行 gcheck 实用程序。
类 ID： 1 事件 ID： 1007	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 内部错误，由于数据库服务器仍在处理损坏的位图页的转换。	在表或索引更改位图页时，数据库服务器遇到内部位图页的不完整修改。 **联机日志：**断言失败或声明警告（附上问题的详细信息）。 **数据库状态：**联机 **用户操作：**注意所有情况并联系 GBase 软件支持。
类 ID： 1 事件 ID： 1008	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 内部错误，由不可转换的位图页导致。	在表或索引更改位图页时，数据库服务器遇到内部位图页修改没有完成的情况。 **联机日志：**断言失败或声明警告（附上问题的详细信息）。 **数据库状态：**联机 **用户操作：**注意所有情况并联系 GBase 软件支持。
类 ID： 1 事件 ID： 1009	4	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： *object* 中页面检查错误	数据库服务器在检查页面读取到内部缓冲区时，检测到不一致。 **联机日志：**断言失败或声明警告（附上问题的详细信息）。 **服务器状态：**联机或脱机，取决于问题的严重程度。 **用户操作：**遵循联机日志中的建

ID	严重性	消息	解释
			议。通常，对类消息或数据库提到的表执行 gcheck -cD 命令。
类 ID：1 事件 ID：1010	4	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 无效的 rowid *rowid*	数据库服务器检测到一个无效的行 ID。 **联机日志：**声明警告，附上发现该问题的位置的描述。 **服务器状态：**联机 **用户操作：**通过在类消息或数据库提到的表中运行 gcheck -cI 命令以修复索引。
类 ID：1 事件 ID：1011	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 关闭 TBLSpace *tblspace_name*	数据库服务器确定该表或索引已关闭。 **联机日志：**断言失败或声明警告（附上问题的详细信息）。 **数据库状态：**联机 **用户操作：**无。数据库服务器将会自动纠正该问题。
类 ID：1 事件 ID：1012	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 无法重建索引 *index_name* ，由于 partnum *partition_number*, iserrno = *error_number*	数据库服务器遇到阻止重建索引的错误。 **联机日志：**断言失败或声明警告（附上问题的详细信息）。 **数据库状态：**联机 **用户操作：**审查 online.log 文件的索引信息，然后手动删除并重建该索引。
类 ID：1 事件	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息：	数据库服务器无法初始化读取操作集的内部结构。 **联机日志：**断言失败或声明警告（附上问题的详细信息）。

ID	严重性	消息	解释
ID： 1013		数据库服务器尝试初始化读取操作集的类型时发生内部错误	**数据库状态：**联机 **用户操作：**审查 online.log 文件的 ISAM 错误代码、表和数据库信息。并再次尝试该操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持。
类 ID： 1 事件 ID： 1014	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 数据库服务器在尝试从 tblspace 页读取记录时发生内部错误	数据库服务器从表或索引读取记录时遇到内部错误。 **联机日志：** 声明带有数据库和表详细信息的警告。 **数据库状态：**联机 **用户操作：**审查 online.log 文件的表和数据库信息。并再次尝试该操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持
类 ID： 1 事件 ID： 1015	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 数据库服务器尝试读取当前记录时发生内部错误	数据库服务器从表或索引读取记录时遇到内部错误。 **联机日志：**声明带有数据库和表详细信息的警告。 **数据库状态：**联机 **用户操作：**审查 online.log 文件的表和数据库信息。并再次尝试该操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持。
类 ID： 1 事件	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 数据库服务器在尝试初始化一组读取缓冲	当数据库服务器尝试初始化一组读取缓冲区时触发了一个内部错误。 **联机日志：**声明带有数据库和表详细信息的警告。

ID	严重性	消息	解释
ID： 1016		区时发生内部错误	**数据库状态：**联机 **用户操作：**审查 online.log 文件的表和数据库信息。并再次尝试该操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持。
类 ID： 1 事件 ID： 1017	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试在位图页上设置新方式时发生内部错误	位图页从较早版本的服务器转换时发生内部错误。 **联机日志：**断言失败或声明警告（附上问题的详细信息）。 **数据库状态：**联机 **用户操作：**无
类 ID： 1 事件 ID： 1018	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试将位图页转化成当前格式时发生内部错误	数据库服务器无法纠正位图页转换期间发生的错误。 **联机日志：** 断言失败或声明警告（附上问题的详细信息）。 **数据库状态：**联机 **用户操作：**注意所有条件，审查 online.log 文件的其他信息，并联系 GBase 软件支持。
类 ID: 1 事件 ID： 1019	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器在轻量级追加操作中尝试修改位图页时，发生内部错误	数据库服务器在轻量级追加操作过程中遇到内部错误并且无法定位需求的位图页。 **联机日志：**断言失败或声明警告（附上问题的详细信息）。 **数据库状态：**联机 **用户操作：**再次尝试该操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持。
类	3	类消息：	当数据库服务器正在执行轻度

ID	严重性	消息	解释
ID： 1 事件 ID： 1020		表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器在尝试执行轻度扫描操作时发生内部错误	扫描操作时，遇到了内部错误。 **联机日志：**断言失败或声明警告（附上问题的详细信息）。 **数据库状态：**联机 **用户操作：**再次尝试该操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持。
类 ID： 1 事件 ID： 1021	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 数据库服务器在尝试执行轻度扫描 I/O 操作时发生内部错误	当数据库服务器正在执行轻度扫描操作时，遇到了内部错误。 **联机日志：**断言失败或声明警告（附上问题的详细信息）。 **数据库状态：**联机 **用户操作：**再次尝试该操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持。
类 ID： 1 事件 ID： 1022	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试验证轻量级追加缓冲区时，报告了内部错误	当数据库服务器正在执行轻度扫描时遇到了内部错误。 **联机日志：**断言失败或声明警告（附上问题的详细信息）。 **数据库状态：**联机 **用户操作：**再次尝试该操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持。
类 ID； 1 事件 ID：	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试在轻量级追加缓冲区中往页中写入下一条记录时，报告了内部错	当数据库服务器正在执行轻度扫描时遇到一个内部错误。 **联机日志：**断言失败或声明警告（附上问题的详细信息）。 **数据库状态：**联机

ID	严重性	消息	解释
1023		误	**用户操作：**再次尝试该操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持。
类 ID： 1 事件 ID： 1024	4	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试打开 tblspace 的轻量级追加时报告了一个内部错误	数据库服务器在 tblspace 中进行轻量级追加操作时遇到了内部错误。 **联机日志：**断言失败 **数据库状态：**联机 **用户操作：**再次尝试该操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持。
类 ID： 1 事件 ID： 1025	4	类消息： 表失败 *'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试加载轻量级追加操作的首个位图页时，报告了内部错误	数据库服务器在轻量级追加操作中遇到了内部错误。 **联机日志：**断言失败：轻量级追加（重做/撤销）：无法找到位图页 **数据库状态：**联机 用户操作： 再次尝试该操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持。
类 ID： 1 事件 ID： 1026	4	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试将轻量级追加操作写入已高速缓存的位图页时，报告了内部错误	数据库服务器在轻量级追加操作中遇到了内部错误。 **联机日志：**断言失败 **数据库状态：**联机 **用户操作：**再次尝试该操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持。
类 ID：	2	类消息： 表失败：*'dbname:"owner".tabname'*	数据库服务器检测到一个内部死锁情况。 **联机日志：**声明标识该数据库和

ID	严重性	消息	解释
1 事件 ID： 1027		特定消息： 数据库服务器中的锁管理器捕获到内部死锁情况	表陷入死锁的警告。 **数据库状态：**联机 **用户操作：** 无
类 ID： 1 事件 ID： 1028	2	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 数据库服务器中的锁管理器捕获到内部死锁情况	数据库服务器检测到一个内部死锁情况。 **联机日志：**声明标识该数据库和表陷入死锁的警告。 **数据库状态：**联机 **用户操作：** 无
类 ID： 1 事件 ID： 1029	4	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 数据库服务器尝试将 tblspace 中的逻辑页号映射到它在 chunk 中位置时，报告了内部错误	数据库服务器无法访问表，因为逻辑页面和其逻辑页号的不一致。 **联机日志：**断言带有页面信息的失败。 **服务器状态：**联机 **用户操作：**在类消息中提到的表中或在数据库中运行 gcheck -cDI 命令，修复报告的问题。然后尝试原操作如果操作又失败，那么注意所有情况并联系 GBase 软件支持。
类 ID： 1 事件 ID： 1030	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器阐释分配变更的信息时，报告了内部错误	当数据库服务器尝试读取内部磁盘结构时，遇到了内部错误。 **联机日志：**声明带有问题详细信息、表和数据库信息的警告。 **数据库状态：**联机 **用户操作：**再次尝试该操作。如果操作又失败，那么注意所有情

ID	严重性	消息	解释
			况并联系 GBase 软件支持。
类 ID： 1 事件 ID： 1031	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试准备要在压缩行上执行的操作列表时，报告了内部错误	当数据库服务器尝试创建内部操作列表把行从压缩版本变换为最近行解压版本时，遇到内部错误。 **联机日志：**声明带有问题详细信息、表和数据库信息的警告。 **数据库状态：**联机 **用户操作：**再次尝试该操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持。
类 ID： 1 事件 ID： 1032	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试在基于新操作指向的行的偏移的操作列表中添加操作时，报告了内部错误	当数据库服务器尝试创建将压缩版的行转换为最近解压版的行的内部操作列表时，遇到了内部错误。 **联机日志：**声明带有问题详细信息、表和数据库信息的警告。 **数据库状态：**联机 **用户操作：**再次尝试该操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持。
类 ID： 1 事件 ID： 1033	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器检测到操作列表不一致时，报告了内部错误	当数据库服务器尝试创建将压缩版的行转换为最近解压版的行的内部操作列表时，遇到了内部错误。 **联机日志：**声明带有问题详细信息、表和数据库信息的警告。 **数据库状态：**联机 **用户操作：**再次尝试该操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持。

ID	严重性	消息	解释
类 ID： 1 事件 ID： 1034	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试释放分区头页时，报告了内部错误	当数据库服务器尝试释放分区头页时，遇到了内部错误。数据库服务器没有释放该头页。 **联机日志：**声明带有问题详细信息、表和数据库信息以及要执行的指定的 gcheck 命令的警告。 **服务器状态：**联机 **用户操作：**审查 online.log 文件的信息并执行给定的 gcheck 命令。
类 ID： 1 事件 ID： 1035	3 或 4	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试验证分区头页时，报告了内部错误	由于 tblspace 页验证错误，数据库服务器无法访问表。 **联机日志：**声明带有该表详细信息的警告。 **服务器状态：**联机 **用户操作：**检查有关指定的表的联机日志信息。在表或数据库中执行 gcheck -pt 命令并纠正发现的错误。再次尝试原操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持。
类 ID： 1 事件 ID： 1036	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器在处理更改表命令过程中尝试修改指定的列的列表时，报告了内部错误	正在修改表时 ，数据库服务器处理与该表相关的指定的列时遇到内部错误。 **联机日志：**声明问题的详细描述、表和数据库信息以及要运行的特定的 gcheck 命令的警告。 **服务器状态：**联机 **用户操作：** 审查online.log 文件的信息并运行给定的 gcheck 命令

ID	严重性	消息	解释
类 ID： 1 事件 ID： 1037	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试记录更改完成并从 tblspace 的头页移除相关联的版本信息时，报告了内部错误	当数据库服务器尝试修改表时遇到了内部错误。 **联机日志：**声明电源问题消息描述、表和数据库信息以及要运行的特定的 gcheck 命令的警告。 **服务器状态：**联机 **用户操作：** 审查online.log 文件的信息并运行给定的 gcheck 命令。
类 ID： 1 事件 ID： 1038	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器检测到缓冲区不一致时，报告了内部错误	数据库服务器在对它所操作的内部缓冲区进行一致性检查时，遇到了内部错误。 **联机日志：**声明电源问题消息描述、表和数据库信息以及要运行的特定的 gcheck 命令的警告。 **服务器状态：**联机 **用户操作：** 再次尝试该操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持
类 ID： 1 事件 ID： 1039	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试构造单个元组转发的行时，报告了内部错误	数据库服务器处理行时遇到内部错误。 **联机日志：**声明带有问题详细信息、表和数据库信息的警告。 **服务器状态：**联机 **用户操作：** 审查 online.log 文件以获得更多信息。再次尝试该操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持
类	3	类消息：	数据库服务器在读取数据的过程中遇到损坏的记录并且它无

ID	严重性	消息	解释
ID： 1 事件 ID： 1040		表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试将数据从一个分区读入读缓存集中时，报告了内部错误	法取回该数据。 **联机日志：** 声明带有错误和参与的 *database:table* 的详细信息的警告。 **服务器状态：**联机 **用户操作：**审查 online.log 文件。该错误的一些实例需要 GBase 软件支持的关注。
类 ID： 1 事件 ID： 1041	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试读取给出 rowid 的数据行时，报告了内部错误	数据库服务器在从索引读取数据的过程中遇到损坏的记录，并且它无法取回该数据。 **联机日志：** 声明带有错误和参与的 *database:table* 的详细信息的警告。 **服务器状态：**联机 **用户操作：** 审查 online.log 文件信息并执行建议的 gcheck 命令。该错误的一些实例需要 GBase 软件支持的关注。
类 ID： 1 事件 ID： 1042	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试将内存中的行更改为最新的模式时，报告了内部错误。	当数据库服务器尝试在已修改的表中将一条记录的旧版转换成新版时，遇到内部错误。 **联机日志：** 声明带有错误和参与的 *database:table* 的详细信息的警告。 **服务器状态：**联机 **用户操作：** 审查 online.log 文件以获得更多信息。再次尝试该操作。如果操作又失败，那么注意所有情况并联系 GBase 软件支持

ID	严重性	消息	解释
类 ID： 1 事件 ID： 1043	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试撤销位图页的修改时，报告了一个内部错误	当数据库服务器尝试还原更改内部位图页的操作时，遇到内部错误。 **联机日志：** 声明带有错误和参与的 *database:table* 的详细信息的警告。 **服务器状态：**联机 **用户操作：** 审查 online.log 文件以获得更多信息，并执行建议的 gcheck 命令。
类 ID： 1 事件 ID： 1044	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试撤销从 tblspace 的头页添加指定列描述符这一操作时，报告了内部错误	数据库服务器在尝试还原添加信息到用于跟踪表的内部结构这一操作时，遇到了内部错误。 **联机日志：** 声明带有错误和参与的 *database:table* 的详细信息的警告。 **服务器状态：**联机 **用户操作：** 审查 online.log 文件以获得更多信息，并执行建议的 gcheck 命令。
类 ID： 1 事件 ID： 1045	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试撤销添加的分区的新版本时，报告了内部错误。	数据库服务器在尝试还原添加信息到用于跟踪表的内部结构这一操作时，遇到了内部错误。 **联机日志：**声明发生的错误和参与的 *database:table:index* 的详细信息的警告。 **服务器状态：**联机 **用户操作：**审查 online.log 文件以获得更多信息并运行建议的 gcheck 命令。
类	3	类消息：	当数据库服务器尝试给表或索

ID	严重性	消息	解释
ID： 1 事件 ID： 1046		表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试为分区号分配文件描述符时，报告了内部错误	引创建新的文件描述符时，遇到了内部错误。 **联机日志：**声明发生的错误和参与的 *database:table:index* 的详细信息的警告。 **服务器状态：**联机 **用户操作：**审查 online.log 文件以获得更多信息并运行建议的 gcheck 命令。
类 ID： 1 事件 ID： 1047	3	类消息： 表失败：*'dbname:"owner".tabname'* 特定消息： 当数据库服务器尝试释放分区号的文件描述符时，报告了内部错误	当数据库服务器尝试释放与表或索引相关的内部数据结构时，遇到内部错误。 **联机日志：**声明具有错误详细信息和 *database:table* 参与的详细信息的警告。 **服务器状态：**联机 **用户操作：**无。数据库服务器会内部纠正该问题。
类 ID： 1 事件 ID： 1048	3	类消息: 表失败*'dbname:"owner".tabname'* 特定消息： 更改表记录出错	数据库服务器无法修改有就地涂改的表的数据库记录。它无法吸入新版本的记录。 **联机日志：**声明发生的错误和参与的 *database:table:index* 的详细信息的警告。 **服务器状态：**联机 **用户操作：**审查 online.log 文件以获得更多信息并运行建议的 gcheck 命令。
类 ID：	3 或 4	类消息: 索引失败：	数据库服务器无法删除一条记录，因为它没有在索引中找到该记录。

ID	严重性	消息	解释
2 事件 ID： 2001		*'dbname:"owner".tabname:idxname'* 特定消息： 删除 Fragid *fragment_id*，没有在 partnum *partition_number* 中找到要删除的 Rowid *rowid*	**联机日志：**声明删除操作失败和发生问题的索引和表的详细信息。 **服务器状态：**联机 **用户操作：**在指定的表、索引或在数据库中运行 gcheck -cI 命令，并纠正找到的错误。重试原操作，如果操作又失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 2 事件 ID： 2002	3	类消息： 索引失败： *'dbname:"owner".tabname:idxname'* 特定消息： 由于索引的不一致性而发生内部错误，该不一致性阻止数据库服务器确定索引中的第一条记录的位置。	数据库服务器检测到不一致的索引，并将它标记为无法使用。 **联机日志：**声明发生的错误和参与的 *database:table:index* 的详细信息的警告。 **服务器状态：**联机 **用户操作：**审查 online.log 文件以获得更多信息并运行建议的 gcheck 命令。
类 ID： 2 事件 ID： 2003	3	类消息： 索引失败： *'dbname:"owner".tabname:idxname'* 特定消息： 由于索引的不一致性而发生内部错误，该不一致性阻止数据库服务器读取索引中的头页。	数据库服务器检测到不一致的索引，并将它标记为无法使用。 **联机日志：**声明发生的错误和参与的 *database:table:index* 的详细信息的警告。 **服务器状态：**联机 **用户操作：**审查 online.log 文件以获得更多信息并运行建议的 gcheck 命令。
类 ID：	4	类消息: 索引失败：	数据库服务器检测到不一致的索引。

ID	严重性	消息	解释
2 事件 ID： 2004		*'dbname:"owner".tabname-idxname'* 特定消息： *object* 中页面检查错误	**联机日志：**各种消息，取决于在哪检测到该问题。例如：在所有 DBSpace TBLSpace 中执行 'gcheck -cD' 时可能产生不一致。 **服务器状态：**联机 **用户操作：**审查 online.log 文件以获得更多信息并在数据库中运行建议的 gcheck -cD 命令。
类 ID： 2 事件 ID： 2005	3	类消息: 索引失败： *'dbname:"owner".tabname:idxname'* 特定消息： 批处理读取索引时发生内部错误,因为数据库服务器有一个不可用的索引键项。	数据库服务器检测到不一致的索引，并将它标记为无法使用。 **联机日志：**声明发生的错误和参与的 *database:table:index* 的详细信息的警告。 **服务器状态：**联机 **用户操作：**审查 online.log 文件以获得更多信息并运行建议的 gcheck 命令。
类 ID： 2 事件 ID： 2006	3	类消息: 索引失败： 'dbname:"owner".tabname:idxname' 特定消息： *index_page* 日志记录太大以致于无法适应逻辑日志缓冲区。建议 LOGBUFF 的最小值为 *number*	服务器检测到索引的一条日志记录对于配置的逻辑日志缓冲区大小太大。 **联机日志：**声明发生的错误和参与的 *database:table:index* 的详细信息的警告。 **服务器状态：**联机 **用户操作：**审查 online.log 文件以获得更多信息并将 LOGBUFF 的 onconfig 值修改为建议的值。
类	3	类消息:	数据库服务器检测到不一致的

ID	严重性	消息	解释
ID： 2 事件 ID： 2007		索引失败： *'dbname:"owner".tabname:idxname'* 特定消息： 基于本机 '*locale_name*' 的对比失败	索引，并将它标记为无法使用。 **联机日志：**声明发生的错误和参与的 *database:table:index* 的详细信息的警告。 **服务器状态：**联机 **用户操作：**检查 online.log 文件以获得更多信息并运行建议的 gcheck 命令。
类 ID： 2 事件 ID： 2008	3	类消息: 索引失败： *'dbname:"owner".tabname:idxname'* 消息： 对比失败	数据库服务器检测到不一致的索引，并将它标记为无法使用。 **联机日志：**声明发生的错误和参与的 *database:table:index* 的详细信息的警告。 **服务器状态：**联机 **用户操作：**检查 online.log 文件以获得更多信息并运行建议的 gcheck 命令。
类 ID： 2 事件 ID： 2009	4	类消息: 索引失败： *'dbname:"owner".tabname:idxname'* 特定消息： 当数据库服务器尝试添加新的项到索引中时，发生内部错误	数据库服务器无法向索引中添加记录。 **联机日志：**声明指定的索引和建议执行的 gcheck 命令。 **服务器状态：**联机 **用户操作：**检查 online.log 文件以获得更多信息并运行建议的 gcheck 命令。
类 ID： 2 事件	4	类消息: 索引失败： *'dbname:"owner".tabname:idxname'*	数据库服务器由于索引中的不一致性而无法检索该索引中正确的项。 **联机日志：**声明指定的索引和建

ID	严重性	消息	解释
ID： 2010		特定消息： 由于索引中的不一致性(它阻止数据库服务器无法定位索引中正确的项目)而产生了内部错误。	议执行的 gcheck 命令。 **服务器状态：**联机 **用户操作：**检查 online.log 文件以获得更多信息并运行建议的 gcheck 命令。
类 ID： 2 事件 ID： 2011	3	类消息: 索引失败： 'dbname:"owner".tabname:idxname' 特定消息： 无法删除索引 *index_name* 的for partnum *partition_number*, iserrno = *error_number*	数据库服务器检测到不一致的索引，并将它标记为无法使用。 **联机日志：**声明发生的错误和参与的 *database:table:index* 详细信息的警告。 **服务器状态：**联机 **用户操作：**审查 online.log 文件以获得更多信息并运行建议的 gcheck 命令。重试原操作，如果操作又失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 2 事件 ID： 2012	3	类消息: 索引失败： *'dbname:"owner".tabname:idxname'* 特定消息： 当数据库服务器尝试将索引键描述符标记为损坏时，发生了内部错误	数据库服务器检测到不一致索引并将它标记为无法使用。 **联机日志：**声明带有所遇到错误和参与的 *database:table:index* 的详细信息的警告。 **服务器状态：**联机 **用户操作：**审查 online.log 文件以获得更多信息、运行建议的 gcheck 命令、修复检测到的问题，然后重新启用该索引。
类 ID：	4	类消息: 索引失败：	数据库服务器无法从索引中删除记录。 **联机日志：**声明指定的索引和建

ID	严重性	消息	解释
2 事件 ID： 2013		*'dbname:"owner".tabname:idxname'* 特定消息： 当数据库服务器尝试从索引中删除一个项时，发生了内部错误	议执行的 gcheck 命令。 **服务器状态：**联机 **用户操作：**检查 online.log 文件以获得更多信息并运行建议的 gcheck 命令。
类 ID： 3 事件 ID： 3001	3	类消息: Blob 失败：*'dbname:"owner".tabname'* 特定消息： blob 描述符中的 tb_sockid 损坏。当前的表是 *'dbname:"owner".tabname'*	
类 ID： 3 事件 ID： 3002	3	类消息: Blob 失败：*'dbname:"owner".tabname'* 特定消息： 不正确的 BLOB 戳记	
类 ID： 3 事件 ID： 3003	4	类消息: Blob 失败：*'dbname:"owner".tabname'* 特定消息： 在 *dbspace_name* BLOB 页面检查错误	当数据库服务器检查在磁盘和内存间移动的页面时，发生失败。 **联机日志：**声明该错误的描述。 **服务器状态：**联机 **用户操作：**再次尝试该操作。如果操作又失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID：	3	类消息: Blob 失败：*'dbname:"owner".tabname'*	

ID	严重性	消息	解释
3 事件 ID： 3004		特定消息： 当尝试从表读取 blob 时，发生内部错误	
类 ID： 3 事件 ID： 3005	3	类消息: Blob 失败：*'dbname:"owner".tabname'* 特定消息： 当尝试从表复制 blob 时，发生内部错误	
类 ID： 4 事件 ID： 4001	4	类消息: Chunk 脱机，但镜像是活动的：*chunk_number* 特定消息： I/O 错误，*error_number* Chunk *'chunk_number'* -- 脱机	读取或写入 chunk 时发生错误。数据库服务器是该 chunk 脱机并切换到在活动镜像 chunk 上以执行所有 I/O 操作。 **联机日志：**声明发生的错误的描述。 **服务器状态：**联机 **用户操作：**检查联机日志以获得更多信息并修复该错误。执行 gspaces -s 命令来恢复脱机的 chunk 。再次尝试该操作。如果操作失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 4 事件 ID：	3 或 4	类消息: Chunk 脱机，但镜像是活动的：*chunk_number* 特定消息： 由于在物理 I/O 操作期间 chunk 没有打	数据库服务器无法访问 chunk ，并切换到在活动镜像 chunk 执行所有 I/O 操作。 **联机日志：**声明描述该错误和发生问题位置的 chunk 的信息。

ID	严重性	消息	解释
4002		开，而发生了内部错误。	**服务器状态：**脱机 **用户操作：**检查联机日志、修改错误并使用 gspaces 实用程序恢复镜像。如果操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 4 事件 ID： 4003	3	类消息: Chunk 脱机，但镜像是活动的：*chunk_number* 特定消息： I/O 错误，*error_number* Chunk '*chunk_number*' — 脱机（合理）	
类 ID： 4 事件 ID： 4004	3	类消息: Chunk 脱机，但镜像是活动的：*chunk_number* 特定消息： Chunk 完整性检查失败	
类 ID： 4 事件 ID： 4005	3	类消息: Chunk 脱机，但镜像是活动的：*chunk_number* 特定消息： 将镜像 Chunk *chunk_number* 添加到空间 '*space_number*' 中。请执行手动恢复。	
类 ID： 5 事件	4	类消息: Dbspace 脱机：'*dbspace_name*' 特定消息：	数据库服务器由于访问 chunk 出错而使 dbspace 脱机。 **联机日志：**如果 dbspace 很重要（例如：rootdbs），那么断言失败。如果 dbspace 不重要，

ID	严重性	消息	解释
ID： 5001		Chunk *chunk_number* 脱机	那么声明警告。并都必须提供脱机的 chunk 和 dbspace 的信息。 **服务器状态：**如果非关键媒介故障，那么服务器处于联机状态。如果关键媒介故障，那么服务器处于脱机状态。 **用户操作：**检查联机日志文件并修复导致 dbspace 脱机的根本问题。您可能需要重新恢复 dbspace。
类 ID： 5 事件 ID： 5002	4	类消息: Dbspace 脱机：*'dbspace_name'* 特定消息： 警告！Chunk *chunk_number* 被脱机测试	gadmin 命令导致该数据库服务器脱机。 **联机日志：**声明指示该 dbspace 已脱机的警告。 **服务器状态：**联机 **用户操作：**无
类 ID： 6 事件 ID： 6016	3	类消息: 内部子系统失败：*'message'* 特定消息： 池不可用。池名称：*pool_name*、地址：*address*	
类 ID： 6 事件 ID： 6017	4	类消息: 内部子系统失败：*'message'* 特定消息： CDR Grouper FanOut 线程终止	Enterprise Replication 发生了问题。 **联机日志：**描述问题的断言。 **服务器状态：**联机 **用户操作：** 遵循联机日志的指示操作

ID	严重性	消息	解释
类 ID： 6 事件 ID： 6018	4	类消息: 内部子系统失败：*'message'* 特定消息： CDR Pager: GBase_8s Paging File 已满：等待 CDR_QDATA_SBSPACE 的额外空间	Enterprise Replication 队列的存储空间已满。 **联机日志：**描述问题的断言。 **服务器状态：**联机 **用户操作：**向 CDR_QDATA_SBSPACE 配置参数指定的 sbspace 添加 chunk
类 ID： 6 事件 ID： 6021	3	类消息: 内部子系统失败：*'message'* 特定消息： 当数据库服务器在其转换期间发现一些指数是旧的格式时，报告内部错误。	
类 ID： 6 事件 ID： 6022	3	类消息: 内部子系统失败：*'message'* 特定消息： 当数据库服务器检查当前服务器在复原过程中任何新的定点更改挂起时,报告了内部错误。	
类 ID： 6 事件 ID： 6023	3	类消息: 内部子系统失败：*'message'* 特定消息： 无法打开索引 '*dbname*:*index_name*'，错误编号 = *error_number*	
类 ID：	3	类消息:	

ID	严重性	消息	解释
6 事件 ID： 6024		内部子系统失败：*'message'* 特定消息： 无法删除索引 '*dbname*:*index_name*'，错误编号 = *error_number*	
类 ID： 6 事件 ID： 6025	3	类消息: 内部子系统失败：*'message'* 特定消息： 无法打开表 '*dbname*:*table_name*'，错误编号 = *error_number*	
类 ID： 6 事件 ID： 6026	3	类消息: 内部子系统失败：*'message'* 特定消息： 无法删除表 '*dbname*:*table_name*'，错误编号 = *error_number*	
类 ID： 6 事件 ID： 6027	3	类消息: 内部子系统失败：*'message'* 特定消息： 当数据库服务器在复原过程中尝试删除 sysmaster 数据库时，报告了内部错误	
类 ID： 6 事件	3	类消息: 内部子系统失败：*'message'* 特定消息：	

ID	严重性	消息	解释
ID： 6030		对于子系统分期 BLOBspace 无效或丢失的名称	
类 ID； 6 事件 ID： 6033	5	类消息： 内部子系统失败：*'message'* 特定消息： 高速缓存读取失败	数据库服务器读取内部高速缓存时遇到错误后关闭。 **联机日志：**断言失败 **服务器状态：**脱机 **用户操作：**启动数据库服务器并再次尝试该操作。如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 6 事件 ID： 6034	3	类消息： 内部子系统失败：*'message'* 特定消息： 无法启动远程服务器	
类 ID： 6 事件 ID： 6035	3	类消息: 内部子系统失败：*'message'* 特定消息： 数据库服务器在处理审计跟踪文件时,报告了错误	
类 ID： 6 事件 ID：	3	类消息: 内部子系统失败：*'message'* 特定消息： *dbspaces_list* 归档终止	

ID	严重性	消息	解释
6036			
类 ID ： 6 事件 ID ： 6037	3	类消息: 内部子系统失败：*'message'* 特定消息： 正在等待的 BLOBSpace 出现逻辑恢复	
类 ID ： 6 事件 ID: 6038	3	类消息: 内部子系统失败：*'message'* 特定消息： 数据库服务器报告了内部错误。用户可能需要查看伴随此 id 的特定消息。	
类 ID ： 6 事件 ID ： 6039	3	类消息: 内部子系统失败：*'message'* 特定消息： 清除已删除项目的错误的页面	
类 ID ： 6 事件 ID ： 6040	3	类消息: 内部子系统失败：*'message'* 特定消息： 清除已删除项目时，缓冲区在错误的状态	
类	5 或 3	类消息：	对于严重性 5 ，数据库服务器缓冲区管理器遇到内部错误并

ID	严重性	消息	解释
ID： 6 事件 ID： 6041		内部子系统错误：*'message'* 特定消息： 数据库服务器中的缓冲区管理器检测到内部错误	关闭或纠正了此问题。 **联机日志：**声明带有发生错误时执行的操作的描述的警告或失败。一般来说，声明警告表示该错误已内部更正。 **服务器状态：**如果错误不可恢复，那么脱机。如果错误已纠正，则处于联机状态。 **用户操作：**如果错误不可恢复，那么启动数据库服务器并再次尝试该操作。如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持。如果该错误已被数据库服务器内部更改，那么无需采取操作。
类 ID： 6 事件 ID： 6042	5 或 2	类消息: 内部子系统错误：*'message'* 特定消息： 当数据库服务器检测到内部缓冲区队列不一致时，报告了内部错误	对于严重性 5，数据库服务器在内部缓冲区队列的处理中检测到不一致，服务器关闭或纠正该问题。 **联机日志：**声明带有发生错误时执行的操作的描述的警告或失败。一般来说，声明警告表示该错误已内部更正。 **服务器状态：**如果错误不可恢复，那么脱机。如果错误已纠正，则处于联机状态。 **用户操作：**如果错误不可恢复，那么启动数据库服务器并再次尝试该操作。如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持。如果该错误已被数据库服务器内部更改，那么无需采取操作。
类	3	类消息:	

ID	严重性	消息	解释
ID： 6 事件 ID： 6043		内部子系统失败：*'message'* 特定消息： 内部文件错误	
类 ID： 6 事件 ID： 6044	3	类消息: 内部子系统失败：*'message'* 特定消息： 当数据库服务器尝试保存该日志缓冲区到系统日志缓冲区时,它自动纠正了内部错误	
类 ID： 6 事件 ID： 6045	5	类消息: 内部子系统失败：*'message'* 特定消息： *'space'* 中对 *'object'* 的逻辑日志记录错误	数据库服务器由于在处理逻辑日志时产生的错误而关闭。 **Online log:** 断言失败（附上该操作的描述和逻辑日志信息）。 **服务器状态：**脱机 **用户操作：**启动数据库服务器。如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持
类 ID： 6 事件 ID： 6046	4	类消息: 内部子系统失败：*'message'* 特定消息： *object* 中页面检查失败	数据库服务器检测到数据不一致。 **联机日志：**各种输出，它取决于检测到问题的位置。例如：DBSpace TBLSpace 在所有 DBSpace TBLSpace 上运行 'gcheck -cD' 时，可能出现不一致。 on all DBSpace TBLSpaces

ID	严重性	消息	解释
			服务器状态：联机 **用户操作：**检查联机日志文件并在数据库中运行建议的 gcheck -cD 命令。
类 ID ： 6 事件 ID ： 6047	3	类消息： 内部子系统失败：*'message'* 特定消息： 重新创建索引时出错	
类 ID ： 6 事件 ID ： 6049	5	类消息: 内部子系统失败：*'message'* 特定消息： 锁类型 *lock_type* 和 *lock_type* 不应该合并	数据库服务器尝试合并不兼容的锁后关闭。 **联机日志：**断言失败（附上数据库服务器尝试合并的锁类型）。 **服务器状态：**脱机 **用户操作：**启动数据库服务器。如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持
类 ID ： 6 事件 ID ： 6050	5	类消息: 内部子系统失败：*'message'* 特定消息： 当数据库服务器在锁定可用列表链中检测到一些损坏时，报告内部错误	数据库服务器在检测到管理可用锁的内部列表的内部结构损坏后关闭。 **联机日志：**断言失败 **服务器状态：**脱机 **用户操作：**启动数据库服务器。如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类	3	类消息:	

ID	严重性	消息	解释
ID： 6 事件 ID： 6051		内部子系统失败： *'message'* 特定消息： 错误 — 关键部分中没有 '等待'的锁!!!	
类 ID： 6 事件 ID： 6052	3	类消息: 内部子系统失败： *'message'* 特定消息： 内部 Tblspace 错误	
类 ID： 6 事件 ID： 6053	3	类消息： 内部子系统失败： *'message'* 特定消息： 会话没有对分区 *partition_name* 独占访问权限。 删除该分区的请求被忽略。	
类 ID： 6 事件 ID： 6054	3	类消息： 内部子系统失败： *'message'* 特定消息： 创建 'sysmaster' 数据库出错	
类 ID： 6	3	类消息： 内部子系统失败： *'message'*	

ID	严重性	消息	解释
事件 ID： 6055		特定消息： SMI 表 partnum *partition_number* 集读取错误	
类 ID： 6 事件 ID： 6056	3	类消息： 内部子系统失败：*'message'* 特定消息： 基于区域 '*locale_name*' 的比较失败	
类 ID： 6 事件 ID： 6057	2	类消息： 内部子系统失败：*'message'* 特定消息： DBSPACETEMP 内部列表未初始化,使用缺省列表	数据库服务器没有创建保存 DBSPACETEMP 信息所必需的结构。 **联机日志：**消息显示该内部 DBSPACETEMP 列表未初始化 **服务器状态：**联机 **用户操作：**无
类 ID： 6 事件 ID： 6058	3	类消息： 内部子系统失败：*'message'* 特定消息： 访问数据源使用的网关（*gateway_name*）可能是处于不一致的状态	
类 ID： 6 事件 ID：	3	类消息： 内部子系统失败：*'message'* 特定消息： 准备参加的 site *site_name* 没有响应	

ID	严重性	消息	解释
6059			
类 ID： 6 事件 ID： 6060	5	类消息： 内部子系统失败： *'message'* 特定消息： 线程退出，持有 *number* 个缓冲区	数据库服务器检测到一个线程持有一个或多个缓冲区后关闭。 **联机日志：**断言失败（附上线程抽样的缓冲区数量）。 **服务器状态：**脱机 **用户操作：**使数据库服务器处于联机状态。如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持
类 ID： 6 事件 ID： 6061	3	类消息： 内部子系统失败： *'message'* 特定消息： 当数据库检测到撤销该事务的日志不适用时，产生的内部错误已被数据库服务器自动更正。	
类 ID： 6 事件 ID： 6062	3	类消息： 内部子系统失败： *'message'* 特定消息： 内部错误—释放的事务条目仍持有锁！	当释放与事务相关的资源时，数据库服务器检测到该事务持有锁。在多数情况下，数据库服务器会释放这些锁。 **联机日志：**断言该事务的警告并声明数据库服务器已经更正该问题。 **服务器状态：**脱机 **用户操作：**如果数据库服务器关闭，那么启动它。
类 ID： 6	3	类消息： 内部子系统失败： *'message'*	

ID	严重性	消息	解释
事件 ID： 6063		特定消息： 用户线程不在 TX 等待列表中	
类 ID： 6 事件 ID： 6064	3	类消息： 内部子系统失败： *'message'* 特定消息： 由于启发式决策,代表指定事务分支的所做的工作可能被启发式完成或提交或回滚或部分提交和部分回滚。	
类 ID： 6 事件 ID： 6065	3	类消息： 内部子系统失败： *'message'* 特定消息： 重建索引时发生错误	
类 ID： 6 事件 ID： 6066	3	类消息： 内部子系统失败： *'message'* 特定消息： 当数据库服务器检查所有的网站以查看启发式回滚是否是失败的原因时,报告了内部错误	
类 ID： 6 事件 ID： 6067	5	类消息： 内部子系统失败： *'message'* 特定消息： 致命性内部错误（递归异常）导致数据库服务器进程意外终止	数据库服务器检测到异常处理的递归调用，立即关闭以避免无限循环。 **联机日志：**断言失败 **服务器状态：**脱机 **用户操作：**启动数据库服务器。

ID	严重性	消息	解释
			如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 6 事件 ID： 6068	5	类消息： 内部子系统失败： *'message'* 特定消息： 致命性内部错误（内部异常）导致数据库服务器进程意外终止	数据库服务器由于不可恢复的内部错误而关闭。 **联机日志：**断言失败（附上导致此问题的异常的信息）。 **服务器状态：**脱机 **用户操作：** 启动数据库服务器。查看在断言失败文件中的该异常信息。如果该异常与用户定义例程有关，那么调查并纠正此用户定义例程。如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 6 事件 ID： 6069	5	类消息： 内部子系统失败： *'message'* 特定消息： 致命性内部错误（主守护进程已死）导致数据库服务器进程意外终止	主守护 oninit 进程停止，该错误可能由操作系统进程的终止导致。 **联机日志：**断言失败 **服务器状态：**脱机 **用户操作：**启动数据库服务器。当终止操作系统进程时，要小心。
类 ID： 6 事件 ID： 6070	5	类消息： 内部子系统失败： *'message'* 特定消息： 致命性内部错误（VP 死机）导致数据库服务器进程意外终止	oninit 进程停止并且数据库服务器关闭。该错误可能由操作系统进程的终止导致。 **联机日志：**断言失败 **服务器状态：**脱机 **用户操作：**启动数据库服务器。当终止操作系统进程时，要小心。

ID	严重性	消息	解释
类 ID： 6 事件 ID： 6071	5	类消息： 内部子系统失败：'*message*' 特定消息： 错误:无法派生辅助服务器线程(MACH11 关机)	辅助服务器关闭但是无法创建线程以正常关闭。 **联机日志：**DR： 关闭此服务器。错误：无法派生辅助服务器线程（MACH11 关机）无法运行 gadmin -ky PANIC: 尝试关闭系统。 **服务器状态：**脱机 **用户操作：**无
类 ID： 6 事件 ID： 6072	3	类消息： 内部子系统失败：'*message*' 特定消息： 一般唯一事件 id ,服务器派生新线程失败	
类 ID： 6 事件 ID： 6073	3	类消息： 内部子系统失败：'*message*' 特定消息： 当数据库服务器无法初始化 GLS 以启动会话时，报告错误	
类 ID： 6 事件 ID： 6074	3	类消息： 内部子系统失败：'*message*' 特定消息： 警告：mt_aio_wait: errno == EINVAL	
类	5	类消息：	数据库服务器由于 KAIO 子系

ID	严重性	消息	解释
ID： 6 事件 ID： 6075		内部子系统失败：’message’ 特定消息： 致命性内部错误(KAIO)导致数据库服务器进程意外终止	统中的错误而关闭。 **联机日志：** 断言失败（附上失败的指定操作）。 **服务器状态：** 脱机 **用户操作：**启动该数据库服务器。如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 6 事件 ID： 6100		一般事件,当数据库服务器隐式地提出断言失败	发生一般内部错误。 **联机日志：**断言失败（附上问题的详细信息）。 **服务器状态：**联机 **用户操作：**查看联机日志并采取建议的操作。数据库服务器可能会自动纠正该错误。重新尝试该操作。如果操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 6 事件 ID： 6300		一般事件,当数据库服务器隐式地提出断言失败	发生一般内部错误。 **联机日志：**断言失败（附上问题的详细信息）。 **服务器状态：**联机 **用户操作：**查看联机日志并采取建议的操作。 重新尝试该操作。如果操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID：		一般事件 ， 当数据库服务器由于内部错误情况而意外终止	发生内部错误并且数据库服务器关闭。

ID	严重性	消息	解释
6 事件 ID： 6500			**联机日志：**断言失败 **服务器状态：**脱机 **用户操作：**启动数据库服务器。检查断言失败以获得更多信息。如果可以，修复标识的问题并重新尝试该操作。如果操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 7 事件 ID： 7001	3	类消息： 数据库服务器初始化失败 特定消息： TABLOCKS 日志记录太大以至于不适合逻辑日志缓冲区。LOGBUFF 建议的最小值为 *size*。 I-STAR(C) 开始准备日志记录太大以至于不适合逻辑日志缓冲区。LOGBUFF 建议的最小值为 *size*。 分区 blob 日志记录太大以至于不适合逻辑日志缓冲区。LOGBUFF 建议的最小值为 *size*。 变更表专栏递减日志记录太大以至于不适合逻辑日志缓冲区。LOGBUFF 建议的最小值为 *size*。	
类 ID： 7 事件 ID： 7002	4	类消息： 数据库服务器初始化失败 特定消息： 无法扩展 *number* 驻留页用于 ROOT chunk 中的 checkpoint 。	数据库服务器无法启动，因为它不能给初始 root chunk 中的内部结构分配更多的空间。 **联机日志：**断言 **服务器状态：**脱机 **用户操作：**再次尝试操作。如果

ID	严重性	消息	解释
		无法扩展 *number* 驻留页用于 ROOT chunk 中的日志。	操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 7 事件 ID： 7003	4	类消息： 数据库服务器初始化失败 特定消息： 转换过程中发生了内部错误。用户可能需要查看采取进一步行动的特定消息。	在升级过程中数据库服务器无法启动，因为在转换过程中发生了内部错误。 **联机日志：**描述该问题的断言。 **服务器状态：**脱机 **用户操作：**查看联机日志和特定消息并采取必要的正确的操作。再次尝试原操作。如果操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 7 事件 ID： 7004	4	类消息： 数据库服务器初始化失败 特定消息： 在尝试转化数据库 tblspace 时发生了内部错误	数据库服务器无法启动，因为在尝试转化数据库 tblspace 时发生了内部错误（它保存了有关实例中的数据库的信息）。 **联机日志：**描述该问题的断言。 **服务器状态：**脱机 **用户操作：**请联系 GBase 软件支持
类 ID： 7 事件 ID： 7005	4	类消息： 数据库服务器初始化失败 特定消息： 试图转换 blob 可用映射页时，发生了内部错误	数据库服务器无法启动，因为当它试图转换 blobspace 可用映射页时发生了内部错误。 **联机日志：**描述该问题的断言。 **服务器状态：**脱机 **用户操作：**请联系 GBase 软件支持。
类 ID：	4	类消息： 数据库服务器初始化失败	数据库服务器无法启动，因为它仍在恢复物理或逻辑日志。如果在恢复完成之前运行 gadmin

ID	严重性	消息	解释
7 事件 ID： 7006		特定消息： 无法打开逻辑日志	-m 或 gadmin -s 命令，这种情况可能发生。 **联机日志：**描述问题的断言。 **服务器状态：**恢复中并正在启动。 **用户操作：**恢复完成后运行 gadmin -m 或 gadmin -s 命令。
类 ID： 7 事件 ID： 7007	4	类消息： 数据库服务器初始化失败 特定消息： 没有找到逻辑日志文件	数据库服务器无法启动因为丢失了一个逻辑日志文件。 **联机日志：**描述该问题的断言。 **服务器状态：**脱机 **用户操作：**从一个备份恢复该数据库服务器。
类 ID： 7 事件 ID： 7008	3	类消息： 数据库服务器初始化失败 特定消息： 警告！LTXHWM 设置为 100% 。这个长事务的高水位将永远不会达到。忽略它们的长度，服务器将不会自动终止这些事务。	
类 ID： 7 事件 ID： 7009	4	类消息： 数据库服务器初始化失败 特定消息： 物理或逻辑恢复活动中	数据库服务器无法启动，因为它仍在恢复物理或逻辑日志。如果在恢复完成之前运行 gadmin -m 或 gadmin -s 命令，这种情况可能发生。 **联机日志：** 描述问题的断言。 **服务器状态：**恢复中并正在启动。 **用户操作：**恢复完成后运行

ID	严重性	消息	解释
			gadmin -m 或 gadmin -s 命令。
类 ID： 7 事件 ID： 7010	4	类消息： 数据库服务器初始化失败 特定消息： *root_dbspace* 尚未物理恢复	数据库服务器无法启动，因为在 rootdbs 物理恢复之前恢复被打断。 **联机日志：** 描述该问题的断言。 **服务器状态：** 脱机 **用户操作：** 恢复 rootdb
类 ID： 7 事件 ID： 7011	4	类消息： 数据库服务器初始化失败 特定消息： *dbspace* 尚未物理恢复	数据库无法启动，因为 dbspace 没有物理恢复。如果数据库服务器尝试在恢复未完成之前启动，那么就会发生这种情况。 **联机日志：** 描述该问题的断言。 **服务器状态：** 脱机 **用户操作：** 启动数据库服务器之前等待直到恢复完成。
类 ID： 7 事件 ID： 7012	4	类消息： 数据库服务器初始化失败 特定消息： *dbspace* 未从相同的归档的 *dbspace* 备份中恢复	数据库服务器无法启动，因为 dbspace 恢复失败。 **联机日志：** 描述该问题的断言。 **服务器状态：** 脱机 **用户操作：** 从一个备份中恢复 dbspace，并前滚必要的日志使数据库回到正确的时间点。
类 ID： 7 事件 ID： 7013	4	类消息： 数据库服务器初始化失败 特定消息： 没有找到日志 *log_number*	数据库服务器无法启动，因为恢复未完成。 **联机日志：** 描述该问题的断言。 **服务器状态：** 脱机 **用户操作：** 启动数据库服务器之前等待直到恢复完成。

ID	严重性	消息	解释
类 ID ： 7 事件 ID ： 7014	4	类消息： 数据库服务器初始化失败 特定消息： 不能跳过逻辑还原。执行了逻辑还原	数据库服务器无法启动，因为逻辑还原未完成。 **联机日志：** 描述该问题的断言。 **服务器状态：** 脱机 **用户操作：**执行逻辑还原（例如：通过使用 gbackuprestore -r -l 命令）并以静默或联机方式启动数据库服务器。
类 ID ： 7 事件 ID ： 7015	4	类消息： 数据库服务器初始化失败 特定消息： 无法变更到联机或 静默方式	数据库服务器无法启动，因为在快速或完整恢复的过程中发生了错误。 **联机日志：** 描述该问题的断言。 **服务器状态：** 脱机 **用户操作：**检查联机日志以获得更多信息。再次尝试原操作。如果操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID ： 7 事件 ID ： 7016	4	类消息： 数据库服务器初始化失败 特定消息： 无法打开主 Chunk ’*chunk_number*’	数据库服务器无法启动，因为它不能访问主 chunk。 **联机日志：** 描述该问题的断言。 **服务器状态：** 脱机 **用户操作：**检查联机日志以获得更多信息。再次尝试原操作。如果操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID ：	4	类消息： 数据库服务器初始化失败	数据库服务器无法启动，因为 chunk 路径的所有者和群组不正确。

ID	严重性	消息	解释
7 事件 ID： 7017		特定消息： Chunk '*chunk_number*' 必须具有 owner-ID "owner_id" 和 group-ID "*group_id*"	**联机日志：** 描述该问题的断言。 **服务器状态：** 脱机 **用户操作：**纠正特定消息中提到的有关 chunk 路径的权限。再次尝试原操作。如果操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 7 事件 ID： 7018	4	类消息： 数据库服务器初始化失败 特定消息： 所有者和群组（660）必须有对 chunk '*chunk_number*' 读/写的权限	数据库服务器无法启动，因为对 chunk 路径的权限不正确。 **联机日志：** 描述该问题的断言。 **服务器状态：** 脱机 **用户操作：**纠正特定消息中提到的有关 chunk 路径的权限。再次尝试原操作。如果操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 7 事件 ID： 7019	4	类消息： 数据库服务器初始化失败 特定消息： 内存分配错误	数据库服务器无法启动，因为它无法分配足够的内存。 **联机日志：** 无 **服务器状态：** 脱机 **用户操作：**确保有足够的内存可用于您指定的数据库服务器的配置。重试原操作。如果操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 7 事件	4	类消息： 数据库服务器初始化失败 特定消息：	数据库服务器无法启动，因为没有足够的空间创建指定的 chunk 。 **联机日志：** 描述该问题的断言。

ID	严重性	消息	解释
ID： 7020		Chunk ’*chunk_number*’ 没有适合的指定空间	**服务器状态：**脱机 **用户操作：**指定一个较小的大小的 chunk 或释放额外的空间
类 ID： 7 事件 ID： 7021	4	类消息： 数据库服务器初始化失败 特定消息： *device_name*：写操作失败	数据库服务器无法启动，因为文件系统没有可用空间。 **联机日志：** 描述该问题的断言。 **服务器状态：**脱机 **用户操作：**确保特定消息中提到的文件系统有足够的空间。重试原操作。如果操作再次失败，那么请注意所有情况并联系 GBase 软件支持
类 ID： 7 事件 ID： 7022	3	类消息： 数据库服务器初始化失败 特定消息： 数据库服务器创建 SMI 表时发生错误	
类 ID： 7 事件 ID： 7023	4	类消息： 数据库服务器初始化失败 特定消息： 无法创建启动配置文件— ’*file_name*’	数据库服务器无法启动，因为它无法创建配置文件。 **联机日志：**描述该错误的声明。 **服务器状态：**脱机 **用户操作：**检查联机日志以获得更多信息并修复该问题。该问题可能是目录的错误权限。重试原操作。如果操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类	3	类消息：	

ID	严重性	消息	解释
ID： 7 事件 ID： 7024		数据库服务器初始化失败 特定消息： 'sysmaster' 数据库将不会被创建/检查	
类 ID： 7 事件 ID： 7025	3	类消息： 数据库服务器初始化失败 特定消息： 警告！物理日志大小 *size* 太小。在活动高峰期可能产生物理日志溢出。建议最小物理日志大小是最大并发用户线程的次数。	
类 ID： 7 事件 ID： 7026	3	类消息： 数据库服务器初始化失败 特定消息： 警告！逻辑日志布局可能导致 __ISN__ 进入锁定状态。建议最小逻辑日志大小是最大并发用户线程的次数。	
类 ID： 7 事件 ID： 7027	3	类消息： 数据库服务器初始化失败 特定消息： 警告！缓冲池大小可能导致 __ISN__ 进入锁定状态。建议最小缓冲池大小是最大并发用户线程的次数。	
类 ID： 7	3	类消息： 数据库服务器初始化失败	

ID	严重性	消息	解释
事件 ID： 7028		特定消息： Checkpoint 日志记录可能没有放入逻辑日志缓冲区。LOGBUFF 建议的最小值是 *size*。	
类 ID： 7 事件 ID： 7029	3	类消息： 数据库服务器初始化失败 特定消息： 临时事务不为空	
类 ID： 9 事件 ID： 9001	4	类消息： 物理恢复失败 特定消息： 物理日志恢复错误	该数据库服务器的物理恢复失败。 **联机日志：**断言失败（附上问题的描述）。 **服务器状态：**联机 **用户操作：**重试该操作或从一个备份恢复 。
类 ID： 10 事件 ID： 10001	3 或 4	类消息： 逻辑恢复失败 特定消息： 回滚错误 *error_number*	逻辑恢复失败，因为数据库服务器无法回滚一个事务。 **联机日志：** 说明错误的详细信息和该日志或日志记录发生错误的位置。 **服务器状态：**联机或脱机，取决于该错误。 **用户操作：**检查该联机日志文件以获得更多信息，并运行建议的命令，例如：gcheck 命令。重新尝试原操作。如果操作失败，请注意所有情况并联系 GBase 软件支持。
类	4	类消息：	该数据库服务器的逻辑恢复失

ID	严重性	消息	解释
ID： 10 事件 ID： 10002		逻辑恢复失败 特定消息： 逻辑恢复终止	败。 **联机日志：**声明有关日记录信息的警告。如果该失败与重要的 dbspace 相关。 **服务器状态：**联机，如果 dbspace 不重要。脱机，如果 dbspace 重要。 **用户操作：**检查联机日志以确定合适的操作，例如：您可能需要重新启动热恢复。
类 ID： 10 事件 ID： 10003	4	类消息： 逻辑恢复失败 特定消息： 偏移量 *log_position* 中的日志 *log_number* 的日志记录（*log_subsystem*:*log_type*）没有回滚	逻辑恢复回滚事务时遇到内部错误。 **联机日志：**描述日志记录的消息。 **服务器状态：**联机 **用户操作：**检查联机日志并确定合适的操作，例如：重新提交该事务。
类 ID： 10 事件 ID： 10004	3	类消息： 逻辑恢复失败 特定消息： '*object*' 中 '*log_subsystem*:*log_type*' 逻辑日志记录错误	
类 ID： 10 事件	4	类消息： 逻辑恢复失败 特定消息： 在逻辑日志恢复过程中,应用日志记录时发	逻辑恢复失败 **联机日志：**声明有关日志记录信息的警告。 **服务器状态：**取决于该失败

ID	严重性	消息	解释
ID： 10005		生内部错误	**用户操作：**检查联机日志并确定合适的操作，例如：重新启动热恢复。
类 ID： 10 事件 ID： 10006	3 或 4	类消息： 逻辑恢复失败 特定消息： 当数据库服务器尝试查找该 tblspace 的描述符时，发生了内部错误	逻辑恢复失败，因为数据库服务器没有找到一个分区的内部文件描述符。 **联机日志：**声明指示发生错误的表和运行 gcheck 命令的说明。 **服务器状态：**联机 **用户操作：**对联机日志中提到的或数据库中的表中运行 gcheck -cDI 命令。
类 ID： 11 事件 ID： 11001	3	类消息： 无法打开 chunk：*'pathname'* 特定消息： 无法打开镜像 Chunk *'chunk_number'*，errno = *error_number*	
类 ID： 11 事件 ID： 11002	3	类消息： 无法打开 chunk：*'pathname'* 特定消息： 无法打开主 Chunk *'chunk_number'*，errno = *error_number*	
类 ID： 12 事件	3	类消息： 无法打开 dbspace：*'dbspace_name'* 特定消息： 错误：没有在表 *table_name* 分片中找到	

ID	严重性	消息	解释
ID： 12001		dBspace *dbspace_name*	
类 ID： 13 事件 ID： 13001	2	类消息： 可能的性能提升 特定消息： 配置的 CPU 轮询线程数量超出 'VPCLASS cpu' 中指定的 CPU VP 的数量。NETTYPE '*protocol*' 轮询线程在 NET VP 上启动。	数据库服务器检测到在服务器初始化时，CPU 虚拟处理器的数量和请求的 CPU 轮询线程数目不匹配。 **联机日志：**有关配置不配不匹配的性能警告 。数据库服务器将使用 NET 虚拟处理器。 **服务器状态：**联机 **用户操作：**检查服务器的配置
类 ID： 13 事件 ID： 13002	2	类消息： 可能的性能提升 特定消息： 由于并行恢复导致事务表溢出	内部结构不足够大到处理该逻辑日志。数据库服务器会推迟处理日志直到结构中出现了更多的空间。 **联机日志：**警告消息指示延迟处理该事务。 **服务器状态：**联机 **用户操作：** 无
类 ID： 14 事件 ID： 14001	3	类消息： Database 故障： *'dbname'* 特定消息： 在变更日志记录方式时，*''dbname''*—错误 *error_number*	
类 ID： 15	3	类消息： 高可用性数据复制失败	

ID	严重性	消息	解释
事件 ID ： 15001		特定消息： DR：在辅助服务器上关闭	
类 ID ： 15 事件 ID ： 15002	3	类消息： 高可用性数据复制失败 特定消息： DR：在主服务器上关闭	
类 ID ： 15 事件 ID ： 15003	3	类消息： 高可用性数据复制失败 特定消息： DR：无法连接辅助服务器	
类 ID ： 15 事件 ID ： 15004	3	类消息： 高可用性数据复制失败 特定消息： DR ：当 DR 没有关闭时，接收到来自远程服务器的请求 [本地类型： *type* ， 当前状态： *state*] [远程类型： *type*]	
类 ID ： 15 事件	3	类消息： 高可用性数据复制失败 特定消息：	

ID	严重性	消息	解释
ID： 15005		DR ：在物理恢复之前接收到连接请求	
类 ID： 15 事件 ID： 15006	3	类消息： 高可用性数据复制失败 特定消息： DR ：本地和远程服务器类型和/或最后更改（LC）不兼容 [本地类型： *type* ， LC：*type*] [远程类型： *type*，LC：*type*]	
类 ID： 16 事件 ID： 16001	2	类消息： 备份完成：*'dbspace_list'* 特定消息： *dbspace_list* 完成，但是没有记录	归档完毕，但是在归档过程中服务器检测到损坏的页面。 **联机日志：**消息指示备份已完成但是检测到损坏的页面。 **服务器状态：**联机 **用户操作：** 不要使用此备份。立即使用较早的备份来还原损坏的 chunk。
类 ID： 16 事件 ID： 16002	2	类消息： 备份完成：*'dbspace_list'* 特定消息： *dbspace_list* 归档完成，但是检测到 *number* 个损坏的页面	归档完毕，但是在归档过程中服务器检测到损坏的页面。 **联机日志：**消息指示备份已完成但是检测到损坏的页面。 **服务器状态：**联机 **用户操作：** 不要使用此备份。立即使用 0 损坏页的较早的备份来还原损坏的 chunk。
类 ID：	2	类消息：	完成列出的 dbspace 归档。 **联机日志：** 消息显示列出的

ID	严重性	消息	解释
16 事件 ID： 16003		备份完成：*'dbspace_list'* 特定消息： *dbspace_list* 归档完成	dbspace 已备份完成。 **服务器状态：**联机 **用户操作：** 无
类 ID： 17 事件 ID： 17001	4	类消息： 备份中止：*'dbspace_list'* 特定消息： 归档检测到页面 *chunk_number*:*page_offset* 已损坏	数据库服务武器检测到损坏并停止备份。 **联机日志：**描述此问题 **服务器状态：**联机 **用户操作：**检查联机日志有关此损坏的信息。再次尝试该操作。如果操作失败，请注意所有的情况并联系 GBase 软件支持。
类 ID： 17 事件 ID： 17002	3	类消息： 备份终止：*'dbspace_list'* 特定消息： 分区 *partition_number* 的页 %d:%d 没有终止	
类 ID： 18 事件 ID： 18001	2	类消息： 日志备份完成：*log_number* 特定消息： 逻辑日志 *log_number* 一备份完成	该逻辑日志已备份。 **联机日志：**消息标识备份的逻辑日志的日志编号。 **服务器状态：**联机 **用户操作：**无
类 ID： 19	3	类消息： 日志备份中止：*log_number*	

ID	严重性	消息	解释
事件 ID： 19001		特定消息： 逻辑日志 *log_number* 一备份终止 *message*	
类 ID： 20 事件 ID： 20001	3	类消息： 逻辑日志已满—需要备份 特定消息： 逻辑日志文件已满—需要备份	
类 ID： 20 事件 ID： 20002	3	类消息： 逻辑日志已满—需要备份 特定消息： 等待下一个逻辑日志文件释放	
类 ID： 20 事件 ID： 20003	3	类消息： 逻辑日志已满—需要备份 特定消息： 逻辑日志已几乎满—需要备份。 在数据复制方案中，这可能会阻止成对的服务器故障恢复。	
类 ID： 21 事件 ID：	3	类消息： 数据库服务器资源溢出：*'resource_name'* 特定消息： 档案 arcbu_next_tbuf() - 缓冲区溢出	

ID	严重性	消息	解释
21001			
类 ID：21 事件 ID：21002	3	类消息： 数据库服务器资源溢出：*'resource_name'* 特定消息： 档案 tcp_logbu_hdr() - 缓冲区溢出	
类 ID：21 事件 ID：21003	3	类消息： 数据库服务器资源溢出：*'resource_name'* 特定消息： 档案 tcp_logbu_trl() - 缓冲区溢出	
类 ID：21 事件 ID：21004	2 或 5	类消息： 数据库服务器资源溢出：*'resource_name'* 特定消息： 物理日志文件溢出	对于严重性 5，该物理日志文件已满并且需要溢出。如果它发生在恢复过程中，那么数据库服务器会尝试扩展该物理日志。 **联机日志：**如果数据库服务器不在恢复状态或其无法扩展该物理日志，那么断言失败。如果数据库服务器在恢复中并且扩展物理日志，那么声明警告。 **服务器状态：**脱机 **用户操作：**无
类 ID：21 事件	3	类消息： 数据库服务器资源溢出：*'resource_name'* 特定消息： 锁定表溢出 - 用户 id %d 、会话 id %d	

ID	严重性	消息	解释
ID： 21005			
类 ID： 21 事件 ID： 21006	5	类消息： 数据库服务器资源溢出：*'resource_name'* 特定消息： 检测到逻辑日志缓冲区溢出	数据库服务器由于逻辑日志已满而关闭。 **联机日志：**断言失败（附上日志记录大小和缓冲区大小）。 **服务器状态：**脱机 **用户操作：**向 onconfig 中的 LOGBUFF 配置参数添加新的值。启动数据库服务器 。
类 ID： 21 事件 ID： 21007	3	类消息： 数据库服务器资源溢出：*'resource_name'* 特定消息： Llog logbu_logfile() - 缓冲区溢出	
类 ID： 21 事件 ID： 21008	3	类消息： 数据库服务器资源溢出：*'resource_name'* 特定消息： Llog logbu_bpage() - 缓冲区溢出	
类 ID： 21 事件 ID：	3	类消息： 数据库服务器资源溢出：*'resource_name'* 特定消息： 无法给用户 id *user_ID*分配线程	

ID	严重性	消息	解释
21009			
类 ID： 21 事件 ID： 21010	3	类消息： 数据库服务器资源溢出：*'resource_name'* 特定消息： 无法给用户 id *user_ID*、会话 id *session_ID* 分配事务	
类 ID： 22 事件 ID： 22001	3	类消息： 检测到长事务 特定消息： XA 事务阻塞，tx *transaction_number*，直到它清除完成	
类 ID： 22 事件 ID： 22002	3	类消息： 检测到长事务 特定消息： 继续长事务（至提交）：tx:	
类 ID： 22 事件 ID： 22003	3	类消息： 检测到长事务 特定消息： 中止长事务：tx:	
类	2	类消息：	逻辑日志已满，不能写入更多的事务。

ID	严重性	消息	解释
ID： 23 事件 ID： 23001		逻辑日志 *'number'* 完整 特定消息： 逻辑日志 *log_number* 完整，时间戳：*timestamp*	**联机日志：**消息指示该逻辑日志已满。 **服务器状态：**联机 **用户操作：**无
类 ID： 24 事件 ID： 24001	3	类消息： 无法分配内存 特定消息： 一般的唯一事件 ID，当服务器开始新线程时分配内存失败	
类 ID： 24 事件 ID： 24002	3	类消息： 无法分配内存 特定消息： 警告：无法分配请求的大缓冲区的大小 *size*	
类 ID： 24 事件 ID： 24003	3	类消息： 无法分配内存 特定消息： 数据库服务器尝试预分配共享内存虚拟片段（依据 SHMVIRT_ALLOCSEG 配置参数的设置）但是无法添加分段。30 分钟内将会输出下一条失败消息	
类 ID：	3	类消息： 无法分配内存	

ID	严重性	消息	解释
24 事件 ID ： 24004		特定消息： 超出消息共享内存	
类 ID ： 24 事件 ID ： 24005	3	类消息： 无法分配内存 特定消息： 超出消息共享内存	
类 ID ： 24 事件 ID ： 24006	3	类消息： 无法分配内存 特定消息： 超出虚拟共享内存	
类 ID ： 24 事件 ID ： 24007	3	类消息： 无法分配内存 特定消息： 没有内存可用于页面清除程序	
类 ID ： 24 事件	3	类消息： 无法分配内存 特定消息： kysearch() ：内存分配错误	

ID	严重性	消息	解释
ID： 24008			
类 ID： 24 事件 ID： 24009	3	类消息： 无法分配内存 特定消息： 锁定表溢出 — 用户 id *user_ID* 、会话 id *session_ID*	
类 ID： 24 事件 ID： 24010	3	类消息： 无法分配内存 特定消息： 无法给用户 id *user_ID* 分配线程	
类 ID： 24 事件 ID： 24011	3	类消息： 无法分配内存 特定消息： 无法给用户 id *user_ID*、会话 id *session_ID* 分配事务	
类 ID： 26 事件 ID： 26001	3	类消息： 动态添加日志文件 *logid* 特定消息： 动态地向 DBspace *dbspace_name* 中添加日志文件 *logid*	

ID	严重性	消息	解释
类 ID： 27 事件 ID： 27001	4	类消息： 需要日志文件 特定消息： 警报：最早的逻辑日志（*log_number*）包含来自打开的事务（*transaction_number*）的记录。逻辑日志记录将会阻塞直到新增一个日志文件。添加日志文件可以使用 glogadmin -a 命令 ，-i （insert）选项，例如： glogadmin -a -d *dbspace* -s *size* -i 然后尽可能尽快完成该事务	数据库服务器需要其他日志文件来继续操作。 **联机日志：**警报：最早的逻辑日志（log_number）包含来自打开的事务（transaction_number）的记录。逻辑日志记录将会阻塞直到新增一个日志文件。添加日志文件可以使用 glogadmin -a 命令 ，-i （insert）选项，例如 glogadmin -a -d dbspace -s size -i 然后尽可能尽快完成该事务。 **服务器状态：**联机 **用户操作：**添加新的逻辑日志
类 ID： 28 事件 ID： 28001	4	类消息： 没有空间存储日志文件 特定消息： 警报：因为最早的逻辑日志（*log_number*）包含来自打开的事务（*transaction_number*）的记录，所以服务器尝试动态地添加日志文件。但是没有空间可用。请添加一个 dbspace 或 chunk 。然后尽快完成该事务	数据库服务器无法动态地添加其他的逻辑日志文件，因为没有足够的空间。 **联机日志：**声明指示没有足够的空间可用于附加的逻辑日志文件警告。 **服务器状态：**联机 **用户操作：**添加新的其他的逻辑日志文件或额外的空间。
类 ID： 28 事件 ID： 28002	4	类消息： 没有空间存储日志文件 特定消息： 警告 — Enterprise Replication 正在尝试动态添加日志文件。但是没有可用的空间。该重放位置可能溢出。	数据库服务器无法动态地添加其他的逻辑日志文件，因为没有足够的空间。 **联机日志：**声明指示没有足够的空间可用于附加的逻辑日志文件警告。 **服务器状态：**联机 **用户操作：**添加新的其他的逻辑

ID	严重性	消息	解释
			日志文件或额外的空间。
类 ID： 29 事件 ID： 29001	2	类消息： 内部子系统：*subsystem* 特定消息： 跳过现有审计文件 *file_name* 到 *file_name*	审计子系统需要变更到一个新的输出文件。 **联机日志：**消息指示该审计文件变更跳过现有的文件。 **服务器状态：**联机 **用户操作：**无
类 ID： 30 - 39	2, 3, or 4	Enterprise Replication 事件。请参阅 Enterprise Replication event alarms	
类 ID： 40 事件 ID： 40001	3	类消息： RSS 报警 特定消息： RSS *server_name* 已添加	
类 ID： 40 事件 ID： 40002	3	类消息： RSS 报警 特定消息： RSS 源 *server_name* 的密码已变更	
类 ID： 40 事件	3	类消息： RSS 报警 特定消息：	

ID	严重性	消息	解释
ID： 40003		RSS *server_name* 已删除	
类 ID： 40 事件 ID： 40004	3	类消息： RSS 报警 特定消息： RSS *server_name* 日志重放位置距离 RSS 源太远	
类 ID: 40 事件 ID： 40005	3	类消息： RSS 报警 特定消息： RSS *server_name* 未确认日志传输	
类 ID： 40 事件 ID： 40006	3	类消息： RSS 报警 特定消息： 从 RSS *server_name* 接收缓冲区时出错 — 关机	
类 ID： 40 事件 ID： 40007	3	类消息： RSS 报警 特定消息： 延迟或停止应用：I/O 写错误： *error_number error_description*	

ID	严重性	消息	解释
列 ID： 40 事件 ID： 40008	3	类消息： RSS 报警 特定消息： 延迟或停止应用：由于错误线程退出	
类 ID： 41 事件 ID： 41001	3	类消息： SDS 报警 特定消息： ERROR ：移除 SDS 节点 *server_name* 超时 — 正在移除	
类 ID： 42 事件 ID： 42001	1	类消息： 发生事件	数据库服务器在验证 tblspace 页面时遇到错误。 **逻辑日志：**声明带有表的详细信息的警告。 **服务器状态：**联机 **用户操作：**检查联机日志以确定发生问题的 *database.owner.tablename* 。在表中运行 gcheck -pt 命令。纠正由 gcheck 实用程序指示的错误并重新尝试该操作。如果操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 43 事件	3	类消息： 连接管理器报警 特定消息：	

ID	严重性	消息	解释
ID： 43001		CM：连接管理器 *name* 的会话异常终止	
类 ID： 43 事件 ID： 43002	3	类消息： 连接管理器报警 特定消息： 连接管理器 *CM_Name* 的 FOC 设置 *FOC_String* 与配置为集群的仲裁故障转移的连接管理器的 FOC 设置不匹配。如果该连接管理器成为活动的仲裁员，那么它的 FOC 会与之前的 FOC 协议不匹配。	
类 ID： 44 事件 ID： 44001	3	类消息： DBSpace 已满： *dbspace_name* 特定消息： 警告：*dbspace_type dbspace_name* 已满	
类 ID： 45 事件 ID： 45001	3	类消息： 分区 *'partition_name'*：没有更多的 extent 特定消息： 分区 *'partition_name'*：没有更多的 extent	
类 ID： 46 事件 ID：	3	类消息： 分区 *'partition_name'*：没有更多的页面 特定消息： 分区 *'partition_name'*：没有更多的页面	

ID	严重性	消息	解释
46001			
类 ID： 47 - 71	3 或 4	Enterprise Replication 事件。请参阅 Enterprise Replication event alarms	
类 ID： 72 事件 ID： 72001	2	类消息： 审计跟踪切换到一个新的文件 特定消息： 审计跟踪切换到 *file_name*	审计系统切换到一个新的输出文件。 **联机日志：** 消息提供了新输出文件的文件名 **服务器状态：**联机 **用户操作：**无
类 ID： 73-77	3 或 4	Enterprise Replication 事件。请参阅 Enterprise Replication event alarms	
类 ID： 78 事件 ID： 78001	3	类消息： 存储池为空。 特定消息： 警告：存储池空间不足	
类 ID： 79 事件 ID： 79001	3	类消息： 动态添加 chunk *chunk_name* 到空间中。 特定消息： 动态添加 *chunk_name* 到空间 '*space_name*' 中。 路径： *path* ，偏移量 *offset_number* 千	

ID	严重性	消息	解释
		字节 大小：*size* 千字节	
类 ID： 80 事件 ID： 80001	2	类消息： 表 *table_name* 的新的分段被添加到 DBspace *dbspace_name* 中	表中自动添加了新的分段，因为表增长的大小大于现有分段大小。 **联机日志：**提供了表名和 dbspace 名称的消息 **服务器状态：**联机 **用户操作：**无
类 ID： 81 事件 ID： 81001	4	类消息： 在备份过程中检测到；逻辑日志文件或 dbspace 损坏。Loguniq 或 Dbspace id: *ID*。 特定消息： 日志备份检测到损坏的逻辑日志文件。 意外的 loguniq ：页号 *log_number*:*page_number* 实际 loguniq ：页号 *log_number*:*page_number* 继续日志备份但是该日志备份无法用于恢复服务器 您应该运行 gcheck 并执行 0 级归档	备份失败，因为数据库服务器检测到逻辑日志文件或 dbspace 损坏。 **联机日志：**声明警告 **服务器状态：**联机 **用户操作：**执行新的 0 级备份
类 ID： 82 事件	3	类消息： 会话 *ID* (*线程*) 网络写操作阻塞至少 30 分钟，这可能指示操作系统问题 特定消息：	

ID	严重性	消息	解释
ID： 82001		会话 *ID* (*线程*) 网络写操作阻塞至少 30 分钟，这可能指示操作系统问题	
类 ID： 83 事件 ID： 83001	3	类消息： SDS：故障转移中止—检测到主服务器仍处于活动状态 特定消息： SDS：故障转移中止—检测到主服务器仍处于活动状态	
事件 ID： 84001	3	类消息： 通用网络故障报警 特定消息： 无法绑定到服务器（*dbservername*）主机（IP 地址或主机名称）的端口 （端口号或服务器名）	主机名称或 IP 地址、服务器名称或端口号可能不正确。该端口可能在使用中。 **数据库状态：**联机，数据库启动过程中。 **联机日志：**声明警告 **用户操作：** 检查 sqlhosts 文件中的主机名或 IP 地址、服务器名称和端口号。确定该端口没有被使用。进行必要的更改并重新启动此服务器
事件 ID： 86001	3	类消息： 空间已达到其最大配置大小 特定消息： 警告:空间 *space_name* 已达到其最大配置大小（*大小* MB）	可扩展的存储空间时配置的最大大小，不能进一步扩大 **服务器状态：**联机 **联机日志：**无 **用户操作：**无需采取操作。如果您想要新增该存储空间的最大大小，请运行带有 modify space sp_sizes 参数的 admin() 或 task() SQL 管理 API 函数，并指定新的最大大小。

严重性 5 的事件警报

严重性 5 事件警报标识数据库服务器已经失败。

ID	消息	解释
类 ID： 6 事件 ID： 6033	类消息： 内部子系统失败：*'message'* 特定消息： 高速缓存读取错误	数据库服务器在读取内部缓存时遇到错误后关闭。 **联机日志：**断言失败。 **服务器状态：**脱机。 **用户操作：**启动数据库服务器并再次尝试该操作。如果操作再次失败，注意所有情况并联系 GBase 软件支持。
类 ID： 6 事件 ID： 6041	类消息： 内部子系统失败：*'message'* 特定消息： 数据库服务器缓冲区管理器检测到一个内部错误	数据库服务器缓冲区管理器遇到一个内部错误而关闭或纠正该问题。 **联机日志：**声明带有发生错误时执行的操作的描述的警告或失败。一般来说，声明警告表示该错误已内部更正。 **服务器状态：**如果错误不可恢复，那么脱机。如果错误已纠正，则处于联机状态。 **用户操作：**如果错误不可恢复，那么启动数据库服务器并再次尝试该操作。如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持。如果该错误已被数据库服务器内部更改，那么无需采取操作。
类 ID： 6 事件 ID： 6042	类消息： 内部子系统失败：*'message'* 特定消息： 当数据库服务器检测到内部缓冲区队列不一致时，报告了内部错误	数据库服务器在内部缓冲区队列的处理中检测到不一致，服务器关闭或纠正该问题。 **联机日志：**声明带有发生错误时执行的操作的描述的警告或失败。一般来说，声明警告表示该错误已内部更正。 **服务器状态：**如果错误不可恢复，那么脱机。如果错误已纠正，则处于联机状态。 **用户操作：**如果错误不可恢复，那么

ID	消息	解释
		启动数据库服务器并再次尝试该操作。如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持。如果该错误已被数据库服务器内部更改，那么无需采取操作。
类 ID： 6 事件 ID： 6045	类消息： 内部子系统失败：*'message'* 特定消息： *'space'* 中的 *object* 逻辑日志记录错误	数据库服务器由于在处理逻辑日志时产生的错误而关闭。 Online log:断言失败（附上该操作的描述和逻辑日志信息）。 **服务器状态：**脱机 **用户操作：**启动数据库服务器。如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 6 事件 ID： 6049	类消息： 内部子系统失败：*'message'* 特定消息： 锁类型 *lock_type* 和 *lock_type* 不应该合并	数据库服务器尝试合并不兼容的锁后关闭。 **联机日志：**断言失败（附上数据库服务器尝试合并的锁类型）。 **服务器状态：**脱机 **用户操作：**启动数据库服务器。如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 6 事件 ID： 6050	类消息： 内部子系统失败：*'message'* 特定消息： 当数据库服务器在锁定可用列表链中检测到一些损坏时，报告内部错误	数据库服务器在检测到管理可用锁的内部列表的内部结构损坏后关闭。 **联机日志：**断言失败 **服务器状态：**脱机 **用户操作：**启动数据库服务器。如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 6 事件 ID： 6060	类消息： 内部子系统失败：*'message'* 特定消息： 线程退出，持有 *number* 个缓冲区	数据库服务器检测到一个线程持有一个或多个缓冲区后关闭。 **联机日志：**断言失败（附上线程持有的缓冲区数量）。 **服务器状态：**脱机 **用户操作：**使数据库服务器处于联机

ID	消息	解释
		状态。如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 6 事件 ID： 6067	类消息： 内部子系统失败：*'message'* 特定消息： 致命性内部错误（递归异常）导致数据库服务器进程意外终止	数据库服务器检测到异常处理的递归调用，立即关闭以避免无限循环。 **联机日志：**断言失败 **服务器状态：**脱机 **用户操作：**启动数据库服务器。如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 6 事件 ID： 6068	类消息： 内部子系统失败：*'message'* 特定消息： 致命性内部错误（内部异常）导致数据库服务器进程意外终止	数据库服务器由于不可恢复的内部错误而关闭。 **联机日志：**断言失败（附上导致此问题的异常的信息）。 **服务器状态：**脱机 **用户操作：**启动数据库服务器。查看在断言失败文件中的该异常信息。如果该异常与用户自定义例程有关，那么调查并纠正此用户自定义例程。如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 6 事件 ID： 6069	类消息： 内部子系统失败：*'message'* 特定消息： 致命性内部错误（主守护进程已死）导致数据库服务器进程意外终止	主守护 oninit 进程停止，该错误可能由操作系统进程的终止导致。 **联机日志：**断言失败 **服务器状态：** 脱机 **用户操作：**启动数据库服务器。当终止操作系统进程时，要小心。
类 ID： 6 事件 ID： 6070	类消息： 内部子系统失败：*'message'* 特定消息： 致命性内部错误（VP 死机）导致数据库服务器进程意外终止	oninit 进程停止并且数据库服务器关闭。该错误可能由操作系统进程的终止导致。 **联机日志：**断言失败 **服务器状态：**脱机 **用户操作：**启动数据库服务器。当终

ID	消息	解释
		止操作系统进程时，要小心。
类 ID： 6 事件 ID： 6071	类消息： 内部子系统失败：*'message'* 特定消息： 错误：无法派生辅助服务器线程（MACH11 关机）	辅助服务器关闭但是无法创建线程以正常关闭。 **联机日志：**DR： 关闭此服务器。错误：无法派生辅助服务器线程（MACH11 关机）无法运行 gadmin -ky PANIC: 尝试关闭系统。 **服务器状态：** 脱机 **用户操作：**无
类 ID： 6 事件 ID： 6075	类消息： 内部子系统失败：*'message'* 特定消息： 致命性内部错误（KAIO）导致数据库服务器进程意外终止	数据库服务器由于 KAIO 子系统中的错误而关闭。 **联机日志：**断言失败（附上失败的指定操作）。 **服务器状态：**脱机 **用户操作：**启动该数据库服务器。如果该操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 6 事件 ID： 6500	一般事件 ，当数据库服务器由于内部错误情况而意外终止	发生内部错误并且数据库服务器关闭。 **联机日志：**断言失败 **服务器状态：**脱机 **用户操作：**启动数据库服务器。检查断言失败以获得更多信息。如果可以，修复标识的问题并重新尝试该操作。如果操作再次失败，那么请注意所有情况并联系 GBase 软件支持。
类 ID： 21 事件 ID： 21004	类消息： 数据库服务器资源溢出：*'resource_name'* 特定消息： 物理日志文件溢出	该物理日志文件已满并且需要溢出。如果它发生在恢复过程中，那么数据库服务器会尝试扩展该物理日志。 **联机日志：**如果数据库服务器不在恢复状态或其无法扩展该物理日志，那么断言失败。如果数据库服务器在恢复中并且扩展物理日志，那么声明警告。

ID	消息	解释
		服务器状态： 脱机 **用户操作：** 无
类 ID ： 21 事件 ID ： 21006	类消息： 数据库服务器资源溢出：*'resource_name'* 特定消息： 检测到逻辑日志缓冲区溢出	数据库服务器由于逻辑日志已满而关闭。 **联机日志：** 断言失败（附上日志记录大小和缓冲区大小）。 **服务器状态：** 脱机 **用户操作：** 向 onconfig 中的 LOGBUFF 配置参数添加新的值。启动数据库服务器 。

连接管理器事件警报 ID

事件警报的类 ID 标识事件的类型。事件 ID 标识特定的事件。

下表列出了连接管理器的事件警报 ID 和消息。

可以设置您自己的警报程序脚本来捕获连接管理器类 ID 和消息，并启动纠正措施或通知。

当您编写连接管理器的警报处理程序时，可以使用 GBASEDBTCMNAME 和 GBASEDBTCMCONUNITNAME 环境变量中设置的值。如果连接管理器提出事件警报，那么连接管理器实例的名称存储于 GBASEDBTCMNAME 环境变量中，连接管理器链接单元名存储于 GBASEDBTCMCONUNITNAME 环境变量中。

事件警报消息会写入连接管理器日志文件。

ID	严重性	消息	解释
类 ID ： 1 事件 ID ： 1001	3	类消息： 连接管理器一般警报 特定消息： 连接管理器停止	连接管理器停止运行。 **联机日志消息：** 连接管理器成功关闭。 **用户操作：** 如有必要，重新启动该连接管理器。
类 ID ： 1 事件	3	类消息： 连接管理器一般警报	连接管理器初始化失败。 **联机日志消息：** 切换守护进程模式失败，连接管理器停止。 错误：初始化失败，连接管理器停

ID	严重性	消息	解释
ID： 1002		特定消息： 连接管理器致命错误	止。 错误：SLA 监听器失败，无法启动连接管理器。 **用户操作：**检查消息日志以获得失败的详细信息。纠正错误后重新启动连接管理器。
类 ID： 1 事件 ID： 1003	3	类消息： 连接管理器一般警报 特定消息： 连接管理器接收信号	连接管理器停止或崩溃。 **联机日志消息：**连接管理器进程接收信号，关闭 **用户操作：**如果连接管理器被信号 9 杀死，那么无需采取操作。否则，将该问题报告给系统管理员。
类 ID： 2 事件 ID： 2001	3	类消息： 故障转移仲裁员警报 特定消息： 故障转移正在进行中	连接管理器故障转移仲裁员启动故障转移事件。 **联机日志消息：**故障转移仲裁员自动进行故障转移。
类 ID： 2 事件 ID： 2002	3	类消息： 故障转移仲裁员警报 特定消息： 故障转移完成	连接管理器故障转移仲裁员已完成故障转移。 **联机日志消息：**故障转移仲裁员自动完成故障转移。
类 ID： 2 事件 ID： 2003	3	类消息： 故障转移仲裁员警报 特定消息： 故障转移禁用	连接管理器的自动故障转移被禁用。 **联机日志消息：**故障转移仲裁员禁用自动故障转移。 **用户操作：** N/A
类	3	类消息：	故障转移进程失败。

ID	严重性	消息	解释
ID： 2 事件 ID： 2004		故障转移仲裁员警报 特定消息： 故障转移仲裁员终止自动的故障转移	**联机日志消息：**故障转移仲裁员终止自动的故障转移 **用户操作：**检查消息日志文件，然后手动启动主服务器或手动执行故障转移。
类 ID： 2 事件 ID： 2005	3	类消息： 故障转移仲裁员警报 特定消息： 故障转移进程处于手动模式	故障转移进程不是自动模式。 **联机日志消息：**故障转移进程处于手动模式
类 ID： 3 事件 ID： 3001	3	类消息： 连接主服务器 特定消息： 无法连接主服务器	连接管理器无法连接主服务器。 **联机日志消息：**无法连接 GBase 8s 服务器。 **用户操作：**纠正设置问题，连接管理器即可自动连接服务器。
类 ID： 3 事件 ID： 3002	3	类消息： 连接主服务器 特定消息： 失去与主服务器的连接	连接管理器与主服务器失联。 **联机日志消息：**检测到失去与 GBase 8s 服务器的连接。 **用户操作：**纠正设置或网络问题，连接管理器即可自动连接服务器。
类 ID： 4 事件 ID： 4001	3	类消息： 连接 ER 节点 特定消息： 无法连接 ER 节点	连接管理器无法连接到 Enterprise Replication 服务器。 **用户操作：** 纠正设置问题，连接管理器即可自动连接 Enterprise Replication 服务器。
类	3	类消息：	连接管理器与 Enterprise

ID	严重性	消息	解释
ID： 4 事件 ID： 4002		连接 ER 节点 特定消息： 失去与 ER 节点的连接	Replication 服务器失联。 **联机日志消息：**检测到失去与 GBase 8s 服务器的连接。 **用户操作：**纠正设置或网络问题，连接管理器即可自动连接 Enterprise Replication 服务器。
类 ID： 5 事件 ID： 5001	3	类消息： 连接一般服务器 特定消息： 无法连接服务器	连接管理器无法连接到高可用集群中的服务器。 **联机日志消息：**无法连接 GBase 8s 服务器。 **用户操作：**纠正该设置问题，连接管理器即可自动连接高可用服务器。
类 ID： 5 事件 ID： 5002	3	类消息： 连接一般服务器 特定消息： 失去与服务器的联系	连接管理器无法连接辅助服务器。 **联机日志消息：**检测到失去与 GBase 8s 服务器的连接。 **用户操作：**纠正该设置问题，连接管理器即可自动连接辅助服务器。

数据库服务器日志中的消息

未编号的消息打印在数据库服务器消息日志（ online.log）中。这些错误消息包含更正操作。

有关错误消息的描述，请使用 finderr 实用程序。

一些消息可能需要您联系 GBase 软件支持。

本章中消息是如何排列的

数据库服务器消息日志消息在本章中以字母顺序排列，按以下附加规则排序：

- 忽略每条消息前面的时间戳记。
- 按字母顺序的排列中忽略字母大小写。
- 忽略空格。
- 忽略引号。
- 忽略前导省略号。
- 如果单词 the 是消息中的第一个单词，那么忽略它。

- 以数字或标点符号开始的消息显示在列表末尾，在标为消息：符号的特殊部分中。
- 某些相关的信息归放在一起，如下：
  - 转换和复原错误消息
  - Enterprise Replication 的转换和复原消息
  - 动态日志消息
  - Sbspace 元数据消息
  - 截断表消息

消息文本后跟消息或消息组的原因和建议的更正操作。

如何查看这些消息

使用以下方法之一查看这些消息：

- 联机消息日志

  要查看发生时显示的消息，使用 tail -f online.log 命令。

- gstat -m 命令

  有关更多信息，请参阅 gstat -l 命令：打印物理和逻辑日志信息 。

要查看与这些未编号消息相关联的错误编号，请查看 sysmaster 数据库中的 logmessage 表：

```
SELECT * FROM logmessage;
```

消息类别

未编号消息存在四个一般类别，但一些消息归入多个类别：

- 常规信息
- 断言失败消息
- 需要管理操作
- 检测到不可恢复错误

技术支持使用断言失败消息辅助故障排除和诊断。其报告的信息通常归入意外事件类别，这些事件可能发展为页可能不发展为由其他错误代码捕获的问题。并且，这些消息是简洁的且通常是极为技术性的。它们可能报告一个或两个孤立的统计信息，但不提供已发生了什么的总体描述。该信息可向技术支持建议可能的研究途径。

消息：A-B

Aborting Long Transaction: tx 0xn.

原因

事务跨越事务高水印（LTXHWM）所指定的日志空间，并正在回滚违例长事务。

操作

不需要附加操作。共享内存中事务结构的地址显示为十六进制。

Affinitied VP mm to phys proc nn.

原因

数据库服务器成功地将 CPU 虚拟处理器绑定到物理处理器上。

操作

不需要任何操作。

Affinity not enabled for this server.

原因

您尝试将 CPU 虚拟处理器绑定到物理处理器上，但您正在运行的数据库服务器不支持进程专用。

操作

从 VPCLASS 配置参数移除专用设置。

Assert Failed: Error from SBSpace cleanup thread.

原因

Sbspace 清除线程在清除游离智能大对象时遇到错误。

操作

请参阅消息日志文件中建议的操作。

大多数时候，在已失败的 sbspace 上运行 gspaces -cl sbspacename 可以成功地清除任何游离智能大对象。如果您遇到不可恢复的错误，请联系技术支持。

Assert Failed: Short description of what failed Who: Description of user/session/thread running at the time Result: State of the affected database server entity Action: What action the database administrator should take See Also: DUMPDIR/af.uniqid containing more diagnostics.

原因

该消息指示内部错误。

操作

ONCONFIG 参数 DUMPDIR 指定的目录中的 af.uniqid 文件包含发送给消息日志的断言失败消息的副本以及当前、相关结果和/或数据缓冲区的内容。该消息中包含的信息是供技术支持使用的。

Begin re-creating indexes deferred during recovery.

原因

在恢复过程中，要创建的索引推迟到恢复完成之后。此消息指示数据库服务器推迟重新创建索引且现在正在创建索引。在数据库服务器重新创建索引期间，它使用共享锁锁定受影响的表。

操作

不需要任何操作。

Building 'sysmaster' database requires ~mm pages of logical log. Currently there are nn pages available. Prepare to back up your logs soon.

原因

您当前没有完成构建 sysmaster 数据库所需的大约数量的可用日志空间。

操作

备份日志。

Building 'sysmaster' database...

原因

数据库服务器正在创建 sysmaster 数据库。

操作

不需要任何操作。

消息：C

Cannot Allocate Physical-log File, mm wanted, nn available.

原因

数据库服务器尝试增加物理日志大小，但是它需要比 dbspace 中可用的连续空间更多的连续空间量。空间量以千字节表示。

操作

必须减小物理日志的大小（使用 PHYSFILE 配置参数），或将物理日志的位置更改到包含足够连续空间的 dbspace 上以容纳物理日志。

Cannot alter a table which has associated violations table.

原因

用户试图添加、删除或修改具有相关联违例表的表中的列。

操作

请不要更改用户表中的列。

Cannot change to mode.

原因

在快速或完全恢复过程中的某些错误阻止了系统更改为联机或静默方式。

操作

请参阅日志文件中以前的消息以获得信息。

Cannot Commit Partially Complete Transactions.

原因

直到 COMMIT 语句被处理之后，删除表或索引的事务才会执行删除（有少数例外）。在这些情况中，写下 beginning commit 日志记录，后跟通常的提交日记记录。如果数据库服务器在这两者之间失败，那么快速恢复进程尝试在您下一次初始化数据库服务器时完成提交。

如果此提交的完成失败，那么数据库服务器生成前述消息。

操作

要确定是否需要执行操作，请查看解释逻辑日志记录中描述的逻辑日志。

Cannot create a user-defined VP class with 'SINGLE_CPU_VP' non-zero.

原因

SINGLE_CPU_VP 设置为非零，且 gadmin 用于创建用户自定义 VP 类。

操作

如果用户自定义 VP 类是必需的，那么停止数据库服务器，将 SINGLE_CPU_VP 更改为零，并重新启动数据库服务器。

Cannot create violations/diagnostics table.

原因

用户发出目标表的 START VIOLATIONS TABLE 语句。数据库服务器无法创建该目标表的违例表。任何以下情况可能是此失败的原因：

- 目标表已有违例表。
- 您在 START VIOLATIONS TABLE 语句中指定了无效的违例表名称。例如：省略了语句中的 USING 子句且如果目标表中的字符数加上四个字符长于标识符的最大长度，那么生成的违例表名称超过标识符的最大长度。
- 您在 START VIOLATIONS TABLE 语句中指定的违例表名称与数据库中现有表名称匹配。
- 目标表包含名称为 gbasedbt_tupleid 、gbasedbt_optype 或 gbasedbt_recowner 的列。由于这些列名与违例表中的 gbasedbt_tupleid 、gbasedbt_optype 或 gbasedbt_recowner 列重复，因此数据库服务器无法创建违例表。
- 目标表是临时表。
- 目标表正在充当某些其他表的违例表。
- 目标表是系统目录表。

操作

要解决此错误，请执行以下操作之一：

- 如果违例表的名称是无效的，那么在 START VIOLATIONS TABLE 语句的 USING 子句中指定违例表的唯一名称。
- 如果目标表包含名称为 gbasedbt_tupleid 、gbasedbt_optype 或 gbasedbt_recowner 的列，请将它们重命名为其他名称。
- 选择不是系统目录表的永久目标表，或不是其他表的违例表的永久目标表。

Cannot insert from the violations table to the target table.

原因

用户发出了试图将行从违例表插入目标表的语句。例如：用户输入以下无效语句：

INSERT INTO mytable SELECT * FROM mytable_vio;

同样，如果目标表具有过滤方式约束，那么您也接收到此错误。

操作

要从该错误进行恢复，请执行以下操作：

- 不要使用过滤约束。
- 停止违例表。
- 将行从违例表插入临时表中，然后将行从临时表插入目标表中。

Cannot modify/drop a violations/diagnostics table.

原因

用户试图更改或删除表，而该表正在充当另一个表的违例表。

操作

请不要更改或删除违例表。

Cannot Open Dbspace nnn.

原因

数据库服务器无法访问指定的 dbspace 。此消息指示打开 tblspace 时发生问题或 dbspace 中初始 chunk 有损坏。

操作

请验证构成该 dbspace chunk 的设备正在正常运行并且已指定给它们正确的操作系统权限（rw-rw----）。您可能需要执行数据恢复。

Cannot Open Logical Log.

原因

数据库服务器无法访问逻辑日志文件。因为它不能访问逻辑日志就无法运行，所以必须解决这个问题。

操作

验证逻辑日志文件驻留的 chunk 设备正在运行并且具有正确的操作系统权限（rw-rw----）。

Cannot Open Mirror Chunk pathname, errorno = nn.

原因

数据库服务器无法打开镜像对的镜像 chunk 。返回了 chunk pathname 和操作系统错误。

操作

有关更正操作的更多信息，请参阅操作系统文档。

Cannot Open Primary Chunk pathname, errorno = nnn.

原因

无法打开镜像对的主 chunk 。返回了 chunk pathname 和操作系统错误。

操作

有关更正操作的更多消息，请参阅操作系统文件。

Cannot Open Primary Chunk chunkname.

原因

无法打开 dbspace 的 initial chunk 。

操作

验证 chunk 设备正在正常运行并具有正确的操作系统权限（rw-rw----）。

Cannot open sysams in database name, iserrno number.

原因

当数据库服务器打开 sysams 系统表时出错。

操作

记下错误 number 并联系技术支持。

Cannot open sysdistrib in database name, iserrno number.

原因

当数据库服务器访问 sysdistrib 系统表时出错。

操作

记下错误 number 并联系技术支持。

Cannot open system_table in database name, iserrno number.

原因

当数据库服务器打开指定的系统表时出错。

操作

记下错误 number 并联系技术支持。

Cannot open systrigbody in database name, iserrno number.

原因

当数据库服务器访问 systrigbody 系统表时出错。

操作

记下错误 number 并联系技术支持。

Cannot open systriggers in database name, iserrno number.

原因

当数据库服务器访问 systriggers 系统表时出错。

操作

记下错误 number 并联系技术支持。

Cannot open sysxtdtypes in database name, iserrno number.

原因

当数据库服务器访问 sysxtdtypes 系统表时出错。

操作

记下错误 number 并联系技术支持。

Cannot Perform Checkpoint, shut system down.

原因

正在尝试恢复镜像 chunk 的线程请求了 checkpoint ，但无法执行该 checkpoint 。

操作

关闭数据库服务器。

Cannot Restore to Checkpoint.

原因

数据库服务器服务无法恢复物理日志从而无法执行快速恢复。

操作

如果数据库服务器未联机，那么从 dbspace 备份执行数据恢复。

Cannot Rollback Incomplete Transactions.

原因

在快速恢复或数据恢复过程中，逻辑日志记录是最先前滚的。然后，将回滚未提交的打开的事务。打开的事务可能会在回滚中失败，对该打开事务的某些修改留在原位。该错误不阻止数据库服务器变成静默方式或联机方式，但可能指示数据库不一致。

操作

要确定是否需要执行任何操作，请使用 glogdump 实用程序检查逻辑日志。

Cannot update pagezero.

原因

当数据库服务器正尝试在复原过程中重写保留页时发生故障。

操作

请参阅日志文件中以前的消息以获得消息，或联系技术支持。

Cannot update syscasts in database name. Iserrno number.

原因

当向 syscasts 系统表中插入数据时发生内部错误。

操作

联系技术支持。

Can’t affinity VP mm to phys proc nn.

原因

数据库服务器支持进程专用，但将虚拟处理器绑定到物理处理器的系统调用失败。

操作

请参阅操作系统文档。

Changing the sbspace minimum extent value: old value value1, new value value2.

原因

当您发出以下命令时，出现此参考消息：

gspaces -ch sbspace -Df "MIN_EXT_SIZE=value1" -y

操作

无。有关更多信息，请参阅 gspaces -ch: 更改 sbspace 缺省规范 。

Checkpoint blocked by down space, waiting for override or shutdown.

原因

Dbspace 在 checkpoint 时间间隔期间已关闭。当发生这种情况时，数据库服务器配置为等待重设。

操作

关闭数据库服务器或执行 gadmin -O 命令重设已关闭的 dbspace 。有关 gadmin 实用程序的更多信息，请参阅 gadmin 实用程序 。

Checkpoint Completed: duration was n seconds.

原因

Checkpoint 已成功完成。

操作

不需要任何操作。

Checkpoint Page Write Error.

原因

数据库服务器在尝试将 checkpoint 信息写入磁盘时检测到错误。

操作

有关解决这种问题的其他帮助，请联系技术支持。

Checkpoint Record Not Found in Logical Log.

原因

逻辑日志或包含该逻辑日志的 chunk 已损坏。数据库服务器无法初始化。

操作

从 dbspace 备份执行数据恢复。

Chunk chunkname added to space spacename.

原因

此消息中的变量具有以下值：

chunkname

是数据库服务器管理器正在添加的 chunk 的名称。

spacename

是数据库服务器管理员正在将 chunk 添加到存储空间的名称。

操作

不需要任何操作。

Chunk chunkname dropped from space spacename.

原因

数据库服务器管理员已从空间 spacename 中删除 chunk chunkname 。

操作

不需要任何操作。

Chunk number nn pathname -\- Offline.

原因

镜像对中所指示的 chunk 已经标记为状态 D 并脱机。镜像对中的其他 chunk 正在成功运行。

操作

立即执行步骤修复 chunk 设备并恢复该 chunk 。将显示 chunk number 和 chunk 设备 pathname 。

Chunk number nn pathname -\- Online.

原因

镜像对中所指示的 chunk 已恢复并已联机（标记状态为 O）。将显示 chunk number 和 chunk 设备 pathname 。

操作

不需要任何操作。

Cleaning stray LOs in sbspace sbspacename.

原因

数据库服务器管理员正在运行 gspaces -cl sbspacename 。

操作

不需要任何操作。

Completed re-creating indexes.

原因

数据库服务器完成了对已推迟索引的重新创建。

操作

不需要任何操作。

Configuration has been grown to handle up to integer chunks.

原因

通过将 CONFIGSIZE 更改为较高的值或将 MAX_CHUNKS 设置为较高的值，数据库服务器管理员已将 chunk 的数量增加到指定的值。

操作

不需要任何操作，更改已成功。

Configuration has been grown to handle up to integer dbslices.

原因

通过将 CONFIGSIZE 更改为较高的值或将 MAX_DBSLICES 设置为较高的值，数据库服务器管理员已将 dbslice 的数量增加到指定的值。

操作

不需要任何操作，更改已成功。

Configuration has been grown to handle up to integer dbspaces.

原因

通过将 CONFIGSIZE 更改为较高的值或将 MAX_DBSPACES 设置为较高的值，数据库服务器管理员已将 dbspace 的数量增加到指定的值。

操作

不需要任何操作，更改已成功。

Continuing Long Transaction (for COMMIT): tx 0xn.

原因

逻辑日志已填充到超出长事务高水印（LTXHWM），但违例长事务正在提交中。在这种情况下，运行事务继续写入逻辑日志并且不进行回滚。共享内存中事务结构的地址显示为十六进制值 tx 0xn 。

操作

不需要任何操作。

Could not disable priority aging: errno = number.

原因

当正在尝试禁用 CPU 虚拟处理器的优先级迟滞时，操作系统调用失败。返回与该失败相关联的系统错误 number 。

操作

请参阅操作系统文档。

Could not fork a virtual processor: errno = number.

原因

虚拟处理器的派生已失败。数据库服务器返回与该失败相关联的操作系统错误 number。

操作

有关确定每个用户可用进程以及系统整体可用进程的最大数量信息，请参阅操作系统文档。

Create_vp: cannot allocate memory.

原因

数据库服务器无法分配新的共享内存。

操作

数据库服务器管理员必须使更多共享内存可用。这种情况可能需要增加 SHMTOTAL 或重

新配置操作系统。此消息通常随给出其他信息的其他信息一起出现。

消息：D-E-F

Dataskip is OFF for all dbspaces.

原因

参考性。

操作

不需要任何操作。

Dataskip is ON for all dbspaces.

原因

参考性。

操作

不需要任何操作。

Dataskip is ON for dbspaces: dbspacelist.

原因

参考性；DATASKIP 对指定的 dbspace 是 ON 。

操作

不需要任何操作。

Dataskip will be turned {ON|OFF} for dbspacename.

原因

参考性；DATASKIP 对指定的 dbspace 是 ON 或 OFF 。

操作

不需要任何操作。

DBSPACETEMP internal list not initialized, using default.

原因

初始化用户指定的 DBSPACETEMP 列表时出错。通常，出现这种情况的原因是内存分配失败。

操作

检查伴随的错误消息。

devname: write failed, file system is full.

原因

由于文件系统 devname 已满，写入失败。

操作

释放 devname 中的一些空间。

Dropping temporary tblspace 0xn, recovering nn pages.

原因

在共享内存初始化过程中，数据库服务器例行公事地搜索遗留下来而未正确清除的临时表。如果数据库服务器找到临时表，那么它删除该表并恢复空间。数据库服务器定位指定的临时 tblspace 并删除它。值 0xn 是 tblspace 的十六进制表示。

操作

不需要任何操作。

Dynamically allocated new shared memory segment (size nnnn).

原因

此状态消息通知您数据库服务器已成功分配了大小为 nnnn 的新共享内存段。

操作

不需要任何操作。

ERROR: NO "wait for" locks in Critical Section.

原因

数据库服务器不允许线程拥有当该线程位于临界段时可能必须等待的锁。任何此类锁请求都遭到拒绝，且向用户返回一条 ISAM 错误消息。

操作

所报告的错误是内部错误。请联系 GBase 8s 技术支持。

Error building sysmaster database. See outfile.

原因

构建 sysmaster 数据库时出错。文件 outfile 包含脚本 buildsmi 的运行结果。

操作

请参阅 outfile 文件。

Error in dropping system defined type.

原因

当更新 sysxtdtypes 、sysctddesc 或 sysxtdtypeauth 系统表时发生内部错误。

操作

请联系技术支持。

Error in renaming systdist.

原因

当尝试查找并重命名 GBase 8s .systdist SPL 例程时发生内部错误。

操作

请联系技术支持。

Error removing sysdistrib row for tabid = tabid, colid = colid in database name. iserrno = number

原因

当更新 sysdistrib 系统表时发生错误。

操作

记下错误 number 并联系技术支持。

Error writing pathname errno = number.

原因

操作系统无法写入 pathname。 Number 是返回的操作系统错误的编号。

操作

调查该操作系统错误的原因。它通常表示没有可用空间用于该文件。也表示目录不存在或没有写权限。

Error writing shmem to file filename (error). Unable to create output file filename errno=mm.Error writing filename errno=nn.

原因

数据库服务器在尝试将共享内存写入 filename 时检测到错误。第一条消息后跟着下两条消息中的一条消息。连个尝试中的任意一个失败的原因是无法创建输出文件或无法写入共享内存的内容。错误是指提示尝试将共享内存写入文件的操作系统错误。nn 值是操作系统错误。

操作

请参阅操作系统文档。

Fail to extend physical log space.

原因

扩展物理日志空间的尝试失败。路径不存在或权限不正确。

操作

使用存在的路径。检查当前工作目录上的权限。您或系统管理源必须给予组在当前工作目录上的执行权限。在群组拥有权限后，重试生成该消息的操作。

Fatal error initializing CWD string. Check permissions on current working directory. Group groupname must have at least execute permission on '.'.

原因

群组 groupname 没有当前工作目录的执行权限。

操作

检查当前工作目录上的权限。您或系统管理员必须予组在当前工作目录上的执行权限。在群组拥有权限后，重试生成该消息的操作。

Fragments dbspacename1 dbspacename2 of table tablename set to non-resident.

原因

tablename 的两个指定分片之一已由 SET TABLE 语句设置为非常驻。

操作

不需要任何操作。

Forced-resident shared memory not available.

原因

您计算机的数据库服务器端口不支持强制常驻共享内存。

操作

不需要任何操作。

Freed mm shared-memory segment(s) number bytes.

原因

当您运行 gadmin 实用程序的 -F 选项以是否未使用的内存之后，数据库服务器向消息日志发送此消息。此消息通知您数据库服务器已成功释放的段数和字节数。

操作

不需要任何操作。

消息：G-H-I

GBase 8s database_server Initialized - Complete Disk Initialized.

原因

已初始化磁盘空间和共享内存。初始化之前存在于磁盘上的任何数据库现在都已不可访问。

操作

不需要任何操作。

GBase 8s database_server Initialized - Shared Memory Initialized.

原因

已初始化共享内存。

操作

不需要任何操作。

GBase 8s database_server Stopped.

原因

数据库服务器已从静默方式变成脱机方式。数据库服务器已脱机。

操作

不需要任何操作。

gcore pid; mv core.pid dir/core.pid.ABORT.

原因

数据库服务器失败过程中的此状态消息提供与虚拟处理器相关联的每个核心文件的名称和位置。

操作

不需要任何操作。

I/O function chunk mm，pagenum nn ，pagecnt aa --> errno = bb.

原因

在尝试从磁盘空间访问数据的过程中发生的操作系统错误。已失败的操作系统功能由 function 定义。发生错误的页的 chunk 编号和物理地址显示为整数。pagecnt 值是指线程正在尝试读取或写入的页数。如果显示 errno 值，那么它是操作系统错误的编号并可能说明该失败。如果 function 指定为 bad request ，那么某些意外事件已导致无效 chunk 或页上的 I/O 尝试。

操作

如果 chunk 状态更改为 D ，或已关闭，那么从其镜像恢复该 chunk 或修复该 chunk 。否则，执行数据恢复。

I/O error, primary/mirror Chunk pathname -- Offline (sanity).

原因

数据库服务器在具有 pathname 的主要 chunk 或镜像 chunk 上检测到 I/O 错误。该 chunk 已脱机。

操作

检查该 chunk 所存储于的设备是否如需要那样正常运行。

已删除索引 idx1 和 idx 2 错误消息。

In-Place Alter Table. Perform EXECUTE FUNCTION sysadmin:task('table update_ipa', 'table_name','database');

原因

当在表中运行定点变更操作时，尝试还原之前版本的数据库服务器。之前版本的数据库服务器无法处理具有多个模式行的表。

操作

在您尝试还原到之前的数据库服务器版本之前，通过更新受影响的表中的行强制执行定点变更操作在表中运行带有 table update_ipa 参数的 SQL 管理 API task() 或 admin() 命令解决暂挂的定点变更操作。

ERROR: Insufficient available disk in the root dbspace to increase the entire Configuration save area.

原因

用户已通过将 CONFIGSIZE 更改为较高的值或将 MAX_DBSPACES 、MAX_DBSLICES 或 MAX_CHUNKS 设置为较高的值来尝试将存储对象的数据增加到特定的值。但数据库服务器没有足够的 rootspace 用于已增加数量的存储对象。存储对象可能是 dbspace 、dbslice 或 chunk 。

操作

增加 root dbspace 的大小或将 CONFIGSIZE 、MAX_DBSPACES 、MAX_DBSLICES 或 MAX_DBSLICES 重新设置为较低的值并重新启动数据库服务器。例如：如果您将 MAX_CHUNKS 设置为 32,768 ，但是 root dbspace 没有足够的空间，那么将 MAX_CHUNKS 设置为较低的值。

Insufficient available disk in the root dbspace for the CM save area. Increase the size of the root dbspace in the ONCONFIG file and reinitialize the server.

原因

原因可能是以下之一：

7. 用户已通过将 CONFIGSIZE 更改为较高的值或将 MAX_DBSPACES 、MAX_DBSLICES 或 MAX_DBSPACES 设置为较高的值来尝试将存储对象的数据增加到特定的值。但数据库服务器没有足够的 rootspace 用于已增加数量的存储对象。存储对象可能是 dbspace 、dbslice 或 chunk 。
8. 用户已转换到需要稍多 rootspace 的数据库服务器版本，但它不可用（不太可能有这种情况）。

操作

执行下列操作之一：

9. 增加 root dbspace 的大小或将 CONFIGSIZE 、MAX_DBSPACES 、MAX_DBSLICES 或 MAX_DBSLICES 重新设置为较低的值并重新启动数据库服务器。例如：如果您将 MAX_DBSPACES 设置为 32,768 ，但是 root dbspace 没有足够的空间，那么将 MAX_DBSPACES 设置为较低的值。
10. 增加 root dbspace 的大小并重新初始化数据库服务器。

Internal overflow of shmid's, increase system max shared memory segment size.

原因

数据库服务器正在初始化共享内存，这时它耗尽了与该段相关联的共享内存标识的内部存储量。

操作

增加最大内核共享内存段大小的值，通常是 SHMMAX 。有关更多信息，请参阅操作系统支持文档。

消息：J-K-L-M

Listener-thread err = error_number: error_message.

原因

监听线程已遇到错误。此消息显示错误编号和消息文本。

操作

有关错误消息的描述，请使用 finderr 实用程序。

Lock table overflow - user id mm session id nn.

原因

当没有锁可用时，线程尝试获得锁。消息中显示了用户 ID 和会话 ID 。

操作

增大 LOCKS 配置参数，并初始化共享内存。

Logical-log File not found.

原因

Root dbspace 保留页中的检查点记录已损坏。

操作

从 dbspace 备份执行数据恢复。

Logical Log nn Complete.

原因

日志 ID 号 nn 所标识逻辑日志已满。数据库服务器自动切换到序列中的下一个逻辑日志文件。

操作

不需要任何操作。

Logical logging vberror for type:subtype in (failed_system).

原因

日志记录已失败。导致该错误的逻辑日志记录如下标识：

type

是逻辑日志记录的类型。

subtype

是日志记录子系统。

failed_system

是指示哪个系统未能进行日志记录的内部函数的名称。

操作

请联系技术支持。

Log Record: log = ll, pos = 0xn, type = type:subtype(snum), trans = xx

原因

数据库服务器在快速恢复或逻辑日志恢复的前滚部分中检测到错误。

导致该错误的日志记录如下标识：

ll

是存储记录的逻辑日志 ID 。

0xn

是日志中的十六进制地址位置。

type

是逻辑日志记录的类型。

subtype

是日志记录子系统。

snum

是子系统编号。

xx

是出现在逻辑日志中的事务编号。

操作

请联系技术支持。

Log record (type:subtype) at log nn, 0xn was not undone.原因

原因

日志撤销因日志损坏而失败。

导致该错误的日志记录如下标识：

type

是逻辑日志记录类型。

subtype

是日志记录子系统。

nn

是存储记录的逻辑日志标识。

0xn

是日志中的十六进制地址位置。

操作

要确定是否需要执行任何操作，请使用 glogdump 实用程序检查该逻辑日志。联系技术支持。

Log record (type:subtype) failed, partnum pnum row rid iserrno num.

原因

发生日志记录失败。

导致该错误的日志记录如下标识：

type

是逻辑日志记录的类型。

subtype

是日志记录子系统。

pnum

是部件号。

rid

是行 ID 。

num

是 iserror 号。

操作

联系技术支持。

Log record (type:subtype) in log nn, offset 0xn was not rolled back.

原因

日志撤销因日志已损坏而失败。

导致该错误的日志记录如下标识：

type

是逻辑日志记录类型。

subtype

是日志记录子系统。

log

是存储记录的逻辑日志 ID 。

offset

是日志中的十六进制地址位置。

操作

要确定是否执行该操作，请使用 glogdump 实用程序检查逻辑日志。联系技术支持。

Logical Recovery allocating nn worker threads thread_type.

原因

数据库服务器确定将用于并行恢复的工作程序线程的数量。变量 thread_type 可以采用值 ON_RECVRY_THREADS 或 OFF_RECVRY_THREADS 。

操作

此状态消息不需要任何操作。如果需要另一数量的工作程序分配用于并行恢复，请更改 ONCONFIG 配置参数 ON_RECVRY_THREADS 或 OFF_RECVRY_THREADS 。

Logical Recovery Started.

原因

逻辑恢复已开始。

操作

此状态消息不需要任何操作。

Maximum server connections number.

原因

具有每个 checkpoint 消息的输出指示自上一次重新启动以来到数据库服务器的并发连接

的最大数量。

操作

此消息帮助客户跟踪许可证的使用以确定何时需要购买更多许可证。有关帮助，请联系技术支持。

Memory allocation error.

原因

数据库服务器耗尽共享内存。

操作

执行下列操作之一：

1. 增加计算机上的交换空间。
2. 检查限制共享内存中核内存参数。
3. 使用 BUFFERPOOL 配置参数中的 buffers 字段降低已分配内存的大小。
4. 增加虚拟内存大小（SHMVIRTSIZE）、已添加段的大小（SHMADD）或全部共享内存大小（SHMTOTAL）。

Mirror Chunk chunkname added to space spacename. Perform manual recovery.

原因

快速恢复、完全恢复或 HDR 辅助服务器已恢复了镜像 chunk 的添加。然而，它不执行自动镜像恢复。管理员必须执行这一操作。

操作

使用 gspaces 实用程序尝试恢复此镜像 chunk 。

Mixed transaction result. (pid=nn user=userid).

原因

仅当数据库服务器涉及到多个事务时才会接收到此消息。此消息指示数据库服务器在准备好用于提交的事务之后，尝试回滚该事务且全局事务完成时不一致。pid 值是协调者进程的用户进程标识号。user 值是是与协调者进程相关联的用户 ID 。

操作

请参阅 GBase 8s 管理员指南 中有关从已失败两阶段提交手动恢复的信息。

mt_shm_free_pool: pool 0xn has blocks still used (id nn).

原因

池取消分配过程中发生内部错误，因为 block 仍与该池相关联。

操作

请联系技术支持。

mt_shm_init: can’t create resident/virtual segment.

原因

创建常驻段或虚拟段失败的原因为：(1) 段大小小于最小的段大小；(2) 段大小大于最大的段大小；(3)分配另一个段将超过允许的全部共享内存大小； (4)当数据库服务器尝试分配段时发生失败。

操作

如果怀疑此错误是由前段中的第 1 或 2 项生成，请联系技术支持。要更正第 3 项，请在 ONCONFIG 配置文件中增加 SHMTOTAL 值。有关第 4 项 ，请参阅逻辑日志文件。

mt_shm_remove: WARNING: may not have removed all/correct segments.

原因

当操作系统试图除去与数据库服务器相关联的共享内存段时，最后一段不等于内部注册的最后一段。这种情况很可能是由于数据库服务器的意外失败。

操作

除去还未清除的所有段。

消息：N-O-P

Newly specified value of value for the pagesize in the configuration file does not match older value of value. Using the older value.

原因

此消息在数据库服务器重新启动时显示。在数据库服务器已经初始化之后更改 ONCONFIG 文件中的 PAGESIZE 值。

操作

数据库服务器使用较旧的 PAGESIZE 值。

Not enough main memory.

原因

数据库服务器在尝试从操作系统获得更多内存空间时检测到错误。

操作

有关共享内存配置和管理的更多信息，请参阅操作系统文档。

Not enough logical-log files, Increase LOGFILES.

原因

在数据恢复过程中，LOGFILES 配置参数的值总是大于或等于逻辑日志文件的总数。在恢复过程的某个时刻，逻辑日志文件的数量超过了 LOGFILES 的值。

操作

在 ONCONFIG 中增加 LOGFILES 的值。

onconfig parameter parameter modified from old_value to new_value.

原因

当数据库服务器共享内存重新初始化时，此消息记载自上一次初始化以来所发生的所有更

改。

操作

不需要任何操作。

oninit: Cannot have SINGLE_CPU_VP non-zero and number of CPU VPs greater than 1.

原因

ONCONFIG 文件包含 num= 值大于 1 的 VPCLASS cpu ，而 SINGLE_CPU_VP 的值非零。当有多个 CPU VP 时，SINGLE_CPU_VP 必须是 0（或省略）。

操作

更正 ONCONFIG 文件并重新启动数据库服务器。

oninit: Cannot have SINGLE_CPU_VP non-zero and user-defined VP classes.

原因

ONCONFIG 文件包含用户定义的 VPCLASS 和非零值的 SINGLE_CPU_VP 。当 ONCONFIG 文件包含用户定义的 VPCLASS 时，SINGLE_CPU_VP 必须是 0（或省略）。

操作

更正 ONCONFIG 文件并重新启动数据库服务器。

oninit: Fatal error in initializing ASF with 'ASF_INIT_DATA' flags asfcode = '25507'.

原因

在数据库服务器的 sqlhosts 文件或注册表中指定的 nettype 值无效或不受支持，或者在数据库服务器的 sqlhosts 文件或注册表中指定的 servicename 无效。

操作

对于每个数据库实例，检查 sqlhosts 文件或注册表中的 nettype 和 servicename 值。检查 ONCONFIG 文件的每个 NETTYPE 参数的 nettype 值。

oninit: Too many VPCLASS parameters specified.

原因

ONCONFIG 文件中指定了太多的 VPCLASS 参数行。

操作

减少 VPCLASS 参数行（如果可能）。如果不可能，请联系技术支持。

oninit: VPCLASS classname bad affinity specification.

原因

VPCLASS 行的专业规范是错误的。专用作为范围进行指定：

对于 m， 使用处理器 m。

对于 m 到 n， 使用范围 m 到 n（包括 m、n）的处理器，其中 m <=n，m >= 0，n >= 0。

操作

更正 ONCONFIG 文件中的 VPCLASS 参数，并重新启动数据库服务器。

oninit: VPCLASS classname duplicate class name.

原因

ONCONFIG 文件中的 VPCLASS classname 具有重复的名称。VP 类名必须唯一。

操作

更正重复的名称并重新启动数据库服务器。

oninit: VPCLASS classname illegal option.

原因

VPCLASS classname 参数中的字段之一是非法的。

操作

更正 ONCONFIG 文件中的参数，并重新启动数据库服务器。

oninit: VPCLASS classname maximum number of VPs is out of the range 0-10000.

原因

VPCLASS 参数行锁指定的 VP 初始数量必须在范围 1 到 10,000 中。

操作

更正该值并重新启动数据库服务器。

oninit: VPCLASS classname name is too long. Maximum length is maxlength.

原因

VPCLASS classname 中的名称字段的长度太长。

操作

选择一个较短的类名，更正 ONCONFIG 文件，并重新启动数据库服务器。

oninit: VPCLASS classname number of VPs is greater than the maximum specified.

原因

由 VPCLASS 参数指定的 VP 初始数量大于同一 VPCLASS 参数指定的最大数量。

操作

更正 VPCLASS 参数，并重新启动数据库服务器。

oninit: VPCLASS classname number of VPs is out of the range 0-10000.

原因

VPCLASS 参数行所指定的 VP 初始数量必须在范围 1 到 10,000 中。

操作

更正该值并重新启动数据库服务器。

gadmin: VPCLASS classname name is too long. Maximum length is maxlength.

原因

动态添加的 VP 类名（由 gadmin -p 指定）太长。

操作

选择一个较短的名称，并重试 gadmin -p 命令。

Online mode.

原因

数据库服务器处于联机方式。用户可以访问所有的数据库。

操作

此状态消息不需要任何操作。

gspaces: unable to reset dataskip.

原因

此错误消息来自 gspaces 实用程序。出于某种原因，该实用程序无法跨数据库服务器实例中的所有 dbspace 更改 DATASKIP 的规范（ON 或 OFF ）。

操作

您不太可能接收到此消息，如果重新启动数据库服务器之后错误仍存在，那么联系技术支持。

Open transaction detected when changing log versions.

原因

数据库服务器在尝试从数据库服务器的以前版本转换数据时检测到打开的事务。

操作

除非日志中的最后一条记录是 checkpoint ，否则不允许转换，必须恢复数据库服务器的以前版本，强制 checkpoint，然后重试转换。

Out of message shared memory.

原因

数据库服务器无法向指定的段分配更多内存。

操作

有关更多信息，请参阅日志文件。

Out of resident shared memory.

原因

数据库服务器无法向指定的段分配更多内存。

操作

有关更多信息，请参阅日志文件。

Out of virtual shared memory.

原因

数据库服务器无法向指定的段分配更多内存。

操作

有关更多信息，请参阅日志文件。

PANIC: Attempting to bring system down.

原因

发生了致命的数据库服务器错误。

操作

查看引起紧急情况的错误、并尝试该错误消息建议的更正操作。有关可解释该失败的其他信息，另见消息日志文件中的其他消息。

Participant site database_server heuristically rolled back.

原因

远程站点在事务达到“准备好提交”阶段之后回滚事务。

操作

您可能需要在其他站点回滚该事务，然后重新启动它。

Physical recovery complete: number pages examined, number pages restored.

原因

此消息在快速恢复过程中显示。number of pages examined 指示存在于物理日志中页映象的数量。number of pages restored 指示已从物理日志恢复的实际页数。已恢复的页数总是小于或等于检查到的数量。

数据库服务器可能在 checkpoint 之间多次物理记录页映象。物理恢复只恢复第一个记录的页映象。

如果页留在内存缓冲池中，那么数据库服务器在每个 checkpoint 物理记录它一次，并在物理日志中存储一个页映象。如果缓冲池太小，那么被更新许多次的页可能会被强制离开缓冲池，进入到磁盘上，然后回到内存中以供下一次更新使用。每次页回到内存中，就再次对它进行物理记录，这导致在物理日志中有重复的页映象。

操作

如果 number of pages examined 远远大于 number of pages restored ，那么增加缓冲池的大小，以减少重复前映象的数量。有关更多信息，请参阅 GBase 8s 性能指南 。

Physical recovery started at page (chunk:offset).

原因

该消息在开始处过程中显示。Chunk 是包含物理日志的 chunk 编号。Offset 是物理日志条目开始处的页偏移量。物理恢复从该点开始恢复页。

操作

不需要任何操作。有关快速恢复的更多信息，请参阅 GBase 8s 管理员指南 。

Portions of partition partnum of table tablename in database dbname were not logged. This partition cannot be rolled forward.

原因

自上一次备份以来发生于操作表的轻附加。

操作

如果您要完全访问该表中的数据，那么需要将该表更改为行，然后更改成想要的表类型。此更改操作除去了由重放未记录操作（例如轻量级追加）所导致的表中的不一致性。

Possible mixed transaction result.

原因

该消息指示返回了错误 -716 ，与该消息相关联的是其上事务结果未知的数据库服务器列表。

操作

有关确定已不一致地实现了事务的信息，请参阅 GBase 8s 管理员指南 。

Prepared participant site server_name did not respond.

原因

联系远程站点 server_name 的尝试太多。在达到几个超时时间间隔之后，可判定该站点已关闭。

操作

验证远程站点是联机的，且已正确配置用于分布式事务。一旦远程站点准备好，就重新初始化该事务。

Prepared participant site server_name not responding.

原因

数据库服务器正在尝试联系远程站点 server_name 。由于某些未知的原因，数据库服务器无法联系该远程站点。

操作

验证远程站点是联机的，且正确配置用于分布式事务。

消息：Q-R-S

Quiescent Mode.

原因

数据库服务器已从某个其他状态进入静默方式。在 UNIX™ 上，只有已作为 gbasedbt 或 root 登录的用户才能与数据库服务器交互。在 Windows™ 上，只有 Gbasedbt-Admin 群组的成员才能与数据库服务器交互。没有用户可以访问数据库。

操作

不需要任何操作。

Read failed. Table name, Database name, iserrno = number

原因

读取指定系统表时出错。

操作

记下错误编号并联系技术支持。

Recovery Mode.

原因

数据库服务器已进入恢复方式。没有用户可以访问数据库，直到恢复完成。

操作

不需要任何操作。

Recreating index: 'dbsname:"owner".tabname-idxname'.

原因

在 DDL 语句隐式或显式建立一个或多个新索引后，但是在下一个 checkpoint 之前数据库服务器异常终止，重新创建新索引延迟直至逻辑恢复完成，而不是按行添加每个索引项行。逻辑恢复结束后，服务器开始并行索引构建以重建它们。该消息显示每个延迟的索引开始重建的时间。（但是如果索引在异常关闭之前已被删除，那么在逻辑恢复之后它将不会被重建，并且不会输出有关该索引的消息。）

操作

不需要任何操作。

Rollforward of log record failed, iserrno = nn.

原因

该消息显示在快速恢复或数据恢复过程中，数据库服务器是否无法前滚特定的逻辑日志记录。数据库服务器也许能够更改为静默或联机方式，但可能导致某些不一致性。有关进一步信息，请参阅恰在这条消息前面的那条消息。iserrno 值是错误编号。

操作

请联系 GBase 8s 技术支持。

Root chunk is full and no additional pages could be allocated to chunk descriptor page.

原因

Root chunk 已满。

操作

要释放 root chunk 中的空间，采取以下操作之一：

- 删除并重新创建 sysmaster 数据库。
- 将用户表从 root dbspace 移动到另一个 dbspace 。
- 对表进行重新分片。

scan_logundo: subsys ss, type tt, iserrno ee.

原因

日志撤销因日志类型 tt 已损坏而失败。

此消息中的变量具有以下值：

ss

是子系统名称。

tt

是逻辑日志记录类型。

ee

是 iserror 编号。

操作

使用 glogdump 实用程序检查逻辑日志以确定是否需要任何操作。联系技术支持。

Session completed abnormally. Committing tx id 0xm, flags 0xn.

原因

仅在数据库服务器正在尝试提交没有当前所有者的事务，且该事务发展成长事务时才发生异常会话结束。数据库服务器派生一个线程来完成该提交。

操作

不需要任何操作。

Session completed abnormally. Rolling back tx id 0xm, flags 0xn.

原因

仅在数据库服务器正在尝试提交没有当前所有者的分布式事务，且该事务发展成长事务时才发生异常会话结束。数据库服务器派生了已回滚该事务的线程。

操作

不需要任何操作。

semctl：errno = nn.

原因

当数据库服务器初始化信号量时发生了错误。返回了操作系统错误。

操作

请参阅操作系统文档。

semget：errno = nn.

原因

对信号量集的分配已失败。返回了操作系统错误。

操作

请参阅操作系统文档。

shmat: some_string os_errno: os_err_text.

原因

附加到共享内存段的尝试已失败。返回系统错误编号和建议的更正操作。

操作

查看更在操作（如果给出的话）并决定是否值得尝试。有关更多信息，请参阅操作系统文

档。

shmctl: errno = nn.

原因

当数据库服务器尝试除去或锁定共享内存段时出错。已返回了操作系统错误编号。

操作

请参阅操作系统文档。

shmdt: errno = nn.

原因

当数据库服务器尝试从共享内存段拆离时出错。已返回了操作系统错误编号。

操作

请参阅系统操作文档。

shmem sent to filename.

原因

断言失败的结果是：数据库服务器将共享内存的副本写到指定的文件中。

操作

无。

shmget: some_str os_errno: key shmkey: some_string.

原因

共享内存段的创建失败，或获取与特定键相关联的共享内存 ID 的尝试失败。返回系统错误编号和建议的更在操作。

操作

参考操作系统文档。

Shutdown (gadmin -k) or override (gadmin -O).

原因

Dbspace 在 checkpoint 时间间隔期间已关闭。当发生这种情况时，数据库服务器配置为等待重设。

当实际发生 checkpoint 时，出现以下消息：Checkpoint 已被关闭空间阻塞，正在等待重设或关闭 。

操作

关闭数据库服务器，或发出 gadmin -O 命令重设关闭的 dbspace 。有关 gadmin 实用程序的更多信息，请参阅 gadmin 实用程序 。

Shutdown Mode.

原因

数据库服务器正在从联机方式转换到静默方式。

操作

不需要任何操作。

Space spacename added.

原因

数据库服务器管理员向数据库服务器添加了一个新的存储空间 spacename 。

操作

不需要任何操作。

Space spacename dropped.

原因

数据库服务器管理员已从数据库服务器上删除了存储空间 spacename 。

操作

不需要任何操作。

Space spacename -\- Recovery Begins(addr).

原因

此参考消息指示数据库服务器正在尝试恢复存储空间。

此消息中的变量具有以下值：

spacename

是数据库服务器正在恢复的存储空间的名称。

addr

是控制 block 的地址。

操作

不需要任何操作。

Space spacename -- Recovery Complete(addr).

原因

此参考消息指示数据库服务器已恢复了存储空间。

此消息中的变量具有以下值：

spacename

是数据库服务器已恢复的存储空间的名称。

addr

是控制 block 的地址。

操作

不需要任何操作。

Space spacename -- Recovery Failed(addr).

原因

此参考消息指示数据库服务器无法恢复存储空间。

此消息中的变量具有以下值：

spacename

是数据库服务器未能恢复的存储空间的名称。

addr

是控制 block 的地址。

操作

不需要任何操作。

sysmaster database built successfully.

原因

数据库服务器已成功构建了 sysmaster 数据库。

操作

不需要任何操作。

Successfully extend physical log space

原因

已成功对指定路径下的文件 plog_extend.servernum 扩展了物理日志空间。

操作

不需要任何操作。

消息：T-U-V

The chunk pathname must have READ/WRITE permissions for owner and group.

原因

chunk pathname 没有正确的所有者和群组权限。

操作

确保已对 chunk 所在的设备指定了正确的权限（-rw-rw---）。

The chunk pathname must have owner-ID and group-ID set to gbasedbt.

原因

Chunk chunkname 没有正确的所有者和群组 ID。

操作

确保 chunk 所位于的设备具有所有者资格。在 UNIX™ 上，所有者和群组都应是 gbasedbt 。在 Windows™ 上，所有者必须是 Gbasedbt-Admin 群组的成员。

The chunk pathname will not fit in the space specified.

原因

Chunk pathname 不适合您所指定的空间。

操作

选择较小的 chunk 大小或释放 chunk 在其中创建的空间。

The DBspace/BLOBspace spacename is now mirrored.

原因

您已成功地将镜像添加到了所指示的存储空间。

操作

不需要任何操作。

The DBspace/BLOBspace spacename is no longer mirrored.

原因

您已结束对指示存储空间的镜像。

操作

不需要任何操作。

The number of configured inline poll threads exceeds the number of CPU virtual processors.

原因

当由 NETTYPE 配置参数指定的内联轮询线程的数量超过了由 VPCLASS 配置参数指定的 CPU 虚拟处理器的数量时，会生成该消息。配置在 CPU 虚拟处理上运行的轮询线程被作为内联轮询线程参考。

操作

更改 VPCLASS 配置参数，增加 CPU 虚拟处理器的数量，或修改 NETTYPE 配置参数，减少内联轮询线程的数量。

This ddl operation is not allowed due to deferred constraints pending on this table and dependent tables.

原因

当您尝试启动违例表且约束处于推迟方式时，返回此错误。

注： 如果启动违例表，然后将约束设置为推迟，那么不返回任何错误。然而，违例立即得到撤销，而不是写入已推迟约束缓冲区。有关更多信息，请参阅 《GBase 8s SQL 指南：语法》 。

操作

如果要启动违例表，那么必须将约束方式更改成立即，或提交该事务。

This type of space does not accept log files.

原因

将逻辑日志文件添加到 blobspace 或 sbspace 是不允许的。

操作

添加逻辑日志文件到 dbspace 。有关更多信息，请参阅 glogadmin -a -d dbspace: 添加逻辑日志文件。

TIMER VP: Could not redirect I/O in initialization, errno = nn.

原因

操作系统无法打开空设备，或复制与打开该设备相关联的文件描述符。已返回系统错误编号。

操作

请参阅操作系统文档。

Too Many Active Transactions.

原因

在数据恢复过程中有太多活动事务。在恢复过程中的某个时刻，活动事务的数量超过了 32 千字节。

操作

无。

Too many violations.

原因

诊断表中的违例数量超过了 START VIOLATIONS TABLE 语句的 MAX VIOLATIONS 子句中指定的限制。当目标表上的单个语句（例如 INSERT 或 UPDATE 语句）向违例表插入的记录比 MAX VIOLATIONS 子句指定的限制多时，向在目标表上发出该语句的用户返回此错误。

操作

要解决此错误，请执行以下操作之一：

- 当启动违例表时，忽略 START VIOLATIONS TABLE 语句中的 MAX VIOLATIONS 子句。此处，您将指定违例表中的行数没有限制。
- 将 MAX VIOLATIONS 设置为较高的值。

Transaction Not Found.

原因

逻辑日志损坏。当已启动了新事务，但是该事务的第一个逻辑日志记录不是 BEGWORK 记录时可能发生这种情况。

操作

请联系技术支持。

Transaction heuristically rolled back.

原因

在已完成两阶段提交中的第一阶段之后，发生回滚事务的试探性决策。

操作

不需要任何操作。

Transaction table overflow - user id nn, process id nn.

原因

当共享内存表中没有可用条目时，线程尝试在事务表中分配一个条目。消息中显示了请求线程的用户 ID 和进程 ID 。

操作

稍后重试。

Unable to create output file filename errno = nn.

原因

操作系统无法创建输出文件 filename 。errno 是返回的操作系统错误编号。

操作

验证该目录存在且具有写入权限。

Unable to extend nn reserved pages for purpose in root chunk.

原因

操作系统无法在 root chunk 中扩展到 nn 个保留页用于 purpose 的。（值 purpose 可以是 checkpoint /日志、 DBSpace 、Chunk 或镜像 Chunk。）

操作

减少所指示的资源的 ONCONFIG 参数，启动数据库服务器并释放主 root chunk 中的一些空间。然后重试相同的操作。

Unable to start SQL engine.

原因

数据库服务器遇到了内存耗尽的情况。

操作

不需要任何操作。

Unable to open tblspace nn, iserrno = nn.

原因

数据库服务器无法打开指定的 tblspace 。（值 nn 是 tblspace 编号的十六进制表示。）

操作

请参阅 ISAM 错误消息编号 nn ，它应能说明为何不能访问 tblspace 。错误消息显示在 GBase 8s 错误消息 中。

The value of pagesize pagesize specified in the config file is not a valid pagesize. Use 2048, 4096 or 8192 as the value for PAGESIZE in the onconfig file and restart the server.

原因

此消息在磁盘初始化时显示。ONCONFIG 文件中指定的 PAGESIZE 值是无效的值。

操作

以有效的 PAGESIZE 值重新启动数据库服务器。

Violations table is not started for the target table.

原因

如果在未启动任何违例表时发出 STOP VIOLATIONS TABLE 语句，您会接收到此消息。

操作

要从此错误进行恢复，那么必须启动目标表的违例表。

Violations table reversion test completed successfully.

原因

当 revtestviolations.sh 脚本已成功完成（未找到打开的违例表）时，在 sysmaster 数据库的 logmessage 表中记录此消息。

操作

不需要任何操作。有关 revtestviolations.sh 的更多信息，请参阅 GBase 8s 迁移指南 。

Violations table reversion test failed.

原因

当数据库服务器找到打开的违例表时，它在 sysmaster 数据库中的 logmessage 表中报告错误 16992 和 16993 并异常终止该复原进程。

操作

当此消息出现时，必须向每个打开的违例表发出 STOP VIOLATIONS TABLE FOR table_name 命令。在关闭所有打开的违例表之后，可以重新启动复原进程。

Violations table reversion test start.

原因

当执行 revtestviolations.sh 脚本时，此消息记录在 sysmaster 数据库的 logmessage 表中。

操作

不需要任何操作。有关 revtestviolations.sh 的更多信息，请参阅 GBase 8s 迁移指南 。

Violations tables still exist.

原因

当找到打开的违例表时，此消息记录在 sysmaster 数据库的 logmessage 表中。

操作

当此消息出现时，必须向每个打开的违例表发出 STOP VIOLATIONS TABLE FOR table_name 命令。在关闭所有打开的违例表之后，可以重新启动复原进程。

Virtual processor limit exceeded.

原因

配置数据库服务器使用的虚拟处理器数超过了所允许的最大数（1000）。

操作

要减少虚拟处理器的数量，修改 VPCLASS 配置参数的值、 NETTYPE 配置参数的值或

都修改。

VPCLASS classname name is too long. Maximum length is maxlength.

原因

该消息指示内部错误。

操作

请联系技术支持。

VPCLASS classname duplicate class name.

原因

该消息指示内部错误。

操作

请联系技术支持。

VPCLASS classname Not enough physical procs for affinity.

原因

VP 类 classname 的专用规范中的物理处理器不存在或脱机。

操作

确保指定的处理器联机。更正指定的 VP 类的专用规范。重新启动数据库服务器。

消息：W-X-Y-Z

WARNING: aio_wait: errno = nn.

原因

当数据库服务器正在等待 I/O 请求以完成时，它在其正在尝试执行的操作上生成错误号 nn 。

操作

请联系技术支持，以获得帮助。

WARNING: Buffer pool size may cause database server to get into a locked state. Recommended minimum buffer pool size is num times maximum concurrent user threads.

原因

缓冲池中没有足够的缓冲区。数据库服务器可能使用了所有可用的缓冲区并导致发生死锁。

操作

将 ONCONFIG 文件中 BUFFERPOOL 参数的 buffers 字段更改为此消息建议的数量。有关 BUFFERPOOL 参数的更多信息，请参阅 BUFFERPOOL 配置参数 。

warning: Chunk time stamps are invalid.

原因

当系统初始化时第一次打开 chunk 时执行 chunk 的稳定情况检查。所指定的 chunk未通

过检查并将变成脱机。

操作

从 dbspace 备份或其镜像恢复 chunk 。

Warning: name_old is a deprecated onconfig parameter. Use name_new instead. See the release notes and the GBase 8s Administrator's Reference for more information.

原因

使用了不推荐使用的 ONCONFIG 参数。您第一次使用不建议使用的参数时显示此消息。其后显示更短的消息。

操作

使用建议的替代 ONCONFIG 参数。

Warning: name_old is a deprecated onconfig parameter. Use name_new instead.

原因

使用了不推荐使用的 ONCONFIG 参数。

操作

使用建议的替代 ONCONFIG 参数。

Warning: Unable to allocate requested big buffer of size nn.

原因

对大缓冲区的内部内存分配已失败。

操作

增加虚拟内存大小（SHMVIRTSIZE）、已添加段的大小（SHMADD）或全部共享内存大小（SHMTOTAL）。

You are turning off smart large object logging.

原因

这些更改将成为新的 sbspace 缺省值。已对 sbspace 进行了更改。gspaces 实用程序将在同一时间读取和更新100 个智能大对象并将这 100 个智能大对象的每个 chunk 作为单个事务进行提交。此实用程序可能要花很长时间才能完成。

操作

当您发出以下命令时出现此参考消息：

gspaces -ch sbspace -Df “LOGGING=OFF“ -y

有关更多信息。请参阅 gspaces -ch: 更改 sbspace 缺省规范 。

消息：符号

HH:MM:SS GBase 8s database server Version R.VV.PPPPP Software Serial Number RDS#XYYYYYY.

原因

此消息指示数据库服务器的启动（在共享内存初始化之后）。

操作

不需要任何操作。

argument: invalid argument.

原因

此内部错误指示向内部传递了无效的参数。

操作

请联系技术支持。

function_name: cannot allocate memory.

原因

数据库服务器无法从内部共享内存池分配内存。

操作

增加虚拟内存大小（SHMVIRTSIZE）、已添加的段的大小（SHMADD）或全部共享内存大小（SHMTOTAL）。

转换和复原错误消息

如果转换复原不成功，那么存储在 online.log 文件中的错误消息将帮助您识别失败的原因和要修复该问题应采取的操作。

Cannot revert new fragment expression for index index, tabid id.

原因

该索引分段定义的版本比要复原的版本更新。

操作

删除有问题的索引分段模式并重试复原。

Cannot revert new table fragment expression for table with id id.

原因

该表分段所定义的版本比要复原到的版本更新。

操作

删除有问题的表分段模式并重试复原。

The conversion of the database name has failed.

原因

指示指定数据库转换失败。

操作

连接该数据库。此操作触发数据库转换。如果失败，那么会出现相关错误消息。请联系技术支持。

Database name is not revertible...

原因

数据库无法通过复原检查之一，且是不可复原的。

操作

执行操作更正作为单独消息显示的错误。

Database name: Must drop trigger (id = id_number) before attempting reversion.

原因

数据库包含的触发器是在比要转换到的版本更新的版本中创建的。

操作

删除带有指定触发器 id 的触发器，然后尝试复原。

The dummy updates failed while converting database name. This may imply data corruption in the database. If so, restore the original database with the tape backup. For more information, see output_file.

原因

在从比 Version 9.2 更早版本的数据库转换过程中，虚拟更新语句运行在受转换数据库中的系统表上。此消息指示运行这些更新语句之一时的失败。

操作

要重试虚拟更新，请运行旧数据库服务器版本的虚拟更新脚本。有关指示信息，请参阅 GBase 8s 迁移指南 。

如果发生了数据损坏，那么使用磁带备份恢复原始的数据库。有关更多信息，请参阅 GBase 8s 备份与恢复指南 。

Error in slow altering a system table.

原因

当执行复原时发生内部错误。

操作

请联系技术支持。

Internal server error.

原因

数据库复原过程中发生了意外错误。

操作

请联系技术支持。

Must drop long identifiers in table name in database name

原因

您正在复原到的数据库服务器版本不支持长度大于 18 字节的标识符。

操作

在尝试复原之前，确保已删除或重命名了该系统中的所有长标识符。

Must drop new database (name) before attempting reversion. Iserrno error_number

原因

系统包含创建于较新数据库服务器版本的数据库。

操作

删除新的数据库并尝试复原。

Must drop new user defined statistics in database name, iserrno number

原因

sysdistrib 系统表中的某些分布使用用户定义的统计信息。您正在复原到的版本不支持此功能。

操作

确保系统中不存在或未使用任何用户定义的统计信息，然后尝试复原。

Reversion canceled.

原因

复原进程因遇到错误而取消。

操作

更正错误原因，并重新启动复原。

There is a semi-detached index in this table, which cannot be reverted.

原因

无法复原此表上的半拆离索引。

操作

要查看所有半拆离索引的列表，请参阅数据库服务器的消息日志。这些索引是无法复原的。要继续复原，请删除这些半拆离索引并重试复原。如果需要，您将需要在复原完成之后重新创建这些索引。

WARNING: Target server version must have a certified Storage Manager installed after conversion/reversion and before bringing up server.

原因

ON-Bar 正在转换或复原。用户必须确保已安装了存储管理器（以目标数据库服务器版本进行了认证）。

操作

无。

Enterprise Replication 的转换和复原消息

在转换和复原过程中，通过 concdr 、revcdr 和 revtestcdr 脚本记录有关 Enterprise Replication 的特定消息。

缺省情况下，这些脚本将消息写到标准输出中。这些消息存储在 $GBASEDBTDIR/etc （在

UNIX™ 上） 或 %GBASEDBTDIR%\etc（在 Windows™ 上）的 concdr.out 、revcdr.out 和 revtestcdr.out 文件中。

CDR reversion test failed; for details look in $GBASEDBTDIR/etc/revtestcdr.out.

原因

Enterprise Replication 是不可复原的。

操作

有关更多信息，请查看 revtestcdr.out 中的消息。在尝试复原之前，修复已报告的问题。

将 revcdr.sh 或 revcdr.bat 脚本的输出打印到标准输出。

Enterprise Replication is not ready for conversion. The Control and TRG send queues should be empty for conversion/reversion to proceed.

原因

控制和事务发送队列（也称作 TRG ）中存在元素。数据库服务器将重复数据发送到目标系统中之前将它发送到 TRG 队列中。

操作

在尝试转换或复原之前等待这些队列清空。有关更多信息，请参阅 GBase 8s Enterprise Replication 指南 。

在转换过程中打印此消息到 concdr.out 或在复原过程中打印到 revcdr.out 中。

Enterprise Replication should be in a stopped state for conversion/reversion to proceed.

原因

Enterprise Replication 应处于已停止状态，以继续转换或复原。

操作

停止 Enterprise Replication。有关更多信息，请参阅 GBase 8s Enterprise Replication 指南 。

在转换过程中打印此消息到 concdr.out ，或在复原过程中打印到 revcdr.out 。

...‘syscdr’ reversion failed; for details look in $GBASEDBTDIR/etc/revcdr.out.

原因

syscdr 数据库复原失败。

操作

在 revcdr.out 文件中查找失败原因，然后在尝试失败复原之前修正该问题。

将 revcdr.sh 或 revcdr.bat 脚本的输出打印到标准输出。

'syscdr' conversion failed. For details, look in $GBASEDBTDIR/etc/concdr.out.

原因

syscdr 数据库的转换已失败。

操作

如果转换失败，那么解决 concdr.out 中报告的问题。从备份恢复 syscdr 数据库并重新尝

试转换。

将 concdr.sh 或 concdr.bat 脚本的输出打印到标准输出。

Syscdr should NOT contain new replicate sets for reversion to succeed.

原因

syscdr 数据库中的新复制集与较旧的版本不兼容。

操作

使用 cdr delete replicateset 命令删除复制集。然后再次运 revcdr.sh 或 revcdr.bat 脚本，以重新尝试复原。

将此消息打印到 revtestcdr.out 。

Syscdr should not contain replicates defined with the --floatieee option for reversion to succeed.

原因

已使用 --floatieee 选项定义了复制。您无法将这些复制复原到较旧的版本。

操作

使用 cdr delete replicateset 命令删除使用 --floatieee 选项定义的复制，然后重新尝试复原。

将此消息打印到 revtestcdr.out 。

动态日志消息

Dynamically added log file logid to DBspace dbspace_number.

原因

下一个活动日志文件包含打开事务的记录。每当数据库服务器动态添加日志，它就记录此消息。示例：已将日志文件 38 动态添加到 DBspace 5 。

操作

尽早完成事务。

Log file logid added to DBspace dbspace_number.

原因

每当管理员手工添加了日志文件，数据库服务器就记录此消息。示例：Log file 97 added to Dbspace 2.

操作

不需要任何操作。

Log file number logid has been dropped from DBspace dbspace_number.

原因

当您删除新添加的日志文件时，数据库服务器记录此消息。示例：Log file number 204 has been dropped from DBspace 17.

操作

不需要任何操作。

Log file logid has been pre-dropped.

原因

当您删除已使用的日志文件时，它被标记为“已删除”（状态 D）且不能再使用。在执行 0 级备份之后，数据库服务器删除该日志文件并可以再利用空间。示例： Log file 12 has been pre-dropped.

操作

要删除该日志文件，请对所有存储空间执行 0 级备份。

Pre-dropped log file number logid has been deleted from DBspace dbspace_number.

原因

备份之后，数据库服务器删除预先删除的日志文件并记录此消息。示例：Pre-dropped log file number 12 has been deleted from DBspace 3.

操作

不需要任何操作。

ALERT: Because the oldest logical log (logid) contains records from an open transaction (transaction_address), the server is attempting to dynamically add a log file. But there is no space available. Please add a DBspace or chunk. Then complete the transaction as soon as possible.

原因

如果数据库服务器因实例已耗尽空间而无法动态添加日志文件，那么它记录此消息。

操作

向现有 dbspace 添加 dbspace 或 chunk 。然后尽早完成该事务。

ALERT: The oldest logical log (logid) contains records from an open transaction (transaction_address). Logical logging will remain blocked until a log file is added. Add the log file with the glogadmin -a command, using the -i (insert) option, as in: glogadmin -a -d dbspace -s size -i. Then complete the transaction as soon as possible.

原因

如果 DYNAMIC_LOGS 参数设置为 1 ，那么数据库服务器将会提示管理员在需要时手工添加日志文件。

操作

使用带有 -i 选项的 glogadmin -a 命令在当前日志文件后面添加日志文件。然后尽早完成该事务。

Log file logid has been pre-dropped. It will be deleted from the log list and its space can be reused once you take level-0 archives of all BLOBspaces, Smart BLOBspaces and non-temporary DBspaces.

原因

当您删除已使用的日志文件时，它标记为“已删除”（状态 D）且不能再使用，且 glogadmin 打印此消息。

操作

要删除该日志文件，请对所有存储空间执行 0 级备份。

sbspace 元数据消息

Allocated number pages to Metadata from chunk number.

原因

数据库服务器已从保留区域释放了指定数量的页并将它们移动至 chunk number 的元数据区域。

操作

不需要任何操作。

Allocated number pages to Userdata from chunk number.

原因

数据库服务器已从保留区域释放了指定数量的页并将它们移动至 chunk number 的用户数据区域。

操作

不需要任何操作。

Freeing reserved space from chunk number to Metadata.

原因

chunk number 中的元数据区域已满。数据库服务器正在尝试从保留区域释放空间给元数据区域。

操作

不需要任何操作。

Freeing reserved space from chunk number to Userdata.

原因

chunk number 中的用户数据区域已满。数据库服务器正在尝试从保留区域释放空间给用户数据区域。

操作

不需要任何操作。

截断表消息

The table cannot be truncated if it has an open cursor or dirty readers.

原因

必须具有对表的互斥访问权。

操作

等待脏阅读器完成或关闭所有打开的游标并重新发出 TRUNCATE TABLE 命令。

The table cannot be truncated. It has at least one non-empty child table with referential constraints.

原因

如果表含有参考约束和至少一行的子表，那么您不能截断它。

操作

在截断该表之前请清空子表。

GBase 8s 中的限制

下列部分罗列 GBase 8s 选定的能力限制和系统缺省值。

UNIX™ 操作系统上的限制

系统级参数限制（UNIX™）

系统级参数	每计算机系统的最大能力
每台计算机的 GBase 8s 系统（依赖于可用的系统资源）	255
可访问的远程站点的最大数目	因机器而异
最大虚拟共享内存段（SHMVIRTSIZE）	2GB（32 位平台）或 4TB（64 位平台）
GBase 8s 共享内存段的最大数目	1024
最大地址空间	因机器而异

表级参数限制（UNIX™）

表级参数（基于 2K 页大小）	每表的最大能力
每页的数据行	255
每分片的数据行	4,277,659,295
每分片的数据页	16,775,134
每分片的数据字节（不包括“智能大对象”（BLOB、CLOB）和在 blospace 中创建的“简单大对象”（BYTE、TEXT））	33,818,671,136
“二进制大对象”BLOB/CLOB 页	4*2*40
“二进制大对象”TEXT/BYTE 字节	4*2*40
行长度	40 MB

表级参数（基于 2K 页大小）	每表的最大能力
列的数目	32 000
每索引分片的最大页数	2,147,483,647
每索引的键部分	16
每功能索引的列	102（对于 C UDR） 341（对于 SPL 或 Java™ UDR）
每索引键的最大字节（对于给定的页大小）：	2K 页大小 = 387 4K 页大小 = 796 8K 页大小 = 1615 12K 页大小 = 2435 16K 页大小 = 3254
SQL 语句大小的最大值	仅受可用内存的限制

存取能力（UNIX™）

存取能力	每系统的最大能力
每 GBase 8s 系统的数据库最大值	21 000 000 000
每 GBase 8s 系统的表的最大值	477 102 080
每 GBase 8s 的最大活动用户数（减去系统线程的最小数目）	32 000 用户线程
每数据库和表的最大活动用户数（还受可用锁的数目的限制，可调整的参数）	32 000 用户线程
一会话中打开数据库的最大数目	32 数据库
每 GBase 8s 系统打开表的最大数目	动态分配
每用户和联接打开表的最大数目	动态分配
每实例打开事务的最大数目	32 767
每 GBase 8s 系统和数据库的锁的最大数	动态分配
页清除程序的最大数目	128
每 dbspace 的最大分区数	4K 页大小：1048445，2K 页大小：1048314（基于 4 比特位图）
递归同义词映射的最大数目	16

存取能力	每系统的最大能力
每用户用 LOCK TABLE 锁定的表的最大数	32
每用户的游标最大数	因机器而异
Enterprise Replication 事务大小的最大值	4 TB
dbspace 大小的最大值	131 PB
sbspace 大小的最大值	131 PB
chunk 大小的最大值	4 TB
chunk 的最大数目	32 766
每 chunk 的 2K 页的最大数目	2 000 000 000
开放“简单大对象”的最大数目(仅适用于 TEXT 和 BYTE 数据类型)	20
B-tree 级别的最大数目	20
决策支持内存的最大数量	因机器而异
对大文件的实用程序支持	17 000 000 000 GB
存储空间(dbspace、blobspace、sbspace 或 extspace)的最大数目	2047

GBase 8s 系统缺省值(UNIX™)

下表中的每一行罗列一个数据库特性，后跟那个特性的 GBase 8s 系统缺省值。

数据库特性	GBase 8s 系统缺省值
表锁模式	页
初始 extent 大小	8 页
下一 extent 大小	8 页
Read-only 隔离级别(随同数据库事务)	Committed Read
Read-only 隔离级别(符合 ANSI 的数据库)	Repeatable Read

GBASE®

南大通用数据技术股份有限公司
General Data Technology Co., Ltd.

微信二维码

技术支持热线：400-013-9696