GBase 8c V5

Technical White Paper

www.gbase.cn

Copyright Notice

Disclaimer

Communication Method

General Data Technology Co., Ltd.

East Tower of Putian Innovation Industrial Park, No.22 Kaihua Road, Huayuan Industrial Park, Tianjin High-tech Zone, Tianjin, China

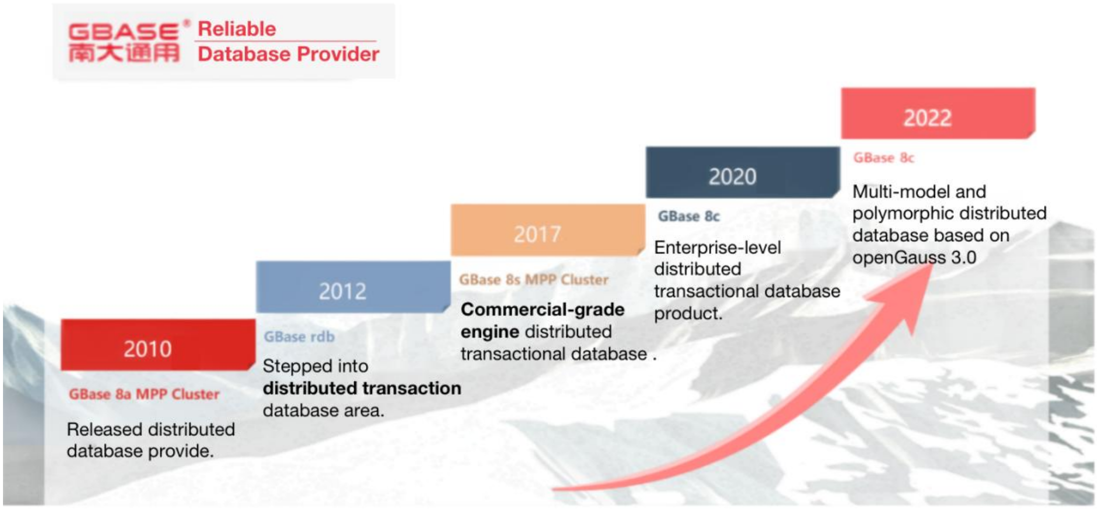
Tel: 400-013-9696 Email: info@gbase.cn

Trademark Statement:

Contents

1 Product Overview

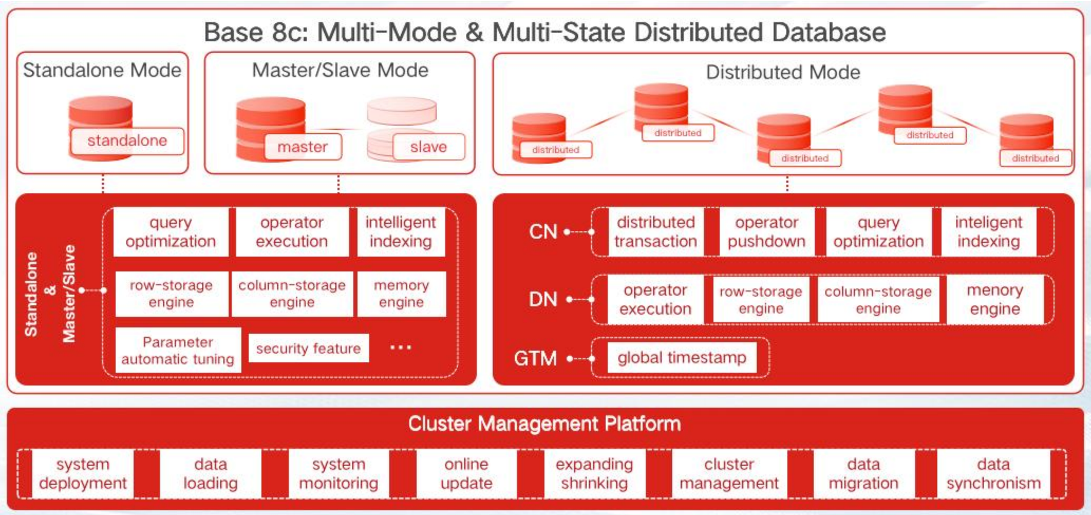
1.1 Development History



General Data Technology Co., Ltd. has been involved in the field of distributed databases since 2010. After years of accumulation and refinement, it launched GBase 8c, a multi-mode distributed database based on openGauss 3.0.

1.2 Product Positioning

GBase 8c is a multi-mode distributed database based on openGauss 3.0. It supports multiple storage modes such as row storage, column storage, and memory, as well as various deployment modes including standalone, master-slave, and distributed. GBase 8c is characterized by high performance, high availability, elastic scalability, and high security. It can be deployed on physical machines, virtual machines, containers, private clouds, and public clouds, providing secure, stable, and reliable data storage and management services for critical industries, internet businesses, and government and enterprise systems.



- ✓ **Multi-mode:** GBase 8c supports three storage modes including row storage, column storage, and memory, as well as three deployment modes including standalone, master/slave, and distributed to meet various business/scene requirements of users.

- ✓ **High Performance:** GBase 8c relies on openGauss kernel's concurrent control technology based on multi-core architecture, combined with Kunpeng hardware optimization and Numa-Aware data structure in key kernel structures, and elastic scalability with up to 0.9 or higher, providing a cluster with high performance capability.

- ✓ **High Availability:** GBase 8c in master/slave mode provides multiple deployment methods including master/slave synchronization, asynchronous, and cascading backup machines; the distributed mode has no single point of failure in the cluster, and high availability can reach 99.999%. The distributed cluster supports two sites and three centers, and supports cross-site active-active deployment to keep business continuously online with no data loss.

- ✓ **Ultimate Elasticity:** GBase 8c supports online expansion and contraction, and the computing storage separation architecture ensures that the cluster's computing and storage capabilities can be linearly improved with the increase of nodes.

- ✓ **Ultimate Security:** GBase 8c supports fully homomorphic computing capability, access control, encrypted authentication, database auditing, dynamic desensitization and other security features, providing end-to-end data security protection.

- ✓ **High Compatibility:** GBase 8c supports standard SQL syntax, fully compatible with PostgreSQL syntax, and highly compatible with Oracle syntax.
- ✓ **Distributed Capability:** GBase 8c supports strong consistency of distributed transactions, supports operator push-down, has multi-threaded parallel business processing capability, and adopts MVCC technology to achieve read-write concurrent processing capability.

1.3 Application Scenarios

- **Key Industries and Core Businesses**

Key industry core systems have strict requirements for database performance, stability, and security. As the business volume of key industry core systems continues to grow, traditional centralized databases have insufficient support for high concurrency and high throughput requirements, and distributed databases have become a powerful tool to solve this problem.

- ✓ GBase 8c has the core capabilities of distributed transaction strong consistency, computing power, and storage capacity linear scalability.
- ✓ Supports multi-site and multi-center, and cross-region active-active, providing 99.999% high availability.
- ✓ The database comes with a visual intelligent operation and maintenance platform, greatly reducing customer operation and maintenance complexity.

- **Internet Business**

With the rapid development of Internet businesses, especially mobile Internet businesses, the performance requirements for databases under massive data concurrency and throughput are getting higher and higher. Moreover, automatic scaling has become a necessary requirement for supporting extreme performance demands such as ‘flash sale’ scenarios for Internet applications.

- ✓ GBase 8c's elastic scaling capability can perform online scaling up and down according to business needs, perfectly coping with the performance requirements of extreme business scenarios.
- ✓ The feature of gray release provides powerful support for frequent online and offline activities of Internet businesses.

- **Government and Enterprise Business Migration to Cloud**

With the deep development of cloud computing and virtualization technologies, the migration of government and enterprise businesses to cloud has become the mainstream architecture choice.

- ✓ GBase 8c supports multiple deployment methods such as physical machines, containers, private clouds, and public clouds, providing flexible choices for government and enterprise business migration to the cloud.
- ✓ Supports row storage, column storage, and memory storage modes, and supports standalone, master-slave, and distributed deployment forms, meeting users' various business needs.
- ✓ GBase 8c supports data transmission encryption and storage encryption, providing powerful guarantees for cloud data security.

1.3.1 Typical Examples of Application Scenarios

Application Scenario	Scenario Characteristics	Advantages
Financial Industry - Credit card bill inquiry - Transaction detail inquiry **Telecommunications Industry** - User bill inquiry - Call detail inquiry **Payment Services** - Credit card transaction recording - Telephone bill details	1. High concurrency level 2. Mainly read operations, less or no involvement of distributed transactions	Basically no cross-node write operations, which can leverage the advantages of distributed systems. By parallel processing across multiple nodes, the entire query pressure can be evenly distributed to all nodes, thereby fully releasing the performance of the entire cluster.
Traditional Industries	1. Large amount of data	1. The data volume is large and

	2. Business requires data strong consistency	requires data splitting, making it suitable for distributed deployment. 2. Strong data consistency is required, and common solutions such as sharding and distributed middleware are difficult to meet, which is suitable for native distributed architecture products.
Significant fluctuations in website traffic -National centralized data submission and reporting	Business has periodic fluctuations and is difficult to handle	1. Leveraging the elastic scalability feature of the distributed architecture, the system's compute nodes can be scaled up during busy times to increase the overall concurrency processing capacity 2. Nodes can be reduced during idle times to save resources and reduce operating and maintenance costs.
GIS platform integration or data processing	Supports GIS features, including storage and access functions of geographic spatial data, and provides fast retrieval of geographic information.	OpenGauss/PG series products have good natural support for GIS, and GBase 8c, as a distributed database product, can provide good underlying support for building "GIS cloud platforms" in related scenarios.

1.4 Technical Features

As a multi-mode and multi-state distributed database product, GBase 8c has strong consistency in global transactions, computation and storage separation, flexible data distribution, flexible deployment, multi-mode and multi-state, online capacity expansion and contraction, online upgrade, high availability of data, high security, cross-site multi-activity, efficient data loading, cluster backup and recovery, easy maintenance, standardization, and compatibility with domestic ecology, etc.

- **Automatic fault-tolerant strongly consistent global transactions**

GBase 8c uses two-phase commit protocol and global timestamps to ensure strong consistency of global transactions. Each cross-node transaction either succeeds or fails completely, and there will be no situations where some node transactions succeed and others fail, achieving strong consistency of global transactions. GBase 8c's transaction processing has automatic fault-tolerant capability. After a node processing a transaction fails, a new node will continue to process the unfinished transaction, without the need for the application program to re-request.

- **Computing and storage separation**

GBase 8c adopts a share-nothing architecture with computing and storage separated. According to business needs, computing and storage capabilities can be horizontally scaled separately to achieve the goal of reducing overall ownership costs.

- **Flexible data distribution**

Users can choose data distribution strategies based on business scenarios to achieve the best match between performance, reliability, and flexibility.

GBase 8c supports replicated tables and distributed tables. Replicated tables are used to store read-only or read-mostly data and can perform joint queries with local and distributed tables, significantly improving query performance. Distributed tables are used to store data with large table scales and distributed to various storage nodes via Hash and other methods to reduce single-table data volume and improve data read-write performance.

- **Flexible deployment methods**

Users can choose to deploy GBase 8c in different environments based on their IT infrastructure construction and comprehensive considerations of performance, convenience, security, and other requirements. GBase 8c supports physical machine deployment, virtual machine deployment, container deployment, private cloud deployment, and public cloud deployment.

- **Multi-Mode**

GBase 8c supports three storage modes: row store, column store, and memory, and three deployment modes: standalone, master/slave, and distributed, meeting various business needs of users.

  - ✓ **Multiple Storage Modes:** GBase 8c supports multiple storage modes to meet the different business needs of various scenarios:

1. Row-based storage engine: designed mainly for OLTP scenarios, such as ordering, delivery, and banking transaction systems;
2. Column-based storage engine: designed mainly for OLAP scenarios, such as data statistics and report analysis systems;
3. In-memory engine: designed mainly for extreme performance scenarios, such as banking risk control scenarios.

- ✓ **Multiple Deployment Modes:** GBase 8c provides multiple deployment modes through a multi-tenant approach to meet various business needs, including standalone deployment, master-slave deployment, and distributed deployment. These deployment modes are managed through a unified operations and maintenance management platform, targeting both enterprise core transactions and future massive transaction scenarios, to create differentiated competitive advantages.

1. Standalone Deployment: GBase 8c supports standalone deployment, which can directly deploy the database on a single server. The advantage of this deployment is its low cost and simple deployment.
2. Master-Slave Deployment: GBase 8c supports the deployment of one master and multiple slaves, which can use synchronous or asynchronous backup methods between the master and slave nodes. This deployment mode is simple to deploy, has high delivery efficiency, and is suitable for scenarios with lower data volume, pursuit of extreme single-node performance, and requiring data backup.
3. Distributed Deployment: Distributed mode supports high availability of distributed full-component redundancy and deployment of calculation-storage separation. It can horizontally scale computing and storage capabilities based on business needs and is suitable for scenarios with large data volume, high concurrency, and pursuit of high data security.

- **Online Scaling**

GBase 8c supports online scaling, which redistributes data automatically and transparently to applications. During online scaling, business operations are not interrupted, ensuring continuous system availability.

- **Data High Availability**

GBase 8c ensures high availability of clusters through redundancy mechanisms, using master-slave replication to guarantee data consistency among multiple data copies. GBase 8c supports synchronous, asynchronous, and semi-synchronous replication modes, allowing users to balance high availability and high performance. GBase 8c supports automatic recovery from faults without human intervention.

- **Security**

GBase 8c provides a comprehensive user, role, and privilege control policy, improving the security of the database cluster.

Supports detailed audit logs, flexible audit policies, and monitoring tools to implement audit management.

Supports various encryption policies, including column-level, table-level, and database-level encryption.

Supports mainstream encryption functions, such as AES, MD5, SHA1, SHA, international standard algorithms.

- **Geo-Distributed Active-Active**

GBase 8c provides a geo-distributed active-active deployment mode to meet strict requirements for high availability. The geo-distributed active-active high-availability deployment can achieve RPO=0 and RTO in seconds, providing room-level and city-level disaster recovery capabilities.

- **Data Efficient Loading**

GBase 8c uses a strategy-based data loading mode, which fully leverages the computing power of all nodes to balance data loading speed and access performance.

- **Easy Maintenance**

Provides graphical deployment configuration, operation monitoring, data synchronization, backup and recovery, and other maintenance functions to reduce maintenance difficulties and improve maintenance efficiency.

- **Standardization**

Supports SQL92, SQL99, SQL2003 ANSI/ISO standards, ODBC, JDBC, ADO.NET, and other interface specifications. It also supports C API, Python API, and other interfaces.

- **Multiple Ecological Support**

GBase 8c, a distributed transactional database, has rich peripheral ecological support:

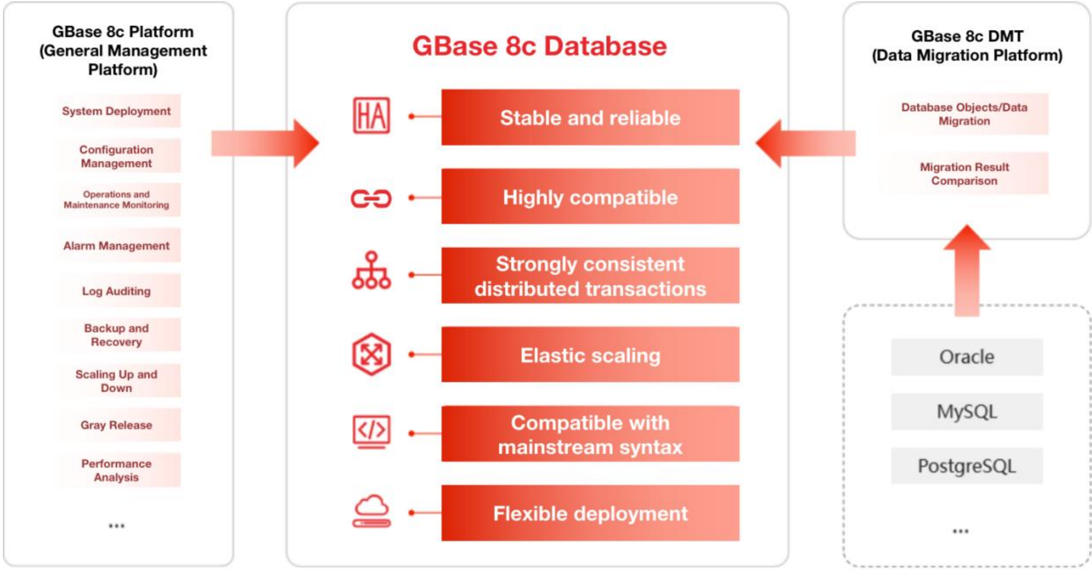
- ✓ Supports powerful geographic information system (GIS). GBase 8c supports the PostGis plug-in, which can effectively support spatial geographic data storage, serve as a spatial database, and efficiently manage spatial data, data measurement, and geometric topology analysis.
- ✓ Supports non-relational data type JSON. GBase 8c is not only a distributed transactional database system, but also supports non-relational data type JSON.
- ✓ Supports Foreign Data Wrappers (FDW). GBase 8c supports users to access data outside the database cluster through SQL by using FDW. The FDW function provides a programming interface, and users can conduct plug-in secondary development to establish data channels between external data sources and databases. Currently, GBase 8c supports oracle_fdw, mysql_fdw, postgres_fdw, as well as non-relational databases such as redis_fdw, mongodb_fdw, big data hive_fdw, hdfs_fdw, etc. Through FDW, GBase 8c can access data from multiple existing data sources.

1.5 Function Overview

Function	Description
SQL	• Supports standard SQL-92/SQL:1999/SQL:2003, supports GBK and UTF-8 and other mainstream character sets; • Supports DDL syntax such as CREATE, ALTER, and DROP; • Supports DML syntax such as SELECT, INSERT, UPDATE, DELETE, MERGE, and supports single-table and multi-table queries.
Data Type	• boolean data type; • integer, smallint, bigint, decimal, real, and other numeric data types; • char, varchar, and other character data types; • date, time, interval, timestamp, and other date and time data types; • bytea, text, and other large object data types; • point, line, path, circle, and other geographic data types.
Database Object	Provides the creation, modification, and deletion operations of commonly used database objects such as databases, tables, indexes, views, stored procedures, triggers, user-defined functions, and synonyms. It supports the creation and deletion of database users/roles, as well as the assignment and revocation of user permissions.
Function	• Supports various standard functions, including control flow functions, string functions, numeric functions, date and time functions, conversion functions, bit functions, encryption functions, information functions, auxiliary functions, aggregate functions, OLAP functions, regular expression functions, etc. • Supports user-defined function extensions in C and Python languages.
Transaction Characteristics	• Supports ACID transactions, supports RC and RR transaction isolation levels. • Supports pessimistic locks, MVCC (multi-version concurrency control), and flashback.
Distributed Characteristics	Supports data sharding, read-write horizontal scaling, and strong consistency distributed transactions.
Intelligent Management Platform	Provides a graphical intelligent management platform.
Interface	Compliant with and supports ODBC, JDBC, ADO.NET, and other interface specifications; supports C API, Python API, and other interfaces.
Security Management	Supports SSL secure network connections, user permission management, password management, security audit, and other functions to ensure the security of the database at the management, application, system, and network layers.

2 Product Family

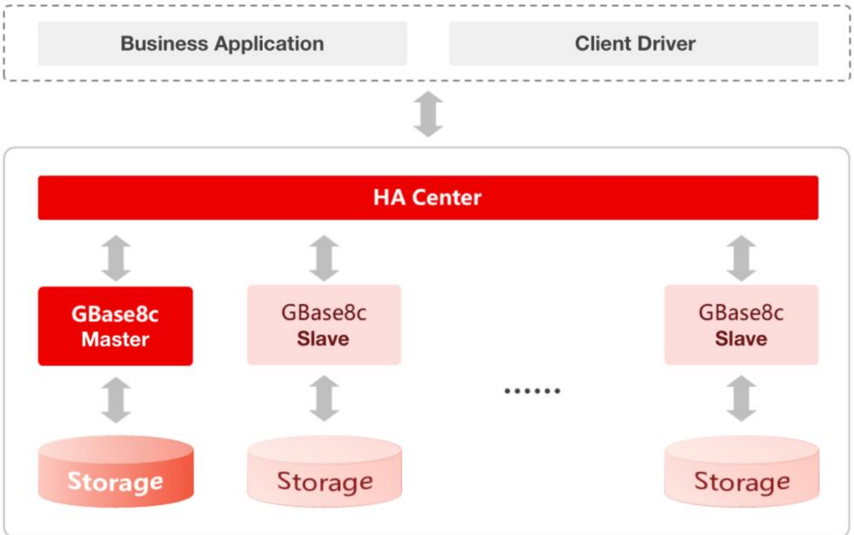
The product family of GBase 8c is shown in the following figure, which includes GBase 8c database, Universal Management Platform, Data Migration Platform, etc.



2.1 GBase 8c Database

2.1.1 Standalone/Master-Slave Architecture

The GBase 8c database in single-node or master-slave mode stores business data on a single physical node, and the high availability and scalability between the master and standby nodes are achieved through synchronous or asynchronous synchronization.

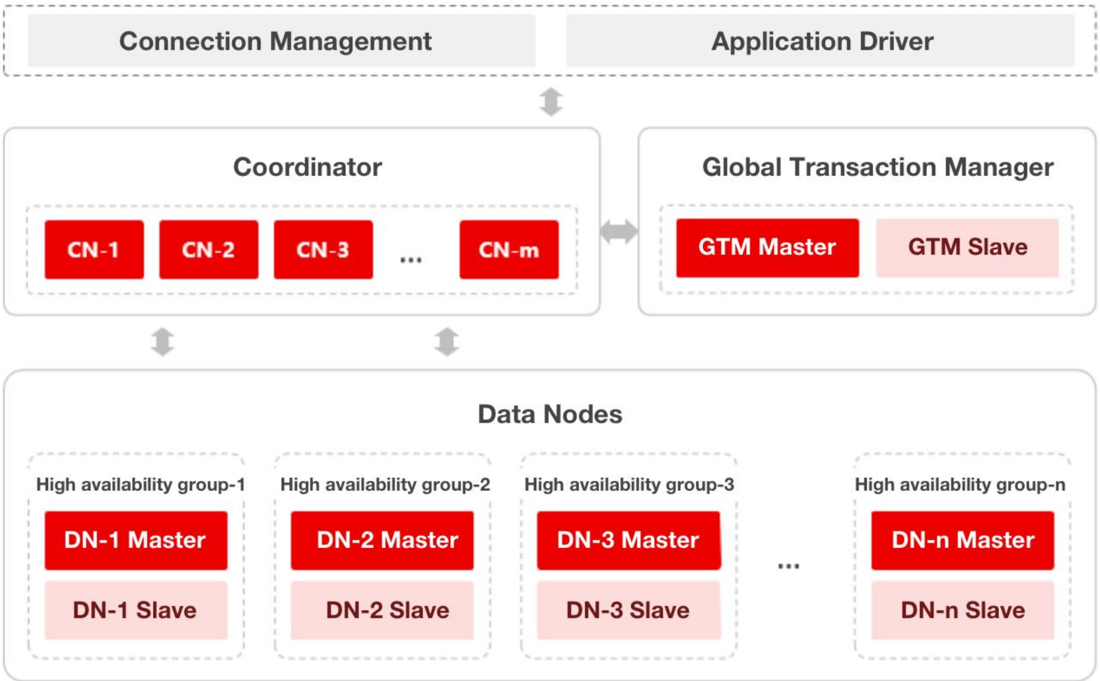


- ✓ **HA Center - High Availability Module:** HA Center is the high availability module of GBase 8c, responsible for determining node status in case of failures and performing state switching.
- ✓ **GBase 8c Master-Slave:** responsible for storing business data (supports row store, column store, and in-memory table storage), executing data queries, and returning execution results to the client driver.
- ✓ **Storage:** local storage resources of the server, responsible for persistently storing data.

2.1.2 Distributed Architecture

The distributed form of GBase 8c database adopts a share-nothing distributed architecture, with separate computing nodes and storage nodes, and communication between nodes through a high-speed network. All nodes have master-slave redundancy to ensure high availability of the system.

Since there is no resource sharing, adding nodes can linearly expand the storage and computing capacity of the cluster, meeting the requirements of business scale growth.

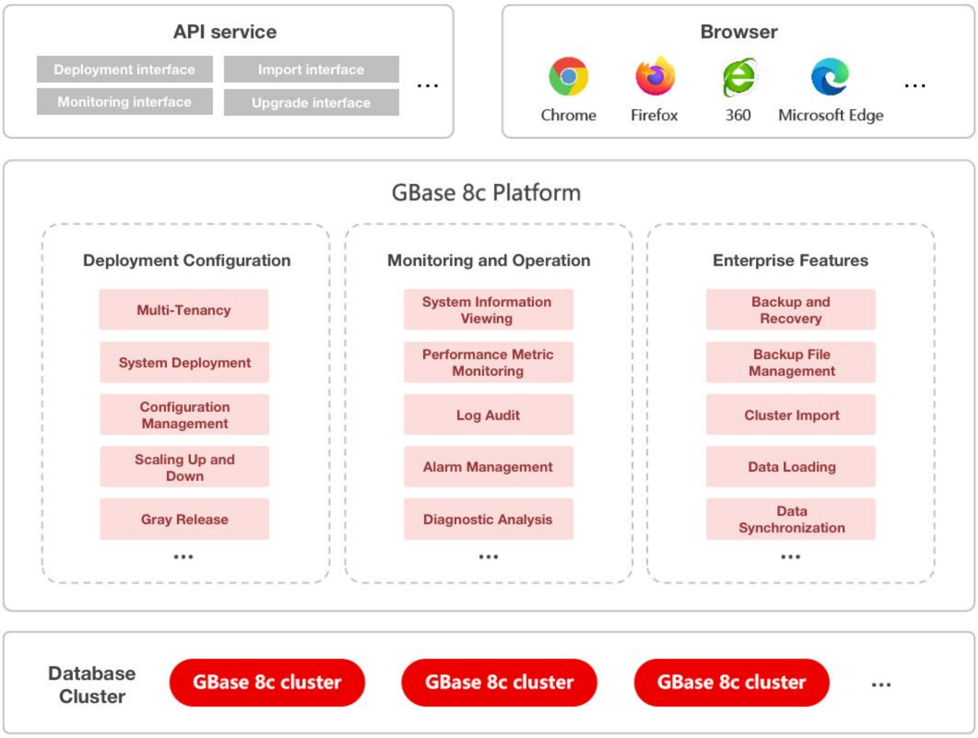


As shown in the above figure, the main nodes of GBase 8c are divided into three categories: Coordinator (CN), Data Node (DN), and Global Transaction Manager (GTM).

- ✓ **Coordinator:** The Coordinator manages the connection with clients, parses the SQL statements sent by clients, generates execution plans, sends the plans to the corresponding Data Nodes for read/write operations, and aggregates the results returned from Data Nodes to the client. For data write operations involving multiple Data Nodes, the Coordinator also coordinates all participants to perform two-phase commit.
- ✓ **Data Node:** The Data Node is the node where the data is actually stored, and it stores database objects such as tables and indexes. The Data Node receives read/write operations issued by the Coordinator and returns the results to the Coordinator for processing. Data Nodes can also communicate with each other to perform data redistribution and join queries across Data Nodes.
- ✓ **Global Transaction Manager:** The Global Transaction Manager manages global transaction IDs and active transaction states to ensure global consistency of the system.

2.2 GBase 8c Platform

GBase 8c Platform, the general management platform of GBase 8c, is a powerful database cluster management platform that provides graphical visualization for multi-tenant resource isolation, system deployment, configuration management, operation monitoring, alarm management, backup and recovery, scaling up and down, and gray release.



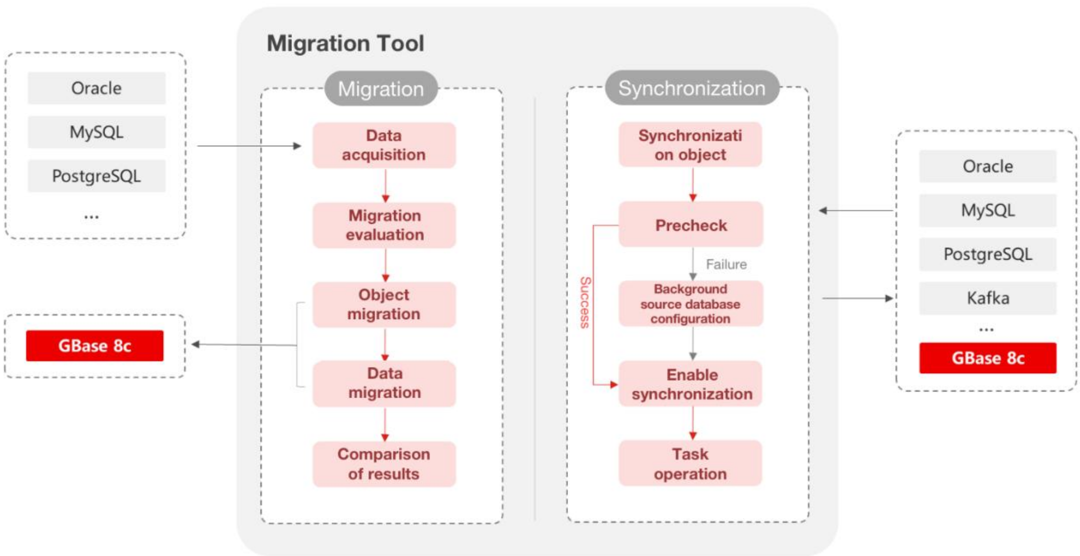
- ✓ **Multi-tenancy:** Supports allocating physical resources such as servers to tenants and creating, configuring, and managing standalone databases or database clusters (including master-slave, high availability distributed architectures) as needed, with resource isolation between tenants.
- ✓ **System deployment:** Supports one-click rapid deployment of database clusters and intuitively displays the execution status and results of each deployment step.
- ✓ **Configuration management:** Supports online parameter configuration management, allowing users to view cluster node parameter configurations through a graphical interface and make online modifications that take effect immediately.
- ✓ **Operation and maintenance monitoring:** Supports monitoring and alerts for database cluster status, providing multiple monitoring dimensions such as cluster, node, and server, deeply displaying the health status of each node, and providing a customizable metrics interface to allow users to monitor the real-time or historical running dynamics of the cluster.
- ✓ **Performance diagnosis:** Supports generating a Workload Diagnosis Report (WDR)

diagnostic report for a specified cluster and supports management operations for diagnostic reports.

- ✓ **Backup and recovery:** Supports using the cluster's data backup/restore feature through a graphical interface, selecting backup/restore methods, storage media, and intuitively displaying backup/restore progress and history.
- ✓ **Scale-out/scale-in:** Supports online scaling of database clusters to horizontally expand/reduce high availability groups or nodes.
- ✓ **Gray release:** Supports gray release of database versions by adjusting the load, enabling nodes to complete version updates in batches online, and intuitively displaying the cluster upgrade progress and process.

2.3 Data Migration Tool

The GBase 8c Data Migration Tool (DMT) is a platform that enables migration and synchronization between heterogeneous databases and the GBase 8c database. It also supports migration evaluation, migration comparison, and other functions.



- ✓ **Data Migration:** GBase 8c DMT supports structure migration, full data migration, and incremental data migration from heterogeneous databases to GBase 8c databases. In addition, during the migration process, it supports functions such as data statistics collection, migration assessment, migration adjustment retry, migration setting, and data-level and object-level migration comparison.

- ✓ **Data Synchronization:** GBase 8c DMT supports real-time data synchronization between GBase 8c and databases such as Oracle, MySQL, PostgreSQL, and Kafka. Users can create synchronization tasks, and the platform backend automatically builds the corresponding synchronization link, mainly supporting the scenario of incremental data synchronization, which synchronizes the source data changes to the target end in real-time.

3 Environment and Technical Specifications

3.1 Software and Hardware Operating Environment

Configuration Item	Configuration Requirements
Server	x86_64 and ARM standard PC servers, PowerLinux servers.
Hard Disk	The hard drive used for installing GBase 8c must meet the following requirements: • At least 4GB for installing GBase 8c application; • Approximately 300MB for metadata storage on each host; • Reserve over 70% and at least 500GB of disk space for data storage. It is recommended to configure the system disk as RAID1 and the data disk as RAID5, and plan for four groups of RAID5 data disks to install GBase 8c. For a single service data disk, it is recommended to use RAID5 (up to 8 disks) or RAID50 (more than 8 disks). Please refer to the hardware manufacturer's manual or online resources for RAID configuration methods. The "Disk Cache Policy" needs to be set to "Disabled" to prevent the risk of data loss in case of a power outage. GBase 8c supports using SSD as the main storage device for the database, which can be deployed in a RAID configuration and supports three types of SSD interfaces, including SAS, SATA, and NVME.
Internal Storage	• Minimum 32GB RAM is recommended for functional debugging. • For performance testing and commercial deployment, it is recommended to have a minimum of 128GB RAM per node. • Complex queries require a high amount of memory, and insufficient memory may occur in high-concurrency scenarios. In this case, it is

	recommended to use machines with large memory or use load management to limit system concurrency.
Storage	Supports local storage (SATA, SAS, NVMe, etc.), array storage (SAN, NAS), and software-defined storage (vSAN, Ceph, etc.).
CPU	• For functional debugging, a minimum of 1x8 cores at 2.0GHz is required. • For performance testing and commercial deployment, a minimum of 2x10 cores at 2.0GHz per node is recommended. • Both CPU hyperthreading and non-hyperthreading modes are supported. **Note:** Currently, GBase 8c supports Intel, AMD, etc. processors.
Network Requirements	• Supports Gigabit Ethernet, 10 Gigabit Ethernet, InfiniBand, and it is recommended to use 10 Gigabit Ethernet for data transmission between nodes. • It is recommended to configure the network card as a dual redundant bond. Please refer to the hardware vendor's manual or the Internet for configuration methods on network card redundancy bonding.
Operating Systems	CentOS, Red Hat, etc.

3.2 Technical Index

Technical Index	Maxinum
Database capacity	Limited by server configuration and quantity
Number of cluster nodes	1024
Number of shards	1024
Maximum table size	2^77 B
Maximum row size	1.6 TB
Maximum field size per record	1 GB
Maximum number of records per table	Unlimited
Maximum number of columns per table	250~1600 (depending on the field type)
Maximum number of indexes per table	Unlimited
Maximum number of columns in a composite index	32
Maximum number of constraints per table	Unlimited
Maximum number of concurrent connections	10000 * CN node number
Number of partitions for partitioned tables	32768 (for range partitioning)/64 (for hash partitioning or list partitioning)
Maximum size per partition for partitioned tables	128 TB * DN node number
Maximum number of records per partition for partitioned tables	Unlimited

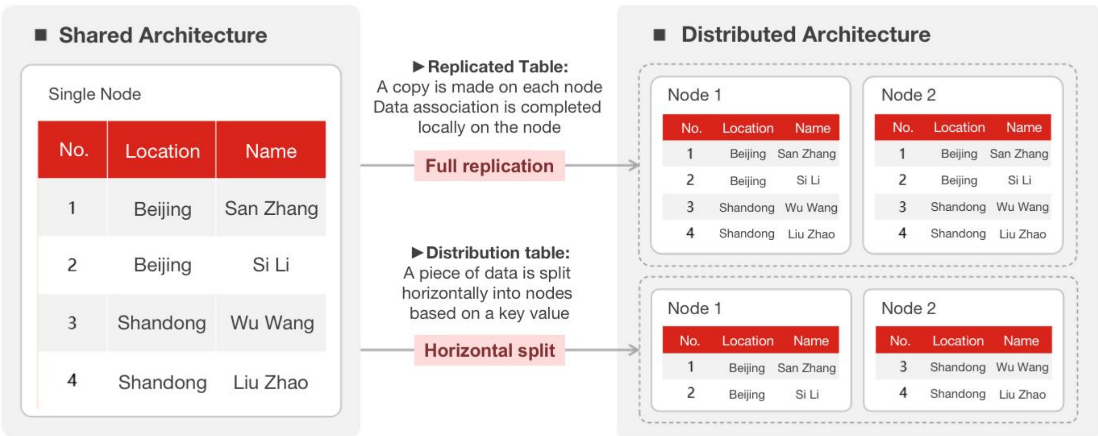
4 Core Technologies

4.1 Adaptive Transaction Processing Mechanism

GBase 8c adopts an adaptive transaction processing mechanism to improve system performance. For transactions that only need to be processed locally, the coordinator processes them according to the local transaction processing procedure without the need for two-phase commit to improve transaction processing efficiency. For transactions that need to be processed across nodes, the coordinator coordinates the participants to perform two-phase commit to ensure global transaction consistency. The entire transaction processing process is transparent to the client.

4.2 Data Distribution Strategy

GBase 8c supports replicated tables and distributed tables to avoid resource contention during parallel computing and improve system performance through data distribution strategies. Replicated tables refer to data that is replicated on each node, and data association is completed locally on the node. Distributed tables refer to splitting a piece of data horizontally into different nodes based on a key value, splitting a single large table into several small tables, and improving system read and write performance.



Replication tables and distribution tables are applicable in the following scenarios:

Table Type	Operation Type	Performance	Usage Scenario
Replication Table	Insertion	Relatively slow	Dictionary table Small table
	Query（JOIN）	Fast/high concurrency/linear improvement	
Distribution Table	Insertion	Fast/high concurrency/linear improvement	Fact table Large table Easily shardable table
	Query (multi-table single-shard)	Fast/high concurrency/linear improvement	
	Query (Single-table multi-shard)	Relatively fast	
	Query (multi-table multi-shard)	Slow	Avoid appearing
Hybrid of Replication and Distribution	Replicated table to a single distributed table JOIN Query	Relatively fast	Dimension table - Fact table Dictionary table - Fact table Small table - Large table

4.3 High Performance

4.3.1 NUMA Optimization

GBase 8c's NUMA optimization technology is designed to optimize resource consumption imbalance and high remote Node latency in concurrent scenarios when the database runs on NUMA architecture CPUs.

By setting the affinity of execution threads to Node nodes, GBase 8c database can bind threads to specified CPU cores to avoid cross-Node scheduling, reduce remote access latency, and improve the database's external processing performance.

At the same time, threads allocate local memory through Node, avoiding the situation where memory on one Node is exhausted while other Node's memory is idle, making full use of CPU, memory, and other resources to improve the database's external processing performance.

4.3.2 Parallel Computing

GBase 8c uses parallel computing technology to improve system performance and throughput, with the following features:

- ✓ The coordinator formulates the distributed execution plan and pushes the operators to the data nodes for parallel processing.
- ✓ Multiple threads are used for parallel processing at each data node.
- ✓ MVCC (Multi-Version Concurrency Control) is used to achieve read-write non-conflict and improve read-write parallel processing capability.
- ✓ Parallel queries are supported to solve the problem of decreased system concurrency caused by long query execution time in complex query scenarios, which affects the database's external service performance.

4.3.3 In-place Updating

PostgreSQL uses the MVCC mechanism for concurrency control:

- ✓ When a delete operation is performed, the database marks the tuple as dead, without physically deleting it.
- ✓ When an update operation is performed, the database writes a new tuple into unused

space and marks the old tuple as dead, without physically deleting it.

- ✓ When there is frequent DML on a table, dead tuples will gradually exhaust the space, and a lot of extra I/O will be generated when doing a full table scan.

GBase 8c uses in-place updating technology:

- ✓ New tuples are placed in the original position, and dead tuples are centrally stored in undo.
- ✓ Vacuum is removed to ensure stable I/O during data recovery.
- ✓ Data space is reduced.

PG stores data using an append-update method, which means that when data is modified, a new record is written instead of modifying the original one. This leads to space inflation, and it is necessary to periodically recycle expired data space. This has always been a weakness of PostgreSQL.

On the other hand, GBase 8c implements an Undo mechanism, which allows data to be updated in place. This brings benefits such as:

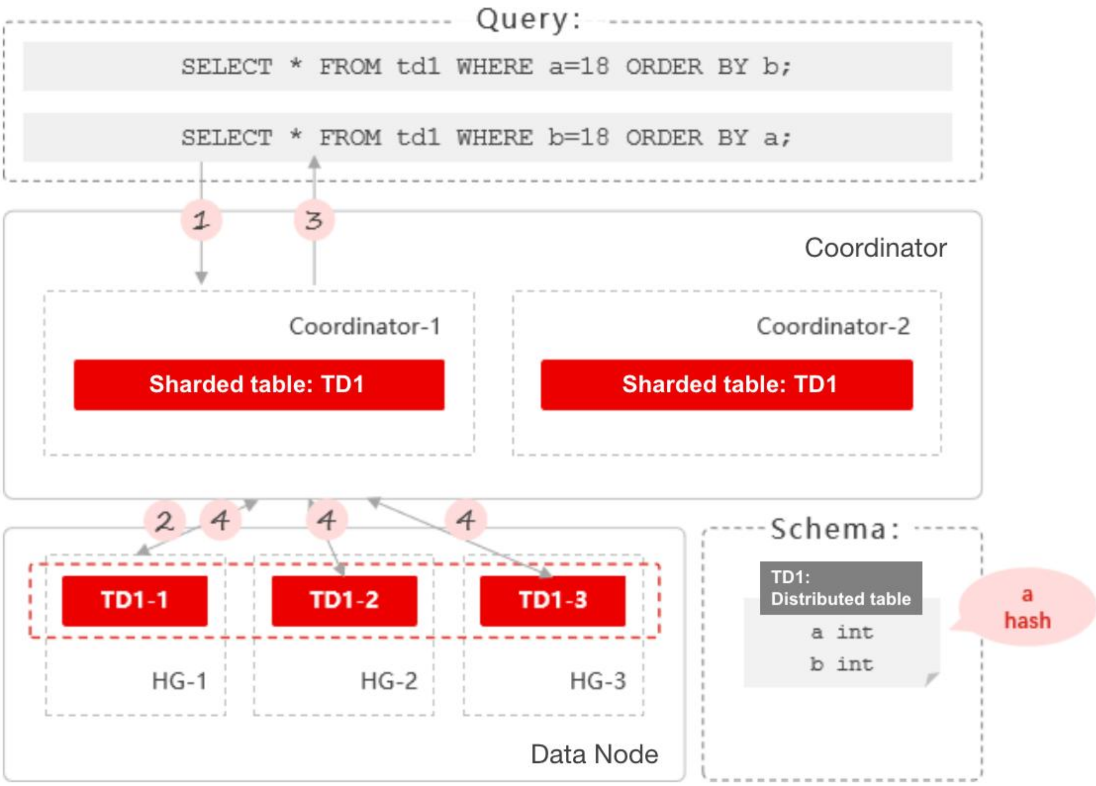
- ✓ High performance: for different loads such as insert, update, and delete, the performance and resource utilization are relatively balanced, with a 10% improvement compared to the Append Update engine.
- ✓ Smooth operation: the performance runs smoothly, and the 8-hour performance degradation value is reduced from 13.8% to 2.5%.
- ✓ Efficient storage: supports maximum in-place updating, and saves an average of 15% to 20% space under the TPCC load. UNDO space is uniformly allocated, centrally recycled, and more efficiently reused, resulting in more efficient and stable storage space usage.

4.3.4 Operator Pushdown

Operator pushdown is one of the key technologies of GBase 8c, which can push down various complex SQL operations and minimize data movement. This is the core advantage compared to the middleware solution based on sharding.

4.3.4.1 Single-table Query Pushdown

For single-table queries, whether or not the WHERE condition of SQL contains a sharding key, the optimizer can generate a pushdown execution plan, including complex operators such as sort/group by, which can all be pushed down.



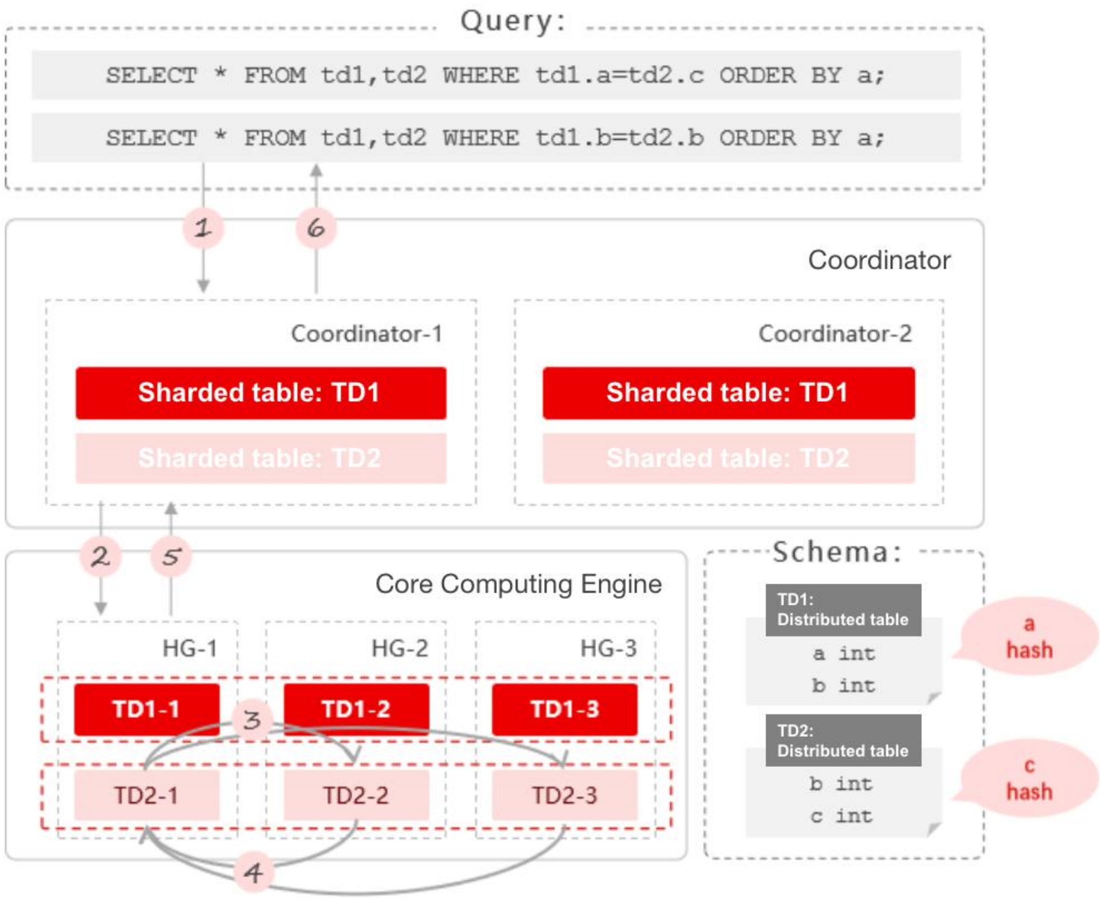
(1) The WHERE condition on the sharding key is directly pushed down to the corresponding data node for execution:

```
gbase=# EXPLAIN SELECT * FROM td1 WHERE a=18 ORDER BY b;
                                    QUERY PLAN
----------------------------------------------------------------
 Remote Fast Query Execution    (cost=0.00..0.00 rows=0 width=0)
   Node/s: dn2
   ->  Sort   (cost=38.44..38.47 rows=11 width=8)
         Sort Key: b
         ->  Seq Scan on td1    (cost=0.00..38.25 rows=11 width=8)
               Filter: (a = 18)
(6 rows)
```

(2) Non-sharding key WHERE conditions: The DN first performs the calculation, and the CN does the result aggregation. The GROUP BY operation can be directly pushed down to the DN.

```
gbase=# EXPLAIN SELECT * FROM td1 WHERE b=18 ORDER BY b;
                                   QUERY PLAN
----------------------------------------------------------------------------
 Remote Subquery Scan on all (dn1,dn2,dn3)   (cost=0.00..1.01 rows=1 width=8)
   ->  Seq Scan on td1   (cost=0.00..1.01 rows=1 width=8)
         Filter: (b = 18)
(3 rows)
```

4.3.4.2 Join Query Pushdown



（1）JOIN conditions on sharding keys are pushed down directly to the corresponding DN for execution:

```
gbase=# EXPLAIN SELECT * FROM td1,td2 WHERE td1.a=td2.c ORDER BY a;
                                    QUERY PLAN
----------------------------------------------------------------------------
 Remote Subquery Scan on all (dn1,dn2,dn3)  (cost=2.04..2.05 rows=1 width=16)
   ->  Sort  (cost=2.04..2.05 rows=1 width=16)
         Sort Key: td1.a
         ->  Nested Loop  (cost=0.00..2.03 rows=1 width=16)
               Join Filter: (td1.a = td2.c)
               ->  Seq Scan on td1  (cost=0.00..1.01 rows=1 width=8)
               ->  Seq Scan on td2  (cost=0.00..1.01 rows=1 width=8)
(7 rows)
```

(2) Non-shard key JOIN conditions, DN performs data exchange directly, avoiding CN becoming a performance bottleneck:

```
gbase=# EXPLAIN SELECT * FROM td1,td2 WHERE td1.b=td2.b ORDER BY a;
                                                 QUERY PLAN
-----------------------------------------------------------------------------------------------------
 Remote Subquery Scan on all (dn1,dn2,dn3)  (cost=2.04..2.05 rows=1 width=16)
   ->  Sort  (cost=2.04..2.05 rows=1 width=16)
         Sort Key: td1.a
         ->  Nested Loop  (cost=0.00..2.03 rows=1 width=16)
               Join Filter: (td1.b = td2.b)
               ->  Remote Subquery Scan on all (dn1,dn2,dn3)  (cost=100.00..101.02
rows=1 width=8)
                     Distribute results by H: b
                     ->  Seq Scan on td1  (cost=0.00..1.01 rows=1 width=8)
               ->  Materialize  (cost=100.00..101.03 rows=1 width=8)
                     ->  Remote Subquery Scan on all (dn1,dn2,dn3)
(cost=100.00..101.02 rows=1 width=8)
                           Distribute results by H: b
                           ->  Seq Scan on td2  (cost=0.00..1.01 rows=1 width=8)
(12 rows)
```

✓ Join pushdown to DN for execution, DN directly performs data redistribution and exchange between each other without the participation of CN. The CBO optimizer selects small table t2 for redistribution.

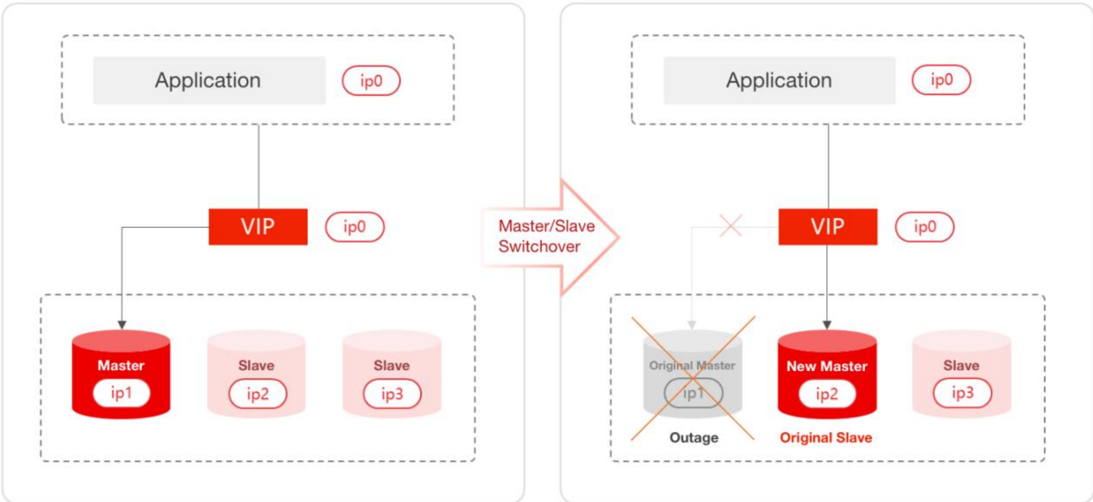
✓ Sort pushdown to DN, CN only needs to do merge sorting, avoiding becoming a performance bottleneck.

4.4 High Availability

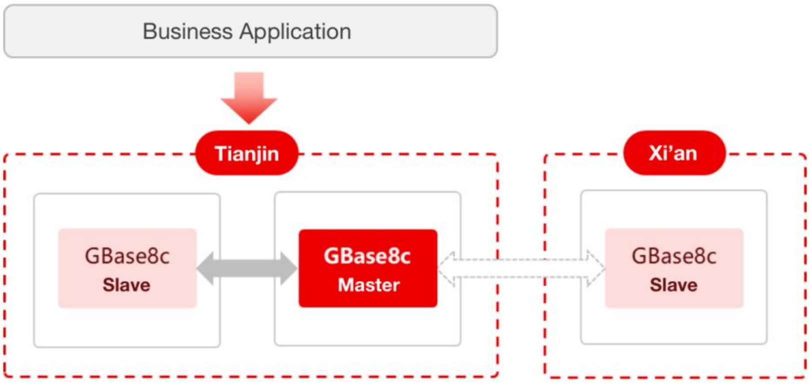
4.4.1 Master-Slave High Availability Architecture

GBase 8c master-slave version supports data redundancy with multiple copies, and data exchange between master and slave replicas is carried out through logs, ensuring that any node failure in the cluster does not affect the database's external service and that data is not lost, satisfying the ACID properties.

When the master node fails (including but not limited to server failure, database service failure, network interruption, power failure, disk failure, etc.), the slave node can automatically upgrade to the master and continue to provide external services. This process is transparent to the application, not limited by the master-slave IP address. That is, when the slave machine is brought up, it will automatically be assigned the IP address that the current application is connecting to, and the entire master-slave switching process is transparent to the application and does not affect external services.



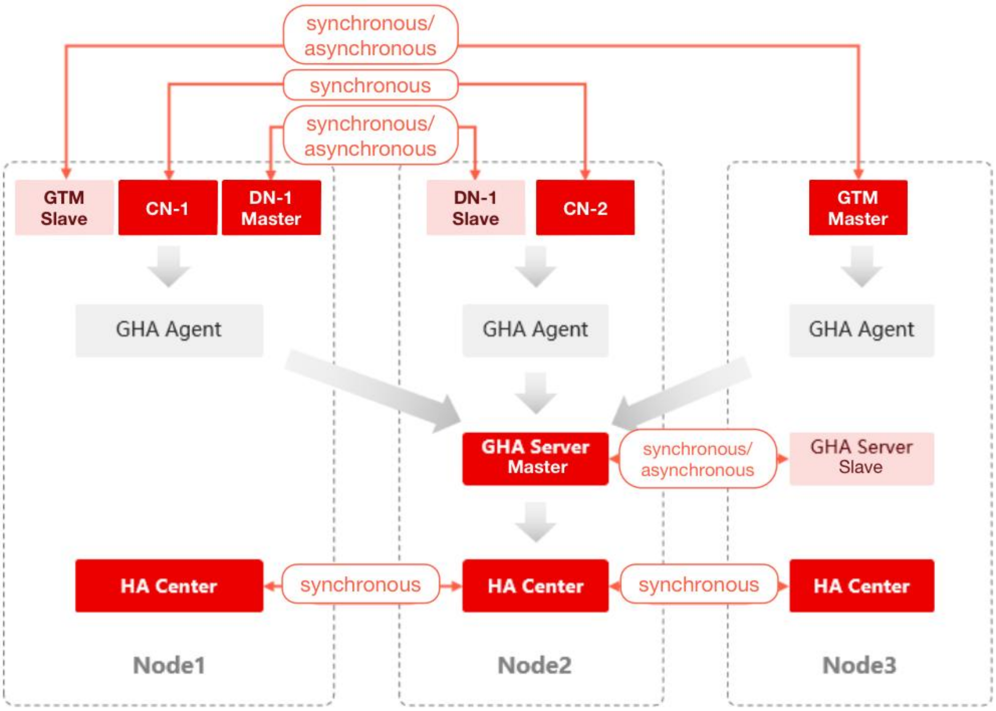
As shown in the figure, GBase 8c provides the VIP (Virtual IP) feature, and the application connects to the VIP directly to access the database master node. When a master node exception occurs, resulting in a master-slave switch, there is no need to manually switch the IP address of the application connection. The cluster will automatically bind the VIP to the new master node to complete the cluster's automatic switch and ensure that the upper-level application is not affected.



GBase 8c's master-slave deployment supports both synchronous and asynchronous backup methods between the master and slave nodes. Generally, remote disaster recovery nodes use asynchronous backup, while local backup nodes use synchronous backup.

4.4.2 Distributed High Availability Architecture

The high availability architecture of GBase 8c is achieved through distributed full-component redundancy, which means that at the software level, component-level redundancy is implemented for each component in the distributed cluster.

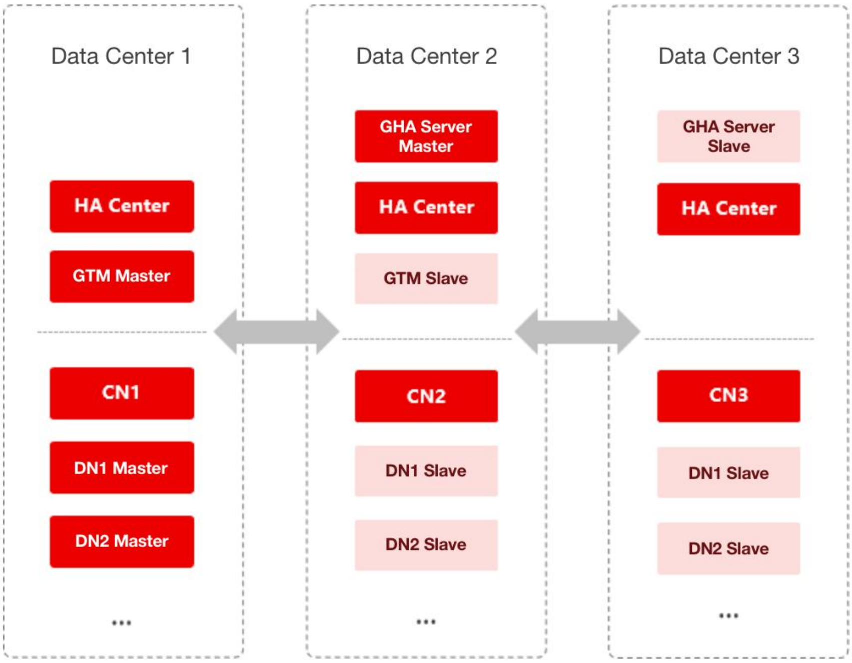


- ✓ **CN:** coordinator, deployed in a completely peer-to-peer manner;
- ✓ **DN:** data node, using a master-slave high availability architecture, with synchronous or asynchronous configuration between the master and slave;
- ✓ **GTM:** global transaction manager, using a master-slave high availability architecture, with synchronous or asynchronous configuration between the master and slave;
- ✓ **HA Center:** cluster state manager, using the Raft replication protocol;
- ✓ **GHA Server:** cluster manager, using a master-slave high availability architecture, with synchronous or asynchronous configuration between the master and slave.

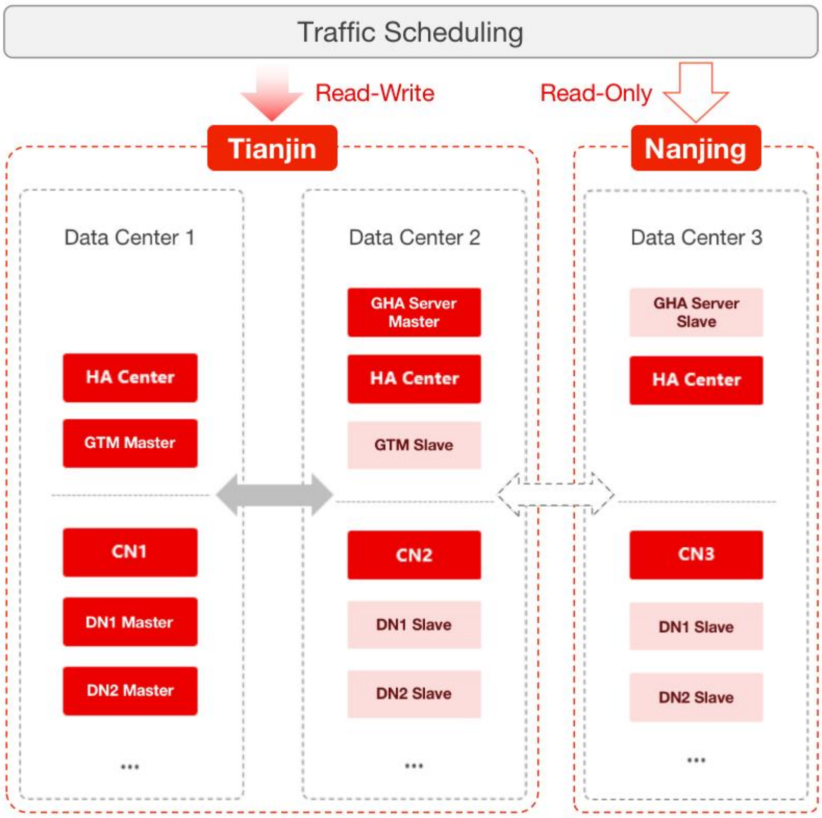
GBase 8c can meet different high availability requirements for databases in various application scenarios.

1. Intra-Data Center Disaster Recovery: adopting the primary/standby hot standby scheme in the same data center can resist hardware-level failures but cannot resist city-level or data center-level disasters. The failover is automatic, with RPO=0 and RTO in seconds.

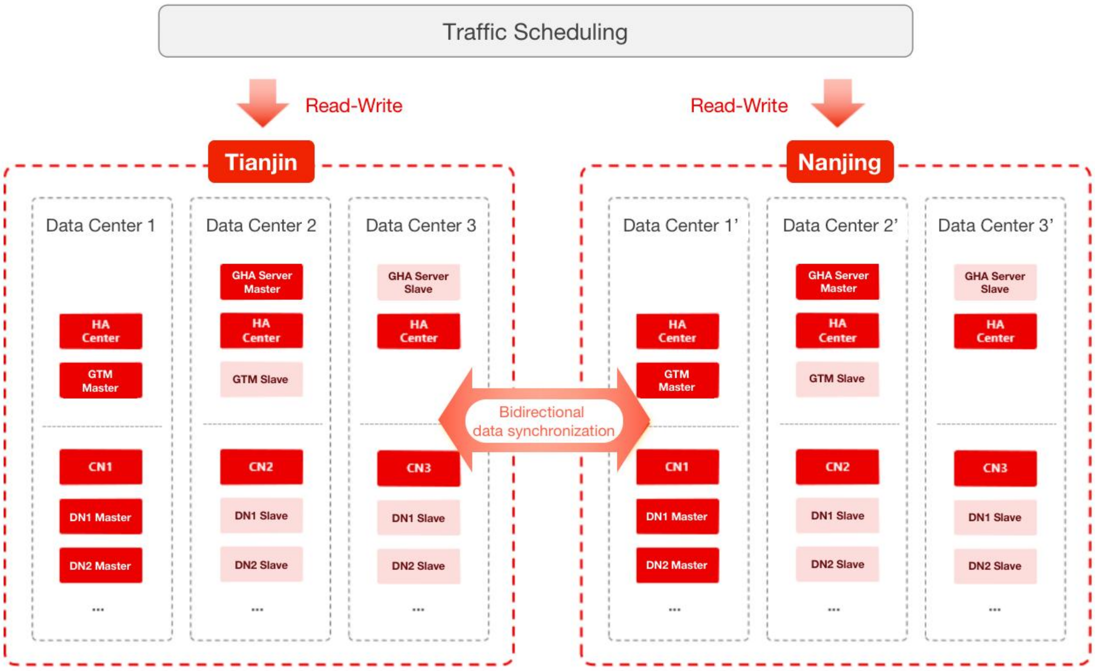
2. Intra-City Disaster Recovery: adopting the primary/standby hot standby scheme within the same city can resist hardware-level failures and data center-level disasters, but cannot resist city-level disasters. The failover is automatic, with RPO=0 and RTO in seconds. The distance between the two data centers should be less than 50 kilometers for intra-city disaster recovery.



3．Disaster recovery across regions: GBase 8c cluster adopts corresponding high availability deployment methods for different nodes, and uses asynchronous replication backup between two locations. It can withstand hardware-level failures and disasters at the level of a data center or a city, and the distance between the two locations can be greater than 1000 kilometers.



4. Cross-Region Active-Active: GBase 8c supports multi-cluster deployment, with data synchronization between clusters in both directions. Data is partitioned and scheduled based on a certain dimension. This deployment can withstand hardware failures, disasters at the level of data centers and cities. The distance between the two regions can be greater than 1000 kilometers.



4.5 Distributed Transactions

GBase 8c provides the ability to perform distributed strong consistency transactions through the use of the Global Transaction Manager (GTM) and local two-phase commit (2PC) technology. Meanwhile, for emerging database businesses that prioritize performance, GBase 8c also supports optional eventual consistency transactions.

Atomicity and Two-Phase Commit Protocol for Distributed Transactions

To ensure the atomicity of distributed transactions and prevent the occurrence of "intermediate state" transactions where some DN submits and some DN rolls back, GBase 8c uses a two-phase commit (2PC) process to implement cross-node distributed transactions.

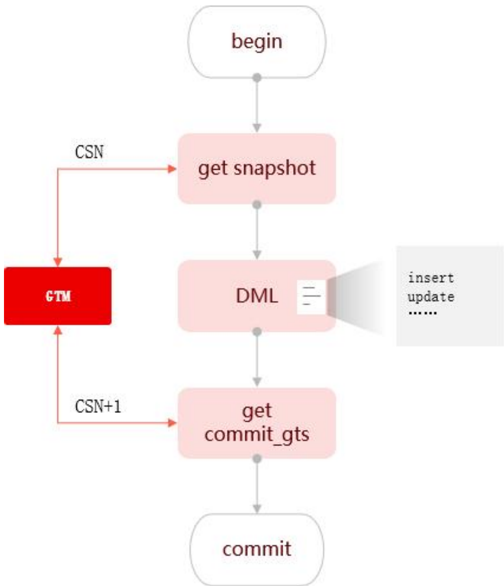
（1）In the prepare phase, all the information and resources required for the submit operation are written to disk and persisted.

（2）In the commit phase, the commit or rollback operation is executed based on the previously prepared commit information and resources.

Once the prepare phase is successfully executed, all the information required for commit is persistently written to disk. Even if a DN encounters an execution error in the subsequent commit phase, it can attempt to commit again from the persistently stored commit information until the commit succeeds. Ultimately, the status of this distributed transaction on all DNs is identical, either all DNs submit or all DNs rollback. Therefore, for external observers, the status change of the transaction is atomic.

Consistency of Distributed Transactions and Global Transaction Management

GBase 8c uses the TSO (timestamp ordering) scheme based on the global transaction commit timestamp to ensure distributed transaction consistency. The processing flow is shown in the following figure:



1. GTM is responsible for maintaining the global timestamp CSN;

2. When a transaction is started, it obtains the current timestamp from GTM.

3. When the transaction is committed, it obtains the timestamp again (CSN+1).

The advantages of this approach are:

✓ Using the global logical timestamp CSN instead of the traditional active transaction list as the global snapshot can greatly reduce the network overhead of all nodes to the transaction manager GTM node, making the global transaction management node no longer an easy bottleneck node for distributed transaction processing.

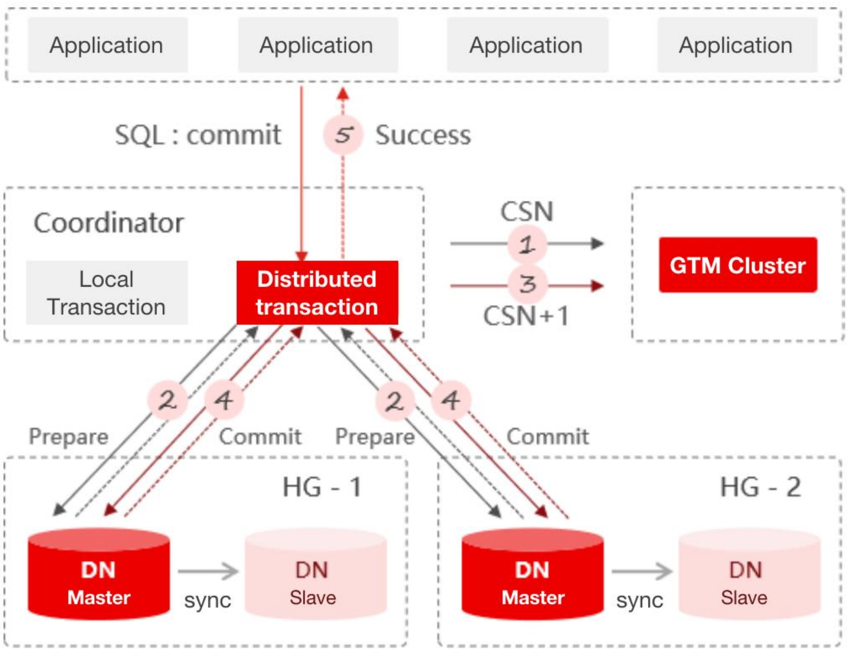
✓ Using the logical timestamp CSN essentially performs internal sorting for all write transactions globally, which provides convenience for handling database bidirectional synchronization and other aspects.

Example:

Begin; //Transaction Start

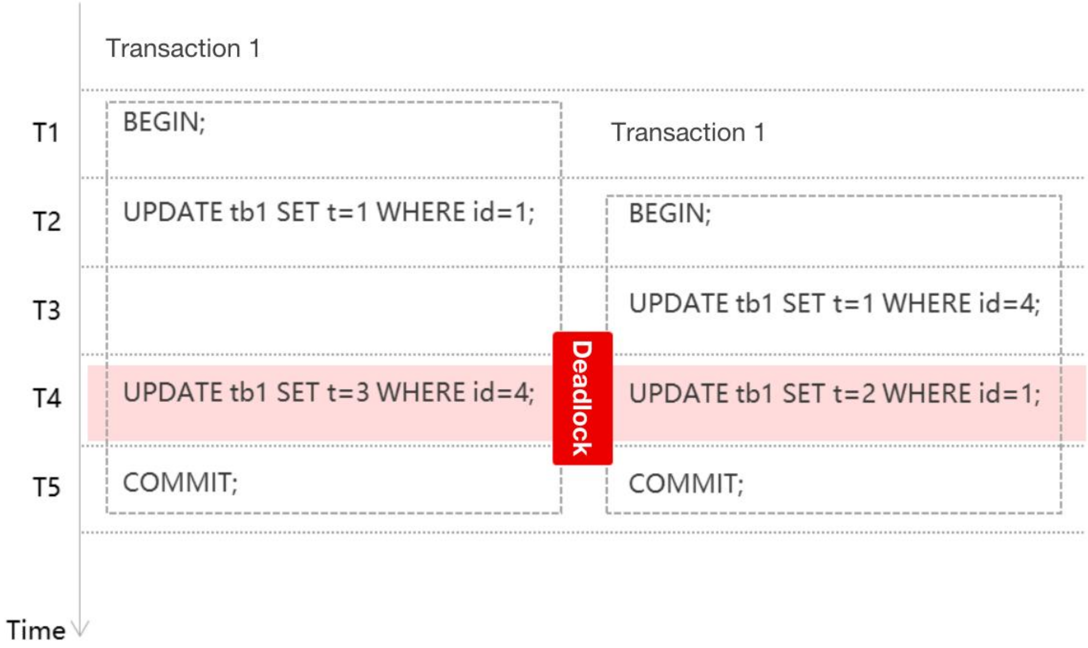
1. Select * from t1 where id = 1; //Single-node query
2. Select * from t1, t2 where t1.id = t2.id;//Cross-node quer
3. insert into t1 values(1, “aaa”); //Single-node write
4. Update t1 set name = ”bbb” where id = 1; //Single-node update
5. Delete from t2 where id < 10; //Cross-node deletion

Commit; //Transaction Commit



4.6 Global Deadlock Resolution

Global deadlock: refers to the situation where multiple database processes on multiple CNs and DN in a database cluster call resources from each other and enter into a circular waiting state. For example:

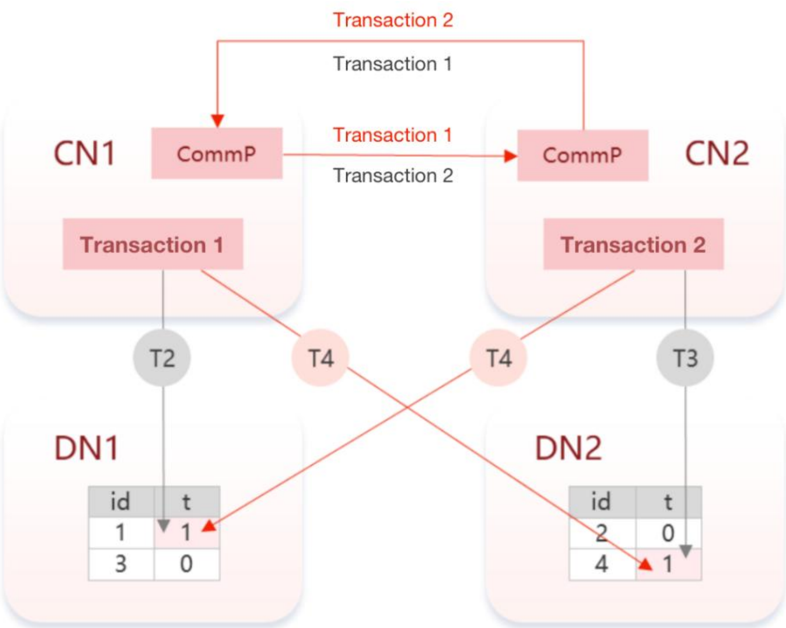


At T1, Transaction 1 begins;

At T2, Transaction 1 updates the t value with id=1, and at the same time, Transaction 2 begins;

At T3, Transaction 2 updates the t value with id=4;

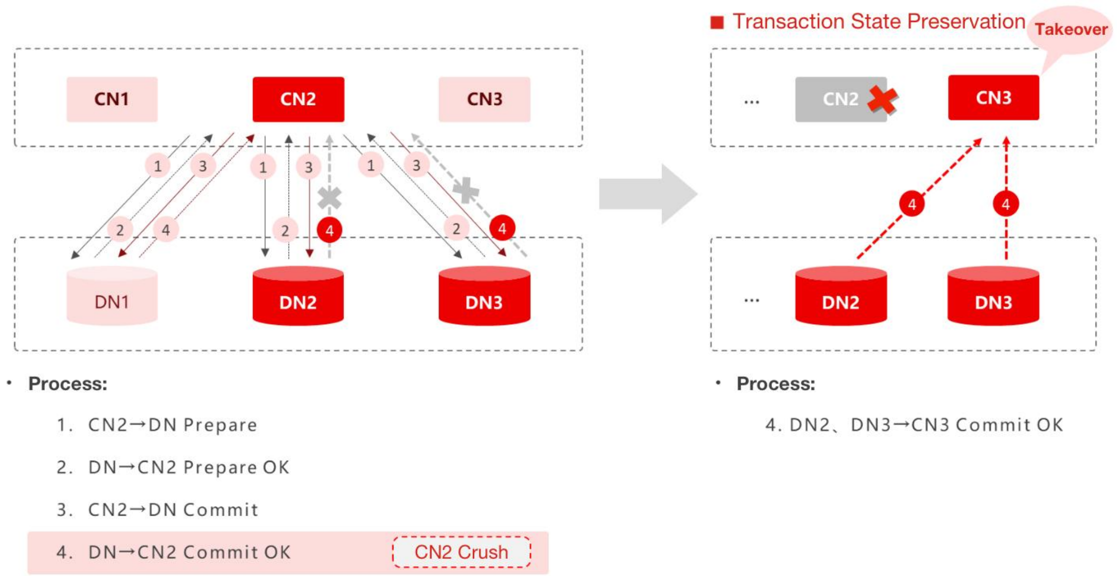
At T4, Transaction 1 needs to update the t value with id=4, while Transaction 2 needs to update the t value with id=1. At this point, the two transactions are in a circular wait, resulting in a global deadlock.



In GBase 8c, after a deadlock cycle is detected between nodes, the operation of exiting the transaction of the first node that discovers the deadlock cycle is performed to resolve the global deadlock issue.

4.7 Transaction State Persistence

GBase 8c has the ability to maintain transaction state. When any coordinator node (CN) fails, the ongoing transaction state of that node is not affected, and the transaction is automatically migrated to another CN to continue running, ensuring that database processing capacity is not interrupted.



In the figure above, when CN3 takes over the transaction, it does not need to repeat the successfully committed transaction state from earlier, but instead continues to complete the state that CN2 failed to complete for this transaction. The entire process is transparent to upper-layer applications, and the failure of any node in the database cluster does not result in deadlock or abnormal waiting situations.

4.8 Global CDC

GBase 8c supports the Global Change Data Capture (CDC) feature, which makes it easy for users to perform full database backups and data extraction. CDC methods for capturing data changes mainly include timestamp, snapshot, trigger, and log. GBase 8c's log-based CDC method does not affect the performance of the source database.

4.9 Backup and Recovery

Massive amounts of business data not only present challenges to data processing and analytical query performance, but also demand higher requirements for data backup and recovery. Without efficient backup and recovery capabilities, in the event of accidents, failures, or disasters, databases cannot be restored in a timely manner, and system and business availability cannot be guaranteed.

The GBase 8c cluster has the ability to perform full and incremental backups and recovery. The backup method and frequency can be configured on the general management platform, and all backup records can be viewed. It provides comprehensive backup and recovery functions based on cluster, database, and table levels, including:

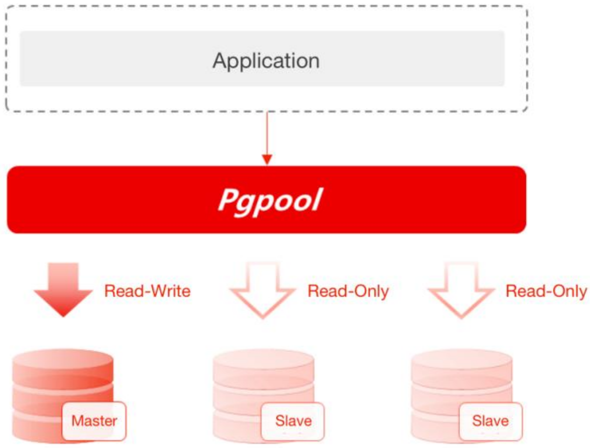
1. Full backup and recovery;
2. Incremental backup and recovery: allows data recovery based on any backup point.

Support Parallel Recovery

When the host logs are transferred to the backup host, the redo recovery distribution thread sends the logs to multiple parallel recovery threads based on the log type and data page of the log operation, which ensures that the redo speed of the backup machine keeps pace with the rate of log generation on the host. This keeps the backup machine in a ready state in real-time, thus achieving instantaneous fault switching.

4.10 Read/Write Separation

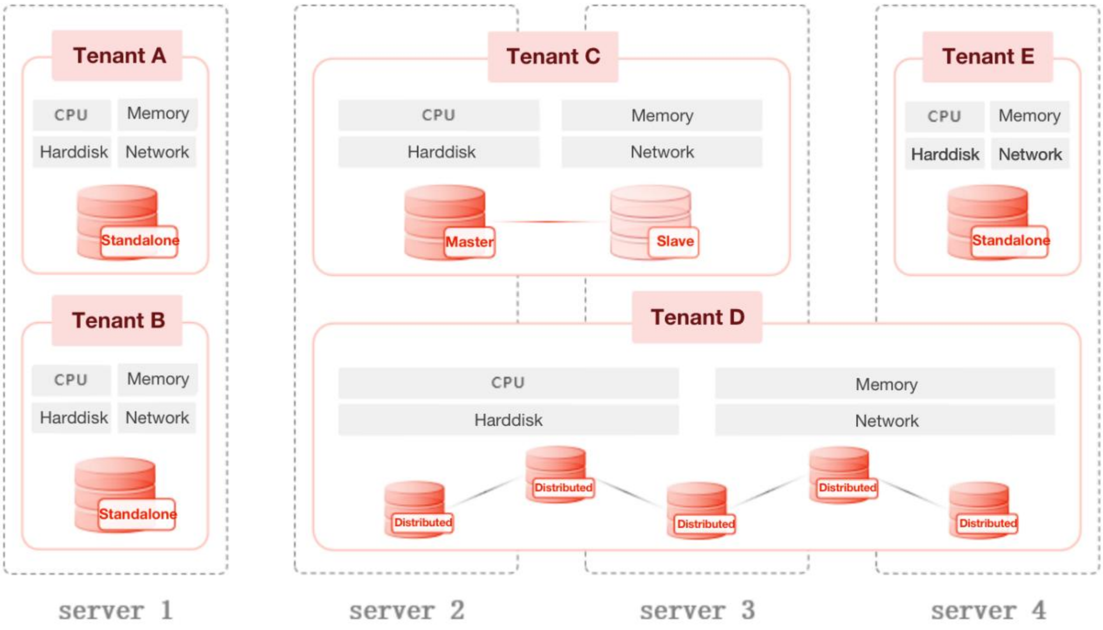
GBase 8c supports statement-level and connection-level read/write separation. Statement-level read/write separation can automatically recognize read and write requests and assign them to the master and slave nodes respectively, while connection-level read/write separation supports configuring read/write connections on the master node and read-only connections on the backup node through connection settings.



5 Advanced Product Features

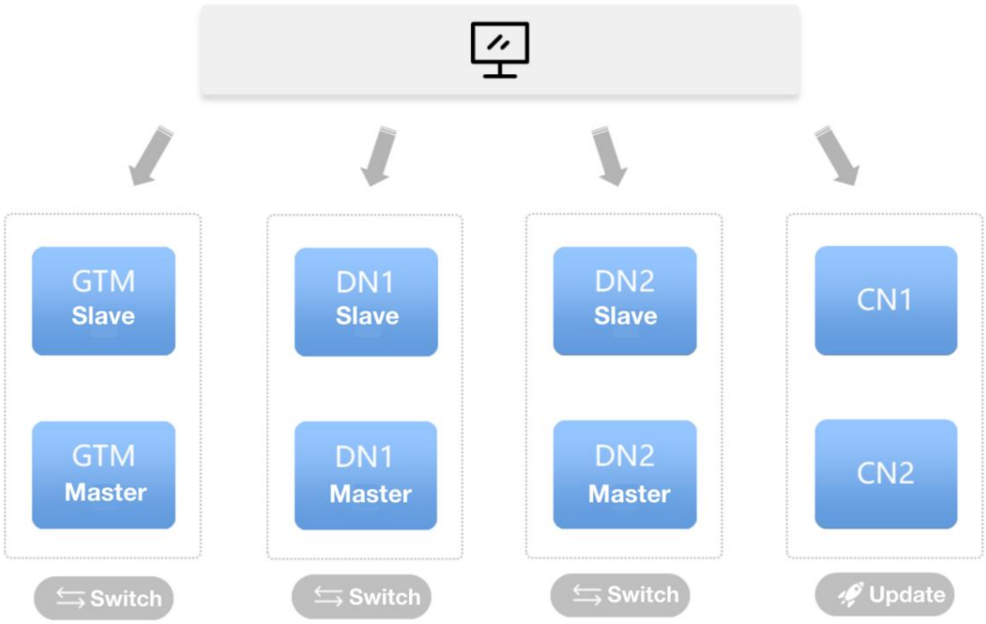
5.1 Multi-Tenancy

GBase 8c supports resource isolation based on multi-tenancy, which ensures that resources are isolated among different tenants and allows setting the resources that each tenant is allowed to use. Each tenant can choose to deploy GBase 8c in a single-node, master-slave, or distributed mode. GBase 8c's multi-tenancy provides effective isolation of CPU, memory, disk, and network resources, offering robust mixed workload management capabilities.



5.2 Gray Release

GBase 8c supports gray release, which allows for database version upgrades and releases while ensuring continuous business availability.



During the gray release process, GBase 8c can switch in seconds and has the ability to maintain system availability.

5.3 Strong Consistency Distributed Transactions

GBase 8c implements distributed transactions through two-phase commit, ensuring the atomicity, consistency, isolation, and durability of all cross-node transactions. Two-phase commit ensures that distributed transactions either all commit successfully or all fail to commit, without the possibility of transactions succeeding on some nodes while failing on others.

For distributed transactions that have been successfully committed, GBase 8c ensures that the same result can be obtained from any node, achieving strong consistency for distributed transactions.

5.4 Multiple Deployment Methods

With the continuous application of virtualization and cloud computing technologies, cloud deployment of businesses has become a trend in IT architecture development, which in turn has put forward requirements for the cloud deployment capabilities of databases.

GBase 8c supports multiple deployment methods, such as physical server deployment, virtual machine deployment, container deployment, cloud bare metal server deployment, private cloud deployment, and public cloud deployment, providing multiple choices for database deployment in different IT architectures.

1. Physical server deployment: Traditional deployment method suitable for customers who require physical isolation of database resources and have not yet cloudified their IT architecture.

2. Virtual machine/container deployment: Customers who want to adopt virtualization deployment but have not yet cloudified their IT architecture can use containers to deploy GBase 8c.

3. Bare-metal deployment on the cloud: Suitable for customers with cloudified IT architecture who also require physical isolation of database resources. GBase 8c can be deployed on bare-metal servers on the cloud.

4. Private/Public cloud deployment: Customers with cloudified IT architecture who do not require physical isolation of database resources can choose to deploy GBase 8c on private or public cloud servers.

5.5 Security Features

GBase 8c has comprehensive security features, including identity authentication and verification, transparent data encryption, separation of powers, and security auditing.

1. **Identity authentication and verification:** In GBase 8c, each database user has a unique user identifier that cannot be duplicated, and this identifier is maintained throughout the entire database lifecycle to ensure its uniqueness. Users must provide their user identifier before using the database and pass the authentication check to enter. The password of the database user is processed using encryption algorithms and stored in the GBase 8c system table. The encrypted password is irreversible, ensuring the security of the password itself.

2. **Transparent data encryption:** GBase 8c's data storage encryption uses in-library encryption, which performs data encryption and decryption processing in the database kernel storage engine and is completely transparent to users. GBase 8c supports storage encryption at different levels, such as library, table, and field, and different data objects can use different encryption keys, such as one table and one key, which can effectively prevent single-point breakthroughs and ensure higher data security.

3. **Security auditing:** GBase 8c has an independent auditing system that can define relevant auditing events, record user-related operations, record auditing data such as user identifiers and identity authentication, perform related auditing analysis and automatic alarms, and support the viewing of auditing data. To facilitate independent auditing and ensure higher system security, GBase 8c has dedicated security auditors for auditing management. Security auditors can use a dedicated auditing operation interface to select auditing events, view relevant auditing data, and handle alarm information.

4．**Separation of powers:** GBase 8c changes the database users from the original single super database administrator to three types of roles: security administrators, auditing administrators, and data administrators. They assume different responsibilities, and each of them does not involve the power range of the other two, thereby realizing the decentralized management of the entire database system, namely, the principle of separation of powers. The specific allocation of responsibilities for these three types of administrator users is roughly as follows: security administrators are mainly responsible for completing the security management functions of the system, auditing administrators are responsible for completing the auditing functions of the system, and data administrators are mainly responsible for completing access control and system maintenance management functions. These three types of administrator users have clear division of labor, mutual restraint, and cooperation, and work together to achieve the security management functions of the database.

5．**Dynamic data masking:** In order to limit unauthorized users' access to private data to some extent, dynamic data masking feature can be used to protect user privacy data. When non-authorized users access data that is configured with dynamic data masking policies, the database returns masked data to achieve the purpose of protecting private data. Without changing the source data, privacy data protection can be flexibly achieved by configuring the filtering scenarios (FILTER), specified sensitive column labels (LABEL), and corresponding masking methods (MASKING FUNCTION) in the masking policy.

5.6 Spatial Database

GBase 8c has spatial database capabilities and provides access to geospatial data related to GIS applications, enabling rapid retrieval of geographic information.

GBase 8c supports both geographic and projected coordinate systems and enables conversion between different coordinate systems. It supports objects and functions specified in the OGC "Simple Features for SQL" specification, as well as geometry objects specified in the "SQL/MM" specification. GBase 8c supports major geospatial data types, including vector, raster, grid, image, network, and topology. It also supports three basic geometric shapes: points, lines, and polygons, and provides spatial indexing through R-tree, function index, and spatial index partitioning.

6 Development Interfaces

GBase 8c interface driver can effectively achieve load balancing of upper-layer application requests.

6.1 ODBC

GBase 8c ODBC is the ODBC driver for GBase 8c, which provides access to all ODBC functions of GBase 8c. GBase 8c ODBC supports ODBC 3.5X Level 1 specification (all APIs + Level 2 features). Users can access the GBase 8c database by calling GBase 8c ODBC driver through ODBC data source administrator, or directly through GBase 8c ODBC driver. In addition, visual programming tools such as C++ Builder, Visual Studio, etc. can also use GBase 8c ODBC to access GBase 8c databases.

The ODBC driver manager is a library that manages communication between ODBC applications and drivers. Its main functions include:

1. Parsing the Data Source Name (DSN);
2. Loading and unloading drivers;
3. Handling ODBC function calls or passing them to drivers.

6.2 JDBC

JDBC (Java Database Connectivity) is a Java API used to execute SQL statements, which can provide a unified access interface for multiple relational databases, and applications can use it to operate data. GBase 8c provides support for JDBC 4.0 features and requires compiling program code with JDK 1.6 and above. JDBC bridging ODBC is not supported.

6.3 ADO.NET

GBase 8c ADO.NET is an interface program that provides a convenient, efficient, and secure interaction between .NET applications and GBase 8c databases. Developers can use any .NET programming language (C#, VB.NET, F#) to operate GBase databases through GBase 8c ADO.NET.

GBase 8c ADO.NET supports the following features:

1. Support for cluster load balancing.
2. Support for all GBase database features, such as views.
3. Support for TCP/IP socket connections on Windows platforms.
4. Support for TCP/IP socket or Linux socket connections on Linux platforms.
5. No need to install a GBase database client, full management functions can be implemented through the GBase 8c ADO.NET class library.

6.4 C API

GBase 8c C API is a C language access library provided by GBase 8c database. Applications can access the GBase 8c database by calling the GBase 8c C API. The GBase C API provides the following functions:

1. Create and disconnect the connection between the client and server.
2. Execute SQL statements directly.
3. Retrieve the result set of executing SQL.
4. Get error information.

6.5 Python API

GBase 8c Python API is an interface driver for connecting and using the GBase 8c database with the Python language. The GBase Python API is written based on the Python Database API Specification standard.

While compatible with the standard interface, it also supports the following features:

1. Support for Python 2.x and Python 3.x.
2. Full support for GBase 8c features.
3. Full support for SQL standard syntax.
4. Support for binary stream insertion and update.
5. Support for batch insertion optimization.
6. Support for executing multiple SQL statements and retrieving multiple result sets.
7. Support for TCP/IP protocol.o

General Data Technology Co., Ltd.

@GBase

@gbaseglobal

Technical support hotline: 400-013-9696