GBASE®

GBase 8a MPP Cluster
Technical White Paper V9.5.3

www.gbase.cn

GBase 8a MPP Cluster Technology White Paper, General Data Technology Co., Ltd.

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

Contents

1 GBase 8a MPP Cluster Product Overveiw

1.1 Product Introduction

General Data Technology's GBase 8a MPP Cluster is a massive distributed parallel processing database cluster system (GBase 8a MPP Cluster), which is developed based on GBase 8a column storage database and features a Shared Nothing architecture. It has high performance, high availability, high scalability and other characteristics, and can provide a cost-effective universal computing platform for data management of various scales. It is widely used to support various data warehouse systems, BI systems, and decision support systems.

1.2 Product Technical Features

The GBase 8a MPP Cluster has the following technical features: federated architecture, large-scale parallel computing, massive data compression, efficient storage structure, intelligent indexing, virtual cluster and mirroring, flexible data distribution, complete resource management, online rapid expansion, online node replacement, high concurrency, high availability, high security, easy maintenance, and efficient loading, specifically as follows:

1) Federated architecture: fully parallel MPP + Shared Nothing federated architecture. The Gcluster scheduling cluster, Data computing cluster, Gcware management cluster and other three functional clusters adopt multi-active deployment, avoiding single point performance bottleneck and single point of failure. The cluster supports a scale of over 1000 nodes. Gcluster scheduling cluster and Gcware management cluster support deployment of 64 nodes. A single virtual subset cluster supports more than 300 data computing nodes. The entire cluster supports over 100PB of

structured raw data, and a single node can support 100TB of raw data without sharing any nodes.

2) Large-scale parallel computing: using distributed execution planner, rule-based and cost-based optimizer, and scheduler based on asynchronous I/O technology, supporting high concurrency, high reliability, and large-scale parallel scheduling.

3) Massive data compression storage: capable of processing structured data of over 100PB; using advanced compression algorithms, data encoding based on column storage, and efficient compression technology to reduce the space required for storing data and correspondingly improve I/O performance. It supports three levels of compression: instance-level, table-level, and column-level, with a compression ratio of up to 1:20.

4) Efficient storage structure: the cluster uses hash, random, and replicate distribution strategies for data distributed storage; adopts column storage structure suitable for analysis optimization; supports row-column mixed storage, effectively improving the query performance under select * multi-column query scenario.

5) Intelligent Indexing: Using high-performance, maintenance-free intelligent indexing technology, the index space accounts for no more than one percent. Intelligent indexing contains statistical information about columns, which can effectively filter data during data retrieval and significantly reduce database disk I/O, greatly improving the query performance of massive data.

6) Virtual Cluster: Through virtual cluster technology, the computing nodes in a large cluster are grouped to form multiple virtual sub-clusters. Each virtual sub-cluster is physically isolated on resources, independently planned and expanded in cluster size and computing resources. Each virtual sub-cluster is logically isolated on the namespace, supporting the

use of multi-tenancy. The virtual cluster provides a unified access entry, metadata view, resource management, execution scheduling, authentication, and permission management for all sub-clusters.

7) Real-time Active-Active Cluster: Using virtual cluster mirroring technology, real-time replication of table-level data is performed between two sub-clusters, achieving real-time active-active cluster capabilities, meeting higher disaster recovery and high availability requirements, and meeting business scenarios such as real-time active-active disaster recovery and read-write load separation. It supports the cross-data-center deployment of the gcluster scheduling cluster, thereby enhancing the high availability capabilities of virtual cluster mirroring in active-active scenarios.

8) Complete Resource Management: Through virtual sub-clusters, horizontal resource isolation is achieved between different businesses, and different sub-clusters carry different businesses using different node resources. By flexibly configuring resource pools and resource usage plans, vertical resource isolation is achieved among different database users, supporting key resources and indicators such as CPU, memory, disk space, disk I/O, and concurrent task numbers. It can provide perfect multi-tenancy capabilities.

9) Online scaling: supports online expansion and contraction of clusters, with the ability to expand or contract any number of nodes at once; supports flexible scaling at multiple levels such as instance, database, and table levels, and allows data to be appended and queried during the expansion process; the expansion process can be monitored, and supports pause, resume, and cancel operations, as well as the ability to adjust the order and priority of expansion tables; online scaling performance is greater than 20TB/hour.

10) High concurrency: supports read-write concurrency, supports loading and

querying data at the same time, and improves concurrency performance with the expansion of scheduling nodes, with a cluster concurrency capacity of more than 1000.

11) High availability: supports high availability at multiple levels such as process, node, and cluster levels; the core process is monitored in real time, and is promptly recovered by the guardian process in the event of a failure; management nodes, scheduling nodes, and computing nodes adopt multi-active deployment, and the cluster can provide normal services when any node abnormally exits, ensuring cluster data consistency; in the event of power failure, switch failure, and other problems, data integrity and consistency can be ensured; supports active-active clusters, using virtual cluster mirroring technology to achieve real-time synchronization of data between two virtual subsets of the same city, with RPO=0 and RTO=0; uses inter-cluster data synchronization to achieve data consistency between remote clusters, achieving remote multi-activity with RPO=0 and RTO<60 seconds. Data is redundantly protected by replicas, supports 1 to 3 data replicas, supports user-specified and adjustable replica numbers, specifies replica distribution methods, and supports automatic detection and recovery of failures.

12) Security: supports user, role, and permission control testing, supports user password strength, whitelist, and user lockout; supports detailed audit logs, and can be flexibly configured to record logs of relevant operations with audit policies; supports transparent encryption of data storage, supports data encryption compression, supports communication encryption, and uses high-level encryption algorithms such as SHA256 and SM4; supports flexible data desensitization within the library, controlling access to original data through permission control; supports encryption functions such as AES_ENCRYPT(), ENCRYPT(), MD5(), SHA1(), and SHA(); supports Kerberos authentication.

13) Efficient Data Loading: Based on a strategy-based data loading mode, it uses replica chain forwarding, P2P multicast transmission, parallel loading of multiple loading machines, etc., with a cluster-wide loading speed of over 30TB/hour. It supports multiple data sources and network protocols, such as HTTP/HTTPS, FTP/SFTP, HDFS, Kafka, and directly loading compressed data formats such as gzip, snappy, and lzo from HDFS.

14) Data Backup and Recovery: Supports instance-level, database-level, table-level full and incremental backup and recovery; supports backing up data to and restoring data from Hadoop; single-node backup and recovery performance is greater than 800GB/hour, and the backup and recovery performance of the entire cluster exceeds 100TB/hour.

15) Data Mining: Built-in data mining algorithms enable deep mining analysis and machine learning based on in-database data and using MPP parallel computing resources, without the need for a separate machine learning platform or data migration. Data mining and machine learning can be carried out using SQL, improving development efficiency. UDFs can be used for flexible and rapid expansion of AI algorithms.

16) Multi-Instance Deployment: Supports deploying multiple computing node instances on one physical machine, fully utilizing the hardware performance of high-end servers and NUMA architecture servers.

17) Standardization: Supports SQL 2003 ANSI/ISO standards and interface specifications such as ODBC, JDBC, ADO.NET; supports interfaces such as C API, Python API, and TCL API; supports OLAP functions.

1.3 Product Basic Feature Introduction

Function	Description
Structured Query Language (SQL)	Compliant with SQL 2003 standard, it supports DDL syntax such as CREATE, ALTER, DROP, and DML syntax such as INSERT, UPDATE, DELETE, MERGE, and supports single table and multi-table join queries.
Data Type	Boolean data type. Numerical data type: INT, TINYINT, SMALLINT, MEDIUMINT, BIGINT, DECIMAL, FLOAT, DOUBLE. Character data type: CHAR, VARCHAR Date type: DATE, TIME, DATETIME, TIMESTAMP Large Object data type: BLOB, LONGBLOB, TEXT
Database Object	Provides operations for creating, modifying, and deleting commonly used database objects such as databases, tables, indexes, views, stored procedures, user-defined functions, and synonyms. It also supports creating and deleting database users/roles and assigning/restricting user permissions.
Functions and Operators	Supports a variety of standard functions, including control flow functions, string functions, numeric functions, date and time functions, conversion functions, bitwise functions, encryption functions, information functions, auxiliary functions, aggregate functions, and OLAP functions (including avg() over(), sum() over(), rank() over(), row_number() over(), cube, rollup, grouping sets etc.), Regular expression functions, etc. Supports user-defined functions extension in C and Python languages.
Row-Column Hybrid Storage	Based on the created physical table, row storage and column storage can be created, modified, and deleted.
Graphical User Interface (GUI) Tools	Provides enterprise management tools and cluster monitoring tools.
Interface	Translation: Compliant with and supporting interface specifications such as ODBC, JDBC, ADO.NET; supporting interfaces such as C API, Python API, TCL API.

2 GBase 8a MPP Cluster Product Architecture

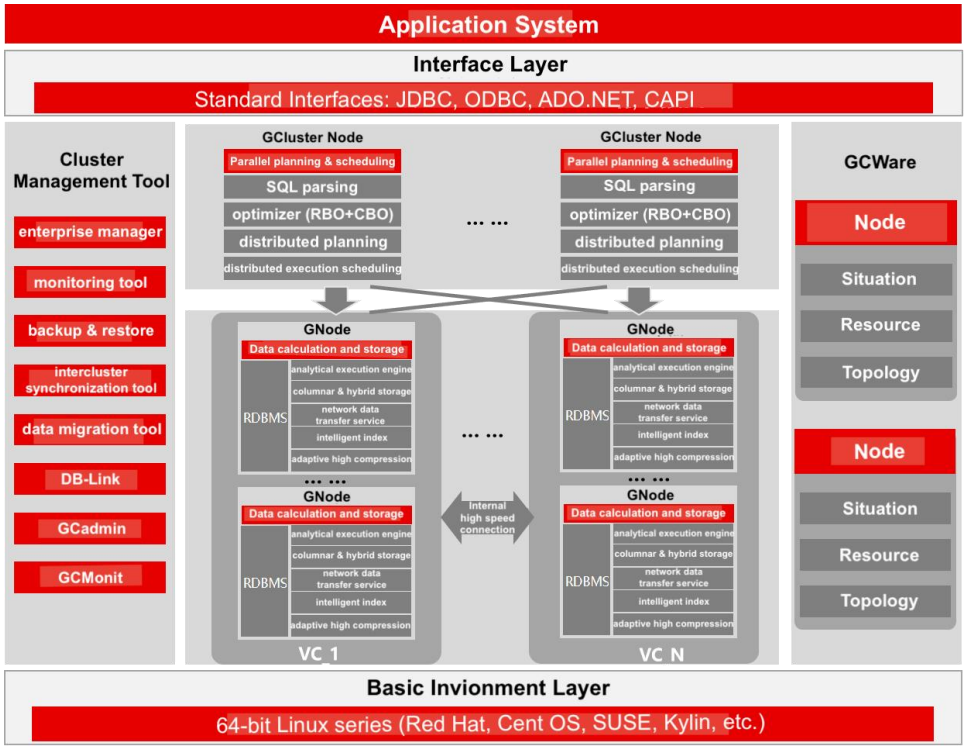

Figure 2-1. GBase 8a MPP Clusterproduct architecture.

GBase 8a MPP Cluster adopts a distributed federated architecture of MPP + Shared Nothing, and the nodes communicate with each other through TCP/IP network and use local disks to store data. It realizes asymmetric deployment, and supports independent deployment and mixed deployment of distributed management cluster, distributed scheduling cluster, and distributed computing cluster. Each node in the system is relatively independent and self-sufficient, and there is no single point bottleneck in the entire system, which has very strong scalability.

Without resource sharing, increasing nodes can linearly expand data capacity and computing power, and can be expanded from a few nodes to thousands of nodes, meeting the requirements of business scale growth.

GBase 8a MPP Cluster consists of three core components: the distributed management cluster GCWare, the distributed scheduling cluster GCluster, and the distributed storage and computing cluster GNode. Their functions are as follows:

GCluster: responsible for SQL parsing, optimization, distributed execution plan generation, and execution scheduling.

GCWare: used to share information (including cluster structure, node status, node resource status, etc.) among GCluster instances on different nodes, and to control the consistency status of data on various nodes during multi-replica data operations.

GNode: the basic storage and computing unit in GBase 8a MPP Cluster. GNode is responsible for the actual storage of cluster data on nodes, and receives and executes SQL execution plans decomposed by GCluster, returning the execution results to GCluster. During data loading, GNode directly receives data from the cluster loading service and writes it to local storage space. When deploying with virtual clusters, different nodes can be physically isolated according to different business characteristics to form different VCs.

VC (Virtual Cluster): a virtual subset cluster composed of a group of computing nodes, which achieves independent management of a single business, including database, tables, and other database objects.

Free Nodes: GNode nodes that do not belong to any VC, usually used as standby machines, to be expanded, or to replace nodes.

In addition to the above core components, there is also the GCMonit component, which is used to monitor the running status of GBase 8a MPP Cluster service programs in real time. Once a change in the process status of a service program is detected, it will execute the corresponding service start-stop script command according to the content in the configuration file to ensure the healthy operation of the service program.

3 GBase 8a MPP Cluster Core Technologies

3.1 Single-Node Columnar Storage Database Technology

3.1.1 Columnar Storage and Hybrid Storage

Data is organized and physically stored on disk in a column-oriented manner. Row-based and column-based database architectures are suitable for different applications, each with its own advantages and disadvantages, as shown in the figure below. Column-oriented architecture has inherent advantages in querying, statistics, and analysis operations.

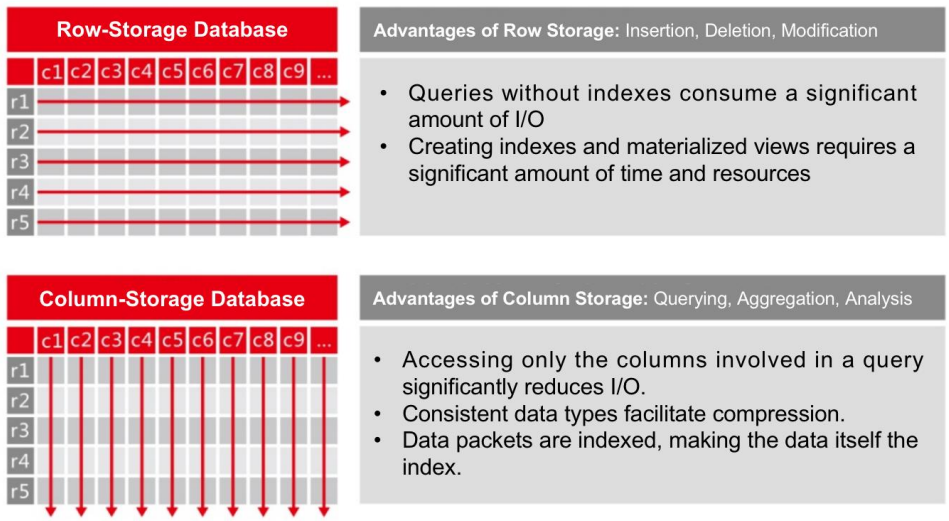

Figure 3-1. omparison between Row Storage and Column Storage.

For example, when analyzing and querying a business table with 100 columns in an analysis system, the reference to the columns typically does not exceed 10. For analysis systems where I/O is the main bottleneck, columnar storage technology can easily bring about an order of magnitude performance improvement compared to traditional row-based databases.

In the face of the I/O bottleneck in analyzing massive amounts of data, analytical databases store table data in a columnar manner, which has advantages in the following areas:

1) Reducing I/O: Data is organized and stored in a columnar fashion, only the columns involved in a query are accessed, which reduces I/O operations. Columns not involved in the query are not accessed and therefore do not generate I/O.

2) High compression ratio: Columnar storage provides high compression ratios ranging from 1:2 to 1:20 and above.

3) Supporting hybrid row-column storage reduces I/O operations for the SELECT * scenarios, improving the system performance.

3.1.2 Efficient Transparent Compression

Efficient and transparent compression technology can automatically select the optimal compression algorithm based on data type and distribution, minimizing the storage space required, reducing I/O consumption, and improving query performance. It supports compression options at the library, table, and column levels, balancing performance and compression ratio flexibly, and the compression and decompression process is transparent to users.

From the perspective of saving I/O resources, for analysis systems where I/O is the main bottleneck, efficient and transparent compression technology can bring about a performance improvement of about an order of magnitude compared to traditional row-based databases.

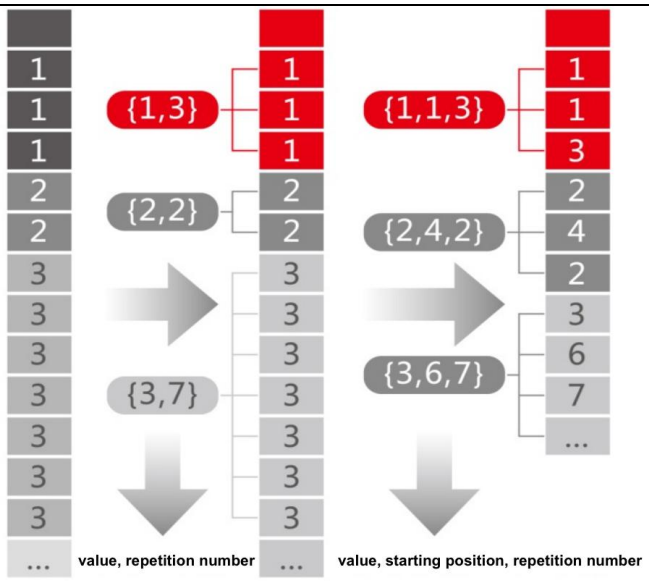

Figure 3-2. Adaptive Transparent Compression Diagram.

1) Compression ratio can reach 1:2 to 1:20 or even better, much higher than row storage;

2) Save 50% to 95% of storage space, greatly reducing data processing energy consumption;

3) Built-in dozens of different levels of compression algorithms, automatically select the best compression algorithm according to data characteristics;

4) The I/O requirements are greatly reduced in the compressed state, and data loading and query performance are significantly improved.

3.1.3 Intelligent Indexing

Intelligent indexing is a type of coarse-grained index that includes data statistics. When data is stored, every 65536 rows of data are packed into a Data Cell (DC) package, and an intelligent index is automatically created for each package when the data is imported. During data querying, statistics values can be obtained and data

filtering can be performed without unpacking the package, which can further reduce I/O and significantly optimize complex queries. As shown in the figure below, the "SmartIndex of Col 1" on the left represents the intelligent index built on the first column data, with each data package (DC) as the unit.

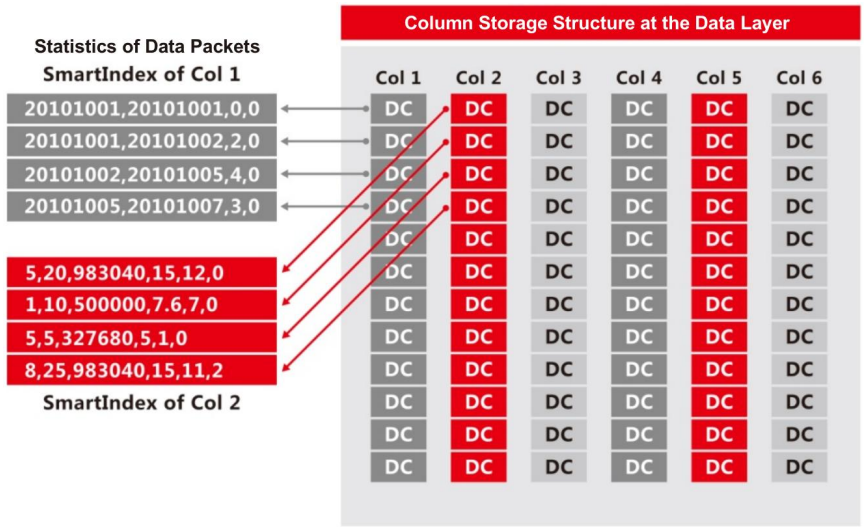

Figure 3-3. Diagram of an intelligent index.

1）indexes without requiring users to manually create and maintain them.

2）Intelligent indexes themselves occupy a small space (about 1%), and do not cause data inflation.

3）Intelligent indexes are established based on data packets, with fast creation speed and unaffected by existing data packets.

Compared with traditional database indexing techniques, intelligent indexes are established on data packets (coarse-grained indexing), and each field is automatically indexed, while traditional indexes are established on each row of data (fine-grained indexing). Therefore, accessing intelligent indexes requires less I/O than accessing traditional indexes. At the same time, the space occupied by intelligent indexes is about one percent of the data, while traditional database indexes can occupy 20-50% of the data.

3.1.4 Parallel Technology

Efficient parallel processing technology has been implemented for data loading, data processing, and data querying, which fully utilizes the parallel processing power of SMP multi-core CPUs to handle massive data. Parallel computing is supported during data scanning and filtering, JOIN calculation, group by aggregation, order by sorting, and query result materialization.

3.1.5 High Performance

The single-node data computing engine automatically selects the optimal execution plan and algorithm through rule-based and cost-based cost evaluation. Leveraging columnar storage, efficient compression, and intelligent indexing technologies, it saves three orders of magnitude in CPU and I/O resource consumption, and achieves query performance 50 to 1000 times higher than traditional row-based databases.

It supports deploying multiple computing nodes on multi-NUMA node servers, binding each computing node to a NUMA node to avoid performance degradation caused by cross-NUMA node operations. Multi-instance deployment can improve performance by more than 1x.

3.2 Massive Parallel Processing (MPP) Technology

GBase 8a MPP Cluster is an asymmetrically deployed federated architecture, where the three core components can be deployed separately. The maximum number of nodes supported for GCluster scheduling cluster and GCWare management is 64, and the GNode computing cluster supports deployment of more than 1000 nodes, capable of handling structured data of over 100 PB.

3.2.1 Massive Parallel Processing

GBase 8a MPP Cluster adopts MPP technology, and its main features are:

1） Distributed parallel planner that combines cluster characteristics, processes operator progress in a distributed manner, and generates an appropriate distributed execution plan.

2） Optimization based on rules and cost to ensure the efficiency of execution plan.

3） The scheduler uses technologies such as asynchronous I/O to ensure efficient and reliable scheduling.

3.2.2 High Availability

GBase 8a MPP Cluster ensures high availability of the cluster through redundancy mechanisms:

1) It provides 1 or 3 copies of data redundancy (3 copies represent 1 main shard and 2 backup shards).

2) Data synchronization between replicas is automatic.

3) The replication engine automatically manages data synchronization.

4) The multi-shard mechanism reduces the bucket effect of node failures.

5) When a node fails, the system automatically switches to another node to ensure business continuity.

It also supports active-active cluster deployment.

Flexible configuration of cluster replica and shard numbers is supported in GBase 8a MPP Cluster, and replicas and shards can be configured on any node in the cluster. More primary and backup replicas can be allocated to nodes with higher performance and larger storage space.

When a node is in an abnormal state, the load of the abnormal server can be evenly distributed among several normal servers where the replicas are located. This can effectively prevent significant performance fluctuations caused by the bucket effect after the switch due to a failure.

Node failures are transparent to applications and do not interrupt business operations. Once the failed node returns to normal, GBase 8a MPP Cluster will recover data from other nodes and provide services immediately after completing the update.

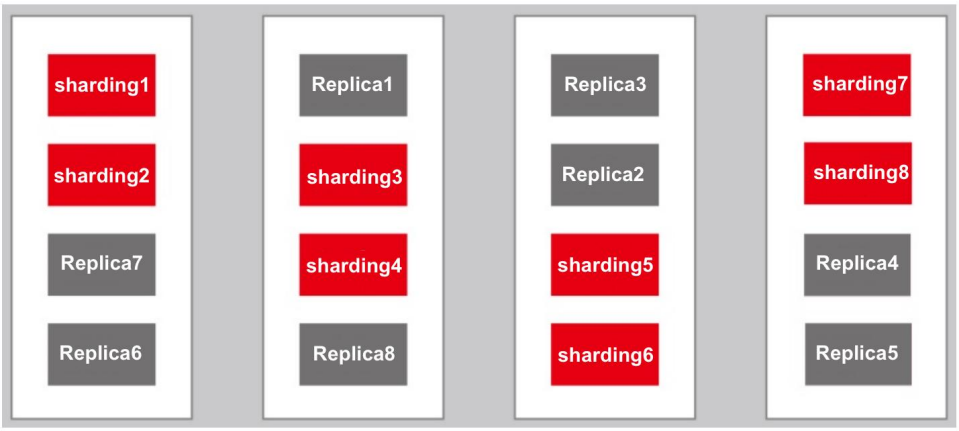

Figure 3-4. Flexible configuration of data sharding and replica sharding.

3.2.3 High-Performance Scalability

GBase 8a MPP Cluster can expand its computing and storage capacity by adding server nodes, supporting clusters with hundreds or thousands of nodes, and the expansion process can be flexibly monitored and managed with features such as monitoring, pausing, resuming, and canceling. The system supports online expansion, with linear performance improvement, without interrupting the current system operation, and supports adding multiple nodes at once. It also supports flexible expansion at different levels such as instance-level, database-level, and table-level.

GBase 8a MPP Cluster has high-performance scaling capabilities:

1) Online dynamic expansion of cluster nodes;

2) Each node can process 100TB of effective data while providing both computing and storage capabilities;

3) Execution scheduling nodes and data computing nodes can be independently scaled as needed.

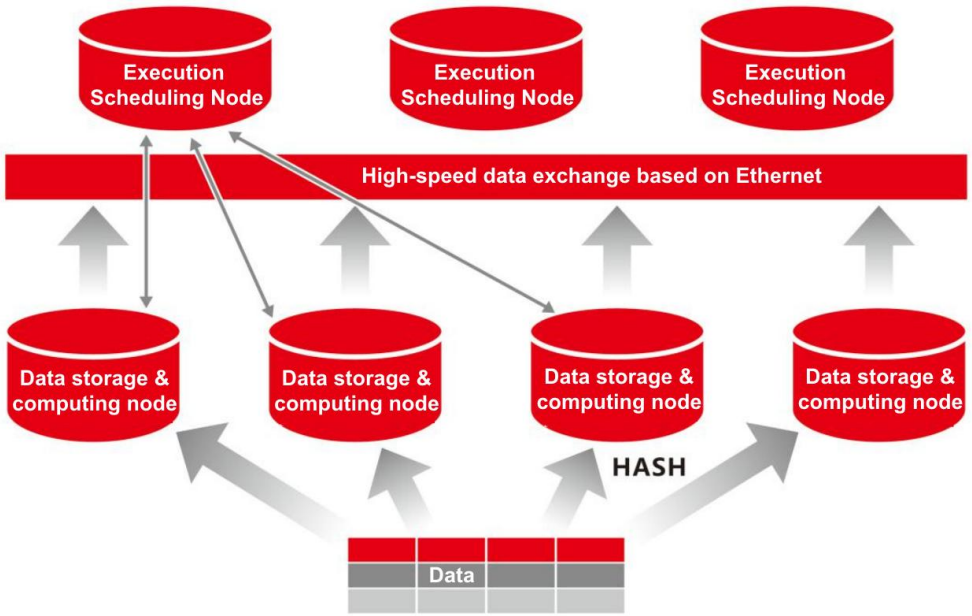

Figure 3-5. GBase 8a MPP ClusterExpansion Technology Diagram.

The last "Execution and Scheduling Node" and "Data Storage and Computation Node" in the above figure are newly expanded nodes based on the original nodes. The scheduling nodes or storage and computation nodes can be expanded. Because GBase 8a MPP Cluster adopts a high-performance single-node MPP architecture, it can ensure smooth expansion and linear performance growth during cluster expansion, as shown in the following figure:

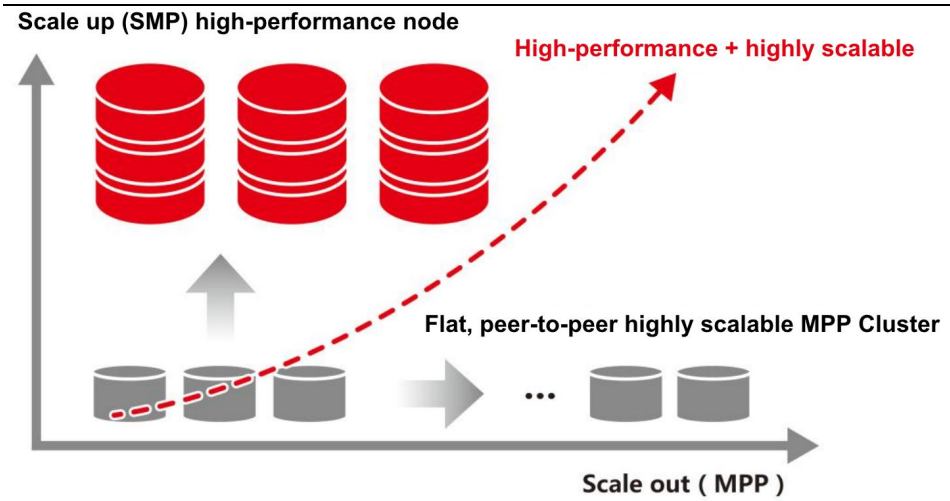

Figure 3-6. GBase 8a MPP Cluster high-performance and high-scalability diagram.

3.2.4 High-Performance Data Loading Capability

GBase 8a MPP Cluster's data loading function achieves high loading performance through parallel processing of data by all data computing nodes and replica forwarding technology:

1）In the case of a single data server, the loading of the cluster can reach the performance limit of the hardware resources of the data server;

2）With the increase of the number of data servers and cluster nodes, the loading performance of the cluster can continue to improve.

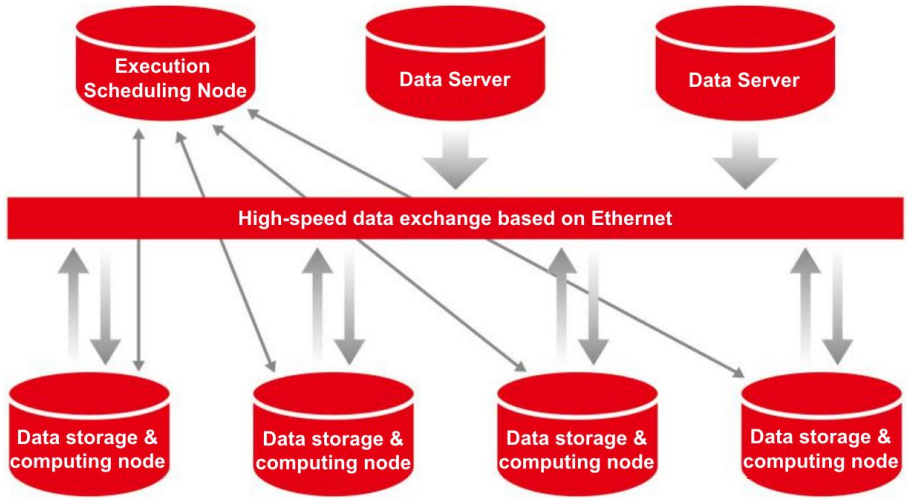

Figure 3-7. GBase8a MPP Cluster data loading diagram.

3.2.5 Multi-Tenancy Capability

GBase 8a MPP Cluster provides multi-tenant capabilities through virtual cluster technology to achieve physical or logical isolation between tenants.

The computing nodes are grouped into multiple virtual subsets within a virtual cluster, with each subset physically isolated on resources and logically separated in the namespace. The virtual cluster provides a unified access entry, metadata view, execution scheduling, authentication, and permission management for all subsets. On each computing node of a virtual subset, different application and user resource quotas and query priorities can be flexibly configured through resource management and resource groups.

With a sound permission authorization and management mechanism, different applications can run in different virtual subsets, and different types of tasks of the same application can run in the same subset, but with different resource quotas and priorities, and necessary isolation is ensured. It's like running in a transparent "sandbox."

3.2.6 Backup and Recovery

Big data not only brings challenges to the performance of data processing, analysis, and querying, but also demands higher requirements for backup and recovery. Due to the massive data volume, if the backup and recovery speed cannot keep up, the database may not be able to be restored in time in case of accidents, faults, or disasters, which will lead to the unavailability of the system and business.

GBase 8a MPP Cluster provides comprehensive backup and recovery functions based on instance-level, database-level, and table-level, including:

1) Full backup and recovery;
2) Incremental backup and recovery: allowing data recovery based on any backup point;
3) Supporting backup to Hadoop and restoring data from Hadoop.

3.2.7 Multi-Instance Deployment

Deploy multiple data computing nodes on a physical server, and each computing node is called a database instance. The diagram below shows the schematic of multi-instance deployment:

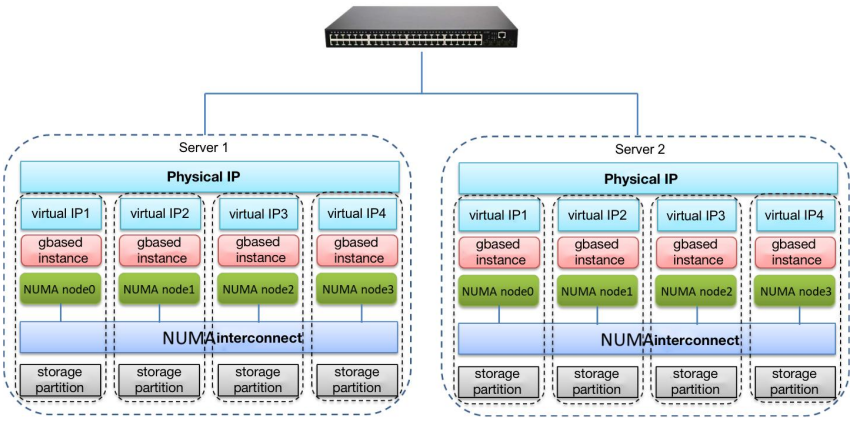

Figure 3-8. GBase 8a MPP Cluster multi-instance deployment diagram.

By installing multiple computing nodes on a physical machine, each computing node corresponds to a NUMA node of the server. Resource isolation and full utilization of CPU, memory, and other resources are achieved through NUMA node binding to avoid performance degradation caused by cross-NUMA node operation. Multiple instance deployment can effectively utilize the performance of high-configuration servers and NUMA architecture servers. Compared with deploying only one computing node on a physical server, it can improve computing performance by more than 1 times.

3.2.8 Service Registration Mechanism

As the cluster size continues to grow, the cost of cluster management will become higher and higher. GBase 8a MPP Cluster supports the service status registration mechanism of data computing nodes, and data computing nodes actively register local service status to the gcware management cluster regularly. This mechanism can improve the accuracy and timeliness of monitoring the state of data computing nodes, and reduce the impact of a small number of abnormal nodes on the cluster.

The information registered by the Data Node to gcware includes:

- The VC to which the data node belongs
- A long connection session ID and node ID used for monitoring the data node's status. After registering the session ID, gcware tracks the data node's heartbeat, and sets its status to abnormal if the heartbeat is interrupted.
- Registration of globally consistent important parameter values, such as gbase_compression_str_method, gbase_compression_num_method, and gbase_segment_size. If the important parameters within the same VC are inconsistent, the data computing node's service cannot be started.

4 GBase 8a MPP Cluster Advanced Features

4.1 Data Loading and Integration

4.1.1 Data Loading

In GBase 8a MPP Cluster, the cluster loading function is directly integrated within the GBase 8a MPP Cluster and does not require additional external loading tools.

The data loading has the following characteristics and advantages:

1）Highly integrated with the cluster;

2）The SQL interface is more in line with users' usage habits;

3）Supports parallel loading of a single table from multiple data sources and supports multiple loading machines for parallel loading of a single table, maximizing loading performance;

4）Supports pulling data from general data servers and supports multiple file transfer protocols such as ftp/sftp/hdfs/Kafka/http/https;

5）Supports multiple data file formats, such as plain text, gzip compression, snappy compression, and lzo compression;

6）Supports loading of plain text, fixed-length text, and loose mode;

7）Supports user-defined column separators, row separators, and date/time formats;

8）Supports error data tracing, which can accurately locate the position of erroneous data in the source file;

9）The loading performance can continue to improve with the expansion of the cluster scale.

4.1.2 kafka Data Integration

The data synchronization system replicates business data from databases such as Oracle and GBase 8s using tools such as Oracle Golden Gate (OGG) and GBase RTSync, and synchronizes it to GBase 8a MPP Cluster through Kafka. To cope with possible spikes in business systems, a Kafka message queue is added to the system as a buffer. The overall process is as follows:

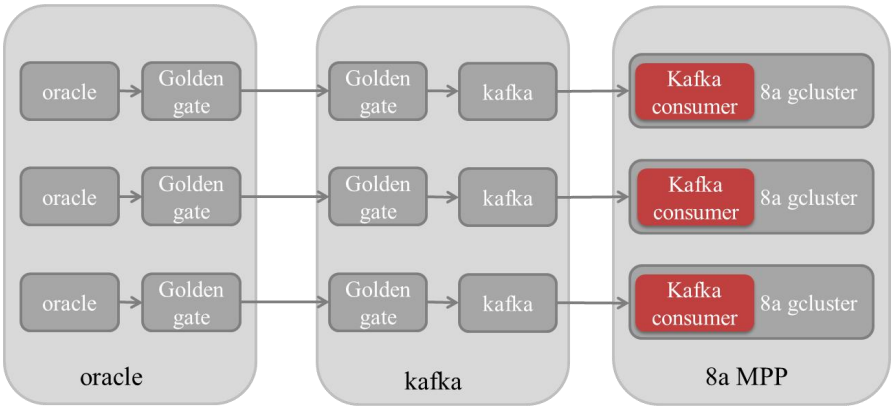

Figure 4-1. Process flow.

The OGG sender (GoldenGate Extract) extracts transaction information from Oracle's online logs and archive logs and generates Trail files. The OGG receiver (GoldenGate Replicat) receives the Trail files, extracts the transaction information, converts it to the target format, and produces transaction messages to Kafka. The consumer consumes transaction messages from Kafka and updates the data to 8a MPP Cluster.

The main function of the Kafka consumer is to synchronize Kafka data to 8a MPP Cluster:

1） Based on the configuration, the business to be synchronized can be specified;

2） During the synchronization process, the function of querying

synchronization status is provided;

3） Implement high availability and transaction data consistency for data synchronization.

4.2 Virtual Clusters and Mirror Clusters

4.2.1 Virtual Clusters

A virtual cluster is developed based on a large-scale cluster, which includes multiple virtual sub-clusters (VCs). Each VC runs independently within the entire cluster and shares a unified entry. With permission, VCs can access each other.

The Coordinator node (including Gcluster node and Gcware node) and VC's data node support running on the same physical node.

Unified Management

By dividing the cluster into virtual sub-clusters, the cluster can reach a scale of thousands of nodes while providing a unified management view, enhancing the cluster's scalability.

Unified access entry to the cluster

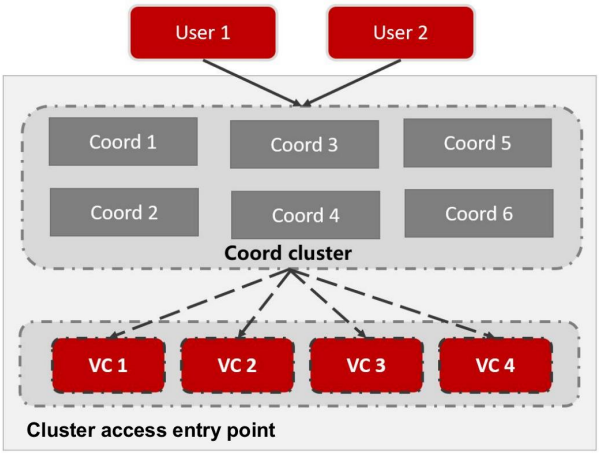

Figure 4-2. Unified access entry.

Users select any Coordinator node in the Coordinator cluster (which includes the Gcluster scheduling cluster and the Gcware management cluster, usually deployed together and referred to as the Coordinator cluster) as the cluster entry point. The Coordinator node determines the default VC based on user input.

Note: The Coordinator node stores metadata for all VC databases and tables.

Business Isolation

Virtual clusters isolate resources vertically in the cluster, enabling physical resource isolation for different businesses in real-world business scenarios.

Transparent Access

Virtual clusters provide a unified access point, and accessing multiple physical clusters is transparent to the application, which appears to be a unified cluster.

4.2.2 Mirror Cluster

The mirror function of the virtual cluster has the following characteristics:

Flexible mirror configuration: supports mirror setting at the database or table level;

Real-time data synchronization: when data changes occur in databases or tables with mirror relationships, the changes are synchronized in real-time on both sides. All tables under mutually mirrored databases will automatically create mirrors. All functions, stored procedures, and views under the database will also be synchronized and created in the target database;

High availability: mirror relationships provide more redundant shards for data, resulting in higher availability; disaster recovery: supports cluster deployment in the same city or different locations.

4.3 Data Security

4.3.1 Data Encryption

GBase 8a MPP Cluster data encryption provides encryption function for database landing data, to meet user security requirements and improve system security. Data encryption is performed at the minimum unit of DC data block in the data file, which can realize encryption requirements at different granularities, such as table-level or column-level.

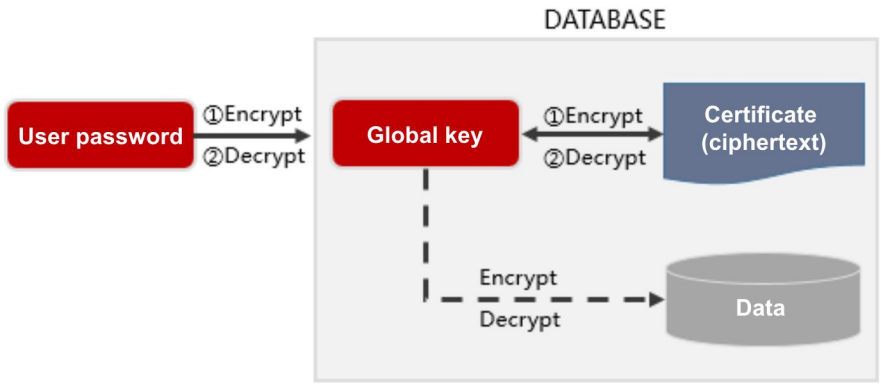

Figure 4-3. Data encrytion process.

The data encryption supports the following features:

- Supports encryption keyword "encrypt" for table creation
- Supports encryption at the table level or column level with different granularities
- Supports querying the encryption attributes of a table
- Supports key certificate management including key certificate creation, opening, closing, password modification, and key conversion operations
- Supports key type conversion, that is, conversion from plaintext keys to

ciphertext keys or from ciphertext keys to plaintext keys

Plaintext keys: do not require a user password and can be generated randomly or entered manually

Ciphertext keys: require a user password and are encrypted and stored based on the password

- Supports querying the current status of the key certificate
- Supports row-level and column-level encryption

4.3.2 Data Masking

GBase 8a MPP Cluster provides a new feature of dynamic data masking, which allows developers or database administrators to control the exposure level of sensitive data and generate data at the database level, greatly simplifying the security design and coding of the database application layer.

According to Permissions and Field Attributes

Users can add masking properties to the fields that need to be masked through SQL syntax, and determine whether to expose the original data to users who have query requirements through user permission control.

Built-in Rules

Dynamic data masking does not actually modify the actual data stored in the table, but applies this feature to control the data returned in the query. Dynamic data masking supports four data masking functions, including default, random, partial, SHA, and keymask. Whether dynamic data masking is enabled depends on the current user's permissions. Users with unmask permissions can access the actual data without being affected by the masking rules, while users without unmask permissions can only access the masked data affected by the masking rules. Masking only applies to projected columns.

4.4 Full-Text Search

GBase 8a MPP Cluster database supports full-text search, which uses a single-word index by default and supports almost all languages, ensuring a 100% query recall rate. Combined with GBase 8a MPP Cluster's unique column storage, compression, and intelligent indexing technologies, it is suitable for retrieval and query applications aimed at massive data.

It supports establishing full-text indexes for external storage documents and provides retrieval. The supported document formats include txt, doc, XML, HTML, PDF, xls, and others.

The main functions include:

1. Index Creation and Search

- GBase 8a MPP Cluster has an embedded full-text search engine that supports the indexing and querying of all text type fields in a table.
- It supports parameterized management, and the processes of index creation, segmentation, index maintenance, and search can be conveniently configured through GBase 8a MPP Cluster's standard configuration files.
- GBase 8a MPP Cluster has an embedded text tokenizer function to achieve single-word segmentation of text columns and search strings, ensuring consistency between their segmentation rules and results and preventing inconsistent segmentation caused by context and language usage.
- It supports synchronous full-text index queries, enabling querying during the index creation process. New data can be indexed in batches, and when the indexed data in the buffer is processed and written to the index file,

users can immediately search for this new content that has been indexed, instead of waiting for all new data to be indexed before querying.

- It supports logical expression queries (AND, OR, NOT), NEAR queries, and logical combination queries between full-text indexed columns and non-full-text indexed fields in a database table.

2. Support DML

- It supports online deletion of character data type columns that have established full-text indexes in a database table.
- It supports synchronously updating full-text indexes after a column data is updated.

3. Support DDL

- It supports automatic invalidation of indexes after a database table's full-text indexed column is deleted.
- It supports index persistence after a database table is renamed.

4.5 Library for In-Database Mining

GBMLLib is an extension library for data mining and machine learning of GBase 8a MPP Cluster. It is added to GBase 8a MPP Cluster as a plugin. Through the machine learning algorithms provided by GBMLLib, GBase 8a MPP Cluster can conduct in-depth analysis and mining of user data and transform user data into user value.

GBMLLib provides SQL-based machine learning algorithms, currently including regression algorithms (linear regression), classification algorithms (logistic regression, support vector machine), and clustering algorithms (K-means). It also provides some basic functions for array operations and linear algebra calculations.

GBMLLib has the following technical features:

- **SQL interface:** GBMLLib provides SQL-based data mining algorithms, and the training, evaluation, and prediction of models are all executed through SQL statements, making it very easy for data analysts to master and combine with their existing skills to fully unleash their creativity and improve work efficiency.
- **In-database analysis:** Unlike other analysis tools that require data to be moved from the database to the analysis node for processing through APIs or ODBC, GBMLLib's analysis algorithms run in the form of database UDF/UDAF inside the threads of GBase 8a MPP Cluster, and are scheduled through GBase 8a MPP Cluster's execution plan to minimize data movement and improve execution speed.
- **Easy to extend:** GBMLLib is added to GBase 8a MPP Cluster as a plugin and adopts an elastic and flexible software architecture, making it easy to add new data mining and machine learning algorithms in the future.

5 GBase 8a MPP Cluster Operating Environment and Technical Indicators

5.1 Hardware Environment

The following hardware environments are supported:

1) Supported server platforms: x86_64 standard PC servers, PowerLinux servers, Inspur K1, Huawei TaiShan, Sugon, Hygon, etc.;

2) Supported storage: Local storage (SATA, SAS, SSD, etc.), array storage (SAN, NAS), SSD, flash cards;

3) Supported network environments: Gigabit, 10 Gigabit, InfiniBand.

5.2 Operating System and Platform

The following operating systems and platforms are supported:

1) Supported CPUs: Intel, AMD, Shenwei, Loongson 3B, Feiteng, X86, Power, Kunpeng, Haiguang, etc.;

2) Supported operating systems: 64-bit operating systems such as CentOS, Red Hat, SUSE, NeoKylin, PowerLinux, Deepin, KylinOS, LINX, NFS-China, i-soft, etc.;

3) Support for virtualization technologies based on x86 and Power, such as VMware ESX and KVM, OpenStack, Docker, etc.

5.3 Interfaces with Third-Party Software Platforms

Compatible with mainstream application software, including:

1) BI tools: MSTR, Cognos, R3 Query, SAS, SPSS, etc.;

2) ETL tools: Datastage, Kettle, Informatica, etc.;

3) Middleware: WebSphere, Tuxedo, Hibernate, Weblogic, Tomcat, etc.

5.4 Technical Specifications

Technical Secification	Description
Single node data processing capacity	100TB
Numeric precision	65
Number of tables	Limited by the file system Ext3 file system has an upper limit of 65535 databases per single instance, while other file systems have no such limit.
Number of columns in each table	2000
Number of rows in each table	140737488355328
Internal length of a row in a table	65534000 bytes
Length of an INTEGER type column	4 bytes
Number of bits representing the year in a date type column	4 bits
Number of characters in a username	16 characters
Length of a CHAR type column	255 characters
Length of a BLOB column	32K bytes
Length of a VARCHAR type column	Varies with the character set For UTF8MB4 and GB18030, is 8192, for GBK and UTF-8, is 10922 characters.
Length of a rowstore column	32KB
Length of a virtual cluster name	255
Number of virtual clusters	64
Length of a database name	48 characters
Length of a table name	56 characters
Length of a column name	64 characters
Length of an index name	64 characters
Length of an alias	255 characters
Encoding format	UTF-8, UTF8MB4, GBK , GB18030

6 GBase 8a MPP Cluster Management Tools

GBase 8a MPP Cluster provides tools such as GBaseDataStudio, cluster monitoring tools, gcadmin command-line tools, DB-Link, and transparent gateways to centrally and uniformly manage the various functions of the database, monitor the system's operating status, resource usage, and task execution, and achieve multiple functions such as fast data loading and efficient data access between clusters.

6.1 Client Tools

6.1.1 GBaseDataStudio

Graphical query and management tool that provides access, control, and management of the cluster environment. With the GBase 8a MPP Cluster Enterprise Manager, the following tasks can be completed:

1) Cluster connection configuration management;
2) Metadata information query and management;
3) Development and execution of SQL, stored procedures, and custom functions.

6.1.2 Cluster Monitoring Tool

The cluster monitoring tool provides reliable monitoring data, timely alarm functions, intuitive trend displays, reliable data distribution views, and displays of the status of database connection threads.

The cluster monitoring tool mainly monitors the running status of the cluster node servers, resource utilization, network communication status, and other information in the single or multiple GBase 8a MPP Cluster deployment environment. It allows

users to effectively monitor the running status and system resources of servers in the cluster environment and can provide reliable basis for monitoring the running status of the cluster and its nodes for users.

6.2 gcadmin Tool

GBase 8a MPP Cluster provides administrators with the gcadmin command-line tool. This tool enables management functions such as adding or removing Distributions and viewing cluster status. Specific features include:

1) Querying cluster status information, such as node IP, node status, and cluster service mode;
2) Querying data distribution information, such as the location of primary and secondary shards, and the total number of shards;
3) Querying cluster lock information, such as lock name, node IP, and lock creation time;
4) Querying node exception status information, such as all error logs and error logs for a table shard on a specific node.

6.3 DB-Link and Transparent Gateway

DB-Link is a remote database connection feature provided by GBase 8a MPP Cluster. Working in collaboration with the Transparent Gateway service, it enables transparent access to data tables in remote GBase 8a MPP Cluster or Oracle databases. Passthrough mode is supported, meaning GBase 8a MPP Cluster does not parse the specified SQL syntax, and the Transparent Gateway service directly pushes the specified SQL to execute the relevant operations on the remote database.

The Transparent Gateway service is an independently running process responsible for connecting GBase 8a MPP Cluster or other databases (through standard JDBC

interfaces) and completing data extraction and push tasks.

Through the DB-Link feature, customers can achieve the following:

1) GBase 8a MPP Cluster applications can access an external data source that supports GBase 8a MPP Cluster or Oracle databases. They can query data from the external data source and perform related operations on data association with local clusters.
2) If the remote data source is a GBase 8a MPP Cluster and there are no network communication restrictions, the system can perform targeted optimizations to improve performance. DB-Link access between different cluster versions is supported, such as supporting the DB-Link feature between v8.6.2build43 and v9.5.3 versions.

6.4 Rsynctool: Cluster Synchronization Tool

The Cluster Synchronization Tool is based on binary file synchronization, which synchronizes data inside the database by parsing and comparing changes in binary files. Incremental synchronization and full synchronization are included. The tool has the following features:

1) Supports both incremental and full data synchronization.

2) Supports read-back verification of disk data.

3) Supports synchronizing the primary shard first and then synchronizing the backup shard after the primary shard is successful.

4) Cluster synchronization supports using ordinary database users for synchronization.

7 GBase 8a MPP Cluster Development Interfaces

The cluster interface driver can effectively balance the load of requests from upper-level applications (the application calls the interface driver's connection string, which is configured with the IP addresses of multiple cluster management nodes, and the interface driver's internal load balancing is performed during connection). The application layer request response node completes SQL parsing and generates execution plans, coordinates the concurrent participation of cluster-related nodes in computing and processing, greatly improving the concurrency of the entire cluster nodes and fully tapping the cluster performance.

7.1 GBase 8a MPP Cluster ODBC

GBase 8a MPP Cluster ODBC is the ODBC driver for GBase 8a MPP Cluster, which provides access to all ODBC functions of GBase 8a MPP Cluster. GBase 8a MPP Cluster ODBC supports ODBC 3.5X level specification (all APIs + 2nd-level features). Users can call GBase 8a MPP Cluster ODBC driver to access GBase 8a MPP Cluster database through ODBC Data Source Administrator or directly. In addition, visual programming tools such as C++ Builder and Visual Studio can also utilize GBase 8a MPP Cluster ODBC for access. GBase 8a MPP Cluster ODBC supports all platforms supported by GBase 8a MPP Cluster, including Windows, Linux, and AIX.

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

1) Resolving Data Source Names (DSN)

2) Loading and unloading drivers

3) Handling ODBC function calls or passing them to drivers

7.2 GBase 8a MPP Cluster JDBC

GBase 8a MPP Cluster JDBC is a driver compatible with JDBC specifications 3.0 and 4.0 (type 4), which means it is a pure Java program that conforms to JDBC 3.0 and 4.0 version specifications and can directly communicate with GBase servers using the GBase protocol.

GBase 8a MPP Cluster JDBC provides access to GBase 8a MPP Cluster interfaces for client applications using the Java programming language.

1) GBase 8a MPP Cluster JDBC supports JDBC specifications 3.0 and 4.0 versions;
2) GBase 8a MPP Cluster JDBC directly communicates with GBase 8a MPP Cluster servers using the GBase protocol;
3) GBase 8a MPP Cluster JDBC has a pass rate of 95% on Sun's JDBC experimental platform.

7.3 GBase 8a MPP Cluster ADO.NET

GBase 8a MPP Cluster ADO.NET is an interface program that provides a convenient, efficient, and secure interaction between .NET applications and the GBase database. It is 100% written in C# and inherits from Microsoft ADO.NET classes. Developers can use any .NET programming language (C#, VB.NET, F#) to operate the GBase database through GBase 8a MPP Cluster ADO.NET.

GBase 8a MPP Cluster ADO.NET supports the following features:

1) Supports cluster high availability and load balancing;
2) Supports all features of the GBase database, such as stored procedures and views;
3) Supports protocol compression, allowing compression of the data stream

between the client and server;

4) Supports TCP/IP socket connection under the Windows platform;

5) Supports TCP/IP socket or Linux socket connection under the Linux platform;

6) No need to install the GBase database client, the full management function can be implemented through the GBase 8a MPP Cluster ADO.NET class library.

7.4 GBase 8a MPP Cluster C API

GBase 8a MPP Cluster C API is a C language access library provided by GBase 8a MPP Cluster database. Applications can access the GBase 8a MPP Cluster database by calling the GBase 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) Operate the database through the prepared statement mode;

4) Get the result set of the executed SQL;

5) Get error information.

7.5 GBase 8a MPP Cluster Python API

GBase Python interface is a driver program that connects and uses the GBase database in the Python language. The GBase Python interface is written based on the Python Database API Specification standard. The interface is compatible with the standard and supports the following features:

1) Supports Python 2.x and Python 3.x

2) Fully supports the features of GBase 8a MPP Cluster

3) Fully supports SQL standard syntax

4) Supports binary stream insertion and update

5) Supports batch insertion optimization

6) Supports executing multiple SQL statements and obtaining multiple result sets

7) Supports TCP/IP protocol

8) Supports mapping between Python's datetime and GBase time types.

GBASE®

General Data Technology Co., Ltd.

@GBase

@gbaseglobal

Technical support hotline: 400-013-9696