GaussDB

Product Bulletin

Issue	01
Date	2025-01-22

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Product Bulletin

1.1 GaussDB Versioning Rules Have Been Updated

Starting October 10, 2024, Huawei Cloud will enhance the versioning specifications of GaussDB engine versions by adding the major version number, V2.0, which represents the second-generation GaussDB architecture.

Change Content

The GaussDB engine version numbers will now include the major version number, V2.0, to indicate the second-generation architecture, while the minor version numbers will remain the same. For details about the versioning rules, see **GaussDB Version Policies**.

Change Impact

The new engine versions will be displayed on the GaussDB console, product documentation, and other user interfaces once the change is made.

The change only concerns the optimization of versioning rules and will not affect the product or customer workloads.

If you have any questions about the new engine versions, please **submit a service ticket** after this notice is released.

1.2 Developer Guide of GaussDB V2.0-1.x Will Be Removed

Notice Content

GaussDB V2.0-1.x instances have been brought offline. To improve user experience, Huawei Cloud plans to remove the Developer Guide of GaussDB V2.0-1.x from the help center on March 15, 2024.

Impacts

Huawei Cloud Help Center will no longer provide the *Developer Guide* for GaussDB V2.0-1.x. *Developer Guide* for GaussDB V2.0-2.x and V2.0-3.x is not affected.

1.3 GaussDB(for openGauss) Is Now Called GaussDB

Background

To strengthen the brand awareness of Huawei-developed relational databases, Huawei Cloud has made some changes to the GaussDB database brand.

Impacts

GaussDB(for openGauss) is now called GaussDB. For details about URL changes, see **URLs of GaussDB Documents Will Be Changed in Huawei Cloud Help Center**.

It takes some time to modify all the content affected by the name change. Before all modifications are complete, the billing information of GaussDB(for openGauss) will be temporarily attached to TaurusDB, but the pricing and resources will not change. The functionality of GaussDB(for openGauss) and your services will not be affected.

After the name change, the original URL of the TaurusDB console will be used by GaussDB. To visit the TaurusDB console, use the new URL.

1.4 URLs of GaussDB Documents Will Be Changed in Huawei Cloud Help Center

Background

The current keyword for identifying GaussDB documents in URLs is **opengauss**, which is different from the service name, so we will change **opengauss** in the URLs of GaussDB documents to **gaussdb**.

Time

New URLs (keyword: **gaussdb**) will be used on November 2, 2023.

Scope

All GaussDB documents in Huawei Cloud Help Center

Impacts

The document content remains unchanged. It does not affect the use of your DB instance.

1.5 GaussDB Billing Items Will Be Split from TaurusDB

To strengthen brand awareness for Huawei-developed relational databases, on January 25, 2024, we plan to make some changes to GaussDB subscription and billing region by region.

Change Content

1. GaussDB subscription and billing will be separated from TaurusDB.
2. After the change, GaussDB billing items originally assigned to TaurusDB, including database compute resources, storage, and backup space, will be assigned to GaussDB.
3. As long as they have not expired, any cash coupons, flexi-purchase coupons, and other discounts you have for the TaurusDB service will also be usable for GaussDB.

Change Impact

After the change, the product information displayed in the subscriptions, SDRs, and bills of newly purchased GaussDB instances will be **GaussDB**, instead of **TaurusDB**. To be clear, this change will not constitute any amendment or change to the agreements, policies, orders, or other agreements (if involved) related to cloud service sales that have taken effect on the Huawei Cloud website, nor will it have any impact on their effectiveness.

If you have any questions, please **submit a service ticket** within one working day after this notice is released.

2 Product Release Notes

2.1 Vulnerability Fixing Policies

After detecting a vulnerability, we will apply a fix or provide a workaround based on Huawei Cloud's vulnerability fixing policy (see section 9.2 of **Huawei Cloud Security White Paper**). This section lists critical vulnerabilities.

For other vulnerabilities, refer to **Kernel Release History**.

2.2 Version Support Bulletin

2.2.1 GaussDB Version Policies

GaussDB is an enterprise-grade relational database from Huawei. It uses a distributed architecture and features hybrid transactional/analytical processing (HTAP) for high performance and intra-city cross-AZ deployment for zero data loss. GaussDB supports petabytes of storage, with more than 1,000 nodes supported per DB instance. This section describes the lifecycle and policies of GaussDB versions.

GaussDB Engine Version Description

GaussDB uses version numbers in dot-decimal notation. The format is **V2.0-***A.BCD*, where:

- **V2.0**: Versions starting with **V2.0** use the second-generation architecture of GaussDB.
- *A*: *A* designates an annual requirement baseline version.
- *BCD*: The first digit (*B*) represents a semi-annual requirement baseline version, and C and D indicate verified patch versions for third-party adaptation.

Lifecycle of GaussDB Versions

Table 2-1 Lifecycle of GaussDB versions

Version	Release	EOM	EOFS	EOS
V2.0-1.4	July 2021	December 2022	December 2023	June 2024
V2.0-2.0	November 2021	December 2022	December 2025	June 2026
V2.0-2.3	June 2022	December 2023	December 2026	June 2027
V2.0-2.7	November 2022	December 2023	December 2026	June 2027
V2.0-2.8	September 2023	December 2023	December 2026	June 2027
V2.0-3.208	July 2023	December 2024	June 2027	December 2027
V2.0-3.222	September 2023	December 2024	June 2027	December 2027
V2.0-3.223	December 2023	December 2024	June 2027	December 2027
V2.0-3.226	May 2024	December 2024	June 2027	December 2027
V2.0-8.102	May 2024	December 2025	June 2028	December 2028
V2.0-8.103	July 2024	December 2025	June 2028	December 2028

NOTE

- Release: when a version was released on Huawei Cloud. Generally, it refers to when it was released for commercial use.
- EOM: End of Marketing, which is when the sales and deployment of a cloud service version are stopped. After this time, the cloud service version is no longer deployed on the live network.
- EOFS: End of Full Support. When EOFS is reached, Huawei Cloud stops fixing newly found defects or providing any new patches for the particular version. Huawei Cloud will conduct root cause analysis for the defects found, but will not fix them until the next version. However, Huawei Cloud will still fix defects and level-1 and level-2 vulnerabilities before EOS.
- EOS: End of Service & Support. When EOS is reached, Huawei Cloud stops providing any technical service or support for the particular version, including locating problems and fixing defects. Problems identified can only be solved through version upgrades. Customers still must comply with Huawei Cloud's lifecycle policy and upgrade software versions in a timely manner.

2.3 Kernel Release History

2.3.1 V2.0-8.*x* Versions

This section describes the V2.0-8.*x* kernel version updates of GaussDB. For details about the version description, version mapping, and how to check the version number, see **DB Engine Versions**.

Table 2-2 What's new in version V2.0-8.201.0

Date	Feature	Description	Specifications and Restrictions
2024-11	Syntax compatible for large bank customers is added to centralized M-compatible isolation framework.	Based on the existing capabilities of the M-compatible isolation framework, the following syntax items are supported: 1. The field function is supported. 2. The following DML operations are supported: a. INSERT, UPDATE, and DELETE statements support the ignore function. b. The LOAD DATA function is supported. 3. The ALTER TABLE table_name TRUNCATE PARTITION syntax is supported.	Key specifications and restrictions: 1. The field function is supported. 2. The following DML operations are supported: a. INSERT, UPDATE, and DELETE statements support the ignore function. b. The LOAD DATA function is supported. 3. The ALTER TABLE table_name TRUNCATE PARTITION syntax is supported. Delivery mode: Centralized

Date	Feature	Description	Specifications and Restrictions
	Syntax rules supported by centralized instances in the M-compatible isolation framework are supplemented.	Based on the existing M-compatible isolation framework, the following syntax and driver compatibility are supported: 1. The ODBC driver adapts to the M-compatible isolation framework. 2. Some system functions are supported, such as JSON_MERGE_PRESERVE(), JSON_MERGE_PATCH(), JSON_ARRAY(), JSON_OBJECT(), JSON_QUOTE(), JSON_CONTAINS(), JSON_CONTAINS_PATH(), JSON_EXTRACT(), JSON_UNQUOTE(), JSON_KEYS(), JSON_SEARCH(), JSON_APPEND(), JSON_ARRAY_INSERT(), JSON_INSERT(), JSON_MERGE(), JSON_REMOVE(), JSON_REPLACE(), JSON_SET(), JSON_DEPTH(), JSON_LENGTH(), JSON_TYPE(), JSON_VALID(), JSON_ARRAY_APPEND(), benchmark(), FROM_BASE64(), TO_BASE64(), make_set(), mid(), ord(), conv(), crc32(), system_user(), password(), IS_IPV4(), IS_IPV6(), sleep(), bin(), CHAR(), format(), LOCATE(), POSITION(), CEIL(),	Key specifications and restrictions: 1. In principle, the SQL syntax is the same as that in MySQL 5.7. 2. Performance: a. The overall SQL statement execution performance is the same as that of MySQL 5.7. (The execution performance of the CREATE TABLE AS SELECT statement is the same as that of the existing GaussDB.) b. The performance of system functions is the same as that of MySQL. (The Numeric data type scenario is an exception. The calculation logic of this type differs greatly from that of the decimal data type in MySQL, and the performance deteriorates by 10%.) c. Columns that contain the ZEROFILL attribute have no impact on integer input performance. The greater the value of **m**, the greater the deterioration of INTEGER(m) output performance. The overall performance is not worse than that of MySQL 5.7. d. The execution performance of

Date	Feature	Description	Specifications and Restrictions
		md5(), default(), ELT(), RANDOM_BYTES(), std(), found_rows(), row_count(), SQL_CALC_FOUND_ROWS, export_set(), ROW_NUMBER and LEAD(). 3. Some GUC parameters, such as **foreign_key_checks**, **sql_mode** and **autocommit**, are supported. 4. Some data types are supported, such as SET, ENUM, BINARY, ZEROFILL, and JSON. 5. Some DML syntaxes are supported, such as the SHOW STATUS, SHOW ENGINES, SHOW INDEX INDEXES KEYS, SHOW TABLE STATUS, and SHOW TABLES statements. The SHOW statement can be used to query the character set and collation. The UPSET statement can be used to set values in the same column for multiple times. The INSERT statement can be used to leave parameters in the VALUES clause empty. The GROUP BY statement supports the WITH ROLLUP function. The UPDATE statement can be used to reference columns with dbname.tblname.coln	operator is the same as that of MySQL 5.7. Delivery mode: Centralized

Date	Feature	Description	Specifications and Restrictions
		ame. The STRAIGHT_JOIN syntax is supported. The SHOW command can be used to view related information. The EXCEPT syntax is supported. The column names returned by the SELECT function and the SELECT expression can be returned in the MySQL format. The desc table statement and user variables are supported. 6. Some DDL operations are supported, such as view-related DDL statements, ALTER TABLE tbl_name ANALYZE PARTITION {partition_names | ALL}, CREATE DATABASE (for specifying the default character set and collation), the syntax of virtual generated columns, the START TRANSACTION WITH CONSISTENT SNAPSHOT statement, the RENAME TABLE tbl_name TO new_tbl_name [, tbl_name2 TO new_tbl_name2]... syntax, ALTER TABLE tbl_name DROP {INDEX | KEY} index_name, CREATE TABLE AS SELECT syntax and CREATE TABLE SELECT syntax (for creating table), and syntax needed in	

Date	Feature	Description	Specifications and Restrictions
		the upgrade script for upgrade. 7. Some views and character sets are supported, such as information_schema.character_sets, information_schema.collation_character_set_applicability, information_schema.collations, information_schema.tables, information_schema.columns, information_schema.global_status, and information_schema.global_variables views, information_schema.partitions, information_schema.processlist, information_schema.schemata, information_schema.session_status, information_schema.session_variables, information_schema.statistics, information_schema.table_constraints, information_schema.engines, and the latin1_bin and latin1_swedish_ci character sequences of the latin1 character set.	

Date	Feature	Description	Specifications and Restrictions
	JDBC protocol is compatible with centralized M-compatible isolation framework.	Compatibility with JDBC protocol is supported. In centralized scenarios, the MySQL JDBC driver can be used to connect to the GaussDB database. GaussDB is compatible with the MySQL JDBC protocol.	Key specifications and restrictions: 1. The database to be connected must be specified in the JDBC connection URL. 2. For system parameters on which MySQL JDBC depends, GaussDB supports only the query function. The function is not completely aligned with MySQL. 3. Performance specifications: basically the same as those for connecting to the MySQL database. The performance of the driver does not deteriorate compared to that in version 505.1. 4. **m_format_dev_version** must be set to **s2**. 5. The following interfaces are supported: **java.sql.Connection**, **java.sql.Statement**, **java.sql.PreparedStatement**, **java.sql.ResultSet**, **java.sql.DatabaseMetaData**, and **java.sql.ResultSetMetaData**. The **java.sql.CallableStatement** interface is not supported. 6. For the **java.sql.Connection** interface, the **prepareCall** method is not supported. 7. For the **java.sql.Statement** interface, the **getWarnings**,

Date	Feature	Description	Specifications and Restrictions
			setMaxFieldSize, **setMaxRows**, and **cancel** methods are not supported. 8. For the **java.sql.PreparedStatement** interface, the **getMetaData** method is not supported. 9. For the **java.sql.ResultSet** interface, the **insertRow** and **updateRow** methods are not supported. 10. For the **java.sql.DatabaseMetaData** class, the following methods are not supported: **getColumnPrivileges**, **getColumns**, **getCrossReference**, **getExportedKeys**, **getFunctionColumns**, **getFunctions**, **getProcedureColumns**, **getProcedures**, **getTablePrivileges**, **getTables**, and **getUserName** 11. PBE packets on the server are not supported. 12. MySQL Connector/J of 5.1.47 or later is supported. Delivery mode: Centralized

Date	Feature	Description	Specifications and Restrictions
	M-compatible isolation framework is compatible with peripheral tools.	This feature provides incremental adaptation for server and client tools in the M-compatible isolation framework scenario. The following are included: 1. gs_rescue and gs_rescue_tui adapt to the M-compatible framework. 2. The logical decoding tool adapts to the M-compatible framework. 3. gs_dump, gs_dumpall, and gs_restore adapt to the M-compatible framework. 4. gs_clean, gs_redis, and gs_roach adapt to the M-compatible framework. 5. The DBMind tool adapts to the M-compatible framework. 6. gs_om, gs_check, gs_replace, gs_upgradectl, gs_expand, and gs_shrink adapt to the M-compatible framework.	Key specifications and restrictions: After related tools adapt to the M-compatible isolation framework, their functions can be used normally. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
	Distributed M-compatible isolation framework has the basic capabilities.	Distributed M-compatible isolation framework supports the SQL syntaxes, views, functions, operators, and data types that have been implemented in the 505.1 centralized mode, and supports the SQL syntax required by the distributed mode. The details are as follows: 1. OM dependency: The **templatem** can be created in distributed mode. 2. The following SQL syntaxes are supported: a. SQL syntaxes that are already supported in the centralized 505.1 M-compatible framework. b. The SQL syntaxes required by the distributed system are supported, including CREATE TABLE, CREATE TABLE AS, CREATE TABLE LIKE, ALTER TABLE, CREATE INDEX, ALTER INDEX, ALTER COORDINATOR, CREATE/ALTER/ DROP NODE, and their related syntax for distributed mode, and the distributed capabilities of ALTER TABLE tblname TRUNCATE PARTITION.	Key specifications and restrictions: 1. Foreign key constraints cannot be created in the distributed mode. 2. Global secondary indexes are supported in distributed mode. 3. Auto-increment columns are not supported in distributed mode. 4. Data types that can be used as distribution keys include bool, char, varchar, tinytext, text, date, datetime, timestamp, datetime, year, tinyint, smallint, mediumint, int, bigint, numeric and decimal. 5. Performance specifications: The SQL statement execution performance is the same as that of the same SQL statement in GaussDB. Delivery mode: Distributed

Date	Feature	Description	Specifications and Restrictions
		3. Supported views: views that are already supported in the centralized 505.1 M-compatible framework are fully supported in the distributed framework. No view is added. The functions of existing views need to be improved. 4. Supported functions: functions that are already supported in the centralized 505.1 M-compatible framework are fully supported in the distributed framework. BIN(), ELT(), FIELD(), INSERT(), LOCATE(), MAKE_SET(), QUOTE() and DEFAULT() are also supported. 5. Supported operators: operators that are already supported in the centralized 505.1 M-compatible framework are fully supported in the distributed framework. 6. Supported data types: data types that are already supported in the centralized 505.1 M-compatible framework are fully supported in the distributed framework. The SET, ENUM, and JSON data types are also supported.	

Date	Feature	Description	Specifications and Restrictions
	GIS is compatible with the M-compatible platform.	The GIS supports some spatial data types and functions of the PostGIS plug-in provided by Yukon in a single platform in centralized M-compatible non-isolation framework.	Key specifications and restrictions: 1. The GIS supports the Arm architecture EulerOS 2.9 in M-compatible mode (non-isolation framework). 2. One spatial data type is supported: GEOMETRY. 3. 84 spatial data functions are supported. Delivery mode: Centralized

Date	Feature	Description	Specifications and Restrictions
	The O-compatible framework is split.	The O-compatible template database (**templatea**) is split from the O-compatible framework. When the O-compatible database is created, **templatea** is used as the template database. **template0** is used as the template database for the M-compatible non-isolation framework, PG-compatible, and TD-compatible databases. (To ensure forward compatibility, the existing O syntax in **template0** is retained.) In addition, independent O-compatible syntax, lexical files, and independent system functions, views, and advanced package files are created. In later versions, the O-compatible functions are added only to the O-compatible template database **templatea** and other O-compatible databases (excluding **template0** and **template1**). The O-compatible functions evolve independently and do not affect databases of other compatibilities.	Key specifications and restrictions: 1. The O-compatible template database is the same as that of **template0** and is forward compatible. 2. Smooth upgrade without affecting user experience is supported. The O compatibility for users before the upgrade can be used properly after the upgrade. The new O-compatible function can be used in subsequent upgrades. 3. When you create an O-compatible database and specify **template0** as the template database, a notice is generated indicating that the O-compatible template database is actually used. 4. The O-compatible template database cannot be connected. 5. **templatea** can be specified as the template database only when an O-compatible database is being created. 6. Before the upgrade, if the original database has the same name as the O-compatible template database or the auxiliary database during the upgrade (**templatea** and **templatea_temp**), an error is reported in advance, prompting the

Date	Feature	Description	Specifications and Restrictions
			user to change the database name. 7. In the upgrade scenario crossing this O-compatible split framework version, the **template0** database in the distributed mode is compatible with MySQL by default. To maintain forward compatibility, the **templatea** database is created using the **template0** template database during the upgrade, causing some differences in compatibility. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
	For O compatibility, system functions such as trunc, mod, and lengthc are supported.	Based on the existing O compatibility, the following are supported: 1. The system function trunc supports function index for input parameters of the timestamp type. 2. The system function USERENV can obtain the **sid** and **instance** parameters. 3. The mod function is supported. 4. The system function sys_connect_by_path is supported. 5. The lengthc function is supported. 6. The wm_concat function is supported. 7. The keep window function and syntax are supported. 8. The system function systimestamp is supported. 9. JDBC allows getBytes and setBytes to read and write the BLOB type.	Key specifications and restrictions: 1. The performance specifications are the same as those of the corresponding system functions in the system. 2. Functions do not consume a large number of resources, and the resource usage is the same as that in database O. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
	For O compatibility, the INSERT ALL INTO syntax is supported.	Based on the existing O compatibility, the INSERT ALL INTO syntax is supported for inserting data into multiple tables.	Key specifications and restrictions: 1. Inserting multiple data records (100,000 to 1 million records) does not consume a large number of resources. The performance is basically the same as that of running INSERT INTO to batch insert records into a single table, with a gap less than 10%. Table types include ordinary tables, temporary tables, and partitioned tables. 2. Inserting data into multiple tables (less than 1,000 tables) does not consume a large number of resources. The performance is basically the same as that of running INSERT INTO for multiple times to insert data into a single table, with a gap less than 10%. Table types include ordinary tables, temporary tables, and partitioned tables. 3. This syntax is supported only in O-compatible mode. 4. error_logging_clause is not supported. 5. The table alias of the subquery cannot be used in into_clause. 6. This syntax can be executed only on tables and not on ordinary views or materialized views. Ordinary tables, temporary tables,

Date	Feature	Description	Specifications and Restrictions
			partitioned tables, and column-store tables are supported. 7. VALUES cannot be followed by multiple lines. 8. Aggregate functions and subqueries are not supported in VALUES. 9. This syntax cannot be executed on a remote table. 10. Do not use the expression of the Table set in the INTO clause. 11. If an IOT table or a table with a bitmap index exists, multiple INTO clauses cannot be executed in parallel. 12. When into_clause specifies a sequence, if nextval is referenced for the first time, the next number of the current value is generated. Otherwise, the same number will always be returned. 13. The functions of the original INSERT syntax are not completely inherited. The RETURNING clause, ON DUPLICATE KEY keyword, IGNORE keyword, and UPSERT clause are not supported. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
	The CREATE TABLE AS syntax for partitioned table, window function ignore nulls, and window function range between are supported in O-compatible mode.	Based on the existing O compatibility, the following are supported: 1. The CASE WHEN statement can be used to subtract dates. 2. The START WITH CONNECT BY statement is supported in distributed mode. 3. The gsql client supports using slashes (/) to separate statements. 4. The PG-compatible syntax ON UPDATE CURRENT_TIMESTAMP is supported. 5. O-compatible window functions and aggregate functions support the IGNORE NULLS and RANGE...BETWEEN syntax. 6. The gs_source system catalog is supported in distributed mode. 7. The CREATE TABLE AS syntax is supported for partitioned tables. 8. Alias can be referenced when executing INSERT to insert data into a single table. 9. There can be no parenthesis after IN in the WHERE condition. 10. In the scenario where the primary key or unique index exists, indexes can be properly matched	Key specifications and restrictions: The performance of the new syntax is the same as that of the original syntax. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
		when equality matching is performed between the bpchar type and the text type.	

Date	Feature	Description	Specifications and Restrictions
	The performance of stored procedures in typical scenarios is improved, and minute-level slow SQL statements can be located.	Based on the existing capabilities of stored procedures, the following are supported: 1. Anonymous blocks can be cached when dynamic statements are executed in stored procedures. Stored procedures support multi-layer PLState. In centralized mode, stored procedures support reuse of nested abnormal transaction blocks. 2. Stored procedure O&M capabilities are enhanced. The pg_stat_activity view can display all statements that are being executed in a stored procedure. The asp view can display the nesting relationship of stored procedure statements. The capability of tracing statements that are being executed in a stored procedure is supported. 3. Memory usage of stored procedures is optimized. The memory function gs_plsql_memory_object_detail for stored procedure compilation objects can obtain the valid memory usage of compilation products in centralized mode. The gs_glc_memory_detail view and function	Key specifications and restrictions: 1. The compilation performance does not deteriorate. 2. Compared with version 505.1 in typical scenario, the performance of a dynamic statement executing an anonymous block is improved by about 50%. 3. The impact of the pg_stat_activity view does not deteriorate the performance of stored procedure. 4. The gs_asp view needs to generate **unique_sql_id**. After related parameters are enabled, the stored procedure performance deteriorates by about 5%. 5. The performance of the stored procedure that is being traced deteriorates by about 5%. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
		glc_memory_detail are extended. The centralized mode support identification of packages, functions, and package compilation searchpath, and the failure causes can be traced. Global variables of compilation products are reconstructed to reduce nested value assignment scenarios and enhance the readability and maintainability of related code. 4. The end character of the stored procedure is enhanced. CREATE PROCEDURE/ FUNCTION is added. PROCEDURE/ FUNCTION in PACKAGE and nested subprograms in various scenarios can end with "END +name;". CREATE PACKAGE/PACKAGE BODY can end with "END;". 5. ECPG stored procedure is enhanced. Based on the existing ECPG, the compatibility of the SQL syntax related to the cursor is enhanced, including the following two aspects: Cursors can use the WHERE CURRENT OF syntax.	

Date	Feature	Description	Specifications and Restrictions
		Cursors can be opened and closed repeatedly.	

Date	Feature	Description	Specifications and Restrictions
	The batch error mode is supported.	The BatchErrorMode can skip errors in batches for batch operations on data. 1. libpq has the batch request and error handling APIs for BatchErrorMode. 2. If an error occurs when a batch request is executed, the system determines whether to stop or continue the remaining data operations in the current batch immediately based on BatchErrorMode. This capability provides two batch request error processing modes: a. STMT_DEFAULT: In this mode, if an error occurs during a batch data operation, the remaining data operations in the current batch are stopped immediately and all changes are rolled back. b. STMT_BATCH_ERRORS: In this mode, if an error occurs during a batch data operation, the system records the error information, continues to finish the remaining data operations, and returns the detailed error information.	Key specifications and restrictions: The BatchErrorMode can skip errors in batches only when a transaction is explicitly started. Delivery mode: Centralized

Date	Feature	Description	Specifications and Restrictions
	Level-2 partitions of partitioned tables support multi-column fields. Partitioned tables support partition-wise join.	Based on the existing capabilities of partitioned table, the following are supported: 1. Partition-wise join and partition-level operator parallelism are supported for partitioned table. Specifically, partition pair scan needs to be supported for partition key association of partitioned tables. In this way, the performance is improved using multi-thread parallel processing capability and computing resource usage is reduced. 2. When a partitioned table is associated with a non-partitioned table, streaming redistribution is supported for data on the non-partitioned table based on the partition key of the partitioned table, reducing data communication overheads and improving execution efficiency. 3. For level-2 range and list partitioning, multi-column partition keys are supported.	Key specifications and restrictions: 1. Partition-wise join supports level-1 partitions, and range and hash partitioning, in both centralized and distributed modes. 2. Partition-wise join is supported for two or more tables with the same partitioning policy. 3. SMP plan is supported for partition-wise join. When the value of **query_dop** does not match the number of partitions, a round-robin distribution of partitioned data to SMP threads is supported. 4. Partition-wise join does not support level-2 partitions. 5. Partition-wise join is supported for partition-based SMP parallel query. In SMP scenarios, the original SMP constraints are inherited. (However, mergejoin, distributed FQS, and remote join query are not supported.) For distributed stream plans, partition-wise join is optimized through SMP only when partitioned tables do not need to cross DN streams. In scenarios where partition data skew is severe, the performance of an SMP plan may deteriorate. 6. Performance specifications: In the

Date	Feature	Description	Specifications and Restrictions
			scenario where 72-core Arm servers with 2 TB memory are used, the TPC-H 20 test case performance is improved by 40% compared with that in the non-partition-wise join scenario when the data volume is 33X, in column-stored or IMCV format, 64 partitions, and 64 jobs are executed concurrently. In partitioned table-related scenarios, the extra overhead of plan generation in the optimizer phase does not exceed 2%. The performance does not deteriorate in the execution phase, which is verified by the TPC-C benchmark of the partitioned table. The performance improvement applies only to DML. You do not need to pay attention to DDL and import and export performance. 7. For level-2 range and list partitioning of partitioned tables, multi-column partition keys is supported. The centralized mode is supported. 8. A maximum of 16 columns of partition keys are supported. Static pruning, dynamic PBE pruning, and dynamic parameterized path pruning are supported.

Date	Feature	Description	Specifications and Restrictions
			9. Automatic list partitioning is supported, and the conversion between autolist and list is supported. The interval multi-column partitioning is not supported. 10. The sparsely partitioned index is supported. 11. Performance specifications: The extra overhead of pruning performance insertion is the same as that of multi-column level-1 partitioning. The DDL execution performance is the same as that of single-column partitioning. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
	Recompilation and reconstruction upon expiration are supported.	Reconstructing compilation dependency records and recompilation upon expiration make the parameter **ddl_invalid_mode** compatible and simplify the compilation recording process. When the parameter is compatible, the system catalog is locked to ensure that no duplicate data exists, and objects can be cascadingly invalidated.	Key specifications and restrictions: 1. Objects that can be imported to the database at a time: package, function, and procedure. 2. Dependent objects that support DDL recompilation upon expiration: package, function, and procedure. 3. Objects that support recompilation upon expiration and ALTER COMPILE: package, function, and procedure. 4. The view can be rebuilt only when the depended function is deleted or rebuilt. Views cannot be invalidated, compiled, or imported to the database at a time. 5. Performance specifications: The performance of executing stored procedures is not affected. The full compilation performance does not deteriorate. Delivery mode: Centralized

Date	Feature	Description	Specifications and Restrictions
	Ustore supports GIN index scan for efficient fuzzy query.	Ustore supports GIN indexes to greatly improve the performance of fuzzy query. It has the following functions: 1. Ustore supports the creation and deletion of GIN indexes. 2. Ustore supports the DML statements of GIN indexes. 3. Ustore supports GIN indexes for efficient fuzzy query in the LIKE condition.	Key specifications and restrictions: 1. Only one opclass is supported: ugin_trgm_ops, which is not the default opclass. 2. **ugin_trgm_ops** supports index keys of the text, varchar, nvarchar2, and clob data types and alias data types of the four types, but does not support the char(n) type. 3. The maximum length of the index column data of **ugin_trgm_ops** is set to **238609291** bytes. 4. The index key of **ugin_trgm_ops** is specified. Its collation cannot be B-compatible collation. Otherwise, an error is reported when an index is created. For details about B-compatible collations, see COLLATE collation. 5. Specifies the index key of **ugin_trgm_ops**. Index scan can be performed only when the LIKE condition is met. 6. When extracting index keys, **ugin_trgm_ops** considers every three characters as a word segment. If the word segment extracted by the index scan condition is fewer than three characters, the specific index key cannot be located during index scan, and all index keys

Date	Feature	Description	Specifications and Restrictions
			are scanned, such as **col LIKE '%ab %'**. 7. The number of index keys cannot exceed 1. Index keys do not support the NULLS FIRST/LAST or INCLUDE syntax. 8. Only Bitmap Index Scan can be used to execute planned scans. Index Scan is not supported. 9. When standby nodes are queried in ultimate RTO mode, UGIN indexes are not selected for scan. 10. If the GUC parameter **gin_fuzzy_search_limit** is set to a non-zero value, an error is reported during UGIN index scan. 11. The following WITH options are supported: **storage_type**, **gin_pending_list_limit**, and **fastupdate**. 12. The incremental buffer (pending list area) cannot be disabled. Therefore, **fastupdate** can only be set to **ON**. If **fastupdate** is set to **OFF**, an error is reported. 13. It cannot be used as the index of a segment-page table, local temporary table, global temporary table, unlogged table, or encrypted table, or as the global index of a partitioned table. 14. It cannot be used as a primary key, unique key,

Date	Feature	Description	Specifications and Restrictions
			or virtual index, or be encrypted. 15.It cannot be rebuilt, created, and deleted concurrently. 16.It does not support concurrent functions, such as parallel creation and parallel scanning. 17.It cannot be created or used in the database where **sql_compatibility** is set to **'M'**. 18.Performance specifications: In a typical configuration scenario with 10 million data records, each record contains 100 tokens. After deduplication, there are 100,000 tokens. You can perform fuzzy query on these records with up to 10 tokens each time, and it will take a maximum of 600 ms to obtain 10,000 results. Delivery mode: Centralized

Date	Feature	Description	Specifications and Restrictions
	The MERGE INTO, UPDATE, and DELETE operators support parallel processing performance improvement.	Supports parallel data update statements, including MERGE INTO, UPDATE, and DELETE.	Key specifications and restrictions: It can be executed in a stored procedure. 1. The UPDATE, DELETE, and MERGE INTO plans of the SMP are the same as those of the SMP under the SELECT operator. That is, the lower layer of the modifytable operator is the stream gather node. 2. Column-store tables are not supported. 3. Global temporary tables are not supported. 4. DML statements containing the RETURNING clause cannot be executed by the SMP. 5. The target Ustore table cannot have a PCR index. 6. The target table cannot contain triggers or foreign keys. 7. If the target table is a partitioned table and does not support automatic partitioning for autonomous transactions, then the target table is not supported. 8. UPDATE statements containing subqueries in the SET VALUE clause are not supported. DELETE statements containing subqueries are not supported. MERGE INTO statements containing subqueries in INSERT or

Date	Feature	Description	Specifications and Restrictions
			UPDATE are not supported. 9. Concurrent UPDATE, DELETE, or MERGE INTO operations with conflicting tuples are not supported. An error will be reported during the execution. 10. The DML statements that operate an Ustore table in a sub-transaction do not support SMP before the upgrade is committed. 11. In typical configuration scenarios, if SMP optimization is enabled, the operator performance is improved by 5 times. Delivery mode: Centralized

Date	Feature	Description	Specifications and Restrictions
	The end-to-end performance of IUD basic operators is improved.	It has the following functions: 1. IUDS performance improvement: a. Noise floor elimination for operators: A series of performance optimizations are made for the noise floor of the main path function and hotspot function, such as reducing the layers of function calls, eliminating redundant logical judgment and operation, reducing context switching, and enhancing cacheline continuity. b. Cache for operators in execution state: The execution status information obtained in the initialization phase is cached. When the plan tree cache is valid, the cached execution status information is used in the execution phase, saving time for the executor, reducing CPU time and repeated memory allocation, and improving performance. c. Memory context framework optimization: In	Key specifications and restrictions: 1. Performance specifications: In typical configuration scenarios, a. in PBE mode, the performance of the insert operator is 0.7 times that of PG16; b. in PBE mode, the performance of the update operator is 0.7 times that of PG16; c. in PBE mode, the performance of the delete operator is 0.6 times that of PG16; d. in PBE mode, the performance of the indexscan operator is 0.6 times that of PG16. 2. Subplans are not supported. 3. Stored procedures are not supported. 4. GPC is not supported. 5. An execution plan contains only the INSERT, UPDATE, DELETE, Result, and IndexScan nodes. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
		GaussDB, when SQL statements are executed, memory contexts are frequently allocated and released, affecting the performance to some extent. This function optimizes the memory context framework. The core design is to reuse the deleted memory context space. 2. Composite unique indexes are compatible with NULL values. In O-compatible mode, NULL values are not equal to other non-empty values but equal to NULL values. During data insertion, if an empty index column is contained and is the same as an existing index column in the table, an error is reported, and the insertion fails.	

Date	Feature	Description	Specifications and Restrictions
	The distributed execution plan supports parameterized paths to avoid query cancellation due to timeout.	Parameterized remote query is supported to improve the performance of imperfect sharding services (table join and related sublinks) in TP scenarios. 1. Nestloop parameterized plan and execution framework. 2. Related sublink plan and execution framework. 3. Management of multiple prepared statements on a single DN.	Key specifications and restrictions: 1. Only PGXC plans are supported. Stream and FQS plans are not supported. 2. For the Nestloop parameterized path, a plan with the parameter **rescan** is generated for the base table only when the index parameterized path is generated for the base table. 3. Stream and PGXC plans cannot be adaptively selected. 4. Performance specifications: The performance of distributed TPC-C is not affected. The performance is better than that of using a stored procedure to split a query into multiple statements that can be delivered and executed on a single node. 5. The performance is improved by 100 times in typical customer scenarios: 1 million rows of data each on two hash distributed tables, and columns **c1** and **c2** have indexes. Delivery mode: Distributed

Date	Feature	Description	Specifications and Restrictions
	Non-stream execution plans are supported for simple queries using GSIs in distributed instances to achieve better performance.	This feature is incrementally developed based on the existing capabilities of distributed global secondary indexes, including: 1. GSI performance improvement: GSI allows simple queries to execute non-Stream distributed plans, and Stream supports PBE. 2. GSI online capability enhancement: Online GSI creation and rebuild are supported.	Key specifications and restrictions: 1. GSI allows simple queries to execute non-stream distributed plans. 2. Performance specifications: In typical scenarios, the performance of the non-stream distributed plan supported by the GSI TABLE ACCESS BY INDEX ROWID is 10 times higher than that of the GSI Stream plan. 3. GSI allows simple queries to execute non-stream distributed plans: You need to set **gsi_fastpath_level** to **partial** or **strict**. Other settings are the same as those for non-stream distributed plans. GSI TABLE ACCESS BY INDEX ROWID does not support hash bucket tables. 4. Stream PBE optimization: The **enable_stream_pbe** parameter must be enabled. The default value is **ON**. 5. When there are no concurrent DML operations, the GSI online creation performance (**case 1**) in GSIBUILD mode is improved by more than 20% compared with that in version 505.1 (**case 2**), where performance refers to the end-to-end latency of online creation, and

Date	Feature	Description	Specifications and Restrictions
			the calculation formula is: (**case 2** – **case 1**)/(**case 1**). 6. Astore is not supported. 7. GSI cannot be created or rebuilt online during the upgrade. 8. Other restrictions are the same as those for online creation and rebuilding of ordinary indexes. For example, long transactions may block online GSI creation. You are advised not to create GSI online when long transactions exist. Online GSI creation and rebuilding are not supported during online scale-out. In high-concurrency scenarios, online GSI creation slows down. You are advised to create GSI online during off-peak hours. If online GSI creation or rebuilding fails due to serious errors (such as FATAL, PANIC, and database faults), indexes (and temporary tables) that are not ready or valid remain, which need to be manually cleared. Delivery mode: Distributed

Date	Feature	Description	Specifications and Restrictions
	In key service scenarios, the ultimate RTO replay speed is improved by 20%, and the flow control performanc e jitter does not exceed 20%.	This feature is incrementally developed based on the existing ultimate RTO and flow control mechanism, including: 1. In the large table truncate and drop scenarios, the mechanism for deleting hash tables such as invalidPage is optimized to improve the replay speed in this scenario. 2. The standby node uses an independent buffer pool for reading **lsninfo** and **blockinfo**, with a dedicated page replacement and flush algorithm. An asynchronous flush thread is added to manage page writes. 3. The flow control algorithm is optimized to reduce the flow control jitter. 4. In ultimate RTO, the restriction that the value of **recovery_parse_work ers** is greater than 1 is removed.	Key specifications and restrictions: 1. In the TRUNCATE/DROP scenario of a large table (greater than 1 TB on a single DN), the replay speed is improved by more than 20% when a DN is faulty and restarted. 2. In typical configuration scenarios (96 vCPUs, 512 GB of memory, NVMe SSD, centralized instance with one primary node and two standby nodes), when 1,500 concurrent updates are performed on the primary node and 50 concurrent full table scans are performed on the standby node, the average replay speed is improved by more than 20%, and the performance of read on standby does not deteriorate. 3. In a typical configuration scenario (Arm EulerOS 2.0 (SP9) 2 channels, three nodes, one primary and two standby nodes, 128 vCPUs, 512 GB memory, NVMe SSD), flow control is enabled. The system runs for 30 minutes in the TPC-C scenario, the tpmC fluctuation is less than 20%. 4. In a typical configuration scenario (16 vCPUs, 64 GB memory, SSD,

Date	Feature	Description	Specifications and Restrictions
			distributed environment with two shards, one primary node and two standby nodes, and the total size of a single table is greater than 500 GB), flow control is enabled. Run **sysbench insert/update/delete** and vacuum operations for 30 minutes. Compared with the scenario where flow control is disabled, the TPS fluctuation is less than 20%. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
	The parallel import performance of gs_loader is improved by five times, and the error tolerance for invalid characters is supported.	This feature is incrementally developed based on the original functions of gs_loader, including: 1. gs_loader supports parallel import using the **parallel** parameter, and the value of **parallel** is used as the degree of parallelism, improving import performance. 2. gs_loader supports error tolerance for invalid characters using the **compatible_illegal_chars** parameter. If a data file contains invalid characters during the import, no error is reported. Instead, the data file is correctly imported to the data table after being processed. 3. The **compatible_illegal_chars** parameter is added to the **\COPY TO** and **COPY TO** commands. If an invalid character exists during the export, no error is reported. Instead, the export is successful after certain processing.	Key specifications and restrictions: 1. When the function supporting non-rollback upon constraint conflicts is enabled: a. If there is no row insert trigger or other non-row insert triggers, the performance does not deteriorate. b. When a row insert trigger is used, the performance deteriorates by 60%. 2. When the rows parameter is specified, the performance does not deteriorate significantly when the total data volume remains unchanged and the number of commit times is less than 1000. 3. After the error table is disabled, the greater the amount of data in the error table, the more significant the performance improvement. 4. When error tolerance for invalid characters is enabled: a. If the data file does not contain invalid characters, the performance does not deteriorate. b. If a data file contains invalid characters, the character conversion performance remains the same as that of

Date	Feature	Description	Specifications and Restrictions
			COPY. Conversion logs are recorded when **enable_log_copy_illegal_chars** is set to **on**. In addition, the log writing overhead is in a linear relationship with the amount of imported data. In an SSD scenario where CPU and memory do not have bottlenecks, the log writing overhead increases by no more than 400s for recording every 10 million rows of data. c. In a scenario where CPU and memory do not have bottlenecks, when the system is fully loaded, the bandwidth occupied by log writing is about 5% of the peak disk I/O bandwidth. 5. When parallel import is enabled, the specifications are as follows (The CPU and memory resources of the client and the CPU and memory resources, idle threads, and network bandwidth of the server meet the required degrees of parallelism and do not have bottlenecks. The total number of data rows is large enough, and the value of **skip** parameter is less than 1% of the total number of data rows.):

Date	Feature	Description	Specifications and Restrictions
			a. When the degree of parallelism is 2, the performance is improved by at least 1.5 times compared with serial import. b. When the degree of parallelism is 4, the performance is improved by at least 3 times compared with serial import. c. When the degree of parallelism is 8, the performance is improved by at least 5 times compared with serial import. d. The improvement of performance decreases as the degree of parallelism increases. e. Each time the degree of parallelism is incremented by 1, the client memory increases by about 10 MB and the server memory increases by about 35 MB. 6. When the function supporting non-rollback upon constraint conflicts is enabled: a. If a table has a BEFORE/AFTER ROW INSERT trigger, a maximum of 10 million rows can be committed at a time. b. Statement-level triggers are not supported. 7. When the number of rows to be committed is specified, it is

Date	Feature	Description	Specifications and Restrictions
			recommended that the number of commit times be less than or equal to 1,000 considering the impact of multiple commits on performance. Frequent commit of a small amount of data affects the data import performance. You are advised to set the **rows** parameter properly to ensure that the amount of data committed each time is greater than 5 MB. For 1 primary + 2 standby instances deployed on common servers with 16 vCPUs and 128 GB of memory and 13 GB of data is imported to a table with five columns, the rate of multiple commits is about 10 MB/s, which is basically the same as that of a single commit (5 MB data is committed each time). 8. Currently, compatibility is supported only when data files contain NUL characters. NUL characters in .ctl control files are not supported. If the .ctl file contains the NUL character, unexpected problems may occur. 9. After the **binary** parameter is set to **true**, the following requirements must be met: a. The data file must be a binary file exported in \COPY binary

Date	Feature	Description	Specifications and Restrictions
			mode. However, the data file exported in this mode has poor compatibility and portability. You are advised to use \COPY to import the data file. b. gs_loader converts the syntax in the control file to the simplest syntax in \COPY binary mode, that is, \COPY *table_name* FROM *'binary_file_path'* BINARY. Only the import mode, table name, as well as control, data, binary, guc_param, and database connection parameters in the control file are parsed. Other parameters are not parsed and do not take effect. c. The command lines and control files of gs_loader must meet the following requirements: Character set configuration is not supported. The WHEN filter and DISCARD operation are not supported. Error data cannot be directly written to **BAD** files when **enable_copy_error_log** is set to **off**. The default value of **errors** is **unlimited**, indicating that encoding exception

Date	Feature	Description	Specifications and Restrictions
			data is recorded by default. The CSV mode is not supported, delimiters and wrappers cannot be specified, and the TRAILING NULLCOLS syntax is not supported. Data type configuration, POSITION configuration, and column expression usage are not supported. The **FILLER**, **CONSTANT**, **SEQUENCE**, and **NULLIF** parameters are not supported. The **skip**, **rows**, and **compatible_nul** parameters are not supported. 10. When error tolerance for invalid characters is enabled: a. If **TERMINATED** or **OPTIONALLY ENCLOSED BY** is converted to spaces or question marks, an error message stating "illegal chars conversion may confuse COPY escape 0x20" will be displayed, prompting you to modify the parameter. b. When GUC parameter **copy_special_character_version** is set to **'no_error'**, its priority is higher than that of **compatible_illegal_chars**. When **copy_special_charac**

Date	Feature	Description	Specifications and Restrictions
			ter_version is set to **'no_error'**, if transcoding is not required, invalid characters are expected to be imported as they are. However, the compatible_illegal_chars API still checks the encoding validity and replaces invalid characters when transcoding is not required. To use compatible_illegal_chars in scenarios where transcoding is not required, leave the GUC parameter **copy_special_character_version** empty. c. The compatible_nul function cannot be enabled at the same time because there is a conflict between the two functions. compatible_illegal_chars converts NUL characters to space characters. When **compatible_nul** is set to **false**, NUL characters are not converted to space characters. 11. When parallel import is enabled: a. The maximum degree of parallelism cannot exceed twice the number of CPU cores on the client. When the client runs in a container, the number of CPUs obtained is the

Date	Feature	Description	Specifications and Restrictions
			number of CPUs on the host, which may be greater than the number of CPUs that can be used by the container. Therefore, you are advised to set the maximum degree of parallelism to a value less than twice the number of CPUs that can be used by the client. b. OPTIONALLY ENCLOSED BY or FIELDS CSV cannot be used to set quotation marks in control files. c. Data in binary format cannot be imported. d. Parallel import is implemented based on data division on the client and cannot ensure that data is imported in sequence. If a table contains auto-increment columns, the sequence of values in the auto-increment columns may be different from that in the data file after the table is imported. 12.When error tolerance for invalid characters is enabled for COPY and \COPY export: a. The error occurs in the scenario where transcoding is required. That is, in scenarios where

Date	Feature	Description	Specifications and Restrictions
			transcoding is not required (the file encoding or client encoding is the same as the server encoding), invalid characters are not processed by compatible_illegal_chars and are exported as they are. b. Invalid character conversion applies only to encodings supported by GaussDB. (For encodings that are not supported, an error is still reported.) 13.gs_loader is applicable only in centralized mode. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
	VerifyLog is enhanced to prevent error pages from infecting the standby node or standby cluster for DR through Xlogs.	This feature can quickly identify error pages, prevent error Xlogs from being generated, and prevent error pages from infecting the standby node or standby cluster for DR through Xlogs. It has the following functions: 1. The GaussDB version number field is added to VerifyLog. 2. When the function is permanently enabled, the anticache verification does not need to wait until the vlog recovery is complete. 3. The recycling verification mechanism is optimized. 4. The build scenario function is adapted. 5. The segment page, UNDO, UNDO segment page, and EXRTO page verification are supported.	Key specifications and restrictions: 1. By default, the function is permanently enabled at the third level, which does not affect the RTO. The performance deteriorates by less than 2% in typical scenarios with default parameters. 2. By default, this function is disabled for devices with small specifications (4-core CPU/16 GB memory, 4-core CPU/32 GB memory, and 8-core CPU/64 GB memory). 3. Verification is not supported in the anticache rebuilding phase after the system is restarted due to a fault. Verification is not supported in database initialization scenario. Non-shared_buffers pages are not supported. Verification is not supported when the anticache memory reaches the upper limit. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
	RTO is optimized for DR switchover of distributed streaming cluster.	This feature enhances the capabilities of distributed streaming replication-based DR. 1. In the distributed deployment scenario of the streaming replication-based remote DR solution, the kernel switchover time is shortened from 1 minute to 30 seconds. 2. In the distributed deployment scenario of the streaming replication-based remote DR solution, the kernel failover time is shorten from 1 minute and 30 seconds to 30 seconds.	Key specifications and restrictions: 1. The existing constraints of the distributed streaming replication-based DR are inherited. 2. The switchover specifications inherit the original prerequisites. If no component is faulty, the RTO is less than 30s. 3. The failover specifications inherit the original prerequisites. If no component is faulty, the RTO is less than 30s. If a minority of components are faulty, the RTO is less than 90s. 4. Only the quorum protocol is supported for the optimization of promoting a standby cluster for DR to primary. 5. The standby cluster for DR must be degraded or normal when it is promoted to primary. Delivery mode: Distributed

Date	Feature	Description	Specifications and Restrictions
	Support of seconds-level centralized DCF DR switchover, minute-level distributed DCF DR switchover, and SM series cryptographic algorithms	Based on the existing DCF capabilities, this feature provides the following new functions: 1. DCF supports seconds-level centralized DR switchover. 2. DCF supports distributed DR networking. 3. DCF supports the DR cluster mode switching between quorum and DCF. 4. DCF supports SM series cryptographic algorithms.	Key specifications and restrictions: 1. The cross-AZ network latency of the primary cluster must be less than or equal to 2 ms, and the remote network latency between the primary and standby clusters must be less than or equal to 100 ms. The DR can run normally within the range of the required network latency. Otherwise, the primary and standby nodes will be disconnected. 2. Establishing the DR Relationship: The performance of the primary cluster deteriorates by no more than 20% before and after a DR relationship is established. 3. When the standby cluster for DR is promoted to primary: a. A certain amount of data can be lost, and the RPO is less than or equal to 10 seconds. b. If no logs are backlogged in the standby cluster for DR (RPO ≤ 10 seconds, RTO ≤ 1 second), instances in the standby cluster for DR are normal, and all components (CN, DN, ETCD, GTM, cm_agent, and cm_server) are normal, the RTO for

Date	Feature	Description	Specifications and Restrictions
			promoting the standby cluster for DR to primary is less than or equal to 90 seconds. c. If no logs are backlogged in the standby cluster for DR (RPO ≤ 10 seconds, RTO ≤ 1 second), some components (CN, DN, ETCD, GTM, cm_agent, and cm_server) of the standby cluster for DR are abnormal but still meet the minority fault case, the RTO for promoting the standby cluster for DR to primary is less than or equal to 120 seconds. 4. When performing a DR switchover: a. If no logs are backlogged in the standby cluster for DR (RPO ≤ 10 seconds, RTO ≤ 1 second), the number of sequences is less than 10, the primary and standby instances are normal, and all components are normal, the DR switchover can be completed within 60 seconds of the kernel time. b. If no logs are backlogged in the standby cluster for DR (RPO ≤ 10

Date	Feature	Description	Specifications and Restrictions
			seconds, RTO ≤ 1 second), the number of sequences is less than 10, and some components of the primary and standby instances are abnormal but still meet the minority fault case, the DR switchover can be completed within 120 seconds of the kernel time. 5. The mode switchover command supports the input of the switchover timeout interval. The default timeout interval is 180 seconds. You are advised to perform the switchover when the service load is light. If the service load is heavy, the mode switchover may not be completed within the timeout interval. 6. Whether the DCF enables SM series cryptographic algorithms and the loaded SM series cryptographic algorithm suite are consistent with those of Quorum. 7. Systems using GMSSL (SSL certificates for SM series cryptographic algorithms) cannot establish secure communication with systems using international SSL certificates. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
	Topology link faults on the data plane of the first aid are reported at minute-level.	The following functions are added to the first aid: 1. The black box detection and alarm capabilities of the database component based on simple SQL statements are supported. 2. The detection and alarm capabilities for topology link faults on the data plane are supported.	Key specifications and restrictions: SQL detection capability specifications and restrictions: 1. This function can be enabled only when the number of CNs in the cluster is less than or equal to the value of **emergence_sql_check_max_cn_count**. 2. Link issues are handled during first aid SQL statement detection only when the source and target ends are normal. 3. If the number of shards in a cluster is less than 10, an alarm can be reported within 1 minute. 4. Only the data plane network can be detected. 5. No detection is performed for a single-node cluster. 6. Arbitration is not performed during upgrade or scale-out. Arbitration is not performed when the DDB is unavailable. 7. Arbitration is not performed during the startup. 8. SQL detection supports only distributed deployment. 9. The standby cluster for DR does not support SQL statement detection, and arbitration is not

Date	Feature	Description	Specifications and Restrictions
			performed during a DR switchover. 10. If the network is normal, automatic switchover is not supported in the following scenarios and switchover needs to be manually performed: a. An error is reported when SQL statements are executed to test links between CNs and some DNs. b. An error is reported when SQL statements are executed to test links between GTMs and some CNs and DNs. c. An error is reported when SQL statements are executed to test links between some CNs. d. An error is reported when SQL statements are executed on the primary DN to test links with the minority of DNs. Specifications and restrictions of the data-plane topology detection capability: 1. Network detection on the data plane is supported to check whether the data plane network is normal. 2. Data plane network detection is not supported for the standby cluster in DR scenarios.

Date	Feature	Description	Specifications and Restrictions
			3. No detection is performed for a single-node cluster. 4. No arbitration is performed during upgrade or scale-out. 5. Arbitration is not performed during the startup. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
	Cluster components can be isolated.	In some fault scenarios (such as network isolation, disk hang, or CM Agent fault), the CM Agent on the faulty node cannot report the fault. Before delivering the arbitration result, the CM isolates the faulty node from the cluster so that it is not connected to any instance in the cluster.	Key specifications and restrictions: 1. RTO for a single instance fault isolation is less than or equal to 30s. 2. The isolation starts after the instance fails to be killed or the network is abnormal for more than 6 seconds. 3. The end-to-end time for fault isolation of a single instance is less than or equal to 7 seconds (5 seconds for communication or storage and 2 seconds for CM). 4. The following scenarios are supported: CN and DN isolate the faulty GTM, DN isolates the faulty CN. 5. Isolation is used only when the process down status cannot be determined. An isolation alarm is generated. In this scenario, you need to manually determine why the process down status cannot be determined. If necessary, you need to manually deliver a kill command or power off the corresponding host. 6. If the isolation operation cannot be completed, subsequent arbitration (GTM leader node election and CN isolation) cannot continue, and the system generates an isolation failure alarm.

Date	Feature	Description	Specifications and Restrictions
			You need to manually handle it as soon as possible. Delivery mode: Distributed

Date	Feature	Description	Specifications and Restrictions
	Position can be specified for logical decoding.	In GaussDB centralized deployment mode, serial/parallel logical decoding supports the decoding of online WALs from a specified position. This position can be an LSN. Logical decoding finds a consistency LSN from the specified LSN, decodes data from the consistency LSN, and outputs data.	Key specifications and restrictions: 1. Performance: a. Performance impact on logical decoding: Compared with logical decoding tasks that depend on system catalogs, the decoding performance of pure DML and DDL transactions does not deteriorate. That is, when other system resources are sufficient, the decoding rate reaches 100 MB/s in general specifications and 300 MB/s in MetaERP scenarios. b. Impact on transactions: The performance of DML does not deteriorate, but the performance of DDL decreases by 20%. (The performance deterioration is related to the I/O capability. ECS: 20%; BMS: 10%.) 2. A position can be specified for decoding online WALs only. 3. You need to adjust the number of online WAL files that can be retained on GaussDB based on the number of WAL files generated by different services every day, so that a position can be specified for decoding the WALs.

Date	Feature	Description	Specifications and Restrictions
			4. Logical decoding actually starts from the consistency LSN. The specific location of the consistency LSN is related to the actual situation of concurrent transactions, for example, long transactions. Only the data modifications generated by transactions since the consistency LSN can be decoded. You need to ensure that the transactions to be decoded start after the consistency LSN. Otherwise, data inconsistency may occur. 5. To configure the retention period of logical decoding data dictionaries, set the GUC parameters. If the data dictionary retention period is shorter than the specified period of generating WALs, decoding may fail due to dictionary data missing. Delivery mode: Centralized

Date	Feature	Description	Specifications and Restrictions
	DBRepair supports seconds-level damaged data page restore, and DBVerify supports consistency check.	1. Capabilities of repairing damaged data pages between the primary and standby nodes: a. The overall DB Repair code framework is optimized. The existing repair capability framework is reconstructed to provide a general entry for HA page repair and provide repair services in proxy mode. b. Auto Block Repair: supports automatic repair of logical bad blocks between the primary and standby nodes. In version 505.1, automatic repair of physical bad blocks between the primary and standby nodes is supported. In this version, the following enhancements are made: Commercial data table types such as hash bucket and segment-page tables are supported. The standby node can automatically pull pages from the primary node for repair when detecting	Key specifications and restrictions: Capabilities of repairing damaged data pages between the primary and standby nodes: 1. The performance of TPC-C and Sysbench services in typical scenarios is not affected. 2. To ensure consistency, the primary or standby node to be repaired needs to wait until the target LSN is replayed. Ideally, it takes less than 1s to obtain a page to be repaired from the remote end. That is, the RTO for repairing a page increases by less than 1s. 3. As the number of physically damaged pages on the standby node increases during log replay, the repair time increases linearly (but still affected by the network and primary node conditions). In extreme scenarios, if the number of damaged pages exceeds a certain threshold, the primary node's libpq library may experience congestion, leading to an exponential increase in repair time. The number of pages that can be repaired at the same time, that is, the maximum capacity of the hash table for repairing pages, is 100 by default. The capacity

Date	Feature	Description	Specifications and Restrictions
		discontinuous LSNs, block loss, and uninitialized scenarios during replay. If write loss or pages not marked as dirty are detected on the primary node during verification, pages are automatically obtained from the standby node for repair. If the primary node detects physical or logical bad blocks during manual page verification, pages are obtained from the standby node for automatic repair. c. The bad block repair system function is enhanced. In the past, many problems occur when view repair functions are actively called. Most of the problems are logical errors caused by lack of input parameter validation, direct operations on correct pages, and manual error rectification. These problems are rectified. 2. Capabilities for supporting manual	can be expanded without a strict upper limit. In extreme scenarios, excessive memory allocation for the bad block hash table may trigger a primary/standby switchover. 4. Automatic and manual repair supports Astore and Ustore. The lite, centralized, and distributed deployment modes are supported. Ordinary row-store tables, segment-page tables, ILM compressed tables, and data pages and index pages of hash bucket tables can be automatically repaired. MOT is not supported. Temporary tables, global temporary tables, and unlogged tables do not involve primary and standby restoration. The undo record page and transaction table page support only append write. Therefore, they can be repaired only when CRC is damaged. The undo meta-information page does not support primary and standby restoration. 5. Automatic block repair depends on the normal communication between the primary and standby nodes and at least one data page on the primary or standby node is normal. The primary node supports automatic

Date	Feature	Description	Specifications and Restrictions
		data correctness and consistency check: a. General CT entry framework for table-level verification data provides scalability, incorporates all current table-level verification capabilities (U-heap, RCR UB-tree, PCR UB-tree, URQ and U-heap consistency verification) into the framework for management, provides a unified exit, and is flexible enough to support future table-level verification capability APIs. b. Data consistency verification system functions support data consistency verification between data tables and indexes using system functions or SQL queries. The supported heap table types include Astore, Ustore ordinary tables, hash bucket tables, and B-tree and UB-tree indexes.	detection and repair of the following types of bad blocks: a. The CRC does not match. b. Anticache or VerifyLog detects write loss. c. The data page is damaged during manual verification. The standby node supports automatic detection and repair of the following types of bad blocks: a. The CRC does not match. b. The LSNs are discontinuous during replay. c. The page is lost or not initialized. 6. After CRC bad blocks are detected in backup scenarios, the kernel repair function gs_verify_and_tryrepair_page is automatically called to repair the bad blocks. 7. Automatic repair is triggered after a damaged page is accessed and detected. The repair of the standby node is controlled by the GUC parameter **standby_page_repair** and is enabled by default. The repair of the primary node is not controlled by a GUC parameter and is enabled by default.

Date	Feature	Description	Specifications and Restrictions
			Capabilities for supporting manual data correctness and consistency check: 1. The performance of TPC-C and Sysbench services in typical scenarios is not affected. 2. To ensure verification accuracy, a level-1 read lock is added to the target table or index for read verification. The performance of index consistency verification is similar to that of index scan. The performance of heap table consistency verification is poor. The efficiency is similar to that of sequential scan and index scan by index. However, block-based execution and resumable execution are supported. 3. The background management and control and periodic scheduling capabilities is supported. You are advised to use scheduled task scheduling during off-peak hours of the standby node or primary node to prevent impact on services on the primary node. To use the I/O resource management and control capability, you need to enable related resource management and control GUC parameters.

Date	Feature	Description	Specifications and Restrictions
			4. Data verification supports Astore and Ustore, but does not support MOT. Centralized and distributed deployment modes are supported. Data verification can be performed on the standby node, but not on the logger node. The verification of common row-store tables, temporary tables, global temporary tables, unlogged tables, ILM compressed tables, TDE tables, and hash bucket tables is supported by heap tables. Indexes support B-tree and UB-tree index verification and are compatible with segment-page storage. Currently, GIN and GiST indexes are not supported. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
	GUC parameter adaption in upgrade scenarios and parallel upgrade of multiple databases are supported.	This feature optimizes the compatibility and duration for upgrade scenarios, including: 1. GUC parameter adaption in upgrade scenarios is supported. 2. Parallel upgrade of multiple databases is supported.	Key specifications and restrictions: 1. In the scenario of parallel upgrade of multiple databases, the degree of parallelism is the number of CPU cores of the upgrade script executor by default. The maximum degree of parallelism is 32. 2. In the same upgrade path, the execution time of the upgrade script for 100 user databases is reduced by 50%. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
	The data rescue tool can be used to parse and restore the ILM and hash bucket tables.	In extreme scenarios without redundancy, such as no backup, no standby node, network isolation, and cluster fails to start, the data rescue tool is used to safeguard customer data security. In this version, the following functions are added: 1. gs_rescue and gs_rescue_tui: support the parsing and restoring of ILM tables, hash bucket tables, and encrypted tables. 2. gs_rescue_tui: adapts to the function of parsing all historical version data in data files. 3. gs_rescue_tui: supports the integrity of end-to-end data restoration in a distributed environment.	Key specifications and restrictions: 1. The ILM tables, hash bucket tables, and encrypted tables can be parsed and restored. 2. Data can be restored and imported to the target database on the web UI. For common data types, the automatic import speed is no lower than 40 MB/s on the server configured with 16 vCPUs, 32 GB of memory, and SSDs. 3. The data rescue tool can parse and restore data files stored on each DN in the distributed database to the target database. 4. Only flushed data is parsed. 5. Offline data parsing cannot ensure data consistency. 6. Data in Xlog or undo records is not parsed. 7. The database must be offline. 8. Astore/Ustore, segment-page-based, and partitioned table data can be parsed. The parsing of ILM tables, hash bucket tables, and encrypted tables is also supported. Column-store and TDE tables are not supported. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
	The compression ratio of the row-store compression feature is improved, and restrictions are removed.	Based on the row-store compression function in earlier versions, this feature supports the following: 1. Metadata compression and constant encoding are introduced to further improve the compression ratio. 2. The restriction on distributed scale-out is removed. Compression is performed when compressed rows in the original table are added to a new node. 3. The M-compatible mode is supported. 4. The CREATE TABLE LIKE syntax can be used to copy the ILM policy of the original table.	Key specifications and restrictions: 1. When metadata compression is enabled, the average compression ratio of typical datasets is improved by 30% compared with version 505.1. 2. The constant coding supports the following storage data types: char, int16, int32, int64, date, timestamp and varlen (variable-length data type). The constant encoding has greater code priority than difference encoding. 3. If the compression policy is not enabled in typical TPC-C scenarios, existing services are not affected. 4. In a typical TPC-C scenario, after setting up an ILM strategy and performing full compression, the performance degradation does not exceed 10% when running TPC-C workloads with 1,000 warehouses and 500 concurrent users. 5. Partial decompression: In the point query scenario where compressed tables are accessed (random query when data is evenly distributed), the time consumed by the LZ4_decompress_safe function decreases by 30%.

Date	Feature	Description	Specifications and Restrictions
			6. The compressed table adapts to distributed scale-out. After scale-out, the space usage of the compressed table remains unchanged. During scale-out, columns are added to the temporary table. As a result, the capacity increases, which is the same as that in the non-compression scenario. Therefore, you need to reserve disk space on the original DN for twice the space of the largest single table, which is the same as that in the non-compression scenario. With the same data volume, the scale-out speed decreases by no more than 50% compared with that of a non-compressed table. 7. The original M-compatible framework uses the same syntax of the row-store compression feature as that in O-compatible mode. 8. During the upgrade observation period, new metadata compression, constant encoding, and LZ4 partial decompression are disabled. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
	In distributed GTM-Lite mode, the strong-consistency read performance of the standby node reaches 80% of the read performance of the primary node.	This feature builds the strong-consistency read performance of the standby node in GTM-Lite mode, including: 1. The distributed GTM-Lite mode supports read on standby in serial and parallel replay modes. 2. The distributed GTM-Lite mode supports read on standby in ultimate RTO replay mode.	Key specifications and restrictions: 1. The query will be canceled in any of the following scenarios when standby nodes are read in ultimate RTO mode: a. Lock conflicts occur during query and replay. Similar to the case of replaying read on standby in serial/parallel mode, the query cancellation is controlled by the **max_standby_streaming_delay** parameter, and the error "canceling statement due to conflict with recovery" is reported. b. The query time exceeds the value of **standby_max_query_time**. c. The forcible recycling of files related to read on standby is triggered, which is controlled by the **standby_force_recycle_ratio** parameter. d. When standby nodes replay logs related to segment-page-based physical space shrinking, queries will be canceled. e. When the stream execution plan is enabled, the query conflicts with the relmap log replay.

Date	Feature	Description	Specifications and Restrictions
			2. When DDL operations are frequently performed on the primary node, the query on the standby node may be slow or even canceled due to conflict with replay. 3. The barrier dotting function must be enabled. If this function is enabled, the cluster performance will be affected, which follows the existing mechanism. 4. In scenarios of low specifications (8 vCPUs or less), ultimate RTO replay occupies a large number of resources. Therefore, you are advised not to enable read on standby in ultimate RTO mode. 5. For read on standby in ultimate RTO mode, query and replay conflict may occur to logs related to the following operations: a. DROP DATABASE b. DROP TABLESPACE c. Clean logs generated by VACUUM d. REINDEX DATABASE 6. Resource management and control: a. Disk space: A threshold is specified for controlling the space used by files related to read on standby in ultimate RTO mode. b. Memory and I/O: For read on standby,

Date	Feature	Description	Specifications and Restrictions
			data can be read from an independent buffer pool and dirty data can be flushed independently. c. CPU: Currently, there is no resource control capability. When the CPU usage of a node is too high (it is recommended that the CPU usage is less than or equal to 70%), the performance of replay and read on standby deteriorate significantly. Delivery mode: Distributed
	ODBC driver can connect to the database to implement read on standby.	ODBC driver can connect to the database to implement read on standby.	Key specifications and restrictions: 1. The performance is the same as that of JDBC in the corresponding scenario. 2. Only CN load balancing in distributed mode is supported. 3. To read on the standby node in distributed mode, the kernel version must be 503.0 or later. 4. Read on standby in a standby cluster for DR is not supported. 5. In distributed mode, the **TargetServerType** parameter cannot be set to a value other than **primary**. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
	libpq can statically call the OpenSSL library, and global loading of SSL certificates is supported.	This feature enhances libpq functions. The following are included: 1. The driver for libpq to statically call the OpenSSL is added to meet customers' requirement for handling OpenSSL version conflicts. 2. The SSL global loading capability is supported. The SSL context environment is initialized only when the server is started. All subsequent session connections share the context.	Key specifications and restrictions: 1. Compared with the libpq that dynamically links the OpenSSL, the libpq that statically linked to the OpenSSL occupies more space, but the APIs and functions remain the same. 2. After the SSL global loading function is enabled, you need to restart the cluster for the certificate to take effect after replacing or revoking the certificate. Delivery mode: Centralized and distributed
	The length of the private key password supported by gs_guc is extended to 128 bits.	The password length supported by gs_guc encrypt is extended to 128 bits, and forward compatibility is maintained.	Key specifications and restrictions: 1. The maximum length of the private key password supported by gs_guc encrypt is extended to 128 bits. 2. The functions are forward compatible. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
	New grouping policy for the JDBC load balancing parameter **autoBalance**.	This feature is incrementally developed based on the existing JDBC capabilities, including: 1. In load balancing mode, **autoBalance** can be set to **shuffleN**. 2. The fault locating capability is enhanced. Logs of main processes (such as establishing connections, executing SQL statements, obtaining results, and closing connections) are modified and added, and exceptions thrown are modified. a. Logs are added for key processes of hot APIs. The log content must contain unique IDs, such as connectionID and SQL ID, to facilitate fault locating. b. Log levels are set based on the log content, and a switch is added to control the log level lower than Warning. c. Exceptions in core APIs and key processes are rectified. Error-level logs are recorded before exceptions are thrown. For exceptions reported by the	Key specifications and restrictions: 1. The shuffleN load balancing policy supports distributed IP address configuration but does not support IPv6. The performance does not deteriorate compared with that when **autoBalance** is set to **priorityN**. 2. The performance does not deteriorate when the fault locating capability added in this version is enabled. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
		kernel, in addition to simple error information, more detailed cause information, including ErrorCode, Position, SQL, OriginalSQL and ErrorMessage, are provided, with reference to Oracle Database.	

Date	Feature	Description	Specifications and Restrictions
	In distributed mode, logical decoding supports DDL and provides data retrieval, comparison, and repair tools.	This feature is incrementally developed based on the logical decoding capability, including: 1. Logical decoding in a distributed cluster supports DDL statement decoding. 2. The kernel provides a data retrieval API. It decodes WALs in a specified LSN range of the cluster and parses data modification actions recorded in WALs into SQL statements. All data fields need to be displayed for DELETE and UPDATE operations. 3. The memory and file handle resources are properly optimized and controlled during the running of the logical decoding task. 4. Logical decoding tasks use snapshots of historical versions to optimize data visibility determination.	Key specifications and restrictions: 1. Logical decoding does not support DDL/DCL/DML hybrid transactions. In hybrid transactions, DML statements after DDL statements cannot be decoded. 2. When there is no DDL statement, the logical decoding performance in the DML-only scenario does not deteriorate. The logical decoding performance in the DDL-only scenario is about 80 MB/s in the standard environment, and that in DDL/DML hybrid transaction scenario is about 80 MB/s in the standard environment. 3. When logical decoding supports DDL, the performance decrease of DDL statements is less than 15%. 4. General decoding restrictions (serial and parallel): DDL statement decoding cannot be performed on local temporary objects, such as GLOBAL temporary tables, LOCAL temporary tables, and temporary schemas. 5. DDL statement decoding in the FOREIGN TABLE scenario is not supported. 6. The DEFAULT of ALTER TABLE ADD COLUMN

Date	Feature	Description	Specifications and Restrictions
			does not support stable or volatile functions. The CHECK constraint expression of CREATE TABLE and ALTER TABLE regarding columns does not support stable or volatile functions. If ALTER TABLE has multiple clauses and one of them has the preceding two situations, the entire ALTER TABLE statement is not parsed reversely. 7. In the decoding scenario of directly connecting to a DN, only logs in the current DN are decoded. 8. Concurrency scenarios are not supported, for example, CREATE INDEX CONCURRENTLY and REINDEX CONCURRENTLY. 9. REINDEX DATABASE and REINDEX SYSTEM are not supported. 10. DDL statement decoding related to views and materialized views and DDL statement decoding related to triggers on views are not supported. 11. If PACKAGE objects are used in CREATE FUNCTION or CREATE PROCEDURE, logical decoding still supports decoding of CREATE/ALTER/DROP FUNCTION or CREATE/ALTER/DROP PROCEDURE. Distributed logical decoding does not

Date	Feature	Description	Specifications and Restrictions
			support PACKAGE. As a result, the execution may fail. 12. If IF NOT EXISTS exists in the statement for creating an object and the object already exists, the statement is not decoded. If IF EXISTS exists in the statement for deleting an object but the object does not exist, the statement is not decoded. 13. DDL statements related to node groups of logical clusters that use only partial shards cannot be decoded. 14. Only the commercial DDL syntaxes earlier than this version are supported. 15. The restrictions on the data retrieval capability are as follows: a. The current network and hardware environment must be normal. b. It is recommended that the size of a single tuple be less than or equal to 500 MB. If the size ranges from 500 MB to 1 GB, an error is reported. c. Data page replication is not supported for log decoding that does not fall into Xlogs. d. Logs before VACUUM FULL cannot be decoded.

Date	Feature	Description	Specifications and Restrictions
			e. The Xlog file before scale-out cannot be decoded. f. The CSN-based replication slot is not supported. 16. Optimized specifications and restrictions on logical decoding resource usage: Memory control is performed on the memory actually used by logical decoding services. Due to the design restrictions on current memory context, the memory context does not always return idle memory blocks to the OS. Delivery mode: Distributed

Date	Feature	Description	Specifications and Restrictions
	Table-level online DDL and online VACUUM FULL.	Based on the online DDL capability supported in earlier versions, this feature supports online modification of column types and precision, length reduction, length extension for the char types, online table-level tablespace movement, online addition of NOT NULL, CHECK, PRIMARY KEY, and UNIQUE constraints, and online VACUUM FULL.	Key specifications and restrictions: 1. When performing operations in this feature, reserve sufficient disk space. That is, the remaining space must be twice the total space of the tables to be operated and the accessories such as indexes on the tables. 2. This feature supports ordinary tables, but does not support database-level objects, index-level objects, partitioned tables, level-2 partitioned tables, segment-page tables, hash bucket tables, temporary tables, or unlogged tables. For objects that are not supported, traditional DDL operations are performed by default. 3. This feature only applies to the following online operations: online modification of column types and precision, length reduction, online table-level tablespace movement, online addition of NOT NULL, CHECK, PRIMARY KEY, and UNIQUE constraints, and online VACUUM FULL. For other DDL operations, the traditional syntaxes are used by default. 4. Long transactions may block online DDL operations. Therefore, you are advised not to

Date	Feature	Description	Specifications and Restrictions
			perform online DDL operations when long transactions exist. After operations in this feature are executed, DQL operations will be blocked and errors will be reported for concurrent DML operations. Ensure that retry upon error reporting is supported. For a service model with continuous retries, the thread pool may be full and cannot provide external services. Therefore, before using this feature, evaluate the tolerance of waiting for locks in the service model and check whether slow SQL statements exist to determine whether this feature can be used. 5. This feature cannot be executed in a transaction or a stored procedure. 6. During the execution of this feature, do not execute DDL operations concurrently. Otherwise, deadlock or errors may occur. In this case, reconnect to the client and try again. 7. After an upgrade from a version that does not support this feature to a version that supports this feature, this feature is not supported during the observation period, and the online DDL operations in this feature do not take

Date	Feature	Description	Specifications and Restrictions
			effect. By default, traditional DDL operations are executed, and a message is displayed indicating that this feature is not supported during the upgrade. 8. During the execution of this feature, a schema named **online$$ddl$$** *[Hash value of (table schema+table name)]* is generated. a. By default, the owner of the schema is the initial user. You are advised not to operate the schema during online DDL operations; otherwise, the online DDL operations may be terminated and an unknown error may occur. b. Users cannot create a schema with the prefix **online$$**. Otherwise, an error message is displayed. If a schema with the same name already exists, online DDL operations do not take effect. By default, traditional DDL operations are executed, and a message is displayed indicating that the current DDL statement cannot be online due to a schema name conflict. c. If a schema with the prefix **online$$** has

Date	Feature	Description	Specifications and Restrictions
			been created and the owner of the schema is the initial user, the GRANT and REVOKE operations cannot be performed on the schema after upgrade to a version that supports this feature. Do not perform CREATE, GRANT, or REVOKE operation on tables or sequences with the prefix **online_ddl** in the schema. Otherwise, an error message will be displayed. You are advised to rename the schema and try again. 9. If any operation of this feature (including online modification of column types and precision, length reduction, online table-level tablespace movement, and online VACUUM FULL) fails, significant table space bloat may occur. Then table may bloat significantly if NOT NULL, CHECK, or PRIMARY KEY constraints are added online no matter whether the operation fails or not. The bloat degree is proportional to the DDL statement execution period and the number of concurrent write operations. You are advised not to perform online DDL operations

Date	Feature	Description	Specifications and Restrictions
			when long transactions exist. 10. If a severe fault, such as FATAL, PANIC, database fault, or node disconnection, occurs during the execution of this feature, the created temporary schema and table appending status may remain. In this case, you need to perform manual cleanup and restoration by using online_ddl_cleanup(). 11. This feature involves contention between user services and online DDL resources. Therefore, this feature is not recommended in heavy-traffic scenarios. In scenarios where resources are limited, fine-grained resource control is recommended. This feature allows you to enable multi-thread execution for accelerating DDL operations. When a user enables or uses resource load management, for example, I/O resource management (for example, **io_limits** > **0** or **io_priority** set to **high**, **medium**, or **low**), if multi-thread execution is enabled for accelerating this feature, resource load management under the user or session will be ignored; in addition, a message is displayed,

Date	Feature	Description	Specifications and Restrictions
			indicating that the current resource load management is invalid and multi-thread acceleration is enabled. The multi-thread execution process of online DDL operations actually follows the resource load management of the initial user. 12. If the table to be operated by this feature has GSIs or the DDL statements contain the BY GLOBAL INDEX syntax, the online DDL operations of this feature do not take effect. Traditional DDL operations are performed by default. 13. This feature inherits the usage restrictions of CREATE TABLE LIKE. For details, see "SQL Reference > SQL Syntax > CREATE TABLE" in *Developer Guide*. 14. This feature does not support the M-compatible mode. 15. In this feature, the column type cannot be changed or converted to the SET data type. 16. In this feature, when the length of the char type is extended for columns, the underlying logic is not modified or rewritten. After online DDL operations are performed, the existing underlying storage of data still retains the old length, and the

Date	Feature	Description	Specifications and Restrictions
			underlying storage of new and modified data complies with the extended length. 17. When operations of this feature (including online modification of column types and precision, length reduction, online table-level tablespace movement, and online addition of NOT NULL, CHECK, PRIMARY KEY, and UNIQUE constraints) are performed, if any data violates the constraints, errors are reported and the error information is different from that of non-online DDL operations. 18. During online DDL operations such as modification of column types and precision, or length reduction, if the MODIFY [COLUMN] column_name data_type syntax is used, and the clause contains any of the CHARSET, COLLATE, FIRST, and AFTER keywords or column constraints, the online DDL operations do not take effect and traditional DDL operations are performed by default. 19. During online DDL operations such as modification of column types and precision, or length reduction, if the CHANGE [COLUMN]

Date	Feature	Description	Specifications and Restrictions
			column_name new_column_name data_type syntax is used, the online DDL operations do not take effect and traditional DDL operations are performed by default. 20. During online DDL operations such as online modification of column types and precision, length reduction, and online addition of NOT NULL, CHECK, PRIMARY KEY, and UNIQUE constraints, data will be verified to check whether the data meets the data type conversion, precision modification, length reduction, or constraint requirements. If the data does not meet the requirements, an error is reported during the DDL operations. Online DDL operations support concurrent DML operations. Concurrent DML operations may delete data that does not meet conversion requirements or violates constraints, but an error may be still reported for the DDL operations. For example, when online DDL operations are performed to scan a table and change the column type, if the data of tuple A in the table does not meet data conversion requirements, an error is reported immediately

Date	Feature	Description	Specifications and Restrictions
			when tuple A is scanned. If tuple A is deleted by a parallel transaction DELETE but tuple A is scanned by an online DDL operation, which is earlier than DELETE, tuple A is not considered as deleted. 21. Adding PRIMARY KEY and UNIQUE constraints online involves online index creation. If an exception (for example, manual cancellation, duplicate unique index key values, insufficient resources, thread startup failure, or lock timeout) occurs during online index creation, the index creation fails. 22. In this case, indexes and temporary tables that are not ready or not valid may remain, occupying system resources. For details about how to handle the exception, see "SQL Reference > SQL Syntax > CREATE INDEX" in *Developer Guide*. 23. Online VACUUM FULL does not support the FREEZE keyword. If both **FREEZE** and **ONLINE** are specified, **VACUUM FULL** is executed offline by default, and a message is displayed indicating that the VACUUM FULL statement cannot be executed online. 24. Other constraints of this feature are the same as

Date	Feature	Description	Specifications and Restrictions
			those of ALTER TABLE in *Developer Guide*. Delivery mode: Centralized and distributed
	Alarm is reported after consecutive login failures or when account is locked.	1. Two alarm reporting types are added. One alarm is reported after an account is locked. The alarm is automatically cleared after the account is unlocked. Another alarm is reported when the number of consecutive login failures exceeds the threshold. The threshold can be configured using the **failed_login_attempts** parameter. After the number of login failures is cleared, the alarm is automatically cleared. 2. The GUC parameter **enable_lock_account** is added to determine whether to lock an account when the number of consecutive login failures reaches the value of **failed_login_attempts**.	Key specifications and restrictions: N/A. For details about the alarm triggering conditions, see the product documentation. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
	ASP supports standby node and scheduled disk flushing, and idle transaction timeout is supported.	Based on the existing ASP and wait event capabilities, the following functions are added: 1. ASP supports standby nodes and scheduled disk flushing. Memory data on the standby node can be flushed to disks on the primary node. In addition, scheduled disk flushing parameters can be set to periodically flush memory data to disks. 2. Based on the timer mechanism, the idle transaction timeout parameter **idle_in_transaction_timeout** is added. The idle transaction timeout capability can be configured to automatically interrupt idle long transactions and solve the problems of backlogged Xlogs and unrecycled dead tuples caused by idle transactions. After an idle transaction times out, information about the timeout transaction is printed, including the start time, thread ID, session ID, and query statement.	Key specifications and restrictions: 1. By default, idle in transaction timeout is enabled for newly delivered instances for 5 minutes, and is disabled for upgraded instances. CNs, DNs and standby nodes are supported. Logger nodes are not supported. 2. Idle sessions exit upon timeout to prevent resource occupation by sessions that are not committed. 3. You can enable this function. Enabling this function does not affect database performance. 4. When the read service is enabled on the standby node, the ASP service is enabled on the standby node by default. You can manually disable the ASP service. 5. When the standby node function is enabled for ASP, the database performance is not affected. 6. You can customize time parameters (1 minute to 60 minutes) to flush ASP memory data to disks. Delivery mode: Centralized and distributed

Date	Feature	Description	Specifications and Restrictions
	Error codes are standardized.	1. Causes and actions are added for more than 3,500 error codes (ERROR level and above) that currently lack these details. 2. For 500 error codes that are frequently encountered by customers, such as syntax errors and parameter errors, the CAUSE and ACTION information is more precise. 3. The err_stack API is provided for printing the call stack. For error whose CAUSE is an internal error and the ACTION is to contact Huawei technical support, this API is called by default to print the call stack.	Key specifications and restrictions: N/A Delivery mode: Centralized and distributed

Table 2-3 What's new in version V2.0-8.102.0

Date	Feature	Description
2024-04	For O compatibility, aggregation-related syntax, table update based on views and subqueries, and comparison operators with spaces are supported.	1. Aggregate functions and ORDER BY can be used at the same time. 2. Nested aggregate functions are supported. 3. You can insert, update, and delete tables in subqueries and views. 4. Spaces are allowed in comparison operators such as "> =" and "< =".
	For O compatibility, system functions and system views are supported.	Based on the existing O compatibility, some system functions and system views are supported, including: 1. Two types of window function syntax are compatible. 2. The sys_guid function is compatible. 3. The compatibility of the time function is improved. The current time instead of the transaction start time can be obtained. 4. The data sorting function array_sort is supported. 5. The SQL syntax FROM DUAL is compatible. 6. The all_directories view is supplemented. 7. BULK COLLECT can be used together with EXECUTE. 8. Syntax related to distributed agg is supported. 9. The row construction syntax => is supported. 10.The EXTRACTVALUE system function is supported.

Date	Feature	Description
	For O compatibility, encoding exceptions and hybrid encoding of special characters are supported.	1. The GBK extended character set zhs16gbk is added. In addition, **client_encoding** can be set to **zhs16gbk** for JDBC and the database. 2. A transcoding function is provided for the new character set zhs16gbk. The conversion relationships are as follows: zhs16gbk to UTF8, UTF8 to zhs16gbk, and GB18030 to zhs16gbk (euro symbol). zhs16gbk (euro symbol) to GB18030. 3. The JDBC driver provides the transcoding function to solve the transcoding problem of the extended character set zhs16gbk. 4. To ensure that customer data containing special characters can be imported and exported to the database, the import and export tool supports the following functions in the case of the same encoding: ● Use **\copy to** to export a binary file. Transcoding and verification are not performed during the export. The header of the binary file contains the original encoding information. ● Use **gsloader** or **\copy from** to import the binary file. Transcoding and verification are not performed during the import. 5. Output is added to ensure that no error is reported during forcible encoding. The output is a placeholder.
	For O compatibility, stored procedures support synonyms, subtypes, dynamic anonymous blocks, and triggers, enhancing commercial capabilities.	In terms of O compatibility, the following contents are added: 1. Triggers support the CREATE OR REPLACE syntax and anonymous block syntax. 2. Anonymous blocks of dynamic statements are enhanced. 3. The stored procedure supports the syntax related to subtype. 4. Synonyms are enhanced.

Date	Feature	Description
	O compatibility supports cross-type integer comparison, bpchar fuzzy match, and system function matching policy optimization.	In O-compatible mode: 1. A cross-type index operator class is added for the integer type (int1/int2/int4/int8/int16/numeric). 2. An index operator class for fuzzy match of bpchar and bpchar is added. 3. The internal matching policy of system functions is optimized.
	For O compatibility, the XMLGEN, STATS, and DESCRIBE advanced packages are supported.	In O-compatible mode, some APIs are supported in the DBMS_XMLGEN, DBMS_STATS, and DBMS_DESCRIBE advanced packages.

Date	Feature	Description
	For M compatibility, commercial requirements such as data types and syntax functions are supported.	1. Data type compatibility: The year data type is supported and the width of integer(p) can be displayed. 2. DDL syntax compatibility: The syntax for deleting primary key and foreign key constraints of a table is supported. The syntax for changing table names is compatible. The syntax for adding partitions is compatible. Auto-increment columns support distributed deployment. The CREATE TABLE LIKE syntax is compatible. The CREATE TABLE syntax is compatible with the ENGINE=InnoDB syntax of MySQL databases. 3. DML syntax compatibility: The INSERT statement now supports the IGNORE keyword, which allows the database to ignore specific errors and continue execution instead of aborting the operation. The LOAD DATA functionality can be used to import data. 4. Operator and expression compatibility: The set names syntax is added to control the character set and collation used for LIKE matching. The source REGEXP pattern and regular expression matching operators are supported to determine whether the source string matches the pattern string. The matching rules are the same as those of the existing ~* operator in GaussDB and may be different in some scenarios. 5. Backquote is supported. The database object name referenced by backquote is case sensitive. 6. The utf8_bin, utf8_general_ci and utf8mb4_0900_ai_ci collations are supported. 7. The following system functions are added for character string processing: strcmp(), log10(), log2(), bin(), elt(), field(), insert(), locate(), make_set(), quote(), space(), and default().
	For M compatibility, new frameworks and protocols are compatible for commercial use.	The new M-compatible framework supports full compatibility with MySQL database syntax in the future, avoiding syntax isolation and forward compatibility such as syntax and keyword occupation in the old framework. The function operator behavior is the same as that of MySQL databases, and the MySQL database protocol is supported.

Date	Feature	Description
	For M compatibility, the existing syntax adapts to the new framework and supports commercial use.	The new M-compatible framework uses the hook mechanism to implement the compatibility function in independent extension and isolate the compatibility function from the GaussDB main process to avoid forward compatibility issues caused by intrusive modification. This feature synchronizes the existing 107 SQL commands to the new framework.
	The JDBC driver supports the streaming read capability.	The GaussDB JDBC driver supports the streaming read capability. In streaming read mode, JDBC does not generate OOM.
	The JDBC driver supports JDK 1.7 and the enhanced JDBC O&M capability.	1. JDBC memory release is optimized to prevent client memory overflow in abnormal connection scenarios. 2. The JDBC driver is used to proactively detect related indicator data on the client and database sides, providing indicator data for database problem demarcation. 3. JDBC adapts to JDK 1.7. 4. The **executeUpdate** command can be used to execute DQL statements and obtain query results.
	The commercial performance of the default configuration parameters is not lower than 1 million tpmC.	The performance of default GaussDB configuration parameters is optimized. The performance value of default GaussDB configuration parameters measured by the standard benchmark (TPC-C) is improved to no less than 1 million tpmC. The capability of locating performance issues is improved.

Date	Feature	Description
	Based on ADIO, the performance is improved by 20% in typical large-capacity scenarios.	In large-capacity scenarios, the AIO-DIO technology and doublewrite removal function are used to fully utilize I/O resources to improve database performance by more than 20%. In addition, online switching from the BIO mode to the ADIO mode is supported.
	The performance of large concurrent write transactions in centralized mode is improved by 50%.	1. The performance of basic write transactions in the case of a large number of concurrent requests is optimized, including insertion, update with indexes, and update without indexes. 2. The performance in scenarios where read operations are more than write operations and hotspot pages exist in the case of a large number of concurrent requests is optimized.
	The performance is optimized by 15% based on stored procedures in typical batch processing scenarios.	The noise floor of the stored procedure is optimized to support SQLBYPASS.
	Concurrent cursor query is supported, improving performance by more than 30% in typical scenarios.	Cursors can be concurrently queried to improve cursor usage efficiency and insert select parallel performance in Ustore.

Date	Feature	Description
	Based on the window function, the performance is improved by six times in typical page turning scenarios.	The projection column of a subquery contains a window function, and the parent query contains filter criteria for the window function. This feature allows the outer filter criteria to be pushed down to the inner subquery.
	For Codegen commercial use, expressions are heavily used in TPC-H calculation, improving typical query performance by 20%.	The commercial use capability of the Codegen is improved. The Codegen is enabled by default to solve the calculation performance problem of complex query expressions.
	Parallel scanning of predicate indexes is supported. In typical scenarios, the performance is 10% higher than that of PG16.	Parallel index scan with predicates (IndexScan and IndexOnlyscan) is supported to improve performance in typical scenarios.
	Local partitioned indexes can be created offline and concurrently between Astore partitions.	Inter-partition parallelism is supported. During local partitioned index creation, steps such as scanning, sorting, and B-tree insertion are performed in parallel. The overall performance (when partition data is evenly distributed) is better than that of the parallel creation solution in the current partition.

Date	Feature	Description
	SPM supports restoration of complex SQL statements.	Based on the plan management function supported by SPM, the following enhancements are made: 1. The outline supports the physical operator control of the SMP plan. 2. The outline supports the subplan that controls the pullup of any sublink. 3. Semi join can be controlled. 4. If more than 12 tables are joined to the SPM outline, the hint takes effect and the plan can be quickly restored.
	DR switchover stability achieves 99% in typical scenarios, ensuring service recovery within 5 minutes.	The internal implementation mechanism and performance are optimized based on typical DR scenarios, effectively improving the DR switchover performance and stability.
	Arterial detection model for first aid is first put into commercial use, which supports slow disk detection.	The arterial detection model is built to identify arterial subhealth problems and provide corresponding measures to improve database HA.
	Client service statements can be terminated with socketTimeout.	When the client is disconnected due to timeout, the GaussDB server can detect the disconnection in a timely manner and terminate the running service statements corresponding to the connection. This prevents session resource stacking and service loss caused by retry due to socket timeout on the service side.

Date	Feature	Description
	Automatic repair based on physical bad blocks: Pages on the standby node can be repaired from the primary node in seconds.	1. In the primary and standby deployment mode, if the target page (data page, index page, undo slot page, or undo page) of the standby node is physically damaged, that is, CRC mismatch occurs, the standby node can pull the target page from the primary node when reading the page during replay. If the page of the primary node is correct, the standby node can be overwritten to perform automatic recovery. 2. This ensures that replay on the standby node is not interrupted and services are unaware of the replay. Backup and restoration or full build is not required. 3. When the page CRC detects that the physical page is damaged, the primary node can be automatically triggered to repair the standby node. This function is enabled by default. 4. DFX is optimized, for example, log printing and page parsing after a page is damaged, and system functions for querying bad blocks.
	PITR modular decoupling and key scenario locating and demarcation are improved.	1. Barrier files can be repaired during startup. 2. Archive log clearing is moved from the kernel side to the gs_roach side, eliminating the impact of the log clearing logic on the gaussdb kernel. 3. Several system functions are added to query and operate archive-related files.
	Automatic list/range partitioning is supported for commercial use.	1. Automatic partitioning is supported. If the inserted data cannot match any existing partition, a new partition is automatically created. Automatic transaction commit is supported in centralized mode. 2. This behavior can be triggered when a sparsely partitioned index is created. The automatic partitioning capability can be enabled or disabled online.

Date	Feature	Description
	Row-store compression supports page-level LMT.	After advanced compression is enabled and an ILM policy is specified for a table, background scheduling is periodically started to scan all rows. After data is frozen, the current timestamp is used as the last modification time of frozen tuples to determine hot and cold data. There is a difference between the timestamp information and the actual last modification time of the tuples. To accurately indicate the LMT, the timestamp corresponding to the LSN of the page where the tuples are located is used to indicate the LMT of the tuples. The timestamp is used as the time basis for determining whether the tuples are cold or hot.
	Based on stored procedures, the global compilation memory usage is reduced by 30% in typical scenarios with a large number of concurrent requests.	In the scenario of a large number of concurrent requests, stored procedures occupy a large amount of memory. Therefore, some improper memory usage needs to be optimized, mainly including structure arrays related to the number of parameters or memory sharing. The memory usage mainly refers to the type description of variables in stored procedures to reduce the memory usage, the concurrent database scale-out capability is improved.
	In typical scenarios with 4 vCPUs and 16 GB of memory, the CPU noise floor of the CM component decreases by 2.75% and the memory usage decreases by 46%.	The CPU and memory of the CM component are optimized for small-scale deployment in typical scenarios with 4 vCPUs and 16 GB of memory.

Date	Feature	Description
	Ustore supports efficient storage of flexible fields.	Enhanced TOAST is a technology used to process oversized fields. It reduces redundant information in TOAST pointers to allow more oversized columns in a single table, and optimizes the mapping between the main table and off-line storage tables. You do not need to use pg_toast_index to store the relationship between main table data and off-line storage table data, reducing storage space. Enhanced TOAST enables split data to be self-linked, eliminating the dependency of the original chunk ID allocation process and greatly improving the write performance.
	Ustore supports large-scale commercial use of TOAST.	1. TOAST table indexes can be upgraded to UB-tree indexes by performing the reindex, reindex concurrently, vacuum full, merge and split operations. 2. gs_parse_page_bypath and pagehack can be used to parse special fields in the TOAST table and indexes. 3. Xlogs can be recorded by SELECT FOR SHARE or SELECT FOR UPDATE.
	TDE supports index encryption and RLS supports expression indexes.	1. TDE supports encrypted indexes and can convert non-encrypted tables to TDE tables. 2. Index scans can take effect when row-level security (RLS) is enabled for tables and predicates contain system functions or Like operators.
	Sensitive data discovery is put into commercial use for the first time, enhancing privacy protection, and providing high security capabilities.	The sensitive data discovery function is implemented by calling functions. By calling different functions, you can specify the scan object and sensitive data classifier to obtain sensitive data of different levels corresponding to the scan object.

Date	Feature	Description
	Tamper-proof Ustore is put into commercial use for the first time.	Ustore can use the tamper-proof ledger database function.
	ABO supports feedback and multi-table cardinality. In typical slow query scenarios, the performance is improved by five times. Cost adaptation is supported. In scenarios where operator selection is inaccurate, the performance is improved by one time.	The adaptive cost estimation function provides the cost estimation capability based on the usual mixed model (UMM) and cost parameter model. The load monitoring monitors model accuracy, implements fast and efficient load management and incremental model update, and ensures the estimation accuracy. The real-time and efficient predicate query feature helps identify the optimal cardinality estimation policy. This feature helps solve the problem that the cost estimation is distorted and the plan is not optimal when the data and execution environment change on the live network.
	An exact row number is displayed when a compilation error is reported.	The logic for calculating line numbers is adjusted to solve the problem that the line number of the function header is separated from that of the function body and line numbers are incorrectly calculated. In this way, the error line number can be obtained accurately.
	Hot patches can be installed for advanced packages.	This feature provides the capability of installing hot patches for advanced packages.

Date	Feature	Description
	Built-in flame graphs support quick performance analysis and locating.	1. Flame graph files can be generated after stacks are manually or automatically collected. 2. Stacks can be automatically and periodically collected and flame graph files can be generated.
	The duration for locating underlying storage exceptions is shortened from weeks to days to solve the problem of missing dirty pages.	The verification and DFX capabilities are added to check for missing dirty pages, improving the fault locating and demarcation efficiency when the underlying storage returns an incorrect version. The fault locating duration is shortened from weeks to days. 1. This feature can be used to solve the problem of missing dirty pages inside the database. No Xlog of an incorrect version is generated. 2. If the underlying storage system does not return the correct page version, you can use the new view of this feature to demarcate the fault.

Date	Feature	Description
	In typical service scenarios, read success on the standby node is 100% and the time for locating read problems on the standby node is shorten from weeks to days.	1. Ultimate RTO of read on standby: The mechanism in earlier versions is optimized so that query cancellation does not need to be triggered in some scenarios, ensuring 100% read success in the following scenarios: • After a large number of records are deleted or updated, VACUUM is performed. As a result, useless data pages at the end of the file are truncated. • Cancellation is caused by relmap update, such as reindex database. 2. The replication slots on the read-only node or cascaded standby node are automatically cleared. After the read-only node is switched over, the primary node automatically clears the replication slots. After the standby node is switched to the cascaded standby node, the replication slots of the original standby node on the primary node remain. As a result, Xlog recycling on the primary node is blocked and is triggered only when the size of Xlogs reaches the value specified by **max_size_for_xlog_prune** (256 GB). 3. Log information is supplemented to help locate read on standby data consistency faults, including snapshot information and key tuple information. GUC parameters are added to control the output information. Gray-box case monitoring is added for read on standby in DML, DDL, and fault scenarios.

Date	Feature	Description
	The troubleshooting and demarcation duration of typical communication module problems is shortened from weeks/days to hours/minutes.	1. This feature improves the capability of locating and demarcating typical communication problems of the GaussDB communication module on the live network and in the test and development environments. For example, the GaussDB server connection is hung or slow, no valid information about pooler functions or views is provided, and the time required for sending and receiving libcomm communication messages cannot be calculated. The following three points are involved: ● The communication link DFX capability of the GaussDB server to respond to connections is enhanced. ● The local and peer information is added to the communication views of the pooler communications library between CNs and between CNs and DNs. ● The libcomm communications library supports the collection of socket statistics of persistent connections between primary DNs in the system view, and supports the display of libcomm communication time when explain performance is used to query performance information related to distributed stream plans. 2. The preceding methods are used to enhance the capability of locating and demarcating typical problems of the GaussDB Kernel communication module. ● Quick fault locating and demarcation of the communication module: from post-event reproduction to in-event recording or troubleshooting. The demarcation duration is shortened from weeks/days to hours/minutes. ● Fault locating methods are enriched for the communication module. Three fault locating methods are added: key time points of the GaussDB server responding to communication links; libcomm persistent connection statistics, and explain performance. The libcomm communication duration in the distributed stream plan can be printed.
	Memory-overcommitted session printing is supported.	A threshold is provided. When the memory usage of a single session or SQL statement exceeds the threshold, detailed memory information (DFX information indicating that the memory usage of a single SQL statement exceeds the threshold) is printed.

Date	Feature	Description
	The DFX performance view supports refined db_time/ wait event duration statistics.	The wait event types of GaussDB modules are optimized to provide more comprehensive fault locating methods. 1. The **db_time** time model is refined, including the network time. The **db_time** parameter is modified to improve the capability of locating and demarcating network time more accurately. 2. Some wait events have nesting relationships. In earlier versions, nesting is not supported. As a result, outer events cannot be reported as wait events, the outer wait event time is lost, and the scenario where the outer wait event takes a long time cannot be located. After nesting is supported, outer events are reported, and the time consumption information is supplemented. 3. Wait events for heavy code blocks and large logic blocks are supplemented to improve the locating and demarcation capabilities of full SQL and ASP in wait events.
	The storage space usage of WDR snapshots is reduced by 40% in typical scenarios.	The WDR snapshot storage space occupation and snapshot space control methods are optimized.
	Astore supports commercial use of hash bucket-based online scale-out.	The online scale-out technology based on the hash bucket table (Astore) is supported. The segment-page database-level data sharding and dynamic log multi-stream technologies are used to implement the online cluster scale-out solution for physical file migration. Large-scale commercial use is supported.

Date	Feature	Description
	In terms of segment-page, Astore supports commercial use of hash bucket-based online scale-out.	1. Segment-page capability improvement: ● Segment-page ultimate RTO supports read on standby (Astore). ● Segment-page supports metadata page repair and file repair. ● Segment-page execution statistics are included in the WDR. ● The **pg_xlogdump** and **pagehack** functions are optimized in segment-page mode. ● Astore flashback (DROP/TRUNCATE) is supported in segment-page mode. ● The **enable_segment** parameter is supported. ● The distributed system supports segment-page system views and functions (including manual residual data clearing functions that support hash bucket). ● In segment-page mode, local temporary tables are stored independently. ● The segment-page ultimate RTO replay performance is optimized. 2. The undo segment-page DFX capability is enhanced. ● Segment-page supports undo page repair and file repair. ● The undo temporary/unlogged table function is enhanced. ● Undo tslot\rdata disconnection protection is supported. ● The run logs are enhanced for the undo segment-page key process. ● Segment-page system views and functions support undo segment-page files.

Date	Feature	Description
	Resolved issues	The following issues are resolved: ● The CN continuously experiences lightproxy core dumps. ● During the switchover process, the base port cannot be connected when the CM performs zombie detection. ● In the connection setup phase, socket timeout in connection does not take effect. ● The service logical decoding task is slow. ● The **dbe_schedule** advanced package ignores February 29 when starting the scheduled task in interval mode. The solution is time-consuming. ● After the specifications are scaled up, the Cgroup is not updated. As a result, the number of CPUs used by the kernel is limited in CPU resource management, and the performance is poor. ● Hints of slow SQL control rules do not take effect for virtual tables. ● In a DR cluster, when the database is normal, a DN exception alarm is reported.

Date	Feature	Description
	Security hardening	The following security vulnerabilities are fixed: CVE-2023-28321 CVE-2023-50471 CVE-2023-46219 CVE-2023-46218 CVE-2023-38546 CVE-2023-4807 CVE-2022-34038 CVE-2023-27533 CVE-2023-27534 CVE-2023-50472 CVE-2023-5363 CVE-2023-29469 CVE-2023-3817 CVE-2023-32082 CVE-2023-28484 CVE-2023-28320 CVE-2023-3446 CVE-2023-28322 CVE-2023-45322 CVE-2023-38545 CVE-2023-27536 CVE-2023-27535 CVE-2021-28235 CVE-2023-45853 CVE-2023-2975 CVE-2023-27538 CVE-2023-36054 CVE-2023-5678 CVE-2023-6237 CVE-2023-6129 CVE-2024-0727 CVE-2024-25062

2.3.2 V2.0-3.*x* Versions

This section describes the V2.0-3.*x* kernel version updates of GaussDB. For details about the version description, version mapping, and how to check the version number, see **DB Engine Versions**.

Table 2-4 What's new in version V2.0-3.226.0

Date	Feature	Description
2024-05	PITR modular decoupling	The startup of GaussDB processes is decoupled from the loading of the barrier file. Even if the barrier file contains errors, it will not affect the startup or availability of GaussDB.
	Improved DFX capability of the PITR module	PITR-related system views and system functions are added to provide convenient O&M methods for O&M personnel. The PITR O&M efficiency is improved. The information collection duration is shortened from hours or minutes to seconds.
	Automatic repair based on physical bad blocks (Pages on the standby node can be repaired from the primary node in seconds.)	When a standby DN detects a page CRC error, it obtains the correct page from the primary node and overwrites the error page at a proper time for repair.
	Enhanced rebuilding capability of the standby node	To accelerate the recovery of faulty standby nodes, it is beneficial to implement incremental builds between standby nodes for transmitting incremental logs and data. Additionally, creating slots during the build process ensures normal log recycling, preventing log accumulation. When log inconsistencies are significant, the primary node retains sufficient logs after the build to avoid prolonged local log replay and prevent repeated build triggers caused by log recycling.
	DR cluster RTO optimization	According to the current mechanism, if an instance in the DR cluster is disconnected, the connection is re-initiated only after all logs are replayed. The disconnection time is significantly affected by the replay speed and the volume of logs to be replayed. Under heavy load, the instances in the DR cluster may be disconnected for a long time and logs in the primary cluster may be accumulated. After the optimization, the CM launches a pre-parsing thread for disconnected instances to quickly scan the full logs of the instance. Then, the connection is initiated from the latest log, significantly reducing the recovery time for primary/standby synchronization.

Date	Feature	Description
	Logging of memory over-limit sessions	Building on the existing multi-level memory management mechanism, this feature adds the capability to log SQL statements when memory usage exceeds predefined limits. This enables the reproduction and analysis of memory over-limit issues, facilitating faster problem diagnosis and resolution. The main functions are as follows: ● When the memory usage of a single SQL statement exceeds the upper limit, an error is reported and the SQL statement details are logged. ● When the memory usage of a single session exceeds the upper limit, an error is reported and the SQL statement details are logged. ● A wait event is added and the current SQL statement is logged before data is flushed to disks when the value of **work_mem** exceeds the upper limit. ● After the global memory usage exceeds the upper limit, the top 10 sessions and threads that occupy the memory are logged. The corresponding thread types and SQL statements are logged as well.
	Logical decoding that adapts to ultimate RTO	The logical decoding function of the standby node needs to access the system catalogs of the historical version. When ultimate RTO is enabled, logical decoding for the standby node needs to be adapted to prevent page recycling and use correct snapshots to access system catalogs.
	Hot patch management	The list interface is connected to the management plane so that the management plane can query the patches that take effect on the kernel side.
	Full SQL data supported by standby nodes of distributed instances	Full SQL data and slow SQL statements can be recorded and checked on the standby DN in a distributed instance. Full SQL data and slow SQL statements can be checked on the standby DN or on the CN by using the **get_global_full/slow_sql_by_timestamp** function.
	Repairing corrupted pages	A manual repair operation is provided for corrupted pages.

Date	Feature	Description
	EXECUTE DIRECT ON supported for querying standby nodes in distributed instances	Currently, distributed instances support readable standby nodes, with standby nodes handling read workloads that require operational monitoring. However, the current O&M operations from the CN cannot directly access the standby DN to query views or functions. As a result, it is difficult to locate faults. Additionally, in the distributed environment, WDR-persisted global views cannot access data from the standby DN, and thus, there is a need to extend access to standby nodes. Therefore, it is necessary to enhance the **EXECUTE DIRECT ON** and **distqry_remote_function_result_handler** functionalities to support connections to one or more standby nodes in a distributed system for performing DFX queries, improving DFX capabilities.
	Enhanced fault locating and demarcating	The system resource check is added to the CM. If the resource usage exceeds the threshold, a log is recorded to facilitate subsequent fault locating.
	Proactive clearing of physical replication slots	Physical replication slots used in primary/standby instances and streaming replication-based disaster recovery solutions may have their logs recycled when the streaming replication connection is broken, as the logs are protected by **max_size_for_xlog_prune**. In such cases, the slots associated with streaming replication become inactive. Even if streaming replication is restored later, the removal of WAL segments can force a full rebuild of the standby node. To prevent this, proactive cleanup of these physical replication slots is performed when log recycling is triggered by reaching the **max_size_for_xlog_prune** limit.
	Multiple SQL throttling rules	In a database system, a certain type of SQL statements may be executed abnormally or the number of concurrent service requests increases sharply, which will occupy a large number of system resources or affect the execution of other workloads, or even the entire database system cannot respond to other service requests. To solve this problem, GaussDB implements SQL throttling to limit the number of concurrent SQL statements of a certain type from multiple dimensions.
	Managing logs of logical replication slots	WAL logs retained in logical replication slots are managed.

Date	Feature	Description
	Security hardening	The following security vulnerabilities are fixed: CVE-2024-2511 CVE-2024-1013 CVE-2024-25062 CVE-2024-0727 CVE-2023-52323 CVE-2023-6237 CVE-2023-46219 CVE-2023-50472 CVE-2023-50471

Table 2-5 What's new in version V2.0-3.223.0

Date	Feature	Description
2023-11	CM disk detection capability enhancement	The capability is enhanced in the following aspect: The CMA uses an independent module for disk detection so that the module can detect the disk status and report the status to the CMS for arbitration.

Date	Feature	Description
	Security hardening	The following security vulnerabilities are fixed: CVE-2023-46218 CVE-2023-5678 CVE-2023-5363 CVE-2023-45853 CVE-2023-45322 CVE-2023-38546 CVE-2023-38545 CVE-2023-4807 CVE-2022-34038 CVE-2023-36054 CVE-2023-3817 CVE-2023-3446 CVE-2023-2975 CVE-2023-28370 CVE-2023-2650 CVE-2023-28320 CVE-2023-28321 CVE-2023-28322 CVE-2023-32082 CVE-2023-1255 CVE-2023-29469 CVE-2023-28484 CVE-2021-28235 CVE-2023-27538 CVE-2023-27536 CVE-2023-27535 CVE-2023-27534 CVE-2023-0466 CVE-2023-0465 CVE-2023-0464 CVE-2023-27533 CVE-2022-4304 CVE-2022-4450 CVE-2023-0215 CVE-2023-0217 CVE-2023-0216 CVE-2022-4203

Date	Feature	Description
		CVE-2023-0401 CVE-2023-0286 CVE-2022-42898 CVE-2022-3996 CVE-2022-40303 CVE-2022-40304 CVE-2022-29824 CVE-2021-37750

Table 2-6 What's new in version V2.0-3.220.0

Date	Feature	Description
2023-08	Optimization on memory usage of stored procedures in global PL/SQL cache	The shared cache based on the stored procedure addresses the high memory usage identified by the stored procedure compatibility. The existing PL/SQL session-level cache is optimized in the following aspects: 1. Global PL/SQL Cache: All sessions share a PL/SQL compilation intermediate product, and their execution status is maintained independently. 2. Syntax node optimization: The data structure of syntax nodes is optimized to reduce the memory usage of a single compilation product. 3. Global PL/SQL cache is adapted for high-concurrency calls. ● The reference count (ref count) is accurate. ● The data structure splitting of compilation products is enhanced. ● The high-concurrency handling capability is enhanced for calling stored procedures. Specifications: 1. If the size of the PL/SQL compilation product cache was M, the size is now reduced to M/N (N indicates the number of long connections). 2. The specifications are equivalent to the existing GPC performance specifications. Restrictions: Applicable only in centralized mode.

Date	Feature	Description
	Processing of stored procedure parameters and recompilation upon expiration	1. Exceptions caused by long **Out** parameter values are handled. 2. Recompilation upon expiration in the SQL syntax is supported. • SQL statements included in a stored procedure must be checked. • Expressions included in a stored procedure must be checked. 3. The validity of values on the right-hand side of the "=" sign are checked. 4. GaussDB is compatible with the syntax that a plus sign (+) is used when a function with multiple input parameters or the input parameter in a stored procedure is wrapped in a new line. 5. The FFIC information is added when a core dump occurs in the stored procedure. 6. The set type capability is enhanced. Element length verification and element precision conversion are supported. The index length of the set type with indexes can be verified. Specifications and restrictions: Invalid objects can be recompiled using scripts. The **schema**/**compile_all** parameter is supported, but the **reuse_settings** parameter is not supported.
	Stored procedure cursors and historical slow SQL statements	1. The INSERT xxx SELECT xxx statement is supported. The SELECT target list expression contains both functions without parameters and functions without parameters as input parameters. (Both centralized and distributed instances are supported.) 2. If the WHERE condition in a SQL statement contains **col=func(***xxx***)**, the performance is the same as that of O databases. 3. Historical slow SQL statements in stored procedures can be viewed (available for centralized instances). Delivery mode: centralized and distributed GaussDB instances

Date	Feature	Description
	Single-replica DR for distributed instances	Single-replica DR is put into commercial use. The specifications and restrictions are as follows: 1. The primary cluster supports only the 1 primary + 2 standby deployment model. 2. The SLA of high availability is not promised. 3. After a DR switchover, the DR cluster cannot be upgraded to the 1 primary + 2 standby deployment model. 4. Only single-node, single-replica clusters that use the Quorum protocol can be used as DR clusters. Delivery mode: centralized and distributed GaussDB instances
	Type conversion, UNION operation, and package adaptation	1. The GaussDB type conversion capability is enhanced. 2. This solves the problem that data and indexes cannot be properly matched when equality matching is performed between the bpchar type (containing multiple spaces) and the text type. 3. This solves the problem that the sorting information is lost when the UNION operation is rewritten in the SQL statement. 4. This solves the problem that the number of lines in the package error message is incorrect. 5. This solves the problem that when an element of the array type exceeds the defined array length, no exception is thrown and only null is returned. Delivery mode: centralized GaussDB instances
	Support of GB18030	1. GB18030 level 3 is supported, including 87,887 Chinese characters. The new government procurement standard that requires the support of GB18030 was enforced on August 1, 2023. 2. GaussDB supports the GB18030-2022 character set. 3. The new national standard GB18030-2022 character set is supported.

Date	Feature	Description
	Streaming DR supported for Dorado-based dual-cluster deployment	The GaussDB dual-cluster (Dorado-based) deployment solution supports streaming DR. 1. Two Dorado-based clusters function as the primary cluster for remote streaming replication to support disaster recovery. 2. Two Dorado-based clusters function as the DR cluster for remote streaming replication to support disaster recovery. 3. In three-cluster mode, functions such as backup, node repair, node replacement, and version upgrade are supported. 4. In the Dorado-based dual-cluster DR switchover scenario, the kernel RTO is less than or equal to 2 minutes. 5. The scenario where the DR and upgrade processes overlap are supported. Delivery mode: centralized GaussDB instances
	Enhanced fault tolerance capability of gs_loader	1. Not all data is rolled back when gs_loader violates constraints. 2. If gs_loader contains incorrect data, not all data needs to be rolled back. The **rows** parameter is supported to specify the number of rows committed in a transaction. 3. If the imported data contains **\n**, a truncated character, or a character that cannot be found in the character set, the original fault tolerance mechanism does not correctly write the error data to the bad file or error table. This issue is rectified by the enhanced mechanism.
	Minor version upgrade supported in the upgrade to be committed scenario of small-scale deployment	In small-scale deployment, minor versions can be upgraded during the service observation period, and rollback by minor version is supported.
	System object privilege hardening for upgrade	During the upgrade process, users' system object privileges may be lost. To prevent this problem, this feature backs up the privileges before the upgrade and restores them afterwards, ensuring that users' system object privileges remain consistent before and after the upgrade.

Date	Feature	Description
	Security hardening	The following security vulnerabilities are fixed: CVE-2023-3817 CVE-2023-32001 CVE-2023-3446 CVE-2023-2975 CVE-2023-28370 CVE-2023-2650 CVE-2023-28320 CVE-2023-28321 CVE-2023-28322 CVE-2023-32082 CVE-2023-1255 CVE-2023-29469 CVE-2023-28484 CVE-2021-28235 CVE-2023-27536 CVE-2023-27538 CVE-2023-27535 CVE-2023-27534 CVE-2023-0466 CVE-2023-0465 CVE-2023-0464 CVE-2023-27533 CVE-2022-4304 CVE-2022-4450 CVE-2023-0215 CVE-2023-0217 CVE-2023-0216 CVE-2022-4203 CVE-2023-0401 CVE-2023-0286 CVE-2022-42898 CVE-2022-3996 CVE-2022-40303 CVE-2022-40304 CVE-2022-29824 CVE-2021-37750

Table 2-7 What's new in versions V2.0-3.201, V2.0-3.207, and V2.0-3.208

Date	Feature	Description
2023-04	START WITH recursion performance optimized	The performance of the following statements is optimized. • START WITH...CONNECT BY statements • START WITH...CONNECT BY statements with the **connect_by_isleaf** field
	Viewing running SQL statements in the current stored procedure	View SQL statements that are being executed in the current stored procedure.

Date	Feature	Description
	Session-level HA attributes	1. Session-level standby node connection detection and strong synchronization of session-level logs are supported. a. Session-level standby node connection detection: After this function is enabled, the system checks whether the standby node can be connected. If the standby node cannot be connected, the primary node reports an error. b. Strong synchronization of session-level logs: After this function is enabled, sessions between the primary and standby nodes are strongly synchronized. 2. Single-node mode: a. Parameters for session-level standby node connection detection can be set, but the function does not take effect. After data is written to the primary node, a success message is returned. b. Parameters for strong synchronization of session-level logs can be set, but the function does not take effect. After data is written to the primary node, a success message is returned. 3. A full build of the standby node is supported. a. During a full build (the standby node cannot be connected): Parameters for session-level standby node connection detection can be set, but the function does not take effect. After data is written to the primary node, a success message is returned. After the parameters for strong synchronization of session-level logs are configured, the function takes effect. After **most_available_sync** is enabled and **keep_sync_window** is configured, new sessions for strong synchronization enter the maximum availability mode if the disconnection duration of the standby node exceeds the value of **keep_sync_window**. b. When the standby node is restoring the logs generated during the full build: Parameters for session-level standby node connection detection can be set, and the function can be enabled. Parameters for forced synchronization of session-level logs can be set, and the function can be enabled. 4. When the network is disconnected, the detection timeliness depends on parameters **tcp_user_timeout** and **wal_sender_timeout**. When

Date	Feature	Description
		the primary and standby nodes are disconnected (including scenarios where the wal receiver process on the standby node exits normally or crashes), the connection status of the standby node is updated synchronously. Specifications: 1. One primary node and one standby node are deployed. The data in the primary node is asynchronously synchronized to the standby node. 2. If the standby node cannot be connected, write request errors are reported during and the service needs to be retried.
	CURRENT OF CURSOR	The CURRENT OF CURSOR syntax is supported. Restrictions for CURSOR statements are as follows: 1. Only simple queries of a single table in the **SELECT** statement are supported. a. Subqueries are not supported. b. LIMIT/OFFSET is not supported. c. START WITH is not supported. d. WITH statements are not supported. 2. Multiple tables are not supported. 3. Partitioned tables are not supported. 4. Column store is not supported. 5. Hash bucket tables are not supported. 6. ORDER BY is not supported. Restrictions for UPDATE statements are as follows: 1. Only a single table can be updated. 2. UPDATE statements can be used only in stored procedures. 3. Multiple tables are not supported. 4. Multiple WHERE clauses are not supported. 5. WITH/USING and ORDER BY are not supported.
	Read-Only detection capability of CM disks enhanced	The capability is enhanced in the following aspects: 1. The read-only status is obtained from the database to ensure accuracy. 2. Read-only quorum arbitrates only the nodes that exceed the threshold. Other nodes are not affected. 3. When the primary node is read-only, a primary/standby switchover is automatically performed. An available standby node is promoted to primary to ensure that the instance can work properly.

Date	Feature	Description
	O&M database connection optimized	Peer authentication is not used. In addition, concurrency conflicts are resolved and O&M performance is improved.
	Unified name GaussDB	The name is changed to GaussDB.
	Automatic start of redistribution after lock timeout during scale-out	The automatic retry mechanism is provided for scale-out. If the lock times out for multiple times, the mechanism automatically exits. After a period of time, scale-out is retried to switch tables during off-peak hours, reducing the impact on user services.
	Software component lifecycle matching the product lifecycle	The software component lifecycle can match the product lifecycle. 1. Update underlying software components (such as open-source, third-party, and self-developed software). 2. Fix security vulnerabilities promptly within the product version lifecycle.
	Single-replica deployment enhanced (for non-production environments)	1. Monitoring metrics are supported. 2. An alarm can be triggered when a node is set to read-only due to the fully occupied disk. a. O&M interface reporting adapts to the single-node mode. b. CM makes the read-only alarm information persistent on DCC.

Date	Feature	Description
	Optimizer adaptation and robustness improved	The operator preference provides assurance if the cost model fails, and multi-version plan selection capability well as intelligent statistics are enhanced, improving the adaptability and robustness of the optimizer. 1. In the cost convergence scenario, the operator preference mechanism is provided to prevent the optimizer from selecting a poor plan that is incorrectly estimated. In this way, the plan performance is improved by 50% when the statistics are invalid. 2. The robustness of intelligent statistics is enhanced, and statistics of multiple columns are integrated to create a statistics model, improving the evaluation performance of high-frequency values by 50%. The overall performance is improved by 20% to 30%. 3. Currently, many slow queries can be resolved only by using service hints. Multi-version plans are required to support more scenarios and provide an autonomous escape mechanism to automatically resolve more than 10% slow queries in specific scenarios. The performance deterioration in the TPC-C test model is less than 5%. This is only available for centralized instances.
	Data restoration in extreme scenarios	In extreme scenarios, users can restore their data by outputting disk data files which need to be identified by the COPY command. Then, data can be imported to a new DB instance as needed. Generally, data restoration is based on existing redundancy technologies. This solution is used only in the following scenarios: 1. Node recovery and primary/standby build 2. Disaster recovery 3. Backup and restoration 4. Hot patch 5. Other cases where the cluster needs to meet the expected requirements Constraints: 1. Only flushed data is parsed. 2. Data consistency cannot be ensured during offline data parsing. 3. Xlog replay cannot be guaranteed. 4. The database is offline. 5. Only heap table data in Astore and Ustore can be parsed.

Date	Feature	Description
	Undo files of Ustore managed by the segment-page mechanism	The undo files of Ustore are managed by the segment-page mechanism. This prevents file handle leakage when a large number of 1 MB undo files are generated.

Date	Feature	Description
	Global secondary indexes for distributed instances	Global secondary indexes for distributed instances 1. Global unique indexes can be created on non-distribution keys. 2. The performance of point queries on non-base table distribution keys is three times that of common index queries. 1. Astore is supported. Constraints: ● Common UB-tree indexes can be created. Local indexes and expression indexes are not supported. ● GSIs whose distribution is the same as that of base tables cannot be created. ● The GSI distribution key in a base table cannot be updated. ● CREATE GSI CONCURRENTLY and PARALLEL are not supported. ● Hash-based GSIs can be created for row-store Astore tables and partitioned tables whose base tables are hash-based. Base tables cannot be replicated tables, bucketed tables, segment-page tables, list/range-based tables, Ustore tables, or column-store tables. GSIs only support hash-based distribution. ● GSI can only be used in distributed instances. ● If **_new$$** or **_NEW$$** is added to a column name of a base table or an identifier (including ctid, xc_node_hash, xmin, xmax or tableoid), a GSI fails to be created for the base table whose column names are repeated. ● The GSI can be created only in GTM-Lite mode. ● IUD returning is not supported for base tables with GSIs. ● TABLE ACCESS BY INDEX ROWID is not supported. ● VACUUM FULL, COPY, GDS, and REINDEX TABLE/DATABASE are not supported. All GSIs will be skipped during execution. ● The following models are not supported: UPSERT, MERGE INTO, HOT UPDATE, max_datanode_for_plan (in PBE mode), logical replication, cluster, and For Update/share/nowait/skip Locked ● Compatibility syntax such as Start With is not supported. ● A maximum of 27 columns (including index keys and distribution keys) are supported.

Date	Feature	Description
		• Other constraints are the same as those of hash distribution, UBTree, LP/FQS, and common indexes.
	CURRENT OF CURSOR	The CURRENT OF CURSOR syntax supports ORDER BY.
	SQL features supported by Ustore in centralized instances	SQL features supported by Ustore in centralized instances: 1. Global temporary tables 2. Column constraints 3. Parallel query 4. Statistics enhancement: Ustore index active page estimation and cost adaptation.
	Out parameter overloading and capability enhancement for stored procedures	1. SQL PATCH scope: DML within a stored procedure 2. Out function overloading scope: Functions with Out parameters can be created and invoked. Only Out parameters in direct value assignment can be returned. For example, in **a = func(b)**, **b** is an out parameter and is returned.
	Overload escape in distributed scenarios	Overload escape is supported in distributed scenarios. 1. After a certain proportion of slow SQL statements in a thread pool are executed, the CANCEL operation is triggered. 2. Control or circuit breaker measures are taken for SQL statements that occupy a large number of resources to prevent normal SQL statements from being executed due to slow SQL statements.

Date	Feature	Description
	SQL PATCH in distributed instances	SQL PATCH in distributed instances Constraints: 1. SQL PATCH operations cannot be synchronized between CNs. 2. SQL PATCH cannot be used for DNs. 3. If the lightproxy plan delivered by a CN to a single node triggers the execution of SQL Patch, the plan is degraded to the FQS plan delivered by the CN to a single node, and SQL PATCH takes effect. 4. Similar to HINT, SQL PATCH affects the capability and specifications of distributed plans. 5. SQL PATCH cannot be used for DNs. For distributed instances, if a stored procedure is pushed down, the DML in the stored procedure generates plans on DNs and is not affected by SQL PATCH.
	Ultimate RTO when standby nodes can process read requests	Based on the ultimate RTO technology, standby nodes can process read requests. Constraints: 1. The DDL statement replay conflicts with the read-only mode of standby nodes. 2. When standby nodes process read requests, they consume CPU, memory, and disk space. The memory and disk space are configured based on GUC parameters. If used memory and disk space exceed the configured GUC values, the system will forcibly reclaim the overused resources.

Date	Feature	Description
	Syntax extension and cost model optimized	Syntax extension and cost model optimized 1. Auto analyze: a. Auto analyze is triggered when a large amount of data is inserted and updated in a short period of time to quickly process recycle task tables. b. By default, the sampling ratio and histogram precision are automatically increased based on the table size to improve the analysis accuracy of large tables. c. The adaptive estimation algorithm is provided to improve the NDV estimation accuracy in typical scenarios. 2. Hints can be used: a. In parameterized paths. b. In Bitmap Scan operations. c. Across query blocks. 1. The Plan Trace function can display the complete process of generating a query plan. This is only available for centralized instances.
	Index creation and rebuild online	[Requirement description] 1. Indexes (including common indexes, global indexes, and local indexes) can be concurrently created and rebuild in Astore. 2. Indexes (including common indexes, global indexes, and local indexes) can be concurrently created and rebuild in Ustore. Specifications: ● The online DML performance deteriorates by no more than 10%. ● Online index fields cannot be added, deleted, or modified. ● Only common tables and level-1 partitioned tables are supported.
	DFX and other features enhanced for Ustore in centralized instances	Centralized instances support the following Ustore features after related whitelist is enabled: 1. DFX capabilities are supplemented, and auxiliary verification for DML operations and REDO/UNDO functions is enhanced. 2. VACUUM FULL is supported. 3. UNDOMETA reduces the frequency and fields 4. FSC structure is embedded and its performance is optimized.

Date	Feature	Description
	Basic operator performance improved	The performance of basic operators is improved. 1. SRF 2. Expression framework flattened 3. AGG optimized 4. INNER UNIQUE optimized 5. Expression operator optimized 6. Parser optimized 7. Printtup optimized 8. Communication performance optimized 9. Index scanning operators optimized 10.Predicates optimized 11.Subquery optimized 12.Noise floor elimination for basic operators 13.Indexes optimized Performance improvement of basic operators: The end-to-end TPC-H performance is improved by 20%, and the TPC-H performance is improved by 40% (including the full table scan operators, expression operators, and aggregation operators).

Date	Feature	Description
	SQL M* syntax compatibility	GaussDB is compatible with some M* features and: 1. Is compatible with RANGE, HASH, and LIST partition syntax of MySQL databases. 2. Allows users to add columns before the first column of a table or after a specified column. 3. Adapts to the MySQL database syntax for modifying column names or definitions. 4. Allows users to configure common character set and collation for schemas, tables, and columns. 5. Adopts to the modified CREATE, ALTER, and DROP EVENT syntax. 6. Is compatible with SELECT INTO. 7. Uses syntax tree of SET TRANSACTION to set the isolation level and read/write mode. The following M* syntax is supported: 1. Centralized instances support the GROUP_CONCAT function. 2. User-defined variables in a session can be set through SET statements. 3. SET is enhanced in global variable configuration. 4. Prefix indexes are supported. 5. Delimiters are supported. 6. DELETE statements can be used for deleting multiple tables.
	Fully-encrypted database framework with software and hardware integrated	Fully-encrypted database framework with software and hardware integrated: Based on the virtual TEE, the fully-encrypted framework can encrypt memory and ensure security isolation. 1. Secure key transmission channel is supported. 2. The fully-encrypted execution framework integrates software and hardware and supports the virtual TEE execution. 3. Server-side encryption operator support range query and sorting operations. A whitelist is used to restrict syntax operations. 4. The JDBC client driver is supported. This is only available for centralized instances.

Date	Feature	Description
	Logical decoding capability enhanced	1. Logical replication supports distributed strong consistency. a. Sequence preserving is ensured when distributed transactions are committed. b. Sequence preserving is ensured in the same transaction on a single DN. c. Sequence preserving is not supported between different DNs of the same transaction. Constraints: This feature is only suitable for GTM-Lite mode. 2. Logical decoding performance views are provided. a. Log sending speed (MB/s). b. Log read speed (MB/s). c. Log parsing speed (MB/s). Constraints: For parallel decoding, only the parallel decoding performance views of the primary nodes for centralized instances are provided. 3. Distributed logical decoding is supported. DRS can directly connect to each standby DN in the service cluster to perform logical decoding that is replayed in the cluster scale-out. 4. The following two monitoring metrics are added: a. Maximum number of transaction lines b. Number of large transactions (with more than 4096 lines) 5. A heartbeat message is added so that external tools can accurately calculate the synchronization latency. a. The latest WAL timestamp is provided for external tools to calculate the latency. b. Parallel decoding provides functions to query the latest WAL timestamp. Serial and parallel decoding provide heartbeat messages. 6. The logical decoding performance can reach 300 MB/s, which optimizes: a. The performance of saving data to the database is optimized. b. Single-slot decoding performance of primary nodes c. Concurrent replay performance of standby nodes d. Single-slot decoding performance of standby nodes

Date	Feature	Description
	Partitioned table functions and performance optimized	1. LIST and RANGE multi-column partitioning are supported. a. The number of LIST partition keys can be increased from 1 to 16. b. The number of RANGE partition keys can be increased from 4 to 16. 1. Range/List/Hash (RLH) partitions: a. RLH partitioned tables mean LIST, RANGE, and HASH tables. b. LIST/RANGE tables support node group distribution. 1. The partition syntax is optimized as follows: a. LIST and level-2 partitions support SPLIT and MERGE functions. b. LIST, HASH, and level-2 partitions support CREATE TABLE ... LIKE statements. Constraints: DDL and DML concurrency across partitions is not supported in distributed instances. 1. Continuous improvement of partitioned table performance: a. The partitioned table performance in services is improved by 30%. b. Partition implicit type and function pruning are optimized. This is only available for centralized instances.

Date	Feature	Description
	SQL O* syntax compatibility improved	1. There are new 167 compatibility items for system views, system functions, and advanced packages. a. System functions: 39 b. System views: 124 c. Advanced package: 4 2. Distributed instances support the Go driver. 3. The dblink is supported. 4. PIVOT/UNPIVOT is supported. 5. The XML data type and related interfaces are supported. 6. Distributed instances support packages. 7. Centralized and distributed instances support ROWNUM. 8. The following 11 XMLDOM interfaces are supported: a. XMLDOM.GETFIRSTCHILD b. XMLDOM.ITEM c. XMLDOM.DOMNODE d. XMLDOM.MAKEELEMENT e. XMLDOM.GETCHILDNODES f. XMLDOM.DOMNODELIST g. XMLDOM.GETNODEVALUE h. XMLDOM.GETLENGTH i. XMLDOM.GETNODENAME j. XMLDOM.GETELEMENTSBYTAGNAME k. XMLDOM.GETTAGNAME l. XMLPARSER.NEWPARSER 9. The functions of tableof are enhanced. a. The array can be nested with tableof. b. Distributed instances support tableof.

Date	Feature	Description
	WDR optimization and DFX support	1. WDR snapshots: a. A statement continues to be executed after it fails to be executed due to timeout, which does not affect the overall execution of a WDR snapshot. b. A WDR snapshot fails during the vacuum process. c. The retention period of WDR snapshots is not 8 days. 2. WDR reports: a. The WDR report generation speed is further accelerated. b. The WDR report format is optimized. c. WDR objects are sorted based on multiple dimensions (such as dead tuples, live tuples, vacuum, analyze, scanned rows). 3. There is an interface for querying GUC parameters such as **search_path** configured in a session. 4. WDR reports can be read by the standby nodes. 5. L0 FULL SQL plans can be viewed.
	Optimization on memory usage of stored procedures in global PL/SQL cache	The shared cache based on the stored procedure addresses the high memory usage identified by the stored procedure compatibility. The existing PL/SQL session-level cache is optimized. 1. Global PL/SQL Cache: All sessions share PL/SQL compilation intermediate products, and their execution status is maintained independently. 2. Syntax node optimization: The data structure of syntax nodes is optimized to reduce the memory usage of a single compilation product. This is only available for centralized instances.
	ODBC Gplan reuse	ODBC supports the JDBC Statement Pool function and Gplan reuse to improve SQL performance.

Date	Feature	Description
	Go drivers for fully-encrypted databases	A fully-encrypted database can: 1. Insert 60,000 rows per second. 2. Query 90,000 rows per second. 3. Update temporary tables. 4. Support HIS key management. 5. Support server-side encryption and decryption functions. 6. Update CMK keys. 7. Support load balancing for JDBC. Go language drivers are available for fully-encrypted databases.
	Minimum RTO due to improved primary/ standby synchronization	The restriction that a standby node can reconnect to the primary node only after the replay is complete is removed, and the RTO performance is optimized.
	User-level audit settings	The audit function is enhanced as follows: 1. User-level audit is supported. The GUC parameter **full_audit_users** is added to set the list of audited users. Audit logs are recorded for all SQL statements executed by users in the list. Administrators' operations can be fully audited to audit administrators' access to user data. 2. The audit operation can be configured based on client information. The GUC parameter **no_audit_client** is added to configure the list of clients that do not need to be audited. Audit logs are not recorded for specified client apps and IP addresses (such as cm_agent@127.0.0.1). 3. Audit logs are recorded for high-risk system function calls.
	Strong security authentication supported by Roach	Roach supports SSL and Kerberos security certificate authentication.

Date	Feature	Description
	Security hardening	The following security vulnerabilities are fixed: CVE-2023-30608 CVE-2023-29469 CVE-2023-28484 CVE-2021-28235 CVE-2023-27538 CVE-2023-27536 CVE-2023-27535 CVE-2023-27534 CVE-2023-0466 CVE-2023-0465 CVE-2023-0464 CVE-2023-27533 CVE-2023-23931 CVE-2023-23915 CVE-2023-23914 CVE-2023-23916 CVE-2022-4304 CVE-2022-4450 CVE-2023-0215 CVE-2023-0286 CVE-2022-42898 CVE-2022-43552 CVE-2022-43551 CVE-2022-40303 CVE-2022-40304 CVE-2022-32221 CVE-2022-42916 CVE-2022-42915 CVE-2022-1941 CVE-2022-35252 CVE-2022-37434 CVE-2022-32205 CVE-2022-32208 CVE-2022-32206 CVE-2022-2097 CVE-2022-32207 CVE-2022-2068

Date	Feature	Description
		CVE-2022-27782 CVE-2022-27781 CVE-2021-22570 CVE-2022-29824 CVE-2022-1292 CVE-2022-27775 CVE-2022-27774 CVE-2022-27776 CVE-2022-22576 CVE-2021-3520 CVE-2021-37750

Table 2-8 What's new in version V2.0-3.103 (whitelist-controlled version)

Date	Feature	Description
2023.3	WDR optimization	Sort top SQL statements by the average duration of a single SQL statement execution. Full SQL statements can be sorted by **avg** based on the top SQL statements in the current WDR report.
	Online specifications change	Change instance specifications online with no downtime. DNs are rebooted in rolling mode, without the need to reboot the instance. This mitigates the impact on services.
	Full SQL tracing on standby nodes and statement-level wait events	1. Support full SQL tracing on standby nodes. 2. Measure time consumption statistics on statement-level wait events for full SQL tracing. Constraints: 1. Full SQL tracing on standby nodes is available only for centralized instances. 2. Time consumption statistics of wait events are measured for L0 full SQL tracing. The performance loss of wait events is less than 3%.
	Domain name configuration for gsql and gsloader	Domain name configuration for gsql and gsloader ensures that multiple domain names can be accessed in HA mode. 1. Multiple domain names can be configured for distributed instances. 2. Centralized instances support leader election.

Date	Feature	Description
	Automatic creation of multi-column statistics	Advantages: 1. Solve the problem of composite index cost calculation. 2. Enhance multi-column statistics. 3. Solve the problem that an incorrect index is selected in typical scenarios.
	Backups created by standby nodes	Move some backup operations from primary nodes to standby nodes to reduce the usage of resources, such as the CPU and disk I/O, of primary nodes.
	SMP supported by global temporary tables	Global temporary tables support SMP.
	Backup and restoration for a single-node instance	Backup and restoration for a single-node instance is supported. This feature meets the requirements of R&D and joint commissioning scenarios. It is used only for SQL development and function joint commissioning. **NOTICE** To use this feature, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console. Constraints: 1. This is only available for centralized instances. 2. Service production environments are not supported. 3. Availability, reliability, and DR (the RTO cannot be ensured) are not committed. 4. O&M functions are not supported, including but not limited to node replacement, scale-out, upgrade, and specification change. 5. Fault injection tests and special tests on performance, pressure, and long-term stability are not supported. 6. There is no requirement on performance or impact on service performance.

Date	Feature	Description
	SQL compatibility enhancement	SQL compatibility is enhanced in the following aspects: 1. Support the following advanced packages: a. UTL_FILE.FILE_TYPE b. DBMS_SQL.VARIABLE_VALUE c. UTL_FILE.FGETATTR d. DBMS_LOB.READ e. DBE_LOB f. DBE_SQL g. DBE_FILE h. DBE_RAW i. DBE_UTILITY 2. Cancel the limitation on concurrent sessions in an autonomous transaction. 3. Check the compilation integrity in stored procedures. a. Check the compilation integrity. b. View the dependency of compiled objects. Invalid objects can be recompiled. 4. Support the following custom types: a. Array b. Tableof c. TableofIndex d. Combination of the preceding three types 5. Support the following 33 system functions: TO_CHAR, SYSDATE, TO_NUMBER, UPPER, CHR, SYS_CONTEXT, TO_DATE, INSTRB, LPAD, ROUND, CAST, TRIM, SUM, ADD_MONTHS, LAST_DAY, MOD, LOWER, RTRIM, MONTHS_BETWEEN, TRUNC, RPAD, LEAST, GREATEST, REGEXP_SUBSTR, LTRIM, TRANSLATE, TREAT, EMPTY_CLOB, USER, REGEXP_REPLACE, AVG, NVL2, REGEXP_COUNT 6. Support the following system views: ALL_ARGUMENTS, ALL_IND_PARTITIONS, ALL_IND_SUBPARTITIONS, ALL_INDEXES, ALL_OBJECTS, ALL_SOURCE, ALL_TAB_COLUMNS, ALL_TAB_PARTITIONS, ALL_TABLES, DBA_ARGUMENTS, DBA_CONS_COLUMNS, DBA_CONSTRAINTS, DBA_DEPENDENCIES, DBA_DIRECTORIES, DBA_HIST_SNAPSHOT, DBA_HIST_SQL_PLAN, DBA_HIST_SQLSTAT, DBA_HIST_SQLTEXT, DBA_IND_COLUMNS, DBA_IND_EXPRESSIONS, DBA_IND_PARTITIONS, DBA_IND_SUBPARTITIONS, DBA_OBJECTS, DBA_PROCEDURES, DBA_SCHEDULER_JOBS,

Date	Feature	Description
		DBA_SCHEDULER_PROGRAMS, DBA_SCHEDULER_RUNNING_JOBS, DBA_SEGMENTS, DBA_SOURCE, DBA_TAB_COLS, DBA_TAB_COLUMNS, DBA_TAB_PARTITIONS, DBA_TABLES, DBA_TRIGGERS, DBA_TYPE_ATTRS, GV$INSTANCE, GV$SESSION, PLAN_TABLE, USER_JOBS, USER_OBJECTS, USER_PROCEDURES, USER_TABLES, V$INSTANCE, V$MYSTAT, V $SESSION, ALL_DEPENDENCIES 7. Enhance output parameters of stored procedures and functions: a. Functions can return the record type. b. The stored procedure that contains the default value and out parameter can be invoked. 8. Improve the performance in stored procedures. a. Large loop scenarios of stored procedures, for example, **for i in 1 .. 100000 loop** b. Autonomous transactions (10 packages, each containing 10 variables, and 100,000 calls) 9. Commit+exception performance optimization: Optimize performance bottlenecks in the commit +exception scenarios and optimize the resource release related to EXCEPTION

Date	Feature	Description
	Readable standby nodes in a distributed instance	Offload read requests from primary nodes to standby nodes in a distributed instance. Constraints: ● Weak read consistency is provided. However, only read consistency is guaranteed only for incremental snapshots within a session, not for those among sessions. ● Session-level parameters are used to distinguish workloads on primary nodes from read workloads on standby nodes. If a standby node is faulty, its workloads cannot be switched to the primary node. ● In abnormal scenarios such as primary/standby switchovers and standby node faults, an error will be reported for the ongoing read service on the concerned standby node, and a service retry is required. ● After the resumption of a standby node or after adding replicas, an error will be reported for the ongoing read service on the concerned standby node, and a service retry is required. ● If no proper standby node can be selected, an error is reported and a service retry is required. ● After standby nodes process read requests, the RTO of the standby nodes is affected. If the RTO exceeds the specified threshold, traffic limiting or circuit breaker needs to be triggered after an error is reported on the service side. ● In DDL data replay scenarios, an API is provided for applications to check whether DDL operations on the standby node are consistent with those on the primary node. If consistency is confirmed, standby nodes can process read requests. Otherwise, a circuit breaker is triggered.
	Online index creation and ultimate RTO supported by Ustore-based centralized instances	The Ustore-based centralized instances support online index creation and ultimate RTO. 1. Indexes can be created online without interrupting services. 2. An ultimate RTO improves the performance of primary/standby instances (The standby nodes in instances cannot process read requests).

Date	Feature	Description
	Upsert and stored procedures supported by GPC	GPC supports upsert and stored procedures. 1. In the pgxc framework, upsert can be executed through gplan. 2. Stored procedures, functions, and packages are supported.
	Performance improvement of basic operators	The performance of basic operators is improved. 1. The seqscan and PI operators are optimized. 2. The cost model is optimized and adjusted.
	Codegen supported by row-store expressions	The Codegen capability of row-store tables is enhanced: Codegen is enabled by default. Users can query whether Codegen is enabled the specified number of rows. By default, all queries are not split.
	Optimizer cost model optimization	The following basic capabilities are optimized: 1. cplan/gplan selection 2. coalesce selectivity estimation 3. nestloop/merge join in some scenarios 4. semi/anti join cost This is only available for centralized instances.
	Optimization on concurrent partitioned table query	Optimize concurrent query on partitioned tables in the following aspects: ● DML/DDL concurrency across partitions is supported. ● Static pruning and dynamic pruning are supported for expressions in partitioned tables. ● Partitioned tables can be used as parameterized paths of internal tables. ● MergeSort query plans are supported.
	MySQL database syntax compatibility improved	GaussDB is compatible with common syntax and APIs of MySQL databases. DELETE and UPDATE statements support the ORDER BY and LIMIT functions.
	Multiple IP addresses for Python drivers	Support multiple IP addresses for Python drivers, so that databases can be properly connected after a primary/standby switchover.

Date	Feature	Description
	PITR enhancement	Enhance PITR capabilities in the following aspects: • Performance deterioration caused by PITR backups is reduced for two-phase distributed transactions in distributed GTM-Lite mode. • PITR backups are not affected if the external storage device is faulty.
	Embedded C preprocessor	Embedded C preprocessor is supported.
	Overload escape in concurrency scenarios	After a certain proportion of slow SQL statements in a thread pool are executed, the CANCEL operation is triggered.
	Data page restoration	Data restoration is enhanced in the following aspects: 1. Data files on the standby node can be restored to prevent file deletion or damage. 2. Damaged pages can be automatically repaired during the creation of backup. There is no impact on the backup.
	GCM encryption and client sorting for fully-encrypted databases	Fully-encrypted databases support GCM encryption and client sorting. 1. The sorting operation is implemented based on clients. 2. GCM encryption and bit transmission are supported. Constraints: Only gsql and JDBC clients are supported.
	SHA256 and SSL that uses SM series cryptographic algorithms	1. The data channel between the client and server supports SSL-encrypted transmission using SM series cryptographic algorithms. Constraints: Only gsql clients are supported. 2. SHA256 encrypts and saves sensitive data in the Shanghai data base project.

Date	Feature	Description
	ANY permission management	Support the following 12 ANY permissions for five objects: ● ALTER ANY TYPE, DROP ANY TYPE ● ALTER ANY SEQUENCE, DROP ANY SEQUENCE, SELECT ANY SEQUENCE ● ALTER ANY INDEX, DROP ANY INDEX ● CREATE ANY TRIGGER, ALTER ANY TRIGGER, DROP ANY TRIGGER ● CREATE ANY SYNONYM, DROP ANY SYNONYM
	Efficient data compression algorithm	With the efficient data compression algorithm, the read and write performance deterioration in the TPC-C scenario is less than 5%, and the data compression ratio of transactional databases reaches 2:1.
	ABO	ABO supports the following service scenarios: 1. Intelligent cardinality estimation: improves the cardinality estimation accuracy of multi-column equality queries. 2. Adaptive plan selection: provides cache multi-plan management and adaptive selection. Specifications: 1. Intelligent cardinality estimation: the ABO statistical algorithm enhanced (efficiency improved by 1*x* and performance improved by 50% in typical scenarios) 2. Adaptive plan selection: the efficiency improved by 1*x* in typical scenarios
	Display of query plans in running state	Display the execution plan specifications using views for specified slow SQL statements.

Date	Feature	Description
	Security hardening	The following security vulnerabilities are fixed: CVE-2022-29824 CVE-2022-27781 CVE-2022-27775 CVE-2022-35252 CVE-2021-37750 CVE-2022-32205 CVE-2022-27776 CVE-2022-32206 CVE-2022-37434 CVE-2022-22576 CVE-2022-27782 CVE-2022-27774 CVE-2022-32207 CVE-2022-2097 CVE-2022-32208 CVE-2022-32221 CVE-2022-42916 CVE-2022-42915

2.3.3 2.*x* Versions

This section describes the 2.*x* kernel version updates of GaussDB.

Table 2-9 2.7 Enterprise Edition

Date	Feature	Description
2022-11	Concurrent DML and TRUNCATE operations between different sessions of a global temporary table	Concurrent DML and TRUNCATE operations are supported between different sessions of a global temporary table.

Date	Feature	Description
	Snapshot backups for standby nodes	Standby nodes support snapshot backups. With typical public cloud resources, it takes less than 8 hours to fully back up and restore 12 TB of data. PITR is also supported. **NOTICE** Whitelist feature. To configure the whitelist permission, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
	Emergency measures for high-latency at the IaaS layer	If the latency increases due to underlying resources, many connections will be created on the service side. As a result, the DN thread pool and dynamic memory will be full. Emergency measures are provided to quickly rectify the fault, preventing long-time impact on services.
	No logic decoding for specific user operations	Logical decoding logs are filtered by user. During the decoding, logical logs about database operations performed by blacklist users are not returned. This ensures that downstream data is not affected by logical replication of blacklist user operations. Username or user OID is required for blacklist user configuration. Logical log transactions contain user information, which provides the basis for downstream services to filter different users.
	Read-only nodes in the cascaded standby server	Read-only nodes provide inventory query services to offload the query load of the primary node. 1. There is one primary node, two standbys nodes, and many read-only nodes in the HA architecture. The read-only nodes can be flexibly deleted. At least 3 read-only nodes are required. (One read-only node is connected to external services, and the other read-only nodes are for high availability.) 2. When network resources are normal, the log replication delay between the primary and read-only nodes is less than 5 seconds. 3. The query SQL statement can be executed for several hours. **NOTICE** Whitelist feature. To configure the whitelist permission, submit an application by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

Date	Feature	Description
	Viewing SQL statements that are being executed in the current stored procedure	The SQL statements that are being executed in the current stored procedure can be viewed, enhancing the DFX capability.
	Security hardening	The following security vulnerabilities are fixed: CVE-2022-32221 CVE-2022-42916 CVE-2022-42915 CVE-2022-2097 CVE-2022-32205 CVE-2022-32206 CVE-2022-32207 CVE-2022-32208 CVE-2022-2068 CVE-2022-24302

Table 2-10 2.3 Enterprise Edition

Date	Feature	Description
2022-06	Logical replication capability enhanced	The logical replication capability is enhanced. Memory management is enhanced based on: 1. Parameter that controls the memory threshold of transactions. When the threshold is exceeded, logical logs are automatically flushed to disks. 2. Parameter that controls the memory threshold of logical replication slots. When the threshold is exceeded, logical logs are automatically flushed to disks. 3. Views or functions that display memory structure of parallel logic decoding.

Date	Feature	Description
	Ustore DFx capability enhanced	1. Online verification: Important data and fields are verified during service running. 2. Offline verification: For onsite problems and data exceptions, if the processes do not exit, **analyze verify** can be used to obtain and output error page information. 3. Module verification: The undo, redo, upage, or ubtree module is classified into three levels: fast, skip recovery, and complete. Key verification analysis can be provided by module. 4. Fault location information enhanced: There is more detailed fault location information in logs (pg_log) and views. The uheap, undo, ubtree, and flashback modules support views. 5. Error message "Snapshot too old" resolved: The message is displayed in a new transaction due to forcible reclamation of undo space.
	SQL PATCH	SQL patches are designed for database administrators, O&M personnel, and other personnel that need to optimize SQL statements. If the execution plan or execution mode of a query statement does not meet expectations, a SQL patch can adjust the query execution plan or mode without directly modifying the statement, to meet the expected effect. In addition, you can flexibly create, delete, or change a SQL patch as needed. In actual GaussDB O&M scenarios, unique SQL IDs are used to locate query statements (normal-stmt and prepared-stmt) and hints are used to optimize query performance. The scenarios for adjusting the statement execution mode are as follows: Scenario 1: Adding hints (such as hint-index and hint-GUC) to statements to adjust the query compilation mode of the optimizer. In scenario 2: Performing plain-text rewriting on query statements in special emergency cases. **NOTICE** SQL-PATCH is not supported for distributed instances.

Date	Feature	Description
	Security hardening	The following security vulnerabilities are fixed: CVE-2022-31799 CVE-2022-27781 CVE-2022-27782 CVE-2022-1292 CVE-2022-29824 CVE-2022-27774 CVE-2022-27776 CVE-2022-22576 CVE-2022-27775 CVE-2021-3520

Table 2-11 2.2 Enterprise Edition

Date	Feature	Description
2022-04	Go driver supported by primary/ standby instances	1. Go driver supports SSL connections to the database. 2. Go driver supports database handles. 3. Go driver supports Stmt. 4. Go driver supports database transactions. 5. Go driver supports query results (Rows and Row). 6. Go driver supports column types.
	SMP for stored procedures	SMP parallel execution plans can be generated and executed for query statements in stored procedures.
	Ustore hardening and optimizatio n	The Ustore adapts to the framework of the parallel index creation.
	WDR report enhanced	The ASP capability is enhanced. The WDR capability is enhanced.
	View O&M capacity enhanced	You can view disk usage details by **pg_ls_waldir()** for wal diagnosis and **pg_ls_tmpdir()** for tmp diagnosis.

Table 2-12 2.1 Enterprise Edition

Date	Feature	Description
2022-02	Ustore commercial use capability improved	1. The Ustore resources can be better managed. The undo memory is optimized and zones are loaded as required. When Ustore is not used, the memory overhead and undo file space are less than 1 MB. 2. Ustore supports WDR reports which include: a. Number of accesses to UHeap pages (including QUERY, MODIFY, DELETE, and PRUNE operations). b. Number of accesses to undo records c. Average access length of undo chains d. Rate at which undo and translot files are flushed to disks e. Rate at which undo files are generated in a specified period. f. Rate at which undo files are reclaimed in a specified period g. Index-related information (number of layers and page visits) 3. Ustore flashback: DROP operations, TRUNCATE operations, and partitioned tables can be flashed back. The baseline for flashback query performance is provided.
	Driver and syntax compatibility enhanced	1. The **select for update wait n** syntax is supported. 2. The conversion rule of the Decode type is compatible with O*. 3. New syntax needs to be connected to GK Smith (a pressure test tool). 4. The following regular expressions are supported: REGEXP_COUNT, REGEXP_INSTR, REGEXP_SUBSTR, and REGEXP_REPLACE. 5. **execute direct on** in query statements supports precompilation. 6. The criteria for updating or inserting data can be specified.
	Online database maintenance without downtime	1. Indexes can be created online without downtime. 2. Adding a standby node does not restart the primary node.

Date	Feature	Description
	Row- and column-store engine capability improved	1. Row-store table query execution is converted to vectorized execution, improving performance in the case of hybrid loads. Under the same resource environments and workloads, the TPC-H performance of vectorized execution is 30% higher than that of the row-store engine and row-store execution framework. 2. Column storage supports MERGE INTO subqueries.
	Logic decoding performance and functions optimized	1. The logic decoding performance is optimized. The DML decoding performance reaches 100 Mbit/s (DDL decoding is not supported). 2. Logical decoding supports parsing DML operations on a specified table. In JDBC, a new parameter is added to support log parsing. (You can call the existing **libpq logical walsender** API to parse logs).
	Performance improvement for enterprise-class application scenarios	1. When the same amount of data is updated concurrently, the average update latency and latency difference on a single DN are almost the same as those on MySQL. 2. There is no obvious periodic fluctuation within 2 hours in the TPC-C benchmark with 8,000 warehouses on an Arm-powered single-node instance with 128 vCPUs and 512 GB SAS SSD or equivalent specifications. 3. walwriter optimization: The performance is optimized under heavy load (the vCPU usage is 60% or higher).
	Enterprise-level capability optimized	1. The IP address, port number, and host name of a dynamic server can be changed. 2. A client driver can be configured with multiple IP addresses of the primary and standby nodes, but is only connected to the primary node. In libpq, after an primary/standby switchover occurs, a standby node can be automatically promoted to primary. 3. The compatibility of gsql with SQLPlus is enhanced. 4. The JDBC interface supports uppercase and lowercase letters.

Date	Feature	Description
	4 GB for CLOB/ BLOB	CLOBs/BLOBs can store 4 GB of data. The performance of the CLOBs/BLOBs smaller than 1 GB is not inferior to that of the previous version. Constraints: More than 1 GB of data can be read and processed only by advanced package functions. An error is reported when more than 1 GB of data is transferred by system functions. Operator and string functions do not support more than 1 GB of data. The maximum buffer size in a stored procedure is 32 KB. DISTINCT, GROUP BY, or ORDER BY operations are not allowed in LOB columns. An advanced package supports a maximum of 4 GB data. To update tables, you need to add **update** in **lob_write**.
	Deletion and update of specified partitions for primary/ standby instances	Primary/standby instances support deletion and update of specified partitions (level-1 partitions for level-1 partitioned tables and level-2 partitions for level-2 partition tables). For the deletion and update of the same amount of data (including point-to-point deletion and batch deletion scenarios), the performance of partitioned tables is the same as that of non-partition tables. After a partition is specified, the global index scan cannot be selected.
	Global cache of syscache system tables	This feature reduces the high memory usage in high-concurrency and complex query scenarios. In the case of any concurrent load, the total memory usage of the GSC or LSC cannot exceed the upper limit specified by the corresponding GUC parameters (GSC: global_syscache_threshold; LSC: local_syscache_threshold). After GSC/LSC is enabled, the loss in the standard benchmark scenario (tpcc/ sysbench) does not exceed 5%.

Date	Feature	Description
	Module function and reliability hardening	1. Capacity expansion hardening: a. Redistribution adaptive lock timeout and lock levels are optimized. b. Redistribution escape method: The kernel can trigger fast service failure. c. Redistribution supports different scenarios, and the mode can be dynamically configured. d. Abnormal residual data can be cleared during redistribution. e. Resumable data transfer is supported. 2. Data reliability a. The primary node restores damaged pages or files from the standby node: **main fork** is used for common row-store tables and **init fork** is used for unlogged tables. After the clog and csnlog files are damaged, a function is called. Currently, the file-level does not support the segment-page format. b. The standby node repairs damaged pages or files from the primary node: Table data files can be automatically repaired during replay. Currently, the file-level does not support the segment-page format. c. If the verification page and file are damaged during the backup, the damaged page and file can be automatically recovered on the standby node by calling a function. d. Logical decoding of a specified LSN segment is supported when the meta information is not changed. 3. Ultimate RTO enabled by default: a. PITR adaptation b. The ultimate RTO flow control is configurable, and the replay speed views are provided. 4. DFX and quality hardening: a. Data consistency mechanism between pgxc_node and cn/dn-handler is hardened. 5. UUID authentication: A verification mechanism is used when the primary and standby nodes are connected. The UUID is generated based on the **dbname** character string for verification.
	PITR for distributed instances	Distributed instances support PITR to further: 1. Enhance backup and restoration capability. 2. Realize resumable recovery.

Date	Feature	Description
	Stored procedures	1. Autonomous transactions support global variables. 2. Cursors can be closed in exceptions. When the JDBC executes stored procedures, the input and output parameters do not need to be registered.
	Permissions	Fine-grained permissions and ANY permissions are supported. The following ANY permissions and syntax are supported: ● CREATE ANY TABLE ● ALTER ANY TABLE ● DROP ANY TABLE ● SELECT ANY TABLE ● INSERT ANY TABLE ● UPDATE ANY TABLE ● DELETE ANY TABLE ● CREATE ANY FUNCTION ● EXECUTE ANY FUNCTION ● CREATE ANY PACKAGE ● EXECUTE ANY PACKAGE ● CREATE ANY TYPE
	Security	1. When an audit file is damaged, audit logs generated after the damage can be queried. 2. An error in an SSL certificate revocation list (CRL) does not affect normal authentication. 3. Audit logs contain transaction IDs, which are used to associate data changes with audit operations. 4. The password of the O&M account is encrypted and flushed to disks.
	Software component lifecycle matches the product lifecycle	New lower-layer software components (such as open-source software, third-party software, and self-developed platforms) are used and will be updated in a timely manner within their lifecycle to match the product lifecycle. Security vulnerabilities can be fixed in a timely manner within the lifecycle of product versions.
	ODBC commercial use	1. Distributed load balancing is supported. 2. The primary/standby switchover is supported.

Date	Feature	Description
	CM enterprise-level capability improved	1. CMS records key arbitration event logs. 2. CMS depends on Huawei-developed DCC component, instead of ETCD, to select the primary node. 3. The arbitration logic is reconstructed to expand the arbitration capability.
	Performance improved	The performance of a full-table scan based on linetitem 1X is improved by 30%, and the response latency is shorter than 1s.
	O&M monitoring capability improved	1. Normalized SQL processing is added to session statistics for diagnosing historical slow SQL statements. 2. The mem trace performance is enhanced to enrich memory fault locating methods. 3. Full-link trace from JDBC to the kernel is supported. 4. Kernel metric collection is optimized to support single-node collection and reporting. 5. The get_node_disk_and_log_status interface can be used in primary/standby instances.
	Paxos enterprise-level capability improved	1. The DCF protocol can prevent frequent switching. 2. You can configure standby Paxos nodes are promoted to primary by priority.
	Security hardening	The following security vulnerabilities are fixed: CVE-2022-23308 CVE-2020-28473 CVE-2021-22947 CVE-2021-22946 CVE-2021-22945 CVE-2021-37750 CVE-2021-36222

