

Relational Database Service

HUAWEI CLOUD MySQL Kernel

Issue 10
Date 2023-09-28



Copyright © Huawei Technologies Co., Ltd. 2023. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks 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 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 Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: <https://www.huawei.com>

Email: support@huawei.com

Contents

1 RDS for MySQL Kernel.....	1
1.1 Kernel Version Description.....	1
1.2 Connection Thread Pool.....	14
1.3 MDL Views.....	18
1.4 Online Varchar Length Increase.....	21
1.5 Ending Idle Transactions.....	21
1.6 DDL Progress Display.....	26
2 RDS for PostgreSQL Kernel.....	27
2.1 RDS for PostgreSQL Kernel Version Description.....	27
2.2 Failover Slot for Logical Subscriptions.....	29
A Change History.....	32

1 RDS for MySQL Kernel

1.1 Kernel Version Description

This section describes the kernel version updates of RDS.

RDS for MySQL 8.0

Table 1-1 RDS for MySQL 8.0 version description

Date	Description
2023-03-15	<p>Updates in RDS for MySQL 8.0.25:</p> <ul style="list-style-type: none">● New features and performance optimized<ul style="list-style-type: none">- The printing of oversized SQL audit logs was optimized.- The security of kernel log printing was enhanced.● Issues resolved Unexpected restarts due to concurrent DDL and DML operations on DB instances were resolved.

Date	Description
2022-09-09	<p>Updates in RDS for MySQL 8.0.25:</p> <ul style="list-style-type: none"> ● New features and performance optimized <ul style="list-style-type: none"> - Connection per thread was supported for killing sessions. - Constraints on memory were added for the Performance Schema. - The performance of SQL Explorer was optimized based on specific scenarios. - Database performance was optimized in specific scenarios when the value of internal_tmp_mem_storage_engine was set to MEMORY. - The compiler was upgraded to GCC 10.3. ● Issues resolved <ul style="list-style-type: none"> - Errors caused by writes to a temporary file were resolved. - Unexpected responses to Common Table Expression (CTE) queries were resolved. ● Security hardening <ul style="list-style-type: none"> - The following security vulnerabilities were fixed: CVE-2021-2417, CVE-2021-2339, CVE-2021-2425, CVE-2021-2426, CVE-2021-2427, CVE-2021-2424, CVE-2021-2383, CVE-2021-2384, and CVE-2021-2410.
2022-06-01	<ul style="list-style-type: none"> ● New features and performance optimized <ul style="list-style-type: none"> - The kernel version was upgraded to 8.0.25. - SQL statement concurrency control was supported. - The compiler was upgraded to GCC 9.3. ● Issues resolved <ul style="list-style-type: none"> - Replication interruption caused by a single oversized binlog was resolved. - Inaccurate innodb_row_lock_current_waits statistics were fixed. - Unexpected restarts due to the use of BLOB fields were resolved. ● Security hardening <ul style="list-style-type: none"> - The following security vulnerabilities were fixed: CVE-2021-2307, CVE-2021-2180, and CVE-2021-2194.

Date	Description
2021-08-07	<ul style="list-style-type: none"> ● New features and performance optimized <ul style="list-style-type: none"> - The static link library for the thread pool improved the performance. - Profile-Guided Optimization (PGO) was enabled. - The MySQL hash algorithm was optimized. - Remarks can be added to a database. - Protection from being modified by DDL was provided for the system database. - The innodb_total_tablespace parameter was added to calculate the number of InnoDB tablespaces. - The InnoDB lock view was provided in information_schema. - The OpenSSL, jemalloc, and curl open-source components were upgraded. ● Issues resolved <ul style="list-style-type: none"> - The issue that XA transactions may restart abnormally after binlog rotation was resolved. - The issue that precompiled SQL statement types were not recorded by SQL Explorer was resolved. - Incorrect execution time statistics for FLUSH PRIVILEGES was resolved. - The issue that audit logs were incorrectly written into other files was resolved.
2021-04-13	<ul style="list-style-type: none"> ● The issue that XA transactions may be lost after unexpected database restart was resolved. ● The adaptive hash segmentation algorithm was optimized. ● The kernel version was upgraded to 8.0.21. ● Security hardening The following security vulnerabilities were fixed: CVE-2020-14697, CVE-2020-14680, CVE-2020-14678, CVE-2020-14663, CVE-2021-2020, CVE-2020-14619, CVE-2020-14591, CVE-2020-14576, and CVE-2020-14539.
2021-01-26	<ul style="list-style-type: none"> ● SQL statement concurrency control was optimized. ● Full SQL collection was optimized.
2020-12-31	<ul style="list-style-type: none"> ● Performance optimized The compiler was upgraded to GCC 9.
2020-12-01	<ul style="list-style-type: none"> ● Performance optimized The efficiency of collecting additional information about slow query logs was improved. ● Issues resolved The issue that replication on the standby DB instance might be interrupted during the rollback of distributed transactions was resolved.

Date	Description
2020-1 1-06	<ul style="list-style-type: none"> Issues resolved Timing error caused by gettimeofday in multiple threads was fixed.
2020-0 9-21	<ul style="list-style-type: none"> Disconnection details were recorded in error logs. Index hints were supported.
2020-0 8-03	<ul style="list-style-type: none"> The execution time and waiting time of large transactions were displayed. Independent connection control was used to manage users. SQL filter was used to restrict the execution frequency of SQL statements during peak hours. Kernel performance was optimized.
2020-0 6-19	<ul style="list-style-type: none"> The kernel version was upgraded to 8.0.20. Kernel performance was optimized.
2020-0 2-15	<ul style="list-style-type: none"> RDS for MySQL 8.0 was put into commercial use. The Huawei Cloud Kunpeng-powered Arm kernel version was released.
2019-1 2-15	<ul style="list-style-type: none"> The kernel version was upgraded to 8.0.17. The parallel index creation was 2.5 times faster.
2019-1 0-15	The issue that the primary/standby replication was abnormal when SQL_MODE was set to PAD_CHAR_TO_FULL_LENGTH was resolved.
2019-0 9-15	<ul style="list-style-type: none"> Thread pools were supported. OpenSSL was upgraded to 1.1.1a. The Common Type System (CTS) syntax create table xx select was supported. The memory usage and CPU time usage of user threads can be queried through show full processlist.

RDS for MySQL 5.7

Table 1-2 RDS for MySQL 5.7 version description

Date	Description
2023-06-28	<ul style="list-style-type: none"> ● New features and performance optimized <ul style="list-style-type: none"> - The kernel version was upgraded to 5.7.41. - Compiler security options were added. ● Issues resolved <ul style="list-style-type: none"> - The replication exception that may occur when an index is added to a reference table and a foreign key is added to another table concurrently was rectified. - The replication exception that may occur when a child table is deleted after a foreign key table is deleted was rectified. ● Security hardening <ul style="list-style-type: none"> - The following security vulnerabilities were fixed: CVE-2023-21963, CVE-2022-32221, CVE-2023-21840, CVE-2022-2097, CVE-2022-21617, CVE-2022-21608, CVE-2022-21592, CVE-2022-21589, CVE-2022-1292, CVE-2022-27778, CVE-2018-25032, and CVE-2022-21515.
2022-09-09	<ul style="list-style-type: none"> ● New features and performance optimized <ul style="list-style-type: none"> - The kernel version was upgraded to 5.7.38. - The compiler was upgraded to GCC 10.3. - Connection per thread was supported for killing sessions. - The threshold for slow query logs can be set based on the lock wait duration. - ALT security was hardened. ● Issues resolved <ul style="list-style-type: none"> - Recovery security of crashed XA transactions on the primary instance was enhanced. - Abnormal instance reboot when Database Proxy is enabled was resolved. - Abnormal reboot caused by failed memory request for plugins was resolved. ● Security hardening <ul style="list-style-type: none"> - The following security vulnerabilities were fixed: CVE-2022-21454, CVE-2022-21417, CVE-2022-21427, CVE-2022-21451, CVE-2022-21444, and CVE-2022-21460.

Date	Description
2022-06-01	<ul style="list-style-type: none"> ● New features and performance optimized <ul style="list-style-type: none"> - The kernel version was upgraded to 5.7.37. - The compiler was upgraded to GCC 9.3. - The OpenSSL and curl open-source components were upgraded. ● Issues resolved <ul style="list-style-type: none"> - Replication interruption caused by a single oversized binlog was resolved. - Unexpected restarts caused by concurrent playback grants to the slave node were resolved. - Replication interruptions caused by hidden auto-increment keys were resolved. - Unexpected restarts during rollback of tables with virtual columns were resolved. - Unexpected restarts during recovery of encrypted tables were resolved. - Inaccurate Seconds Behind Master values in specific scenarios were resolved. ● Security hardening <ul style="list-style-type: none"> - The following security vulnerabilities were fixed: CVE-2022-21367, CVE-2022-21304, and CVE-2022-21344.
2022-01-26	<ul style="list-style-type: none"> ● New features and performance optimized <ul style="list-style-type: none"> - Hidden auto-increment keys were supported. ● Issues resolved <ul style="list-style-type: none"> - Replication exceptions caused by repeated commitment of XA transactions were resolved. - Inaccurate innodb_row_lock_current_waits statistics were fixed.
2021-11-26	<ul style="list-style-type: none"> ● New features and performance optimized <ul style="list-style-type: none"> - Restrictions on the length of a single record for SQL Explorer were removed. - Application Lossless and Transparent (ALT) Phase I was supported for RDS for MySQL. ● Issues resolved <ul style="list-style-type: none"> - Memory problems of the thread pool in extreme scenarios were resolved. - Occasional suspension of XA transaction playbacks on a standby node was resolved.

Date	Description
2021-08-07	<ul style="list-style-type: none"> ● New features and performance optimized <ul style="list-style-type: none"> - The static link library for the thread pool improved the performance. - Profile-Guided Optimization (PGO) was enabled. - The MySQL hash algorithm was optimized. - Remarks can be added to a database. - Protection from being modified by DDL was provided for the system database. - I/O latency information can be queried in error logs. - Minidumps were supported. - The kernel version was upgraded to 5.7.33. - The OpenSSL, jemalloc, and curl open-source components were upgraded. ● Issues resolved <ul style="list-style-type: none"> - The issue that the replication on the standby node may be interrupted due to playback order preserving was resolved. - The issue that XA transactions may restart abnormally after binlog rotation was resolved. - The issue that precompiled SQL statement types may not be recorded by SQL Explorer was resolved. - Incorrect execution time statistics for FLUSH PRIVILEGES was resolved. - The issue that audit logs were incorrectly written into other files was resolved. ● Security hardening <p>The following security vulnerabilities were fixed: CVE-2021-2011, CVE-2021-2178, and CVE-2021-2202.</p>
2021-04-13	<ul style="list-style-type: none"> ● The issue that XA transactions may be lost after unexpected database restart was resolved. ● The adaptive hash segmentation algorithm was optimized. ● The kernel version was upgraded to 5.7.32.
2021-01-26	<ul style="list-style-type: none"> ● New features <p>The real client address can be displayed when proxy is used.</p> ● Issues resolved <p>Full SQL collection was optimized.</p> <p>The issue that revoking permissions might cause permission inconsistency between primary and standby DB instances was resolved.</p> <p>MySQL 8.0 optimizations on Instant Add Column was incorporated.</p>

Date	Description
2020-1 2-31	<ul style="list-style-type: none"> ● Performance optimized <ul style="list-style-type: none"> - The efficiency of collecting additional information about slow query logs was improved. - The compiler was upgraded to GCC 9. ● Issues resolved The issue that replication on the standby DB instance might be interrupted during the rollback of distributed transactions was resolved.
2020-1 2-01	The conflict frequency between fil_sys mutual exclusions was reduced.
2020-1 1-06	<ul style="list-style-type: none"> ● New features <ul style="list-style-type: none"> - The compiler was optimized. - The UTF-8 encoding efficiency was improved on non-Arm platforms. ● Issues resolved Timing error caused by gettimeofday in multiple threads was fixed.
2020-0 9-21	<ul style="list-style-type: none"> ● The kernel version was upgraded to 5.7.31. ● SQL filter was used to restrict the execution frequency of SQL statements during peak hours.
2020-0 8-03	<ul style="list-style-type: none"> ● Kernel performance was optimized. ● The recycle bin was supported. ● Issues that may occur during local disk data cleanup were resolved.
2020-0 7-09	<ul style="list-style-type: none"> ● Kernel performance was optimized. ● Users' operation history can be recorded in error logs. ● Distributed transaction stability was improved.
2020-0 6-30	<ul style="list-style-type: none"> ● Kernel performance was optimized. ● Local disk logs whose size exceeds the local disk capacity were stored on cloud disks. ● The buffer pool memory initialization module was optimized to improve the initialization efficiency. ● The thread security of some operations on Arm was improved.
2020-0 5-30	<ul style="list-style-type: none"> ● New features <ul style="list-style-type: none"> - Index hints were supported. - Full SQL logs can be captured. ● Issues resolved Occasional database connection failures were resolved.
2020-0 4-30	Kernel performance was optimized.

Date	Description
2020-03-30	<ul style="list-style-type: none"> ● The kernel version was upgraded to 5.7.29. ● Kernel performance was optimized. ● Threshold pools supported statistics on I/O waits.
2020-02-15	<ul style="list-style-type: none"> ● Replication dual-channel: A replication status channel was added to accurately determine the replication status when the primary database crashes, ensuring that transactions are not lost. ● Optimized ROW_IMAGE mode: The binlog size was reduced, and migration and SQL flashback were supported.
2019-12-15	<ul style="list-style-type: none"> ● The progress of adding columns or indexes can be obtained through information_schema.innodb_alter_table_progress. ● Long transactions: The transaction execution time Trx_Executed_Time can be obtained through show processlist. ● Online extension of the string field length: The VARCHAR field length was changed from COPY to INPLACE by default. ● Abundant InnoDB deadlock information: The complete onsite information when deadlock occurs at the InnoDB layer can be viewed through show engine innodb status.
2019-10-15	<ul style="list-style-type: none"> ● Performance optimized <ul style="list-style-type: none"> – The Huawei Cloud Kunpeng-powered Arm kernel version was released. ● New features <ul style="list-style-type: none"> – The kernel version was upgraded to 5.7.27. – Instant Add Column was supported. Columns are quickly added to a table without copying data and without occupying disk space and disk I/Os. Data can be updated in real time during peak hours. – Metadata lock (MDL) view was supported. The MDL information held or waited by the thread can be obtained through information_schema.metadata_lock_info.
2019-08-15	<p>For Jemalloc memory management, the glibc memory management module was replaced to reduce memory usage and improve memory allocation efficiency.</p>
2019-06-15	<ul style="list-style-type: none"> ● The kernel version was upgraded to 5.7.25.

Date	Description
2019-05-15	<ul style="list-style-type: none"> ● New features <ul style="list-style-type: none"> – When sync_binlog and innodb_flush_log_at_trx_commit are set to values other than 1, the crash-safe replication of the standby database is guaranteed. In sysbench high-concurrent write-only request scenarios, the replication delay between primary and standby DB instances is almost zero. ● Issues resolved <ul style="list-style-type: none"> – The issue that when relay_log_recovery was set to ON, the standby database could not be rebooted after being killed in certain scenarios was resolved. – The issue that the primary/standby replication was abnormal when SQL_MODE was set to PAD_CHAR_TO_FULL_LENGTH was resolved. – The issue that performance_schema statistics were repeatedly measured was resolved. – The issue that when ORDER BY queries were performed on tables related to information replication in Performance_schema, the return value was empty was resolved.
2019-01-15	<ul style="list-style-type: none"> ● The issue that data was inconsistent and data replication was interrupted after the flush operation was performed on read replicas was resolved. ● The issue that the replication thread of the standby database was suspended due to statements like REPAIR or OPTIMIZE was resolved.
2018-11-15	<ul style="list-style-type: none"> ● The kernel version was upgraded to 5.7.23. ● Temporary tables can be created or deleted in transactions when GTID is enabled. ● Table-level multi-threaded slaves (MTS) parallel playback was supported.
2018-07-15	<ul style="list-style-type: none"> ● The kernel version was upgraded to 5.7.22. ● Thread pools were supported. ● The Common Type System (CTS) syntax create table xx select was supported. ● Operator pushdown: The aggregation operator is pushed down to the storage engine layer to improve the execution speed of count() and sum(). ● Kill idle transactions: Transactions that have been idled for a long time are automatically killed after the timeout threshold is reached. ● The memory usage and CPU time usage of user threads can be queried through show full processlist.

RDS for MySQL 5.6

Table 1-3 RDS for MySQL 5.6 version description

Date	Description
2023-03-15	<p>Updates in RDS for MySQL 5.6.51:</p> <ul style="list-style-type: none"> ● New features and performance optimized <ul style="list-style-type: none"> - The printing of oversized SQL audit logs was optimized. - The security of log printing was enhanced. ● Issues resolved <ul style="list-style-type: none"> - The replication exception that may occur when an index is added to a reference table and a foreign key is added to another table concurrently was rectified.
2022-09-09	<ul style="list-style-type: none"> ● New features and performance optimized <ul style="list-style-type: none"> - Connection per thread was supported for killing sessions. ● Issues resolved <ul style="list-style-type: none"> - Abnormal instance reboot when Database Proxy is enabled was resolved. - main.proxy_connect buffer overflow issues were resolved. - Abnormal reboot caused by failed memory request for plugins was resolved.
2022-06-01	<p>Inaccurate innodb_row_lock_current_waits statistics were fixed.</p>
2021-08-07	<ul style="list-style-type: none"> ● New features <ul style="list-style-type: none"> - Remarks can be added to a database. - Protection from being modified by DDL was provided for the system database. - The OpenSSL and jemalloc open-source components were upgraded. ● Issues resolved <ul style="list-style-type: none"> - The issue that the synchronization may be interrupted after password change was resolved. - The issue that audit logs were incorrectly written into other files was resolved.
2021-04-13	<ul style="list-style-type: none"> ● The issue that the replication on the standby node may be interrupted due to playback order preserving was resolved. ● The kernel version was upgraded to 5.6.51. ● Security hardening The vulnerability patches for MySQL 5.6 Community Edition are no longer released because the patches for this version are not released anymore.

Date	Description
2021-01-26	<ul style="list-style-type: none"> ● New features The real client address can be displayed when proxy is used. ● Issues resolved The issue that a syntax error was reported during the select 1 for update execution was resolved. Full SQL collection was optimized.
2020-12-31	SQL filter was reconstructed to improve usability.
2020-11-06	The kernel version was upgraded to 5.6.50.
2020-09-23	SQL filter was used to restrict the execution frequency of SQL statements during peak hours.
2020-08-03	The kernel version was upgraded to 5.6.49.
2020-07-09	<ul style="list-style-type: none"> ● Local disk logs whose size exceeds the local disk capacity were stored on cloud disks. ● Users' operation history can be recorded in error logs.
2020-05-30	The buffer pool memory initialization module was optimized to improve the initialization efficiency.
2020-04-30	Occasional database connection failures were resolved.
2020-03-30	<ul style="list-style-type: none"> ● Full SQL collection was supported. ● The compiler was upgraded to 7.3. ● The kernel version was upgraded to 5.6.47.
2020-02-15	<ul style="list-style-type: none"> ● Replication dual-channel: A replication status channel was added to accurately determine the replication status when the primary database crashes, ensuring that transactions are not lost. ● Optimized ROW_IMAGE mode: The binlog size was reduced, and migration and SQL flashback were supported.
2019-12-15	<ul style="list-style-type: none"> ● Long transactions: The transaction execution time Trx_Executed_Time can be obtained through show processlist. ● Online extension of the string field length: The VARCHAR field length was changed from COPY to INPLACE by default. ● Abundant InnoDB deadlock information: The complete onsite information when deadlock occurs at the InnoDB layer can be viewed through show engine innodb status.
2019-10-15	<ul style="list-style-type: none"> ● The kernel version was upgraded to 5.6.45. ● The memory usage and CPU time usage of user threads can be queried through show full processlist. ● Kill idle transactions: Transactions that have been idled for a long time are automatically killed after the timeout threshold is reached.

Date	Description
2019-08-15	For Jemalloc memory management, the glibc memory management module was replaced to reduce memory usage and improve memory allocation efficiency.
2019-06-15	<ul style="list-style-type: none"> ● The kernel version was upgraded to 5.6.43. ● The audit function was supported.
2019-05-15	<ul style="list-style-type: none"> ● The issue that the replication delay Seconds_Behind_Master was inaccurate in certain scenarios was resolved. ● The issue that the primary/standby replication was abnormal when SQL_MODE was set to PAD_CHAR_TO_FULL_LENGTH was resolved.
2019-01-15	<ul style="list-style-type: none"> ● The issue that data was inconsistent and data replication was interrupted after the flush operation was performed on read replicas was resolved. ● The issue that the replication thread of the standby database was suspended due to statements like REPAIR or OPTIMIZE was resolved. ● The issue that an error occurred on the grant select(column_name) statement replication threshold was resolved.
2018-11-15	<ul style="list-style-type: none"> ● The kernel version was upgraded to 5.6.41. ● Temporary tables can be created or deleted in transactions when GTID is enabled. ● Table-level multi-threaded slaves (MTS) parallel playback was supported.
2018-07-15	<ul style="list-style-type: none"> ● New features <ul style="list-style-type: none"> - The kernel version was upgraded to 5.6.40. - Thread pools were supported. - The Common Type System (CTS) syntax create table xx select was supported. ● Issues resolved <ul style="list-style-type: none"> - The issue that binlog and relay log names depended on the PID file names was resolved. - The issue that the empty relay_log_basename parameter value resulted in primary/standby replication failures was resolved. - The issue that the force index syntax became invalid in the group by xx order by xx limit n1,n2 scenario was resolved.

RDS for MariaDB Kernel Version Description

The kernel version of RDS for MariaDB is based on open-source MariaDB 10.5.16, which is a stable (GA) version of MariaDB. By optimizing the performance,

security, and storage engine, RDS for MariaDB is suitable for various applications and scenarios.

The following table lists the detailed information about RDS for MariaDB versions.

Table 1-4 RDS for MariaDB version description

Date	Description
2023-05-01	RDS for MariaDB was released for open beta testing (OBT). The kernel version was based on 10.5.16.

1.2 Connection Thread Pool

Introduction

When there are a large number of concurrent database connections, a large number of resources are occupied, and the performance of the MySQL server deteriorates significantly. RDS for MySQL provides a connection thread pool that uses a few active threads to serve a large number of database connections. This decouples connections from execution and improves database performance in high-concurrency scenarios.

Characteristics

RDS for MySQL connection thread pool provides the following benefits:

- A large number of database connections can be processed, and resource contention and context switches are reduced.
- The number of concurrent transactions is limited. When the database load is heavy, transactions that are being executed are preferentially guaranteed.
- Connections are processed quickly to prevent thread exceptions.
- When a transaction is waiting for I/Os and locks, CPU resources and other connections are released.

Thread Pool Operations

- Querying thread pool parameters

Run **show variables** to query thread pool parameters.

```
show variables like 'threadpool%';
+-----+-----+
| Variable_name      | Value      |
+-----+-----+
| threadpool_enabled | ON         |
| threadpool_high_prio_tickets | 4294967295 |
| threadpool_idle_timeout | 60         |
| threadpool_long_conn_time | 2          |
| threadpool_max_threads | 100000     |
| threadpool_oversubscribe | 3          |
| threadpool_prio_kickup_timer | 1000       |
| threadpool_rec_launch_time | ON         |
| threadpool_size     | 128        |
| threadpool_slow_conn_log | ON         |
```

```
| threadpool_slow_conn_log_interval | 30 |
| threadpool_slow_launch_time      | 2   |
| threadpool_stall_limit           | 500 |
+-----+-----+
```

Table 1-5 Thread pool parameters

Parameter	Description
threadpool_enabled	Enables or disables thread pools.
threadpool_high_prio_tickets	Number of tickets held by a high-priority thread.
threadpool_idle_timeout	Idle time before a thread is destroyed, in seconds.
threadpool_long_conn_time	If the login time exceeds the value of this parameter, the login information is printed in logs.
threadpool_max_threads	Maximum number of threads that can be created in a thread pool.
threadpool_oversubscribe	Maximum number of extra threads that can be created in a thread group.
threadpool_prio_kickup_timer	Maximum duration (in milliseconds) in a low-priority queue.
threadpool_rec_launch_time	Records the thread launch time.
threadpool_size	Number of thread groups.
threadpool_slow_conn_log	Whether to record slow logins in error logs.
threadpool_slow_conn_log_interval	Recording frequency. After a slow login is recorded, the system does not record logins within this interval.
threadpool_slow_launch_time	If the login or query time is greater than the value of this parameter, the value of threadpool_slow_launch_request in status increases by 1.
threadpool_stall_limit	Interval for checking whether a thread group is busy.

Table 1-6 Thread pool parameters that can be modified

Parameter	Dynamic Parameter	Data Type	Value Range	Description
threadpool_enabled	Yes	boolean	[ON,OFF]	<ul style="list-style-type: none"> ON: Enables the thread pool. OFF: Disables the thread pool.
threadpool_oversubscribe	Yes	integer	[1,50]	Maximum number of extra threads that can be created in a thread group.
threadpool_size	Yes	integer	[1,512]	Number of thread groups.

- Querying thread pool status

Run **show status** to query the thread pool status.

```
show status like 'threadpool%';
+-----+
| Variable_name          | Value |
+-----+
| Threadpool_active_connections | 65 |
| Threadpool_active_threads   | 5 |
| Threadpool_avg_launch_time  | 0 |
| Threadpool_dump_threads     | 0 |
| Threadpool_idle_threads     | 63 |
| Threadpool_running          | ON |
| Threadpool_slow_launch_request | 0 |
| Threadpool_threads          | 237 |
| Threadpool_threads_high_water_mark | 1075 |
| Threadpool_waiting_threads   | 57 |
| Threadpool_worst_launch_time | 692548 |
+-----+
```

Table 1-7 Thread pool status

Status	Description
Threadpool_active_connections	Number of active connections in a thread pool
Threadpool_active_threads	Number of active threads in a thread pool
Threadpool_avg_launch_time	Average waiting time, in milliseconds
Threadpool_dump_threads	Number of dump threads

Status	Description
Threadpool_idle_thread	Number of idle threads in a thread pool
Threadpool_running	Whether a thread pool is running
Threadpool_slow_launch_request	Number of times that the slow_launch_request is exceeded
Threadpool_threads	Total number of connections in a thread pool
Threadpool_threads_high_water_mark	Number of historical high threads
Threadpool_waiting_threads	Status of the waiting thread pool
Threadpool_worst_launch_time	Worst launch time, in milliseconds

Performance Tests

Table 1-8 Performance tests of different threads

Model	Threads	Thread Pool Enabled	QPS	Latency (ms)
oltp_update_non_index	32	Yes	5932.47	7.84
oltp_update_non_index	64	Yes	10074.11	9.39
oltp_update_non_index	128	Yes	18079.61	10.65
oltp_update_non_index	256	Yes	27439.38	14.46
oltp_update_non_index	512	Yes	33007.96	28.16
oltp_update_non_index	1024	Yes	30282.13	51.94
oltp_update_non_index	2048	Yes	29836.86	95.81

1.3 MDL Views

Introduction

MySQL Community Edition cannot obtain table MDLs when performance_schema was disabled. If **Waiting for metadata lock** is displayed, blocking DML or DDL, you may need to reboot DB instances because the association among sessions cannot be identified. This has an impact on service running.

In complex service scenarios, such problems will frequently occur if exclusive operations like DDL and LOCK Table are performed on database metadata, bringing troubles to you.

To resolve the problems, Huawei Cloud RDS for MySQL introduces the MDL view, enabling you to view MDLs that each session is holding and waiting for. You can effectively diagnose the system and identify the problematic sessions, minimizing the impact on services.

Description

The MDL view is displayed as a system table. The table is named **metadata_lock_info** and contained in the **information_schema** database. The table structure is as follows.

```
desc information_schema.metadata_lock_info;
+-----+-----+-----+-----+-----+-----+
| Field      | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| THREAD_ID  | bigint(20) unsigned | NO   |     | 0       |       |
| LOCK_STATUS | varchar(24)         | NO   |     |         |       |
| LOCK_MODE  | varchar(24)         | YES  |     | NULL    |       |
| LOCK_TYPE  | varchar(30)         | YES  |     | NULL    |       |
| LOCK_DURATION | varchar(30)        | YES  |     | NULL    |       |
| TABLE_SCHEMA | varchar(64)        | YES  |     | NULL    |       |
| TABLE_NAME | varchar(64)         | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

Table 1-9 metadata_lock_info fields

No.	Field Name	Type	Description
0	THREA D_ID	bigint(20) unsigned	Session ID.
1	LOCK_S TATUS	varchar(24)	Two statuses of MDL: <ul style="list-style-type: none"> ● PENDING: The session is waiting for the MDL. ● GRANTED: The session has obtained the MDL.
2	LOCK_ MODE	varchar(24)	MDL mode, such as MDL_SHARED, MDL_EXCLUSIVE, MDL_SHARED_READ, and MDL_SHARED_WRITE.

No.	Field Name	Type	Description
3	LOCK_TYPE	varchar(30)	MDL type, such as Table metadata lock, Schema metadata lock, Global read lock, and Tablespace lock.
4	LOCK_DURATION	varchar(30)	MDL range. The value options are as follows: <ul style="list-style-type: none"> ● MDL_STATEMENT: statement-level MDLs ● MDL_TRANSACTION: transaction-level MDLs ● MDL_EXPLICIT: global-level MDLs
5	TABLE_SCHEMA	varchar(64)	Database name. For some global-level MDLs, this parameter is left empty.
6	TABLE_NAME	varchar(64)	Table name. For some global-level MDLs, this parameter is left empty.

Examples

Scenario: If no transaction is committed for a long time, DDL operations are blocked, and then all operations on the same table are blocked.

Table 1-10 MDL view example

Table Name	Session			
	Session 2	Session 3	Session 4	Session 5
t1	begin; select * from t1;	-	-	-
t2	-	begin; select * from t2;	-	-
t3	-	-	truncate table t2; (blocked)	-
t4	-	-	-	begin; select * from t2; (blocked)

The **show processlist** command output shows information about threads and MDL views.

- Session 4 is waiting for an MDL on table t2.
- Session 3 holds a transaction-level MDL on table t2. If the transaction hold by session 3 is not committed, session 4 will be kept blocked.

You only need to run the **commit** command on session 3 or kill session 3 to keep services running.

1.4 Online Varchar Length Increase

Introduction

Varchar is a set of character data. The native MySQL only supports varchar whose length is no more than 256 bytes. To increase its length to more than 256 bytes, copy data to new tables and lock the tables to prevent data writes during the length increase. Huawei Cloud RDS for MySQL has no limitations on the varchar length and allows you to increase it online.

Constraints

This function is available only in RDS for MySQL 5.6 (kernel version 5.6.46 or later) and 5.7 (kernel version 5.7.27 or later).

Category

- Increase the varchar length to no more than 256 bytes.

```
create table t1(a varchar(10));
Query OK, 0 rows affected (0.03 sec)
alter table t1 modify a varchar(100),ALGORITHM=INPLACE, LOCK=NONE;
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warning: 0
```
- Increase the varchar length from less than 256 bytes to more than 256 bytes.

```
create table t1(a varchar(100));
Query OK, 0 rows affected (0.05 sec)
alter table t1 modify a varchar(300),ALGORITHM=INPLACE, LOCK=NONE;
Query OK, 0 rows affected (0.11 sec)
Records: 0 Duplicates: 0 Warning: 0
```
- Increase the varchar length beyond 256 bytes.

```
create table t1(a varchar(300));
Query OK, 0 rows affected (0.08 sec)
alter table t1 modify a varchar(500),ALGORITHM=INPLACE, LOCK=NONE;
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warning: 0
```

1.5 Ending Idle Transactions

Introduction

If an idle transaction is not committed for a long time, its rollback will consume database resources and performance. If a large number of idle transactions are not committed and not rolled back for a long time, the performance loss to a database is severe especially during peak hours.

Characteristics

Huawei Cloud RDS for MySQL supports idle transaction disconnection when a rollback timed out. This function has the following characteristics:

- Different parameters are used to control different types of transactions.
- When idle transactions timed out, they are automatically rolled back and disconnected.

Description

Run **show variables** to query related parameters.

```
show variables like '%idle%';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| idle_readonly_transaction_timeout | 0     |
| idle_transaction_timeout      | 0     |
| idle_write_transaction_timeout  | 0     |
+-----+-----+
```

Table 1-11 Parameter description

Parameter	Type	Value Range	Dynamic Validation	Description
idle_readonly_transaction_timeout	integer	Positive integer	Yes	Time in seconds that the server waits for idle read-only transactions before killing the connection. If this parameter is set to 0 , there is not timeout threshold for idle read-only transactions.

Parameter	Type	Value Range	Dynamic Validation	Description
idle_transaction_timeout	integer	Positive integer	Yes	<p>Time in seconds that the server waits for common idle transactions before killing the connection. If this parameter is set to 0, there is not timeout threshold for common idle transactions.</p> <p>NOTE</p> <p>The parameters idle_readonly_transaction_timeout and idle_write_transaction_timeout have higher priorities than the parameter idle_transaction_timeout.</p> <ul style="list-style-type: none"> If you set a value for idle_readonly_transaction_timeout or idle_write_transaction_timeout and validate the value, idle_transaction_timeout becomes invalid. If only the parameter idle_transaction_timeout has been set and validated, the value of this parameter is used as the timeout interval for read and write operations on transactions.
idle_write_transaction_timeout	integer	Positive integer	Yes	<p>Time in seconds that the server waits for idle read/write transactions before killing the connection. If this parameter is set to 0, there is not timeout threshold for idle read/write transactions.</p>

Application Scenarios

The parameters are set as follows.

```
show variables like 'idle%';
```

```
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| idle_readonly_transaction_timeout | 5     |
| idle_transaction_timeout      | 10    |
```

```
| idle_write_transaction_timeout | 15 |
+-----+-----+
```

- Setting **idle_readonly_transaction_timeout**

Set **idle_readonly_transaction_timeout** to 5.

- Run the **begin** statement to start a transaction and run a query statement. The following information is displayed.

```
begin;
Query OK, 0 rows affected (0.00 sec)
select * from t1;
+-----+-----+-----+
| a | b | c | d |
+-----+-----+-----+
| 1 | b | 303 | d |
+-----+-----+-----+
1 row in set (0.00 sec)
```

- Wait for five seconds and run a query statement again. The following information is displayed.

```
select * from t1;
+-----+-----+-----+
| a | b | c | d |
+-----+-----+-----+
| 1 | b | 303 | d |
+-----+-----+-----+
1 row in set (0.00 sec)
select * from t1;
ERROR 2006(HY000): MySQL server has gone away
```

- Setting **idle_transaction_timeout**, **idle_readonly_transaction_timeout**, and **idle_write_transaction_timeout**

Set **idle_transaction_timeout** to 10, **idle_readonly_transaction_timeout** to 0, and **idle_write_transaction_timeout** to 0.

```
show variables like '%idle%';
+-----+-----+-----+
| Variable_name | Value |
+-----+-----+-----+
| idle_readonly_transaction_timeout | 0 |
| idle_transaction_timeout | 10 |
| idle_write_transaction_timeout | 0 |
+-----+-----+-----+
3 rows in set (0.01 sec)
```

- Read-only transactions

When **idle_readonly_transaction_timeout** is set to 0, the **idle_transaction_timeout** parameter takes effect.

- Run the **begin** statement to start a transaction and run a statement to query the table data. The following information is displayed.

```
begin;
Query OK, 0 rows affected (0.00 sec)
select * from t1;
+-----+-----+-----+
| a | b | c | d |
+-----+-----+-----+
| 1 | b | 43 | d |
+-----+-----+-----+
1 row in set (0.00 sec)
```

- Wait for 10 seconds and run a query statement again. The following information is displayed.

```
select * from t1;
ERROR 2006(HY000): MySQL server has gone away
```

- Read/write transactions

When **idle_write_transaction_timeout** is set to **0**, the **idle_transaction_timeout** parameter takes effect.

- i. Run the **begin** statement to start a transaction, insert data, and run a query statement within 10 seconds. The following information is displayed.

```
begin;
Query OK, 0 rows affected (0.00 sec)
INSERT INTO t1(a,b,c,d) VALUES (1,'b',FLOOR( 1 + (RAND()*1000)) , 'd');
Query OK, 1 rows affected (0.00 sec)
select * from t1;
+----+-----+-----+-----+
| a  | b  | c  | d  |
+----+-----+-----+-----+
| 1  | b  | 425 | d  |
+----+-----+-----+-----+
1 row in set (0.00 sec)
```

- ii. Wait for 10 seconds and run a query statement again. The following information is displayed.

```
select * from t1;
ERROR 2006(HY000): MySQL server has gone away
```

- iii. Independently run a statement to query the table. If the following information is displayed, the transaction has been rolled back.

```
select * from t1;
Empty set (0.00 sec)
```

- **Setting `idle_write_transaction_timeout`**

Set **idle_write_transaction_timeout** to **15**.

- a. Run the **begin** statement to start a transaction and then insert a data record. The following information is displayed.

```
begin;
Query OK, 0 rows affected (0.00 sec)
INSERT INTO t1(a,b,c,d) VALUES (1,'b',FLOOR( 1 + (RAND()*1000)) , 'd');
Query OK, 1 rows affected (0.00 sec)
```

- b. Run a query statement within 15 seconds of the time range specified by **idle_write_transaction_timeout**. The following information is displayed.

```
select * from t1;
+----+-----+-----+-----+
| a  | b  | c  | d  |
+----+-----+-----+-----+
| 1  | b  | 987 | d  |
+----+-----+-----+-----+
1 row in set (0.00 sec)
```

- c. Wait for 15 seconds and run a query statement again. The following information is displayed.

```
select * from t1;
ERROR 2006(HY000): MySQL server has gone away
```

- d. Reconnect the transaction to the database and run a query statement. If the following information is displayed, the transaction has been rolled back.

```
select * from t1;
Empty set (0.00 sec)
```

1.6 DDL Progress Display

Introduction

DDL operations on large tables are time-consuming. However, MySQL Community Edition does not provide you with any information about the DDL execution phase and progress, which may cause great troubles to you.

To solve this problem, Huawei Cloud RDS for MySQL launches the DDL progress display feature. You can query the **INFORMATION_SCHEMA.INNODB ALTER_TABLE_PROGRESS** table to view the execution phase and progress of DDL statements in real time.

Characteristics

Table 1-12 INNODB ALTER_TABLE_PROGRESS table columns

Column	Description
THREAD_ID	Thread ID
QUERY	ALTER TABLE SQL statements
START_TIME	DDL start time
ELAPSED_TIME	Elapsed time (s)
ALTER_TABLE_STAGE	ALTER TABLE stage events
STAGE_COMPLETED	Completed work at the current stage
STAGE_ESTIMATED	Estimated work at the current stage

In order of occurrence, ALTER TABLE stage events include:

- **stage/innodb/alter table (read PK and internal sort):** Read the primary key.
- **stage/innodb/alter table (merge sort):** Sort by primary key. This process may take a long period of time because temporary files are generated.
- **stage/innodb/alter table (insert):** Insert the sorted data into the table.
- **stage/innodb/alter table (log apply index):** Apply DML logs generated during DDL execution to the created or modified index.
- **stage/innodb/alter table (flush):** Flush data to the disk.
- **stage/innodb/alter table (log apply table):** Apply DML logs generated during DDL execution to the created or modified table.
- **stage/innodb/alter table (end):** Finish the remaining work.

2 RDS for PostgreSQL Kernel

2.1 RDS for PostgreSQL Kernel Version Description

The following table lists the release dates and updates of community versions supported by RDS for PostgreSQL.

Table 2-1 Kernel version release history

Released In	Supported Community Versions	Updates
2023-08	<ul style="list-style-type: none"> • 14.8 • 13.11 • 12.15 • 11.20 	<p>Main updates:</p> <ul style="list-style-type: none"> • Synchronized the latest code in the community to fix security vulnerabilities such as CVE-2023-2454 and CVE-2023-2455 and functional problems. <p>More release notes:</p> <ul style="list-style-type: none"> • Release Notes on PostgreSQL 14.8, 13.11, 12.15, and 11.20
2023-04	<ul style="list-style-type: none"> • 14.6 • 13.9 • 12.13 • 11.18 • 10.23 	<p>Main updates:</p> <ul style="list-style-type: none"> • Supported the pgl_ddl_deploy plugin. • Upgraded the zhparser plugin. <p>More information about plugins:</p> <ul style="list-style-type: none"> • Different kernel versions support different plugins. For details, see Supported Plugins.

Released In	Supported Community Versions	Updates
2022-12	<ul style="list-style-type: none"> • 14.6 • 13.9 • 12.13 • 11.18 • 10.23 	<p>Main updates:</p> <ul style="list-style-type: none"> • Synchronized the latest code in the community to fix security vulnerabilities and functional problems. <p>More release notes:</p> <ul style="list-style-type: none"> • Release Notes on PostgreSQL 14.6, 13.9, 12.13, 11.18, and 10.23
2022-07	<ul style="list-style-type: none"> • 14.4 • 13.7 • 12.11 • 11.16 • 10.21 	<p>Main updates:</p> <ul style="list-style-type: none"> • Introduced PostgreSQL 14. • Synchronized the latest code in the community to fix security vulnerabilities such as CVE-2022-1552 and functional problems. <p>More release notes:</p> <ul style="list-style-type: none"> • Release Notes on PostgreSQL 14.4 • Release Notes on PostgreSQL 13.7, 12.11, 11.16, and 10.21
2022-04	<ul style="list-style-type: none"> • 13.6 • 12.10 • 11.15 • 10.20 • 9.6.24 	<p>Main updates:</p> <ul style="list-style-type: none"> • Synchronized the latest code in the community to fix security vulnerabilities and functional problems. <p>More release notes:</p> <ul style="list-style-type: none"> • Release Notes on PostgreSQL 13.6, 12.10, 11.15, 10.20, and 9.6.24
2021-04	<ul style="list-style-type: none"> • 13.2 • 12.6 • 11.11 • 10.16 • 9.6.21 • 9.5.25 	<p>Main updates:</p> <ul style="list-style-type: none"> • Introduced PostgreSQL 13. • Synchronized the latest code in the community to fix security vulnerabilities and functional problems. • Supported failover slots. For details, see Failover Slot for Logical Subscriptions. <p>More release notes:</p> <ul style="list-style-type: none"> • Release Notes on PostgreSQL 13.2, 12.6, 11.11, 10.16, 9.6.21, and 9.5.25
2020-03	12.2	/
2019-12	9.5.19	/
2019-11	<ul style="list-style-type: none"> • 11.5 • 10.10 • 9.6.15 	/

Released In	Supported Community Versions	Updates
2019-08	<ul style="list-style-type: none"> • 11.4 • 9.6.13 • 9.5.17 	/
2019-07	<ul style="list-style-type: none"> • 11.2 • 10.8 	/
2019-06	9.5.15	/
2019-03	<ul style="list-style-type: none"> • 10.6 • 9.6.11 	/
2019-01	11.0	/
2018-11	10.3	/
2018-03	9.6.5	/
2018-02	9.6.3	/
2017-10	9.5.5	/

2.2 Failover Slot for Logical Subscriptions

Scenarios

RDS for PostgreSQL synchronizes information about logical replication slots specified as failover slots from the primary DB instance to the standby DB instance to ensure logical subscriptions remain connected after a primary/standby switchover.

Information about logical replication slots will not be transferred to the new primary DB instance during a primary/standby switchover. If there is a primary/standby switchover, the logical subscriptions are disconnected, and you will need to manually create new slots. Failover slots make it possible to synchronize all logical slots from the primary DB instance to the standby DB instance, preventing logical subscriptions from being disconnected after a primary/standby switchover.

Precautions

- This function is supported in RDS for PostgreSQL 12.6 and later minor versions, and all minor versions of RDS for PostgreSQL 13 and later major versions.
- This function supports only logical subscriptions. It does not support physical subscriptions.
- This function introduces new log types. Therefore, if you enable this function for your instance, old slot information will be retained when you restore data

to a new instance from backups. You need to manually delete the old slot information after the restoration is complete.

How to Use

Run the following SQL statement on the publishing client to create a failover slot:

Table 2-2 Commands and parameters

Version	Command	Parameters
RDS for PostgreSQL 12 and RDS for PostgreSQL 13	SELECT * FROM pg_create_logical_replication_slot('slotname', 'pgoutput', false, true);	<ul style="list-style-type: none"> • <i>slotname</i> is the name of the logical slot. • <i>pgoutput</i> is the name of the plugin. You can also change it to another supported plugin. • The third parameter specifies if the slot is a temporary slot. • The fourth parameter specifies if the slot is a failover slot.
RDS for PostgreSQL 14 or later	SELECT * FROM pg_create_logical_replication_slot('slotname', 'pgoutput', false, false, true);	<ul style="list-style-type: none"> • <i>slotname</i> is the name of the logical slot. • <i>pgoutput</i> is the name of the plugin. You can also change it to another supported plugin. • The third parameter specifies if the slot is a temporary slot. • The fourth parameter specifies whether to enable two-phase commit. • The fifth parameter specifies if the replication slot is a failover slot.

To create a failover slot, the value of the third parameter must be **false** and that of the last parameter must be **true**. If you do not specify the last parameter, the created slot is not a failover slot.

How to Check

Run the following SQL statement on the publishing client to query the replication slot information of the database:

```
select * from pg_get_replication_slots();
```

- Check the **failover** field. If the value is **true**, the replication slot is a failover slot.
- If there is no the **failover** field or its value is **false**, the replication slot is not a failover slot.

Example of a Complete Logical Subscription

- Create a table on the publishing client.
create table tableName(id int primary key, num int);
- Create a publication on the publishing client.
create publication pubName for table tableName;
- Create a failover slot on the publishing client. (For RDS for PostgreSQL 12 or 13, delete the fourth parameter.)
SELECT * FROM pg_create_logical_replication_slot('slotname', 'pgoutput', false, false, true);
- Insert data into the table on the publishing client.
insert into tableName values(1,1);
insert into tableName values(2,2);
- Create a table on the subscription client.
create table tableName (id int primary key, num int);
- Create a subscription on the subscription client and specify the name of the failover slot.
create subscription subName connection 'host=192.168.0.10 dbname=postgres user=root port=5432 password=xxxxxxx' publication pubName with(copy_data=true,create_slot=false,slot_name= slotname);
- Query data on the subscription client and check whether the data is subscribed to.
select * from tableName;
- Perform a primary/standby switchover.
Insert data into the table on the publishing client and view the data on the subscription client. The logical subscription is not disconnected.

A Change History

Released On	Description
2023-09-28	This issue is the tenth release, which incorporates the following change: Added RDS for PostgreSQL Kernel .
2023-09-21	This issue is the ninth release, which incorporates the following change: Added RDS for MariaDB version updates in Kernel Version Description .
2023-06-28	This issue is the eighth official release, which incorporates the following change: Added RDS for MySQL 5.7 version updates in Kernel Version Description .
2022-09-09	This issue is the seventh official release, which incorporates the following change: Added version updates of RDS for MySQL 8.0, 5.7, and 5.6 in Kernel Version Description .
2022-06-01	This issue is the sixth official release, which incorporates the following change: Added version updates of RDS for MySQL 8.0, 5.7, and 5.6 in Kernel Version Description .
2022-01-26	This issue is the fifth official release, which incorporates the following change: Added RDS for MySQL 5.7 version updates in Kernel Version Description .
2021-11-26	This issue is the fourth official release, which incorporates the following change: Added RDS for MySQL 5.7 version updates in Kernel Version Description .

Released On	Description
2021-08-07	This issue is the third official release, which incorporates the following change: Added version updates of RDS for MySQL 8.0, 5.7, and 5.6 in Kernel Version Description .
2021-06-10	This issue is the second official release, which incorporates the following change: Optimized description of the <code>idle_transaction_timeout</code> parameter in Ending Idle Transactions .
2021-04-13	This issue is the first official release.