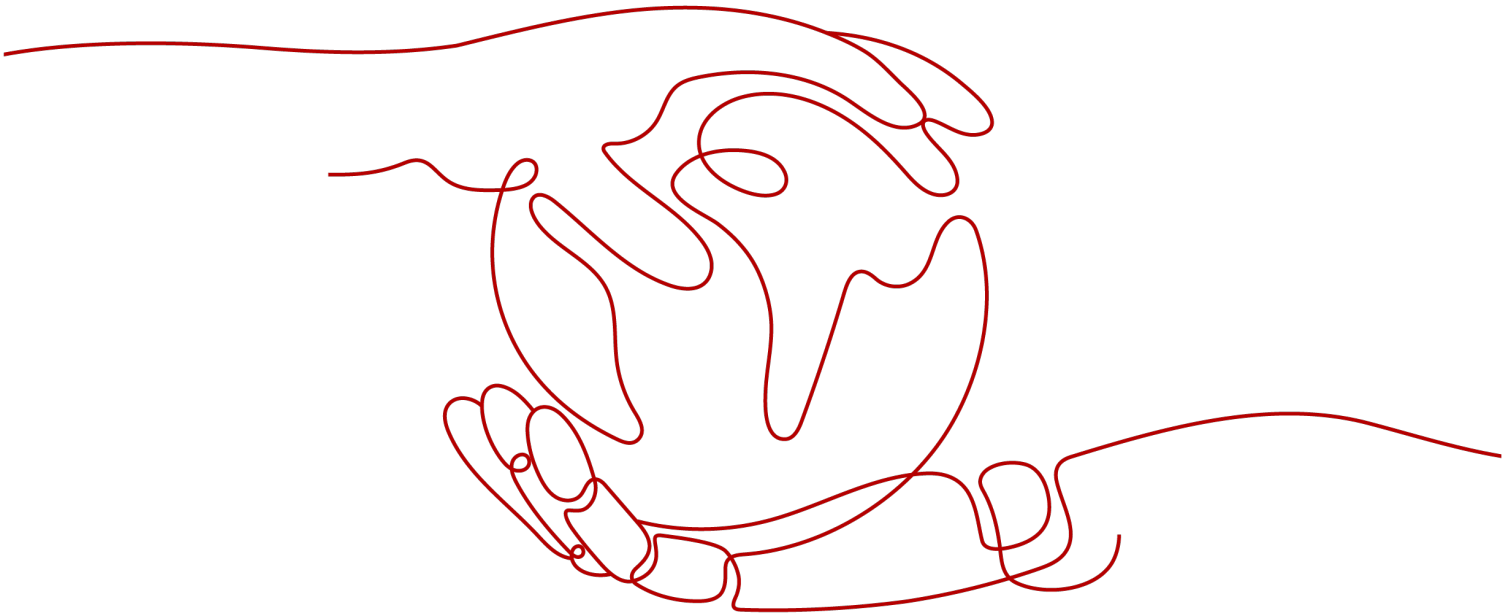
RDS for MySQL

Kernels

Issue	01
Date	2025-10-31



HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2025. 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 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.

Contents

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

1 RDS for MySQL Kernel Version Description

This section describes the kernel version updates of RDS.

- **RDS for MySQL 8.0**
- **RDS for MySQL 5.7**
- **RDS for MySQL 5.6**

RDS for MySQL 8.0

Table 1-1 RDS for MySQL 8.0 version description

Date	Description
2025-10-20	The kernel version was officially upgraded to 8.0.41. • New features and performance optimized – The community version 8.0.41 was introduced. For details, see **Changes in MySQL 8.0.41**. – If a large number of data operations (such as inserting, updating, or deleting millions of rows) are being performed in a single transaction and the transaction is terminated, the system enters the rollback phase. The rollback usually takes the same time as the original transaction execution. During this process, database locks are continuously held and a large number of I/O resources are occupied, which severely affects subsequent transactions. To avoid such performance problems, this version can identify and intercept large-scale transactions that may cause risks in the execution plan estimation phase and remind users of the risks, ensuring the database stability and high availability. For details, see **Intercepting Write Transactions During the Optimization Phase**. • Issues resolved – The partition scanning issue in certain level-2 partition scenarios was resolved.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Date	Description
2025-07-28	The kernel version was upgraded to 8.0.41. • New features and performance optimized The community version 8.0.41 was introduced. This version is available only for OBT. For details, see **Changes in MySQL 8.0.41**. • Issues resolved – The issue that a process exception may occur when an overlong dynamic masking rule is added was resolved. – The read-only performance in certain scenarios was optimized when too many dynamic masking rules are applied.
2025-04-23	Updates in RDS for MySQL 8.0.32: • Issues resolved – The issue that the database-level recycle bin does not take effect in certain scenarios was resolved.
2025-01-14	Updates in RDS for MySQL 8.0.32: • Issues resolved – The issue that the kernel reports a warning for passwords that do not meet complexity requirements was resolved. – The issue that certain **root** privileges are removed after a DB instance is upgraded from 8.0.28 to 8.0.32 was resolved. – Audit logging issues in certain scenarios were resolved. – The minimum value of **max_execution_time** was changed from 60 to 1 (unit: second). **NOTE** • To upgrade your DB instance to 8.0.32, **submit a service ticket** to apply for required permissions. • The performance of MySQL 8.0.32 deteriorates in some scenarios due to **instant_col**. For details, see **Bug 111538**.
2024-10-28	Updates in RDS for MySQL 8.0.32: • Issues resolved – The statement outline was optimized to support the EXPLAIN ANALYZE statement. – The deadlock problem in the 8.0.28.*x* community edition was resolved. – Some bugs in SQL statement concurrency control were fixed. **NOTE** • To upgrade your DB instance to 8.0.32, **submit a service ticket** to apply for required permissions. • The performance of MySQL 8.0.32 deteriorates in some scenarios due to **instant_col**. For details, see **Bug 111538**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Date	Description
2024-06-18	Updates in RDS for MySQL 8.0.28: • Issues resolved The issue that a deadlock may occur during **SHOW PROCESSLIST** execution due to LOCK_thd_security_ctx introduced by sessions was resolved. For details, see **Bug #32320541**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Date	Description
2024-05-23	The kernel version was upgraded to 8.0.32. • New features and performance optimized – Dynamic privileges of MySQL 8.0 allow the **SUPER** user to grant advanced administrative privileges, such as SESSION_VARIABLES_ADMIN and SYSTEM_VARIABLES_ADMIN, to other users. In RDS for MySQL 8.0.32, users are prevented from assigning risky permissions that may cause instance exceptions to themselves. – When columns are added to or deleted from a table, the default algorithm can be changed from INSTANT to INPLACE or COPY. • Issues resolved The issues resolved are the same as those of the community edition. For details, see **Changes in MySQL 8.0.32**. **NOTE** • To upgrade your DB instance to 8.0.32, **submit a service ticket** to apply for required permissions. • The performance of MySQL 8.0.32 deteriorates in some scenarios due to **instant_col**. For details, see **Bug 111538**. • Security hardening – The following security vulnerabilities were fixed: CVE-2018-25032, CVE-2021-22570, CVE-2022-21412, CVE-2022-21413, CVE-2022-21414, CVE-2022-21415, CVE-2022-21417, CVE-2022-21418, CVE-2022-21423, CVE-2022-21425, CVE-2022-21427, CVE-2022-21435, CVE-2022-21436, CVE-2022-21437, CVE-2022-21438, CVE-2022-21440, CVE-2022-21444, CVE-2022-21451, CVE-2022-21452, CVE-2022-21455, CVE-2022-21459, CVE-2022-21460, CVE-2022-21462, CVE-2022-21478, CVE-2022-21479, CVE-2022-21509, CVE-2022-21515, CVE-2022-21517, CVE-2022-21522, CVE-2022-21525, CVE-2022-21526, CVE-2022-21527, CVE-2022-21528, CVE-2022-21529, CVE-2022-21530, CVE-2022-21531, CVE-2022-21534, CVE-2022-21537, CVE-2022-21538, CVE-2022-21539, CVE-2022-21547, CVE-2022-21553, CVE-2022-21556, CVE-2022-21569, CVE-2022-21592, CVE-2022-21594, CVE-2022-21599, CVE-2022-21604, CVE-2022-21605, CVE-2022-21607, CVE-2022-21608, CVE-2022-21611, CVE-2022-21617, CVE-2022-21625, CVE-2022-21632, CVE-2022-21633, CVE-2022-21635, CVE-2022-21637, CVE-2022-21638, CVE-2022-21640, CVE-2022-21641, CVE-2022-27778, CVE-2022-32221, CVE-2022-39400, CVE-2022-39402, CVE-2022-39403, CVE-2022-39408, CVE-2022-39410, CVE-2023-21836, CVE-2023-21863, CVE-2023-21864, CVE-2023-21865, CVE-2023-21866, CVE-2023-21867, CVE-2023-21868, CVE-2023-21869, CVE-2023-21870, CVE-2023-21871, CVE-2023-21872, CVE-2023-21873, CVE-2023-21874, CVE-2023-21875, CVE-2023-21876, CVE-2023-21877,

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Date	Description
	CVE-2023-21878, CVE-2023-21879, CVE-2023-21880, CVE-2023-21881, CVE-2023-21882, CVE-2023-21883, CVE-2023-21887, CVE-2023-21912, CVE-2023-21913, CVE-2023-21917, CVE-2023-21963, CVE-2023-22015, CVE-2023-22026, CVE-2023-22028, and CVE-2023-22084.
2023-12-01	Updates in RDS for MySQL 8.0.28: ● New features and performance optimized – SQL statement concurrency control rules are applied to all users (the **root** user is excluded in the earlier version). To exclude certain users, contact customer service. – The **information_schema.rds_sql_filter_info** system table is added, which allows you to view the current number of concurrent SQL statements and the number of historical SQL statements throttled by SQL statement concurrency control. ● Issues resolved The issue that the audit log plugin did not record prepared statements was resolved.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Date	Description
2023-09-01	The kernel version was upgraded to 8.0.28. • New features and performance optimized – SQL statement concurrency control was optimized. – The asynchronous purge performance of large files was optimized. • Issues resolved – The bugfix of the community version 8.0.28 was introduced. For details, see **Changes in MySQL 8.0.28**. – Data inconsistency caused by parallel DDL was resolved. – The memory leakage and thread suspension issues of audit logs were resolved. • Security hardening – The following security vulnerabilities were fixed: CVE-2019-17543, CVE-2020-1971, CVE-2021-22901, CVE-2021-22922, CVE-2021-22923, CVE-2021-22924, CVE-2021-22925, CVE-2021-22926, CVE-2021-22945, CVE-2021-22946, CVE-2021-22947, CVE-2021-2339, CVE-2021-2340, CVE-2021-2342, CVE-2021-2352, CVE-2021-2354, CVE-2021-2356, CVE-2021-2357, CVE-2021-2367, CVE-2021-2370, CVE-2021-2372, CVE-2021-2374, CVE-2021-2383, CVE-2021-2384, CVE-2021-23841, CVE-2021-2385, CVE-2021-2387, CVE-2021-2389, CVE-2021-2390, CVE-2021-2399, CVE-2021-2402, CVE-2021-2410, CVE-2021-2412, CVE-2021-2417, CVE-2021-2418, CVE-2021-2422, CVE-2021-2424, CVE-2021-2425, CVE-2021-2426, CVE-2021-2427, CVE-2021-2429, CVE-2021-2437, CVE-2021-2440, CVE-2021-2441, CVE-2021-2478, CVE-2021-2479, CVE-2021-2481, CVE-2021-3449, CVE-2021-35537, CVE-2021-35546, CVE-2021-35575, CVE-2021-35577, CVE-2021-35583, CVE-2021-35591, CVE-2021-35596, CVE-2021-35602, CVE-2021-35604, CVE-2021-35607, CVE-2021-35610, CVE-2021-35612, CVE-2021-35622, CVE-2021-35623, CVE-2021-35624, CVE-2021-35625, CVE-2021-35626, CVE-2021-35627, CVE-2021-35628, CVE-2021-35629, CVE-2021-35630, CVE-2021-35631, CVE-2021-35632, CVE-2021-35633, CVE-2021-35634, CVE-2021-35635, CVE-2021-35636, CVE-2021-35637, CVE-2021-35638, CVE-2021-35639, CVE-2021-35640, CVE-2021-35641, CVE-2021-35642, CVE-2021-35643, CVE-2021-35644, CVE-2021-35645, CVE-2021-35646, CVE-2021-35647, CVE-2021-35648, CVE-2021-3711, CVE-2022-0778, CVE-2022-21245, CVE-2022-21249, CVE-2022-21253, CVE-2022-21254, CVE-2022-21264, CVE-2022-21265, CVE-2022-21278, CVE-2022-21297, CVE-2022-21301, CVE-2022-21302, CVE-2022-21303, CVE-2022-21304, CVE-2022-21339,

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Date	Description
	CVE-2022-21342, CVE-2022-21344, CVE-2022-21348, CVE-2022-21351, CVE-2022-21352, CVE-2022-21358, CVE-2022-21362, CVE-2022-21367, CVE-2022-21368, CVE-2022-21370, CVE-2022-21372, CVE-2022-21374, CVE-2022-21378, CVE-2022-21595, CVE-2022-21600, and CVE-2023-21950.
2023-03-15	Updates in RDS for MySQL 8.0.25: ● New features and performance optimized – 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. Users can no longer be granted the **connection_admin** permission.
2022-09-09	Updates in RDS for MySQL 8.0.25: ● New features and performance optimized – 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 – Errors caused by writes to a temporary file were resolved. – Unexpected responses to Common Table Expression (CTE) queries were resolved. ● Security hardening – 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.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Date	Description
2022-06-01	• New features and performance optimized – 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 – 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 – The following security vulnerabilities were fixed: CVE-2021-2307, CVE-2021-2180, and CVE-2021-2194.
2021-08-07	• New features and performance optimized – Static connection in 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_tablespaces** 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 – 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.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Date	Description
2021-04-13	• 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	• SQL statement concurrency control was optimized. • Full SQL collection was optimized.
2020-12-31	• Performance optimized The compiler was upgraded to GCC 9.
2020-12-01	• 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.
2020-11-06	• Issues resolved Timing error caused by gettimeofday in multiple threads was fixed.
2020-09-21	• Disconnection details were recorded in error logs. • Index hints were supported.
2020-08-03	• 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.
2020-06-19	• The kernel version was upgraded to 8.0.20. • Kernel performance was optimized.
2020-02-15	• RDS for MySQL 8.0 was put into commercial use. • The Huawei Cloud Kunpeng-powered Arm kernel version was released.
2019-12-15	• The kernel version was upgraded to 8.0.17. • The parallel index creation was 2.5 times faster.
2019-10-15	The issue that the primary/standby replication was abnormal when **SQL_MODE** was set to **PAD_CHAR_TO_FULL_LENGTH** was resolved.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Date	Description
2019-09-15	• Thread pools were supported. For details, see **Connection Thread Pool**. • 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
2025-04-23	Updates in RDS for MySQL 5.7.44: • Issues resolved – The issue that metadata at the server layer was inconsistent with that at the InnoDB layer due to non-atomic DDL was resolved by clearing the metadata at the server layer.
2025-01-14	Updates in RDS for MySQL 5.7.44: • Issues resolved – The issue that the kernel reports a warning for passwords that do not meet complexity requirements was resolved. – Audit logging issues in certain scenarios were resolved. – The minimum value of **max_execution_time** was changed from 60 to 1 (unit: second).
2024-10-28	Updates in RDS for MySQL 5.7.44: • Issues resolved – Some bugs in SQL statement concurrency control were fixed.
2024-04-11	The kernel version was upgraded to 5.7.44. • The changes are the same as those of the community edition. For details, see **Changes in MySQL 5.7.44**. • Security hardening – The following security vulnerabilities were fixed: CVE-2023-22028, CVE-2023-22084, and CVE-2023-38545.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Date	Description
2023-12-01	• New features and performance optimized – The kernel version was upgraded to 5.7.43. – SQL statement concurrency control rules are applied to all users (the **root** user is excluded in the earlier version). To exclude certain users, contact customer service. – The **information_schema.rds_sql_filter_info** system table is added, which allows you to view the current number of concurrent SQL statements and the number of historical SQL statements throttled by SQL statement concurrency control. • Issues resolved The issue that the audit log plugin did not record prepared statements was resolved. • Security hardening – The following security vulnerabilities were fixed: CVE-2022-43551, CVE-2023-21912, CVE-2023-21980, CVE-2023-22007, CVE-2023-22015, CVE-2023-22026, and CVE-2023-22053.
2023-09-01	Updates in RDS for MySQL 5.7.41: • New features and performance optimized – SQL statement concurrency control was optimized. – Slow memory release of jemalloc, which causes out of memory (OOM) exceptions, was optimized. • Issues resolved – The memory leakage and thread suspension issues of audit logs were resolved. – Too large gap lock ranges were resolved.
2023-06-28	• New features and performance optimized – The kernel version was upgraded to 5.7.41. – Compiler security options were added. • Issues resolved – 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 – 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.

Date	Description
2022-09-09	• New features and performance optimized – 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 – 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 – 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.
2022-06-01	• New features and performance optimized – 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 – 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 – The following security vulnerabilities were fixed: CVE-2022-21367, CVE-2022-21304, and CVE-2022-21344.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Date	Description
2022-01-26	• New features and performance optimized – Hidden auto-increment keys were supported. • Issues resolved – Replication exceptions caused by repeated commitment of XA transactions were resolved. – Inaccurate innodb_row_lock_current_waits statistics were fixed.
2021-11-26	• New features and performance optimized – 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 – Memory problems of the thread pool in extreme scenarios were resolved. – Occasional suspension of XA transaction playbacks on a standby node was resolved.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Date	Description
2021-08-07	• New features and performance optimized – Static connection in 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 – 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 The following security vulnerabilities were fixed: CVE-2021-2011, CVE-2021-2178, and CVE-2021-2202.
2021-04-13	• 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	• New features The real client address can be displayed when proxy is used. • Issues resolved Full SQL collection was optimized. The issue that revoking permissions might cause permission inconsistency between primary and standby DB instances was resolved. MySQL 8.0 optimizations on Instant Add Column was incorporated.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Date	Description
2020-12-31	• Performance optimized – 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.
2020-12-01	The conflict frequency between fil_sys mutual exclusions was reduced.
2020-11-06	• New features – 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.
2020-09-21	• The kernel version was upgraded to 5.7.31. • SQL filter was used to restrict the execution frequency of SQL statements during peak hours.
2020-08-03	• Kernel performance was optimized. • The recycle bin was supported. • Issues that may occur during local disk data cleanup were resolved.
2020-07-09	• Kernel performance was optimized. • Users' operation history can be recorded in error logs. • Distributed transaction stability was improved.
2020-06-30	• 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.
2020-05-30	• New features – Index hints were supported. – Full SQL logs can be captured. • Issues resolved Occasional database connection failures were resolved.
2020-04-30	Kernel performance was optimized.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Date	Description
2020-03-30	• The kernel version was upgraded to 5.7.29. • Kernel performance was optimized. • Threshold pools supported statistics on I/O waits.
2020-02-15	• 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	• The progress of adding columns or indexes can be obtained through **information_schema.innodb_alter_table_progress**. For details, see **DDL Progress Display**. • 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. For details, see **Online Varchar Length Increase**. • 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	• Performance optimized – The Huawei Cloud Kunpeng-powered Arm kernel version was released. • New features – 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**. For details, see **MDL Views**.
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	• The kernel version was upgraded to 5.7.25.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Date	Description
2019-05-15	• New features – 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 – 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	• 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	• 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	• The kernel version was upgraded to 5.7.22. • Thread pools were supported. For details, see **Connection Thread Pool**. • 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. For details, see **Ending Idle Transactions**. • The memory usage and CPU time usage of user threads can be queried through **show full processlist**.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

RDS for MySQL 5.6

Table 1-3 RDS for MySQL 5.6 version description

Date	Description
2024-10-28	Updates in RDS for MySQL 5.6.51: • Issues resolved – DB instance crashes due to the crash bug of the **row_search_mvcc** function in the open-source community were resolved by backporting. For details, see **Bug #96610**. – The calculation of long-running transactions after XA transactions are committed was optimized.
2023-09-01	Updates in RDS for MySQL 5.6.51: • New features and performance optimized – SQL statement concurrency control was optimized. • Issues resolved – Thread suspension for audit logs was resolved.
2023-03-15	Updates in RDS for MySQL 5.6.51: • New features and performance optimized – Printing of oversized SQL audit logs was optimized. – The security of log printing was enhanced. • Issues resolved – 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	• New features and performance optimized – Connection per thread was supported for killing sessions. • Issues resolved – 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	Inaccurate innodb_row_lock_current_waits statistics were fixed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Date	Description
2021-08-07	• New features – 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 – 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	• 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.
2021-01-26	• 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	• 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	• Full SQL collection was supported. • The compiler was upgraded to 7.3. • The kernel version was upgraded to 5.6.47.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Date	Description
2020-02-15	• 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	• 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. For details, see **Online Varchar Length Increase**. • 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	• 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. For details, see **Ending Idle Transactions**.
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	• The kernel version was upgraded to 5.6.43. • The audit function was supported.
2019-05-15	• 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	• 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.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Date	Description
2018-11-15	• 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	• New features – The kernel version was upgraded to 5.6.40. – Thread pools were supported. For details, see **Connection Thread Pool**. – The Common Type System (CTS) syntax **create table xx select** was supported. • Issues resolved – 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.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2 Main Kernel Functions

2.1 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     |
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
| 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 2-1 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.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Table 2-2 Thread pool parameters that can be modified

Parameter	Dynamic Parameter	Data Type	Value Range	Description
threadpool_enabled	Yes	boolean	[ON,OFF]	• **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 2-3 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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

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 2-4 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

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2.2 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 2-5 metadata_lock_info fields

No.	Field Name	Type	Description
0	THREAD_ID	bigint(20) unsigned	Session ID.
1	LOCK_STATUS	varchar(24)	Two statuses of MDL: • **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.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

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: • **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 2-6 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)

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Case Analysis

Description

After TRUNCATE operations on table t2 are blocked, SELECT operations on table t2 are also blocked in the service process.

Problem Analysis

- **Without the MDL view**

  If DDL operations are blocked, run the **show processlist** command. Information in the following figure is displayed.

```
show processlist;
+------+--------+--------------+--------+-----------+----------
+----------------------------------|-------------------------|
| Id   | User   |  Host        | db    |  Command  |   Time   |   State                       |Info                  |
+---------------+----------------------+-----------+----------+-----------------------------------
+-------------------------|
| 2    | root   |  localhost   |  test  |  Sleep    |   73     |                               | Null                 |
| 3    | root   |  localhost   |  test  |  Sleep    |   63     |                               | Null                 |
| 4    | root   |  localhost   |  Null  |  Query    |   35     | Waiting for table metadata lock   | truncate
table test.t2  |
| 5    | root   |  localhost   |  test  |  Query    |   17     | Waiting for table metadata lock   | select * from
test.t2   |
| 6    | root   |  localhost   |  test  |  Query    |   0      | starting                      | show processlist       |
+------+--------+--------------+--------+-----------+----------
+----------------------------------|-------------------------|
```

  According to the preceding thread list:

  – When executing TRUNCATE, session 4 is blocked by the table metadata lock held by other sessions.

  – When executing SELECT, session 5 is also blocked by the table metadata lock held by other sessions.

  – You cannot determine which session blocks session 4 and session 5.

  In this case, killing other sessions randomly will cause great risks to online services. Therefore, you can only wait for other sessions to release the MDL.

- **With the MDL view**

  Run the **select * from information_schema.metadata_lock_info** command to view the MDL information. The following information is displayed.

```
select * from information_schema.metadata_lock_info;
+-------------+-------------+-------------------------+---------------------+-------------------+----------------
+----------------+
| THREAD_ID   | LOCK_STATUS |  LOCK_MODE              | LOCK_TYPE           |  LOCK_DURATION    |
TABLE_SCHEMA  |   TABLE_NAME   |
+-------------+-------------+-------------------------+---------------------+-------------------+----------------
+----------------+
| 2           | GRANTED     |  MDL_SHARED_READ        | Table metadata lock |  MDL_TRANSACTION  |
test          |   t1           |
| 3           | GRANTED     |  MDL_SHARED_READ        | Table metadata lock |  MDL_TRANSACTION  |
test          |   t2           |
| 4           | GRANTED     |  MDL_INTENTION_EXCLUSIVE | Global read lock    |  MDL_STATEMENT
|               |              |
| 4           | GRANTED     |  MDL_INTENTION_EXCLUSIVE | Schema metadata lock |
MDL_TRANSACTION  |  test          |              |
| 4           | PENDING     |  MDL_EXCLUSIVE          | Table metadata lock |                |  test         |
t2           |
| 5           | PENDING     |  MDL_SHARED_READ        | Table metadata lock |                |  test
|   t2           |
+-------------+-------------+-------------------------+---------------------+-------------------+----------------
+----------------+
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

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.

2.3 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.

Supported Versions

You are advised to use the latest minor version. For details about how to upgrade a minor version, see **Upgrading a Minor Version**.

Table 2-7 Supported versions for online varchar length increase

Varchar Length	RDS for MySQL 5.6	RDS for MySQL 5.7	RDS for MySQL 8.0
Less than 256 bytes	Not supported	Supported	Supported
From less than 256 bytes to more than 256 bytes	Supported	Supported	Not supported
More than 256 bytes	Not supported	Supported	Supported

Category

- Increase the varchar length to less 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
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

- 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
```

2.4 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 2-8 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.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Type	Value Range	Dynamic Validation	Description
idle_transaction_timeout	integer	Positive integer	Yes	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. **NOTE** The parameters **idle_readonly_transaction_timeout** and **idle_write_transaction_timeout** have higher priorities than the parameter **idle_transaction_timeout**. ● 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	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.

Application Scenarios

The parameters are set as follows:

```
show variables like '%idle%';
+-----------------------------------+-------+
| Variable_name                     | Value |
+-----------------------------------+-------+
| idle_readonly_transaction_timeout | 5     |
| idle_transaction_timeout          | 10    |
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

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

- Setting **idle_readonly_transaction_timeout**

  Set **idle_readonly_transaction_timeout** to **5**.

  a. 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)
  ```

  b. 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.

    i. 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)
    ```

    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
    ```

  – Read/write transactions

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

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)
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2.5 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.

Constraints

This function is only available to RDS for MySQL 5.7.

Characteristics

Table 2-9 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.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

2.6 Intercepting Write Transactions During the Optimization Phase

Introduction

In MySQL, if a single transaction involves a large number of data operations (such as inserting, updating, or deleting millions of rows), it may hold locks for extended periods and heavily consume I/O resources, severely affecting the normal execution of subsequent operations. Users often have to forcibly terminate the session, which will trigger rollback operations. The rollback typically takes as long as the original transaction execution. During this process, the database locks are continuously held and I/O resources are still occupied, further blocking other workloads.

To avoid such performance bottlenecks, Huawei Cloud RDS for MySQL intercepts large-scale write transactions before they are executed, preventing the system performance from deteriorating due to excessive data changes. This feature promptly alerts you to potential risks, helping ensure database stability and high availability. However, enabling this feature may slightly affect write performance.

Prerequisites

- For RDS for MySQL 5.7, the kernel version must be 5.7.44.241207 or later.
- For RDS for MySQL 8.0, the kernel version must be 8.0.41.250900 or later.

Parameters

Table 2-10 Parameter description

Parameter	Dynamic Parameter	Data Type	Value Range	Description
rds_max_data_mutation_num_per_dml	Yes	integer	[0, 10000-9999999999999]	Maximum number of rows that can be affected by INSERT, UPDATE, or DELETE. If this number is exceeded, an error is reported. This is a session-level parameter. The default value is **0**, indicating that no statements will be intercepted. The value can be **0** or [10000, 9999999999999].

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

Parameter	Dynamic Parameter	Data Type	Value Range	Description
rds_data_mutation_counting_mode	Yes	enum	[PRECISE, ESTIMATED]	Evaluation mode for the number of rows affected by INSERT, UPDATE, or DELETE. If the accurate row count cannot be obtained, the estimated value of the optimizer is used in ESTIMATED mode, and no interception is performed in PRECISE mode. This is a session-level parameter. The default value is **ESTIMATED**.

Restrictions

Only statement-level restrictions are supported. Risks are identified before the following statements are executed:

Table 2-11 Restrictions

SQL Statement	Restrictions
INSERT ... VALUES	Only the number of rows added by the VALUES statement is limited, not the number of rows actually inserted.
REPLACE ... VALUES	Only the number of rows added by the VALUES statement is limited, not the number of rows actually inserted.
INSERT ... SELECT	• Only the number of output rows estimated by the optimizer for the SELECT statement is limited. • If the SELECT statement results are not cached and the execution plan contains multiple-table joins, no restriction is imposed.
REPLACE ... SELECT	• Only the number of output rows estimated by the optimizer for the SELECT statement is limited. • If the SELECT statement results are not cached and the execution plan contains multiple-table joins, no restriction is imposed.

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

SQL Statement	Restrictions
CREATE TABLE ... SELECT	• Only the number of output rows estimated by the optimizer for the SELECT statement is limited. • If the SELECT statement results are not cached and the execution plan contains multiple-table joins, no restriction is imposed.
UPDATE TABLE ...	• Only a single table is supported. • Only the number of output rows estimated by the optimizer for the entire table or filtered by the WHERE condition is limited.
DELETE FROM TABLE ...	• Only a single table is supported. • Only the number of output rows estimated by the optimizer for the entire table or filtered by the WHERE condition is limited.

As SQL statements are not executed, the system can only rely on the optimizer's estimated values for interception. The interception cannot be absolutely accurate. However, the accurate row count can be obtained in the following scenarios:

- The INSERT ... VALUES operation is performed on a table without any restriction.
- In the INSERT/REPLACE ... SELECT or CREATE TABLE ... SELECT statement, the SELECT results need to be cached (for example, materialized temporary tables and SORT cache) before being inserted.
- Before the UPDATE or DELETE statement is executed, the row IDs of all rows to be updated need to be cached (for example, temporary files and SORT cache).
- In a single-table query, indexes (such as ref or range) can be directly used for access, without extra filtering.

In the preceding scenarios, regardless of whether the PRECISE or ESTIMATED mode is configured, the system determines whether to intercept the statements based on the obtained number of rows.

Examples

The following are examples of intercepting write transactions in the optimization phase:

1. Table structure

```
mysql> SHOW CREATE TABLE t1\G
*************************** 1. row ***************************
       Table: t1
Create Table: CREATE TABLE `t1` (
  `id` int NOT NULL AUTO_INCREMENT,
  `category` int NOT NULL,
  `value` int NOT NULL,
  `flag` enum('active','inactive') DEFAULT 'active',
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
  `created_at` timestamp NOT NULL DEFAULT '2025-05-05 01:01:01',
  PRIMARY KEY (`id`),
  KEY `idx_category` (`category`),
  KEY `idx_value` (`value`),
  KEY `idx_flag` (`flag`),
  KEY `idx_category_value` (`category`,`value`)
) ENGINE=InnoDB AUTO_INCREMENT=120109 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci
1 row in set (0.00 sec)

mysql> CREATE TABLE t2 LIKE t1;
Query OK, 0 rows affected (0.15 sec)
```

2. Example of intercepting the INSERT ... SELECT statement

```
mysql> EXPLAIN FORMAT=TREE INSERT INTO t2 SELECT * FROM t1 WHERE category=1;
+-----------------------------------------------------------------------------------------------------------+
| EXPLAIN                                                                                                   |
+-----------------------------------------------------------------------------------------------------------+
| -> Insert into t2
    -> Index lookup on t1 using idx_category (category=1)  (cost=6276 rows=60112)
 |
+-----------------------------------------------------------------------------------------------------------+
1 row in set (0.02 sec)

mysql> SET rds_max_data_mutation_num_per_dml=60111;
Query OK, 0 rows affected (0.00 sec)

# If a single table uses the ref index without filter criteria and the estimated value exceeds the
threshold, the statement is intercepted in ESTIMATED mode.
mysql> SET rds_data_mutation_counting_mode='ESTIMATED';
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO t2 SELECT * FROM t1 WHERE category=1;
ERROR 4167 (HY000): The number [60112] of data mutations in the SQL statement has exceeded the
maximum limit. Please check the configuration parameter 'rds_max_data_mutation_num_per_dml'.
mysql>

# If a single table uses the ref index without filter criteria and the estimated value exceeds the
threshold, the statement is intercepted in PRECISE mode.
mysql> SET rds_data_mutation_counting_mode='PRECISE';
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO t2 SELECT * FROM t1 WHERE category=1;
ERROR 4167 (HY000): The number [60112] of data mutations in the SQL statement has exceeded the
maximum limit. Please check the configuration parameter 'rds_max_data_mutation_num_per_dml'.
mysql>

mysql> EXPLAIN FORMAT=TREE INSERT INTO t2 SELECT * FROM t1 WHERE category=1 AND
flag='active';
+-----------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------+
|
EXPLAIN
                    |
+-----------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------+
| -> Insert into t2
    -> Filter: (t1.flag = 'active')  (cost=3270 rows=30056)
        -> Index lookup on t1 using idx_category (category=1)  (cost=3270 rows=60112)
 |
+-----------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------+
1 row in set (0.00 sec)

mysql> SET rds_max_data_mutation_num_per_dml=30055;
Query OK, 0 rows affected (0.00 sec)

# If there are filter criteria that cannot be indexed and the estimated value exceeds the threshold, the
statement is intercepted in ESTIMATED mode.
mysql> SET rds_data_mutation_counting_mode='ESTIMATED';
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO t2 SELECT * FROM t1 WHERE category=1 AND flag='active';
ERROR 4167 (HY000): The number [30056] of data mutations in the SQL statement has exceeded the maximum limit. Please check the configuration parameter 'rds_max_data_mutation_num_per_dml'.
mysql>
mysql> SET rds_data_mutation_counting_mode='PRECISE';
Query OK, 0 rows affected (0.00 sec)

# If there are filter criteria that cannot be indexed and the estimated value exceeds the threshold, the statement is not intercepted in PRECISE mode.
mysql> INSERT INTO t2 SELECT * FROM t1 WHERE category=1 AND flag='active';
Query OK, 32275 rows affected (37.11 sec)
Records: 32275  Duplicates: 0  Warnings: 0

mysql>
```

3. Example of intercepting the UPDATE statement

```
mysql> EXPLAIN FORMAT=TREE SELECT * FROM t1 WHERE category=1;
+-----------------------------------------------------------------------------+
| EXPLAIN                                                                     |
+-----------------------------------------------------------------------------+
| -> Index lookup on t1 using idx_category (category=1)  (cost=4710 rows=44448)
 |
+-----------------------------------------------------------------------------+
1 row in set (0.00 sec)

mysql> SET rds_max_data_mutation_num_per_dml=44447;
Query OK, 0 rows affected (0.00 sec)

# If a single table uses the ref index without filter criteria and the estimated value exceeds the threshold, the statement is intercepted in ESTIMATED mode.
mysql> SET rds_data_mutation_counting_mode='ESTIMATED';
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE t1 SET value=value+100 WHERE category=1;
ERROR 4167 (HY000): The number [44448] of data mutations in the SQL statement has exceeded the maximum limit. Please check the configuration parameter 'rds_max_data_mutation_num_per_dml'.
mysql>

# If a single table uses the ref index without filter criteria and the estimated value exceeds the threshold, the statement is intercepted in PRECISE mode.
mysql> SET rds_data_mutation_counting_mode='PRECISE';
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE t1 SET value=value+100 WHERE category=1;
ERROR 4167 (HY000): The number [44448] of data mutations in the SQL statement has exceeded the maximum limit. Please check the configuration parameter 'rds_max_data_mutation_num_per_dml'.
mysql>
mysql>  EXPLAIN FORMAT=TREE SELECT * FROM t1 WHERE category=1 AND flag='active';
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| EXPLAIN                                                                                                                                                               |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| -> Filter: (t1.flag = 'active')  (cost=2487 rows=22224)
    -> Index lookup on t1 using idx_category (category=1)  (cost=2487 rows=44448)
 |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------+
1 row in set (0.01 sec)

mysql>
mysql> SET rds_max_data_mutation_num_per_dml=22223;
Query OK, 0 rows affected (0.00 sec)

mysql>

# If there are filter criteria that cannot be indexed and the estimated value exceeds the threshold, the statement is intercepted in ESTIMATED mode.
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
mysql> SET rds_data_mutation_counting_mode='ESTIMATED';
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE t1 SET value=value+100 WHERE category=1 AND flag='active';
ERROR 4167 (HY000): The number [22224] of data mutations in the SQL statement has exceeded the
maximum limit. Please check the configuration parameter 'rds_max_data_mutation_num_per_dml'.
mysql>

# If there are filter criteria that cannot be indexed and the estimated value exceeds the threshold, the
statement is not intercepted in PRECISE mode.
mysql> SET rds_data_mutation_counting_mode='PRECISE';
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE t1 SET value=value+100 WHERE category=1 AND flag='active';
Query OK, 32275 rows affected (43.55 sec)
Rows matched: 32275  Changed: 32275  Warnings: 0
```

4. Example of intercepting the DELETE statement

```
mysql> EXPLAIN FORMAT=TREE SELECT * FROM t1 WHERE category=2;
+-----------------------------------------------------------------------------+
| EXPLAIN                                                                     |
+-----------------------------------------------------------------------------+
| -> Index lookup on t1 using idx_category (category=2)  (cost=6276 rows=60112)
 |
+-----------------------------------------------------------------------------+
1 row in set (0.00 sec)

mysql>
mysql> SET rds_max_data_mutation_num_per_dml=60111;
Query OK, 0 rows affected (0.00 sec)

# If a single table uses the ref index without filter criteria and the estimated value exceeds the
threshold, the statement is intercepted in ESTIMATED mode.
mysql> SET rds_data_mutation_counting_mode='ESTIMATED';
Query OK, 0 rows affected (0.00 sec)

mysql> DELETE FROM t1 WHERE category=2;
ERROR 4167 (HY000): The number [60112] of data mutations in the SQL statement has exceeded the
maximum limit. Please check the configuration parameter 'rds_max_data_mutation_num_per_dml'.
mysql>

# If a single table uses the ref index without filter criteria and the estimated value exceeds the
threshold, the statement is intercepted in PRECISE mode.
mysql> SET rds_data_mutation_counting_mode='PRECISE';
Query OK, 0 rows affected (0.00 sec)

mysql> DELETE FROM t1 WHERE category=2;
ERROR 4167 (HY000): The number [60112] of data mutations in the SQL statement has exceeded the
maximum limit. Please check the configuration parameter 'rds_max_data_mutation_num_per_dml'.
mysql>
mysql> EXPLAIN FORMAT=TREE SELECT * FROM t1 WHERE category=2 AND flag='active';
+-----------------------------------------------------------------------------------------------------------------
----------------------+
| EXPLAIN                                                                                                        |
+-----------------------------------------------------------------------------------------------------------------
----------------------+
| -> Filter: (t1.flag = 'active')  (cost=3270 rows=30056)
    -> Index lookup on t1 using idx_category (category=2)  (cost=3270 rows=60112)
 |
+-----------------------------------------------------------------------------------------------------------------
----------------------+
1 row in set (0.12 sec)

mysql> SET rds_max_data_mutation_num_per_dml=30055;
Query OK, 0 rows affected (0.00 sec)

# If there are filter criteria that cannot be indexed and the estimated value exceeds the threshold, the
statement is intercepted in ESTIMATED mode.
mysql> SET rds_data_mutation_counting_mode='ESTIMATED';
Query OK, 0 rows affected (0.00 sec)
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.

```
mysql> DELETE FROM t1 WHERE category=2 AND flag='active';
ERROR 4167 (HY000): The number [30056] of data mutations in the SQL statement has exceeded the maximum limit. Please check the configuration parameter 'rds_max_data_mutation_num_per_dml'.
mysql>

# If there are filter criteria that cannot be indexed and the estimated value exceeds the threshold, the statement is not intercepted in PRECISE mode.
mysql> SET rds_data_mutation_counting_mode='PRECISE';
Query OK, 0 rows affected (0.00 sec)

mysql> DELETE FROM t1 WHERE category=2 AND flag='active';
Query OK, 32040 rows affected (41.72 sec)
```

 Copyright © Huawei Cloud Computing Technologies Co., Ltd.