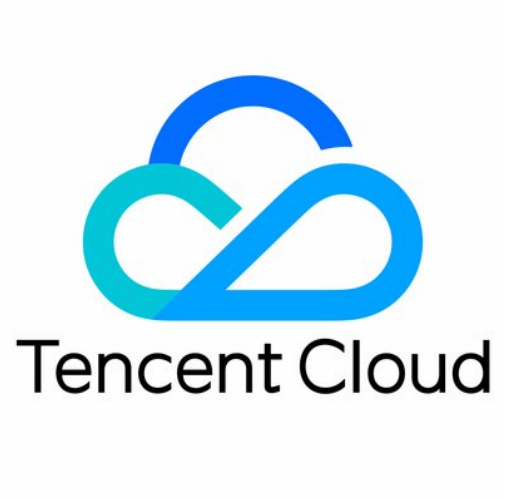
TencentDB for MySQL

Tencent Kernel TXSQL

Product Documentation



Contents

Tencent Kernel TXSQL Overview

Last updated : 2020-08-25 11:36:33

TXSQL is a MySQL kernel branch maintained by the TencentDB team and is fully compatible with native MySQL. It provides various features similar to those in the MySQL Enterprise Edition, such as enterprise-grade transparent data encryption, auditing, thread pool, encryption function, and backup and restoration.

TXSQL not only deeply optimizes the InnoDB storage engine, query performance, and replication performance, but also improves the ease of use and maintainability of TencentDB for MySQL. While providing all the benefits of MySQL, it offers more enterprise-grade advanced features such as disaster recovery, restoration, monitoring, performance optimization, read/write separation, transparent data encryption, and auditing.

The following provides more information about TXSQL:

- For details on the TencentDB for MySQL kernel version updates, please see Kernel Version Updates.
- TencentDB for MySQL supports the `instant` algorithm to quickly add columns in big tables, while avoiding data replication. For more information, please see Quick Column Adding Feature.
- The kernel minor versions of TencentDB for MySQL can be upgraded automatically or manually. For more information, please see Upgrading Kernel Minor Version.
- You can use a CVM instance to log in to a TencentDB for MySQL instance and check its kernel minor version. For more information, please see How do I check the kernel minor version?.

Kernel Version Updates

Last updated : 2021-04-02 17:08:55

This document describes the MySQL kernel version updates. For information on how to upgrade the kernel, please see Upgrading Kernel Minor Version.

MySQL 8.0

20200630

New features

- Supports async deletion of big tables. You can clear files asynchronously and slowly to avoid business performance fluctuation caused by deleting big tables. To apply for this feature, please submit a ticket.
- Supports automatic killing of idle tasks to reduce resource conflicts. To apply for this feature, please submit a ticket.
- Supports transparent data encryption (TDE).

Bug fixes

- Fixed the error where switch failed due to inconsistent checkpoints between `relay_log_pos` and `master_log_pos` .
- Fixed the data file error caused by asynchronously storing data in the disk.
- Fixed the hard error when `fsync` returned `EIO` and retries were made repeatedly.
- Fixed the crash caused by phrase search under multi-byte character sets in full-text index.

MySQL 5.7

20201231

New features

- Supports using `NOWAIT` and `SKIP LOCKED` in `SELECT FOR UPDATE/SHARE` statements.
- Supports dynamic setting of thread pooling mode or connection pooling mode by using the `thread_handling` parameter.
- Supports source-replica buffer pool synchronization.
- Supports monitoring of user connection status. Monitoring items include sync/async IO, memory, log size, CPU time, lock duration, etc.

Performance optimizations

- Optimizes the transaction subsystem to improve the high concurrency performance.
- Optimizes the time to start crash recovery for large transactions.
- Optimizes redo log flushing.
- Optimizes the buffer pool initialization time.
- Optimizes UTF8/UTF8MB4 string efficiency.
- Optimizes audit performance.
- The value of `gtid_purged` does not have to be empty.
- Optimizes the backup lock. `LOCK TABLES FOR BACKUP`, `LOCK BINLOG FOR BACKUP`, and `UNLOCK BINLOG` are supported. `FLUSH TABLES WITH READ LOCK` is used to take a backup of the database, but it blocks the whole database from providing service. In contrast, the three statements above use a lightweight backup lock to ensure data consistency during physical/logical backup while allowing the database to providing service.
- Optimizes the `drop table` operations on big tables.

Bug fixes

- Fixed the hang issue when querying `performance_schema`.
- Fixed the overflow of the `digest_add_token` function.
- Fixed the crashes when accessing ibuf using the `truncate table` statement.
- Fixed incorrect queries when `const` in `left join` statements was calculated earlier than it should.

20200930

Performance optimizations

- Optimizes the backup lock.
  `FLUSH TABLES WITH READ LOCK` is used to take a backup of the database, but it blocks the whole database from providing service. Therefore, a lightweight backup lock is provided in this version.
- Optimizes the `drop table` operations on big tables.
  The `innodb_fast_ahi_cleanup_for_drop_table` parameter helps significantly reduce the time it takes to clean up adaptive hash indexes when dropping big tables.

Bug fixes

- Fixed the crashes when accessing ibuf using the `truncate table` statement.
- Fixed cold backup failures when the quick column adding feature was enabled.
- Fixed performance degradation caused by frequently releasing InnoDB memory table objects.
- Fixed incorrect queries when `const` in `left join` statements was calculated earlier than it should.

- Fixed the core issue caused by rule class name conflict between SQL throttling and query rewrite.
- Fixed the concurrent update issue caused by the `insert on duplicate key update` statement in multiple sessions.
- Fixed the `duplicate key error` caused by concurrently inserting data into auto_increment columns.
- Fixed the crashes caused by evicting InnoDB memory objects.
- Fixed concurrency security issues caused by hotspot update.
- Fixed the coredump issue when enabling the thread pool after jemalloc was upgraded to v5.2.1.
- Fixed the incomplete audit log due to fwrite error-free handling.
- Fixed the error where mysqld_safe failed to print logs when it was started as `root`.
- Fixed the increase in the size of the DDL log file caused by `alter table exchange partition`.

20200701

Bug fixes

- Fixed the `INNOBASE_SHARE index mapping` error.

20200630

New features

- Supports using `NOWAIT` and `SKIP LOCKED` in `SELECT FOR UPDATE/SHARE` statements.
- Supports large transaction optimization, which can solve such problems as source-replica delay and backup failures caused by large transactions.
- Optimizes audit performance: async audit is supported.

Bug fixes

- Fixed the overflow of the `digest_add_token` function.
- Fixed the instance crash caused by `insert blob`.
- Fixed the source-replica replication interruption when a hash scan failed to find the record while updating the same row in an event.
- Fixed the hang issue when querying `performance_schema`.

20200331

New features

- Added the official MySQL 5.7.22 JSON series functions.
- Supports the Hotspot Update feature for ecommerce flash sale scenarios.
- Supports SQL throttling.
- Supports encryption with custom KMS keys.

Bug fixes

- Fixed the crash caused by phrase search under multi-byte character sets in full-text index.
- Fixed the crash of the CATS lock scheduling module in high-concurrence scenarios.

20190830

New features

- Supports skipping the corrupted data and continuing to parse when a binlog is corrupted. If the source instance and binlog are both damaged, this feature helps restore data from the replica database for use as much as possible.
- Supports syncing data from non-GTID to GTID mode.
- Supports querying the "user thread memory usage" by executing the `show full processlist` statement.
- Supports quick column adding for tables. This feature does not replicate the data or use disk space/IO and can implement changes during peak hours.
- Supports persistent auto-increment values.

Bug fixes

- Fixed the error where replication would be interrupted if the column name in a `GRANT` statement contained reserved words.
- Fixed the error where SQL execution efficiency dropped when reverse scan was performed on a partitioned table.
- Fixed the error where the query result had an exception due to data inconsistency when using virtual column index and primary key.
- Fixed the error where data was missing due to InnoDB primary key range queries.
- Fixed the error where the system crashed when a DDL statement was executed for a table with spatial indexes.
- Fixed the error where source/replica disconnection occurred when the binlog size was too large and the file length in the heartbeat information exceeded the limit.
- Fixed the error where other events could not be executed as scheduled when an event was deleted.
- Fixed the error where the aggregate query result was incorrect.

20190615

New features

- Supports transparent data encryption (TDE).

20190430

Bug fixes

- Fixed the error where null pointer reference occurred when the LONGTEXT feature was used in subqueries.
- Fixed the error where source/replica disconnection occurred due to hash scan.
- Fixed the error where the replica I/O thread was interrupted due to source binlog switch.
- Fixed the crash caused by the use of `NAME_CONST` .
- Fixed the illegal mix of collation error caused by character set.

20190203

New features

- Supports async deletion of big tables. You can clear files asynchronously and slowly to avoid business performance fluctuation caused by deleting big tables. To apply for this feature, please submit a ticket.
- Supports CATS lock scheduling.
- Supports creating and deleting temp tables and CTS syntax in transactions when GTID is enabled. To apply for this feature, please submit a ticket.
- Supports implicit primary keys. To apply for this feature, please submit a ticket.
- Supports users without super privileges to kill sessions of other users by configuring the `cdb_kill_user_extra` parameter (default value: `root@%`).
- Supports enterprise-grade encryption functions. To apply for this feature, please submit a ticket.

Bug fixes

- Fixed the error where replication was interrupted when binlog cache file ran out of space.
- Fixed the hard error when `fsync` returned `EIO` and retries were made repeatedly.
- Fixed the error where replication was interrupted and could not be recovered due to GTID holes.

20180918

New features

- Supports automatic killing of idle transactions to reduce resource conflicts. To apply for this feature, please submit a ticket.
- Supports automatically changing the storage engine from MEMORY to InnoDB: if the global variable `cdb_convert_memory_to_innodb` is `ON` , the engine will be changed from MEMORY to InnoDB when a table is created or modified.
- Supports invisible indexes.

- Supports memory management with jemalloc, which can replace the jlibc memory management module to reduce memory usage and improve allocation efficiency.

Performance optimizations

- Optimizes binlog switch to reduce the `rotate` lock duration and improve system performance.
- Increases the crash recovery speed.

Bug fixes

- Fixed the error where an event became invalid due to source/replica switch.
- Fixed the crash caused by `REPLAY LOG RECORD` .
- Fixed the error where the query result was incorrect due to loose index scans.

20180530

New features

- Supports SQL auditing.
- Supports table-level concurrent replication. To apply for this feature, please submit a ticket.

Performance optimizations

- Optimizes replica instance locks to improve the performance synchronization of replica instances.
- Optimizes the pushdown of the `select ... limit` statement.

Bug fixes

- Fixed the error where switch failed due to inconsistent checkpoints between `relay_log_pos` and `master_log_pos` .
- Fixed the crash caused by `Crash on UPDATE ON DUPLICATE KEY` .
- Fixed the "Invalid escape character in string." error when a JSON column was imported.

20171130

New features

- Supports the `information_schema.metadata_locks` view to query the MDL grant and wait status in the current instance.
- Supports the `ALTER TABLE NO_WAIT | TIMEOUT` syntax to grant DDL operations wait timeout. To apply for this feature, please submit a ticket.
- Supports the thread pool. To apply for this feature, please submit a ticket.

Bug fixes

- Fixed the error of `innodb_buffer_pool_pages_data` overflow by calculating it based on `bytes_data`.
- Fixed the error where speed limit plugin became unavailable in async mode.

MySQL 5.6

20201231

Bug fixes

- Fixed the error (error code: 1032) caused by hash scans.
- Fixed the error where REPLACE INTO does not update AUTO_INCREMENT columns in row-based replication.
- Fixed the memory leak caused by not freeing up the memory requested for parsing SQL statements.
- Fixed the error where the sql_mode check is skipped when running CREATE TABLE AS SELECT.
- Fixed the error where the sql_mode check is skipped when inserting default values.
- Fixed the error where the sql_mode check is skipped when running UPDATE with bound parameters.

20200915

New features

- Supports SQL throttling.

Performance optimizations

- Optimizes the initialization acceleration of buffer pool.

Bug fixes

- Fixed the hang issue of `rename table` on both source and replica.
- Fixed the crash when `event_scheduler` was set to `disable` and `cdb_skip_event_scheduler` was changed from `on` to `off`.
- Fixed the `sync_wait_array` assertion failure when the maximum number of connections of `tencentroot` was not counted in `srv_max_n_threads`.
- Fixed the crash of source-replica parallel replication caused by the system table structure inconsistency between TencentDB for MySQL v5.6 and other cloud vendors' MySQL v5.6.
- Fixed the `INSERT ON DUPLICATE KEY UPDATE THE WRONG ROW` error.
- Fixed the error of `index_mapping`.
- Fixed the MTR failure.

- Fixed the source-replica replication interruption when a hash scan failed to find the record while updating the same row in an event.

20190930

New features

- Supports querying the "user thread memory usage" by executing the `show full processlist` statement.

Bug fixes

- Fixed GTID holes caused by the replication filter of the replica.
- Fixed the error where source/replica disconnection occurred when the binlog size was too large and the file length in the heartbeat information exceeded the limit.
- Fixed the illegal mix of collation error caused by character set.
- Fixed the error where the source/replica disconnection occurred due to hash scan.
- Fixed the crash caused by the use of `NAME_CONST` .
- Fixed the error where the replica I/O thread was interrupted due to source binlog switch.
- Fixed the error of incompatible backups due to `innodb_log_checusum` .

20190530

Bug fixes

- Fixed the error where dirty data might be read in RC mode.
- Fixed the error where replica instance replay might fail due to the deletion of temp table.
- Fixed the error of deadlock under high concurrency.

20190203

New features

- Supports async deletion of big tables. You can clear files asynchronously and slowly to avoid business performance fluctuation caused by deleting big tables. To apply for this feature, please submit a ticket.
- Supports users without super privileges to kill sessions of other users by configuring the `cdb_kill_user_extra` parameter (default value: `root@%`).
- Supports creating and deleting temp tables and CTS syntax in transactions when GTID is enabled. To apply for this feature, please submit a ticket.

Performance optimizations

- Optimizes the replication and replay of partitioned tables to improve efficiency.

Bug fixes

- Fixed the error of data inconsistency between source and replica due to insufficient temporary space.
- Fixed the error of suspended hot record updates.
- Fixed the error where the `Seconds_Behind_Master` value had an exception during concurrent replication.

20180915

New features

- Supports automatically changing the storage engine from MEMORY to InnoDB: if the global variable `cdb_convert_memory_to_innodb` is `ON`, the engine will be changed from MEMORY to InnoDB when a table is created or modified.
- Supports automatic killing of idle transactions to reduce resource conflicts. To apply for this feature, please submit a ticket.

Bug fixes

- Fixed the crash caused by `REPLAY LOG RECORD`.
- Fixed the error of time data inconsistency between source and replica due to decimal precision issues.

20180130

New features

- Supports the thread pool. To apply for this feature, please submit a ticket.
- Supports dynamically modifying replication filtering rules for replica nodes.

Performance optimizations

- Reduces performance fluctuation caused by `drop table`.

Bug fixes

- Fixed the error where the database crashed due to authentication password strings.

20180122

New features

- Supports SQL auditing.

Bug fixes

- Fixed the error of integer overflow.
- Fixed the error caused by queries using full-text index.
- Fixed the error where the replica crashed during replication.

20170830

Bug fixes

- Fixed the error where binlog speed limit became invalid in async mode.
- Fixed the error where the `buffer_pool` status had an exception.
- Fixed the error where `SEQUENCE` and implicit primary key conflicted.

20170228

Bug fixes

- Fixed the character encoding bug in `drop table` .
- Fixed the error where special symbols such as decimal point in a database or table could not be properly filtered by the `replicate-wild-do-table` statement.
- Fixed the error where SQL threads exited too early after the replica had a `rotate` event.

20161130

Performance optimizations

- Splits the `lock_log` lock to reduce the time used by lock log and improve the concurrency performance.
- Separates the ACK thread of the source to improve the response time.
- Prohibits the user thread from being killed while waiting for ACK in order to prevent phantom reads.
- Fixes the unnecessary `lock_sync` lock when `sync_binlog != 1` .

Quick Column Adding Feature

Last updated : 2021-03-29 11:41:59

This document describes how to use the `instant` algorithm to quickly add columns in big tables, while avoiding data replication. This feature does not replicate the data or use disk capacity/IO, and can implement changes in real time during peak hours.

Limits

- Supported instance architecture and version: two-node or three-node MySQL v5.7
- Supported kernel minor version: 20190830 and later

> **Note :**
>
> Newly purchased instances use the latest kernel minor version by default. For more information on how to view the kernel minor version, see How do I check the kernel minor version?. For more information on kernel updates, see Kernel Version Updates.

Directions

Log in to the database and use the following syntax to quickly add a column:

```
ALTER TABLE t1 ADD COLUMN c1 int, algorithm=instant;
```

> **Note :**
>
> - The `innodb_alter_table_default_algorithm` parameter is used to specify the default `ALTER TABLE` algorithm. If `INSTANT` is configured, there is no need to specify the `algorithm=instant` syntax for `ALTER TABLE`. Currently, you cannot directly modify the default value of this parameter. To modify it, please submit a ticket.
> - The `innodb_alter_table_default_algorithm` parameter can be configured as `INPLACE` (default value) or `INSTANT`.