Copyright Notice

©2013-2019 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice

All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud’s products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.
Contents

TXSQL Kernel
  Overview
  Kernel Version Updates
  Quick Column Adding Feature
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, monitoring, performance optimization, read/write separation, transparent data encryption, and auditing.

**Related documentation**

- Kernel Version Updates
- Quick Column Adding Feature
- Upgrading Kernel Minor Version
Kernel Version Updates

Last updated: 2020-06-15 13:00:00

MySQL 5.7

20190830

New features

- Supports skipping the corrupted data and continuing to parse when a binlog is corrupted. If the master instance and binlog are both damaged, this feature helps restore data from the slave 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 copy the data or use disk capacity/IO, and can implement changes in real time during peak hours.
- Supports persistent auto-increment values.

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 master/slave 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

Fixes
- Fixed the error where null pointer reference occurred when the LONGTEXT feature was used in subqueries.
- Fixed the error where master/slave disconnection occurred due to hash scan.
- Fixed the error where the slave I/O thread was interrupted due to master 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.

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 tasks 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` holdlock duration and improve system performance.
• Increases the crash recovery speed.

**Fixes**
• Fixed the error where an event became invalid due to master/slave 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 slave instance locks to improve the performance synchronization of slave instances.
• Optimizes the pushdown of the `select ... limit` statement.

**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.
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

20190930

New features
- Supports querying the "user thread memory usage" by executing the `show full processlist` statement. To apply for this feature, please [submit a ticket](#).

Fixes
- Fixed GTID holes caused by the replication filter of the slave.
- Fixed the error where master/slave 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 master/slave disconnection occurred due to hash scan.
- Fixed the crash caused by the use of `NAME_CONST`.
- Fixed the error where the slave I/O thread was interrupted due to master binlog switch.
- Fixed the error of incompatible backups due to `innodb_log_checksum`.

20190530

Fixes
- Fixed the error where dirty data might be read in RC mode.
- Fixed the error where slave 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.

Fixes
• Fixed the error of data inconsistency between master and slave 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 tasks to reduce resource conflicts. To apply for this feature, please submit a ticket.

Fixes
• Fixed the crash caused by REPLAY LOG RECORD.
• Fixed the error of time data inconsistency between master and slave 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 slave nodes.

Performance optimizations
• Reduces performance fluctuation caused by drop table.

Fixes
• Fixed the error where the database crashed due to authentication password strings.
20180122

**New features**
- Supports SQL auditing.

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

20170830

**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

**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 slave 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 master 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

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

- Instance version: MySQL 5.7 High-Availability Edition and Finance Edition
- Kernel minor version: 20190830 and above

Instructions

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

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

- 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`. 