

SchoonerSQL™, a Full High-Availability High-Performance Build of MySQL and InnoDB Version 5.1.3

Release Notes

February 27, 2011

Table of Contents

▪ Introduction.....	1
▪ Notes.....	1
▪ About SchoonerSQL™.....	1
▪ Main Features	1
▪ Known Issues.....	2
▪ Resolved Issues/Enhancements (customer reported and reported in 5.1.x Release Notes) ..	6
▪ Release History.....	7
▪ Technical Support	7

Introduction

These Release Notes highlight the fixes, enhancements as well as the known issues in this SchoonerSQL™, a full high-availability, high-performance build of MySQL and InnoDB version 5.1.

Notes

Prior to installing the product, please read the following important notes:

1. This SchoonerSQL™ 5.1 release is based on MySQL v5.1.52 and InnoDB plug-in v1.0.9.
 2. This release has been tested under CentOS/RHEL 5.4/5.5/6.2
 3. You have 96 hours after installation to use SchoonerSQL™ 5.1.3 before you will need a license. After that time, it will stop functioning. Please contact Schooner to obtain a valid license.
-

About SchoonerSQL™

SchoonerSQL™ is a fully integrated technology solution that provides order-of-magnitude improvements in performance, scalability, service availability, and the potential for significant reduction in total cost of ownership (TCO) for datacenters. By seamlessly integrating the innovative Schooner Operating Environment (SOE) with best-of-breed hardware, SchoonerSQL™ delivers industry-leading performance, data durability, and database consistency. It provides comprehensive data and service availability through a rich set of high-availability (HA) and disaster-recovery (DR) features, including replication, backup and restore. It also greatly simplifies the scaling needed to meet rapidly growing data requirements by reducing the need for sharding across multiple masters and allowing the consolidation of slaves. SchoonerSQL™ is fully compatible with existing MySQL applications and tools, making it easy to integrate into any existing MySQL environments.

Main Features

The Schooner SchoonerSQL 5.1.3 release has the following features/enhancements on top of the release 5.1.2

Enhancements

- SchoonerSQL™ makes sure that the first instance started after whole cluster down is the one with the latest data in order to avoid data loss. It returns error if user starts an instance, which does not have latest data, first.
- SchoonerSQL™ allows configuring Asynchronous Master failover timeout. The default timeout is 30 seconds. It can be changed by setting the variable `app.asyncfailover.master.failover` in the file `/opt/schooner/admin/ac/scm/conf/scm.properties`.

Fixes for issues reported by customers and internal QA

The 5.1.3 release fixes some issues reported in the 5.1, 5.1.1, 5.1.2 release notes.

Known Issues

The following are the major known issues found in this SchoonerSQL™ 5.1 release:

Engine issues:

1. Issue #9267: The setting `SQL_MODE=NO_AUTO_VALUE_ON_ZERO` is not supported. If this setting is used and value 0 is inserted for an auto increment column followed by other non zero values, it is possible that the slave would crash with the following message in the `schooner-mysql.err` log.

20216 15:42:56 [ERROR] Slave SQL: Could not execute Write_rows event on table test.t1; Duplicate entry '1' for key 'PRIMARY', Error_code: 1062; handler error HA_ERR_FOUND_DUPP_KEY; the event's master log FIRST, end_log_pos 16542, Error_code: 1062

120216 15:42:56 [Warning] WSREP: RBR event 2 apply warning: 121, 172512

120216 15:42:56 [ERROR] WSREP: Failed to apply app buffer: ??=0, seqno: 172512, status: WSREP_FATAL

at galera/src/replicator_smm.cpp:apply_data():94

at galera/src/replicator_smm.cpp:apply_trx_ws():359

120216 15:42:56 [Note] wsrep recv thread exiting (code:6)

120216 15:42:56 [Note] WSREP node consistency compromised, abort server if not shutting down already

120216 15:42:56 - mysqld got signal 6 ;

The workaround is to remove the `NO_AUTO_VALUE_ON_ZERO` setting.

2. Issue 8443: Stored functions do not replicate properly in either synchronous or asynchronous mode.
3. Issue 8569, 8657: In certain conditions where a network is completely unstable, if a recovering slave splits from group before finding a donor instance and tries to rejoin again with the original group, all the `mysqld` instances in the cluster will abort. The workaround is to start the instance one by one if the network is unstable.
4. Issue 8610: `SAVEPOINT` is not supported.
5. Issue 8646: SchoonerSQL™ synchronous slave recovery and provisioning will fail if DDL statements are executed during the recovery process.
6. Issue 8665: A write transaction may be aborted when a slave joins or leaves the cluster. The MySQL client may fail upon this abort. The workaround is the client to retry the same transaction again.

7. Issue 8558: If a transaction contains both InnoDB and MyISAM statements, binlog entry won't be generated for the corresponding transaction. This will result in inconsistent data in the asynchronous slaves replicating from SchoonerSQL™ cluster.
8. Issue 8777: A very large transaction(~100G) may cause an asynchronous slave to fail with the error "log event entry exceeded max_allowed_packet; Increase max_allowed_packet on master".
9. Issue 7844: In conditions involving heavy DDL transactions along with heavy DML transactions the synchronous slave may abort.
10. Issue 8235: The multi master asynchronous replication between two active cluster groups does not work.
11. Issue 8228: If the mysql tmp directory is inside the data directory, the full recovery removes it and that would cause the instance to fail at startup.
12. Issue 8240: A recovering synchronous slave will fail if ALTER TABLE is executed during the recovery.
13. Issue 8315: The master instance may core dump during the shutdown sequence. This is not harmful because the database will be recovered during the next instance restart.
14. Issue 7485: Loads involving heavy DDL transactions along with heavy DML transactions may cause a recovering slave to hang or abort or may cause a hang in transactions at master.
15. Issue 8853: The server will at times write the log enter "Failed to apply app buffer:..." during shutdown. This error is not harmful and can be ignored.
16. Issue 8235: The multi master asynchronous replication between two synchronous groups does not work.
17. Issue 9023: In rare instances, if the second phase of the slave recovery is not finished before the allocated recovery buffer, the slave may not come up and would exit with the following message in the MySQLd error log file(schooner-mysql.err).

```
"[ERROR] WSREP: gcs/src/gcs.c:_handle_timeout():1063: Unplanned timeout! (tout:
1324888709783045165, now: 1324888709782538000)
[Note] WSREP: RECV thread exiting -110: Connection timed out"
```

The workaround is to increase the recovery buffer size and restart the slave. The recovery buffer size can be increased by setting the option `gcs.recv_q_hard_limit` in the `my.cnf` variable `wsrep_provider_options`.

Example: The following setting changes the recovery buffer size to 100G

```
wsrep_provider_options =
gcs.recv_q_hard_limit=100G;gcs.recv_q_soft_limit=0.75;gcs.max_throttle=0;...
```

18. Issue 8999: The mysqld process may crash while an instance is being created and instance creation will fail due to this. The GUI and CLI would report that instance creation is failed and the MySQLd error log(schooner-mysql.err) file would have the following message when this issue occurs.

```
"/opt/schooner/ac_5.1/mysql/libexec/mysqld(handle_segfault+0x322)
[0x5d8f12]/lib64/libpthread.so.0[0x3ecce0eb10]
/lib64/libc.so.6(gsignal+0x35) [0x3ecc630265]/lib64/libc.so.6(abort+0x110)
[0x3ecc631d10]/opt/schooner/ac_5.1/mysql/libexec/mysqld [0x785832]
/opt/schooner/ac_5.1/mysql/libexec/mysqld [0x692945]
/opt/schooner/ac_5.1/mysql/libexec/mysqld(replicated_trax(THD*, char const*)+0x12b)
[0x692e5b]
/opt/schooner/ac_5.1/mysql/libexec/mysqld(MYSQL_BIN_LOG::write(Log_event*)+0x702)
[0x698742]"
```

```

/opt/schooner/ac_5.1/mysql/libexec/mysqld(THD::binlog_query(THD::enum_binlog_query_type,
char const*, unsigned long, bool, bool, int)+0xff) [0x5c71cf]
/opt/schooner/ac_5.1/mysql/libexec/mysqld(change_password(THD*, char const*, char const*,
char*)+0x472) [0x6a5bf2]
/opt/schooner/ac_5.1/mysql/libexec/mysqld(set_var_password::update(THD*)+0x1e) [0x5fe5ee]
/opt/schooner/ac_5.1/mysql/libexec/mysqld(sql_set_variables(THD*, List<set_var_base>*)+0x7d)
[0x60d4ad]
/opt/schooner/ac_5.1/mysql/libexec/mysqld(mysql_execute_command(THD*)+0x3f67)
[0x5f4b17]
/opt/schooner/ac_5.1/mysql/libexec/mysqld(mysql_parse(THD*, char*, unsigned int, char
const**) +0x4e7) [0x5f9767]
/opt/schooner/ac_5.1/mysql/libexec/mysqld(dispatch_command(enum_server_command, THD*,
char*, unsigned int)+0x184e) [0x5fafbe]"

```

The workaround is to delete the instance and add it again using GUI/CLI.

19. Issue 9017: In certain conditions where a synchronous replication group is functioning as an asynchronous slave and the replication network is unstable, the synchronous replication group might ignore one or more transactions replicated asynchronously from asynchronous master. This may make the data in the synchronous replication group inconsistent with asynchronous master. The MySQLd error log file(schooner-mysql.err) would have log entries with transaction ID, binlog file and position for ignored transactions as below

"[Warning] removed old transaction 2752885 from ./binlog.000040:211622375"

The above entry means that the transaction with the transaction ID 2752885 was removed from local binlog ./binlog.000040 at position 211622375.

The following are possible remedies for this issue

1. Find the transaction corresponding to the ignored transaction ID from Asynchronous master's binlog and apply the transaction manually to the synchronous replication group.
2. If the above remedy is not possible, the following method could be used.
 - (1) Stop asynchronous slave
 - (2) Repair the affected table. Possible ways to repair are
 - * Dump the affected table using mysqldump from asynchronous master and restore it to synchronous replication group
 - * Restore the database from previous backup to sync group and re-attach to Asynchronous master
 - (3) Restart the asynchronous slave

Contact schooner support if this issue is encountered.

20. This issue pertains to CentOS 6.2 support only: Due to incompatible disk controller firmware on IBM systems with CentOS6.2, the system's performance may be erratic and result in frequent heartbeat misses which will eventually lead to split brains. The fix for this issue is to make sure the firmware for the storage controller is updated to the latest release. If the problem persists, please contact IBM support.
21. Issue 9165: The binary log may be inconsistent with the database when system restarts after a power off. This could cause incremental recovery to fail or an inconsistent database in the cluster. In Mysqld the binlog is not durable by default (this means that binlog events that were in the file system cache will be lost when the system is powered off). But incremental recovery requires that the binary log is durable so that it is consistent with the database when the system is powered off.

So in the cases where frequent power offs are expected, one of the following workarounds may be implemented:

(1) Disable incremental recovery:

Set "schooner_rec_incremental = 0" in my.cnf of all instances

(2) Make the binary log durable:

set "sync_binlog=1" in my.cnf of all instances

22. Issue 9171: In certain remote conditions where a network partition between Master and a recovering slave occurs, the writes at master may hang.

The following steps can be done to make mysqld come out of the hang without restarting the instance

(1) set schooner_set_last_committed_timeout = 1 in the mysqld prompt

(2) set schooner_set_last_committed_timeout = 10000000 after mysqld is back to normal

23. Issue 9233: The synchronous master instance hangs while importing data from a mysqldump file, which contains ALTER TABLE statements.

The workaround is to import the data while slave is down and start the slave after importing is completed.

24. Issue 9294: In certain remote conditions, importing data to a table using LOAD IN FILE statement while the server is handling write traffic to other tables may cause the server to hang.

The workaround is to import the data to the table while the server is not handling an write traffic

25. Issue 9162: In certain remote cases write transactions are stuck at master and causes mysqld to run out of connection. The following messages are printed in schooner.err log when this issue occurs.

120130 12:30:32 InnoDB: Warning: cannot find a free slot for an undo log.

InnoDB: many active transactions running concurrently?

120130 12:30:32 InnoDB: Warning: cannot find a free slot for an undo log. Do you have too

InnoDB: many active transactions running concurrently?

This issue would occur either one or both of the following reasons

- There are 1024 connections doing long write transaction with autocommit=0
- If innodb transactions log flushing rate is slower (due to slower hard-disks) than the rate of write transactions and number of client connections are high (>500).

The mysqld would recover from this state automatically after flushing catches up and write transactions complete. To reduce the probability of this issue occurring, the number if undo-slots can be increased by setting the following in my.cnf
innodb_extra_undoslots=4096.

GUI/ CLI/ configuration related issues:

1. Issue 8789: If a replication group does not have any live instances and more than one instances started at the same time through CLI, all instances except the master instance may be brought down. The workaround is start master instance first and wait for it to become READY and then start all slave instances.
2. Issue 8774: If SchoonerSQL™ is stopped on more than one node in a particular order and started in the reverse order one by one, *the SchoonerSQL™ Administrator* will show incorrect states for the mysqld instances until SchoonerSQL™ is started on all the nodes.
3. Issue 8773,8681: Changing configuration while the administration network is partitioned due to network split would result an inconsistent management view and configuration. The workaround is to not change the configuration while thje administration network is in a partitioned condition.

4. Issue 8365: If a donor instance dies during slave recovery, the slave will abort and have corrupted data. Restarting the slave again while one or more new donors are available may keep failing. The workaround is manually clean the txlog directory and restart the slave.
5. Issue 8555: Changing configuration through simultaneous SchoonerSQL™ Administrator sessions is not supported.
6. Issue 8389: After an instance is migrated to a new node, the migrated instance is not manageable through the CLI. The workaround is to execute "list mysql" command in the CLI first and then go to CLI instance management section for managing the migrated instance.
7. Issue 8239: Online backup (based on xtrabackup) does not work if the data directory contains CSV files.
8. Issue 7862: If a database is restored from a full and incremental backup, automatically connecting to an asynchronous master from the last position remembered would cause data inconsistency because incremental backups do not record the binlog file and offset. The workaround is not to use the auto reconnect feature.
9. Issue 7758: Selective restore from incremental backups does not work. All incremental backups will always be restored.
10. Issue 8679: In certain cases if multiple nodes go down at same, alert configuration may be lost and alerts will not be sent thereafter. The workaround is to reconfigure the alerts.
11. Issue 9164: After upgrading to SchoonerSQL 5.1.2 the Schooner Administrator (GUI) is no longer accessible. To resolve this issue, restart the httpd service on all upgraded nodes by issuing "service httpd restart".
12. Issue 9332: After upgrading to SchoonerSQL 5.1.2, a new variable can be found in the my.cnf file named "ws.tmp_dir" which defaults to /schooner/data/tmp/dbX if there are more than one instances configured in the system. If the instance's data directory is not /schooner/data/ then the instance would report errors and instance startup may fail

Resolved Issues/Enhancements (customer reported and reported in 5.1.x Release Notes)

Engine issues:

1. Issue 9009: A synchronous slave may abort while importing data from a mysqldump file, which contains ALTER TABLE statements.
2. Issue 9177: In certain conditions, Master may crash if network partition occurs during slave provisioning and recovery
3. Issue 8542: In certain remote conditions, the recovering slave hangs in the phase 1 of the recovery

GUI/ CLI/ configuration related issues:

1. Issue 9163: After upgrading to SchoonerSQL 5.1.2, a new variable can be found in the my.cnf file named "ws.tmp_dir" which defaults to /schooner/data/tmp/dbX. If the instance's data directory is not /schooner/data/ then the instance would report errors and instance startup may fail.
2. Issue 9005: If the password for the MySQL account "admin" is changed via GUI, the SchoonerSQL™ charting module would report errors.
3. Issue 9031: In certain conditions, some of the SchoonerSQL™ recovery process might not be

cleaned up on the slave instance when the slave dies during first and second phase of the recovery, which may cause restarting the slave to fail. The GUI/CLI would report that instance startup is failed with this issue is encountered.

The workaround is to manually kill the sticky processes (like `wsrep_sst_xtrabackup`) and then restart the slave

4. Issue 9172: After upgrading to SchoonerSQL 5.1.2 from SchoonerSQL 3.x, the `schooner_license` file is not recognized. The workaround for this issue is to manually move the `schooner_license` file from `/opt/schooner/ac_3.1/mysql` to `/opt/schooner/ac_5.1/mysql`.
5. Issue 9025: The password of admin account is not updated in the data collection module (emt) when user changes the password.
6. Issue 9056: GUI does not report error if a backup operation fails and does not take failed backup in to account for rotation
7. Issue 9065: Failed to stop and remove a scheduled backup
8. Issue #9091: Backup of selective binlog files does not work.
9. Issue 9038: Asynchronous master failover does not work if both asynchronous master and slave die at the same time.

Release History

- 02/27/2011: SchoonerSQL™, a full high-availability, high-performance build of MySQL and InnoDB v5.1.3 release.
- 01/31/2011: SchoonerSQL™, a full high-availability, high-performance build of MySQL and InnoDB v5.1.2 release.
- 12/29/2011: SchoonerSQL™, a full high-availability, high-performance build of MySQL and InnoDB v5.1.1 release.
- 11/30/2011: SchoonerSQL™, a full high-availability, high-performance build of MySQL and InnoDB v5.1 release.
- 06/20/2011: Schooner Active Cluster™, an enhanced distribution of the Schooner Appliance for MySQL® Enterprise™ v5.1 release.
- 04/27/2011: Schooner Active Cluster™, an enhanced distribution of the Schooner Appliance for MySQL® Enterprise™ v3.1 release.
- 02/07/2011: Schooner Appliance for MySQL® Enterprise™ with InnoDB v3.1 release.
- 06/11/2010: Schooner Appliance for MySQL® Enterprise™ with InnoDB v2.5.0 release.
- 02/26/2010: Schooner Appliance for MySQL® Enterprise™ with InnoDB v2.1 Service Pack 3 release.
- 02/16/2010: Schooner Appliance for MySQL® Enterprise™ with InnoDB v2.1 Service Pack 2 release.

Technical Support

Technical support for Schooner products is available from the following sources:

- **Phone:** (408) 773-7500 (main)/ (877) 888-5064 (Sales & Support)
- **Fax:** (408) 736-4212

- **Email:** support@schoonerinfotech.com
- **Web:** <http://www.schoonerinfotech.com/support>