SchoonerSQL™

Frequently Asked Questions

1. What is SchoonerSQL™?

SchoonerSQL is a full OLTP relational database that is 100% compatible with widely-used MySQL® and its standard InnoDB storage engine. Schooner applied unique and powerful optimizations and enhancements to MySQL and InnoDB. These include multi-threaded synchronous and asynchronous replication deeply integrated into InnoDB and immediate automated failover and recovery over LAN or WAN. This gives higher availability with better performance than any other MySQL distribution whether the MySQL database is on hard drives, a SAN, or flash memory. SchoonerSQL also provides dramatically simpler management of clustering and replication than traditional MySQL 5.1 or 5.5.

2. What is different about SchoonerSQL replication?

High service availability, high data integrity, and high performance are critical for mission-critical deployments. Traditional MySQL high-availability (HA) approaches can result in lost or inconsistent data, costly downtime, complex administration, and low performance. Schooner developed parallel synchronous replication with instant automated fail-over to solve these traditional MySQL problems. SchoonerSQL replication eliminates lost or stale data, provides immediate and automatic fail-over, dramatically simplifies administration, and provides high performance and easy scalability while minimizing downtime.

3. Is SchoonerSQL compatible with my existing applications and data?

Yes. SchoonerSQL is 100% compatible with all MySQL and InnoDB applications and databases. No schema or hardware changes or any other kind of migration is required.

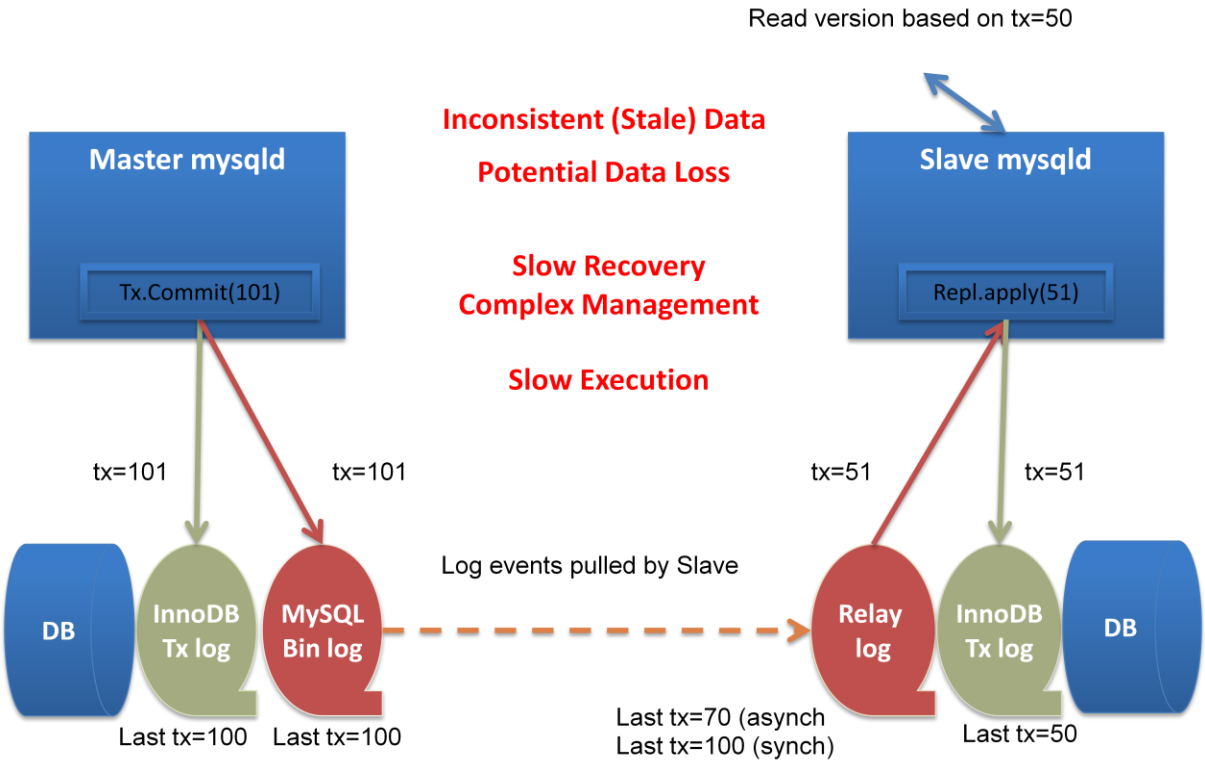
4. Do I need special hardware to run SchoonerSQL?

No. SchoonerSQL works on all Dell, HP, and IBM x86 Red Hat Enterprise Linux / CentOS servers and blades, with the database on hard drives, SAN, or flash memory. SchoonerSQL does contain numerous optimizations to get the most from flash memory, making it the fast and easy path to flash if you want to increase performance and reduce power consumption. In large clusters with many slaves and high master updates, using 10Gb Ethernet connections can further increase your performance.

5. What's the matter with MySQL asynchronous and semi-synchronous replication?

As shown in the figure below, MySQL asynchronous and semi-synchronous replication are based on loose coupling between the master and slaves. As update transactions execute on the master their statements or row data are written to a MySQL Bin Log and gradually transmitted to the

slaves. There, the modifications are eventually applied thereby replicating the modifications that were made on the master. The slaves operate independently from the master, deciding how much to read and from which point in the master bin log. The slaves serially apply the changes using a single thread (to insure ACID compliance) while also servicing other read transactions (in order to provide read transaction scalability against the master's data). With traditional MySQL 5.1 asynchronous replication, the master does not wait for the slaves at all. With newer MySQL 5.5 semi-synchronous replication the master waits for one slave to acknowledge that it has received and queued into its relay log (but not applied) the binlog data before the master completes the transaction.



Loosely-Coupled Traditional MySQL
Asynchronous and Semi-Synchronous Replication Architecture

Loosely-coupled MySQL asynchronous and semi-synchronous replication has a bad impact on service availability, data integrity, administration, scalability and execution efficiency, and cost of ownership:

Reduced Service Availability

- When a master fails, fail-over to a slave is stalled until all transactions in the relay logs have been committed and a new master established, and the remaining slaves reconfigured.

Reduced Data Integrity

- Since slaves are arbitrarily behind the master in reading and applying changes, slaves can give old (stale) data while servicing read transactions.
- Slaves may not have the latest committed data from the master, resulting in data loss when the master fails (the semi-synchronous replication in MySQL 5.5 provides a partial solution to this problem compared to MySQL 5.1 asynchronous replication).
- Checksums in binary and relay log are not generated and persisted to permanent storage, which makes data corruption possible (to be addressed in MySQL 5.6).

High Administrative Complexity

- Recovery from a Master failure, hardware and software upgrades, slave migrations or additions, and schema changes require complex, error-prone, tedious, manual processes, and are a large burden on the responsible DBAs.
- There is no single point of management, monitoring, trouble-shooting and tuning.

Poor Performance

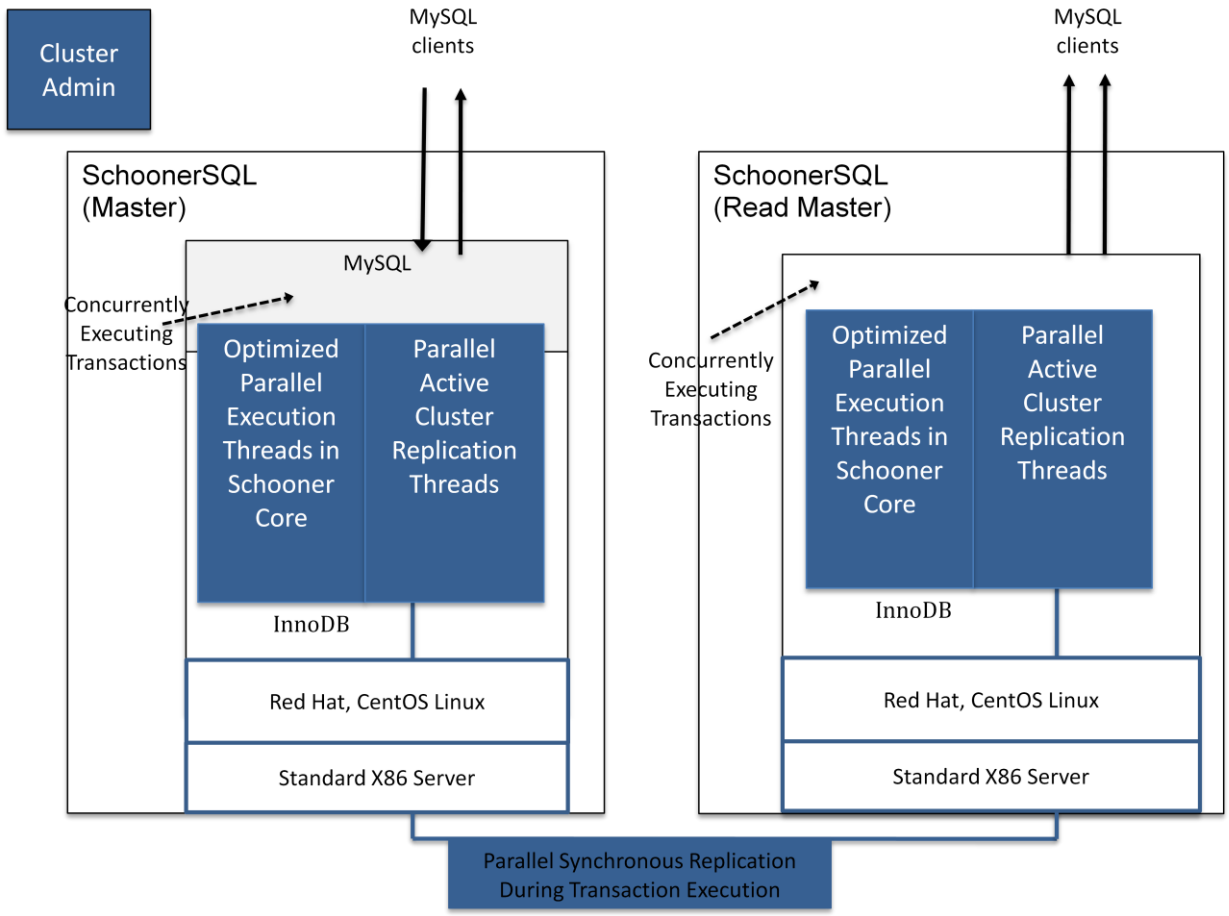
- Applying committed transactions from the relay log on slaves for a database is single-threaded (so as to provide serial consistency), resulting in low utilization and throughput of slaves. This creates slave server sprawl in order to support the required read transaction throughput levels.
- Master throughput must be limited to match the slaves' performance so the slave databases are not too far out of date (otherwise the recovery time could be very long when the master fails). This results in low master update transaction throughput, forcing additional database sharding.

High Cost

- Low master and slave utilization coupled with high administrative and consulting costs increase capital and operating expenses, and downtime reduces revenue

6. How does SchoonerSQL replication work?

SchoonerSQL adds synchronous replication that is deeply integrated into MySQL and InnoDB to deliver complete data integrity, high performance, cluster-wide consistency, and fast automated fail-over and recovery. This is shown in the architecture diagram below.

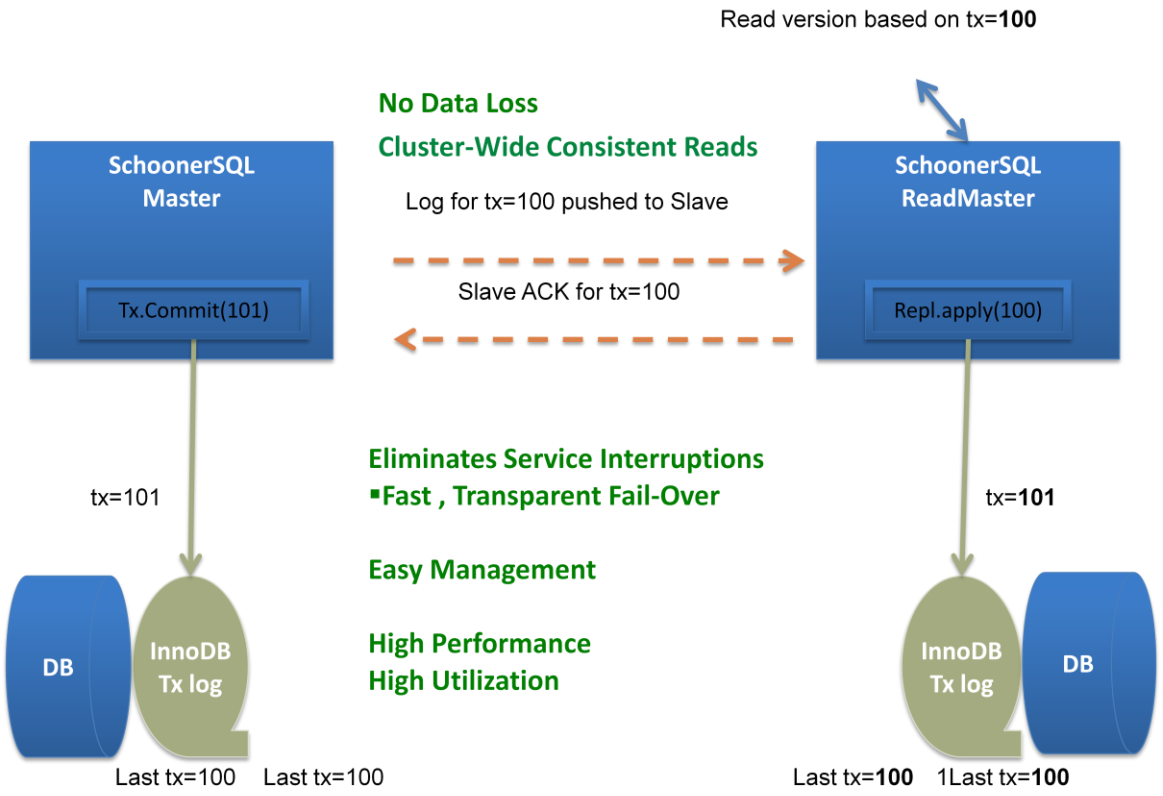


Tightly-Coupled SchoonerSQL Synchronous Replication Architecture

Schooner has deep expertise in bringing the parallelism available from multi-core processors to bear on MySQL. SchoonerSQL replication uses that parallelism to concurrently communicate, replicate, and apply master update transactions to all the replicas (Read Masters) with extremely high throughput and low latency. No two-phase commit is required since all updates are initiated on the master. When the master commits a transaction, all read masters are guaranteed to have received and committed the update. When a master failure is detected, fail-over is automatic and completes within a few seconds with no service interruption and no data loss. A read master is automatically promoted to become the new master, the master's VIPs (Virtual IP addresses) are instantly and automatically switched to the newly promoted read master, and updates continue to be processed without any service interruption. If a read master fails, its load is automatically switched and load balanced to surviving nodes. Once a failed master or read master is repaired, it is automatically brought current with the master and read masters, then automatically made active and load balanced.

The SchoonerSQL Administrator, through its graphical user interface (GUI) running in any web browser or though its Command Line Interface (CLI), provides a global point for easy and powerful administration, monitoring and tuning. Hardware and software upgrades, on-line

consistent back-ups, and instance migration are accomplished with a simple point-and-click or a CLI command, all without service interruption.



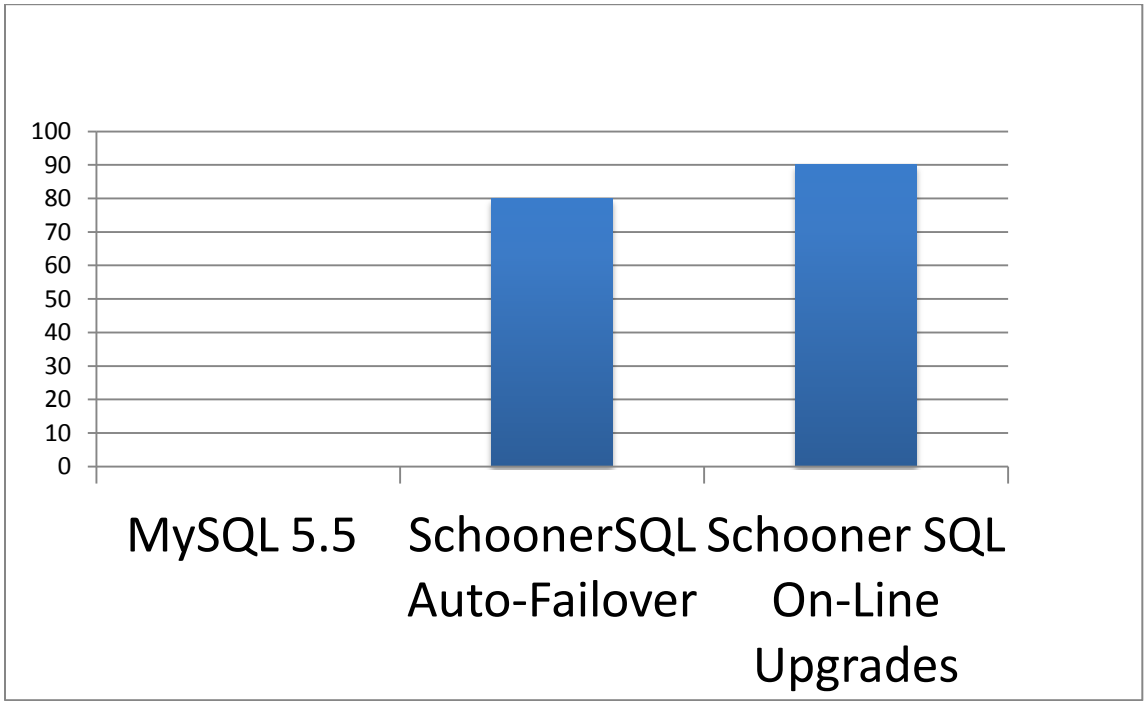
Tightly-Coupled SchoonerSQL

7. Does SchoonerSQL replication use two-phase commit?

No, because it is not required. Two-phase commit is a type of consensus protocol that is used in distributed databases to coordinate concurrent transactions to preserve ACID consistency among potentially conflicting updates. In SchoonerSQL replication all updates are initiated on the master, which eliminates the need for a two-phase commit between masters and read masters since there is no chance for conflicting update transactions on the read masters. The read masters are allowed to execute read transactions, and a read transaction that is using data in a pending update transaction is held off until the update transaction is synchronously committed across the cluster. The SchoonerSQL replication approach reduces complexity and improves performance since it avoids both application rollbacks and the cross-node communication to implement a two-phase commit protocol. Centralizing updates on the master is common practice in most MySQL deployments, and SchoonerSQL masters provide very high update performance so scalability is not an issue.

8. How much does SchoonerSQL replication improve service availability?

SchoonerSQL replication includes automated fail-over and recovery and elimination of downtime from on-line upgrades or migrations. This reduces downtime by 85% to 95% compared to the asynchronous or semi-synchronous replication in MySQL 5.5 and 5.6.



Percent Cumulative Downtime Reduction from SchoonerSQL

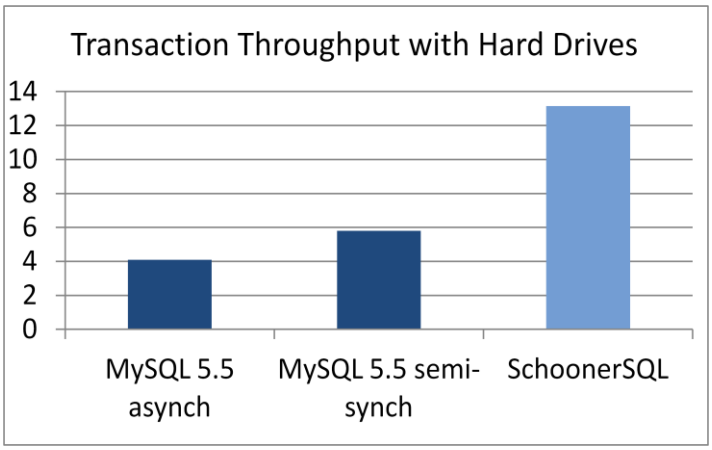
9. How does the performance of SchoonerSQL replication compare to MySQL 5.5 asynchronous or semi-synchronous replication?

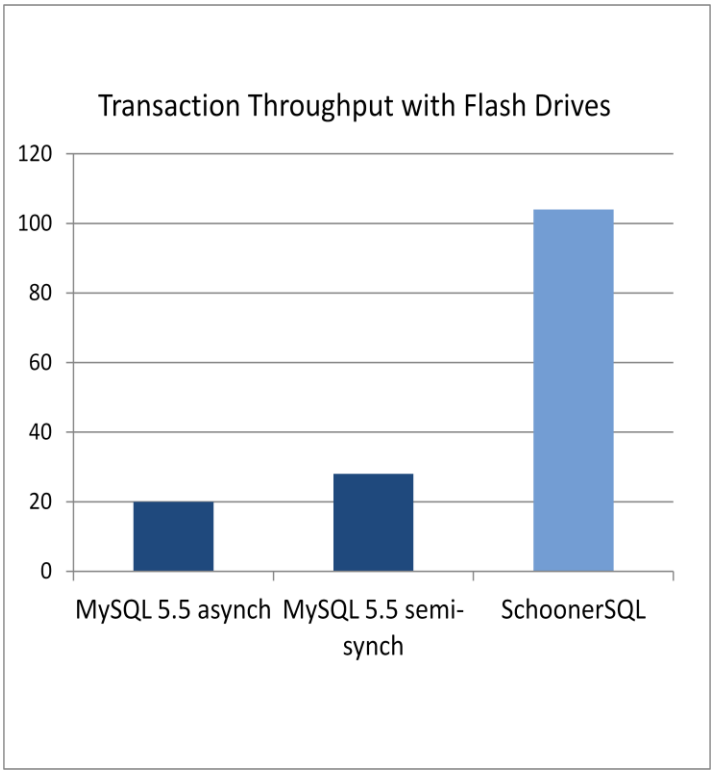
SchoonerSQL increases performance over MySQL 5.5 by 3 times when the database is stored on hard disk drive systems, and by 4 times when the database is stored in flash, when measured using the DBT2 open-source OLTP version of TPC-C.

SchoonerSQL uses high parallelism and high currency during both transaction execution and cluster-wide synchronous replication to deliver highly efficient master and read master utilization and excellent performance. This is clear from performance benchmarking using the DBT2 open-source OLTP version of TPC-C, set for 1000 warehouses using 32 connections, with zero think time. The following charts show the performance throughput, measured in thousands of transactions per minute. In each case all measurements were done on the same hardware: a 2 node Master - Slave configuration running in steady state after two or more hours of warm-up. The master and read master each ran on a standard 2 socket Westmere server with 72GB DRAM.

Measurements were made with two storage configurations: one with the database stored in 8 typical (15,000 RPM) hard disk drives (HDDs) and the other with the database stored in flash

memory (using Fusion-io). The performance of the MySQL 5.5 master is the throughput at which slave lag is stable and does not exceed 1 second. (Allowing slave lag to grow can result in unbounded binlog size and recovery time, and is not an acceptable practice with MySQL). Since SchoonerSQL guarantees zero lag on all read masters due to synchronous replication and immediate, transparent fail-over. Master updates can execute at full throughput.

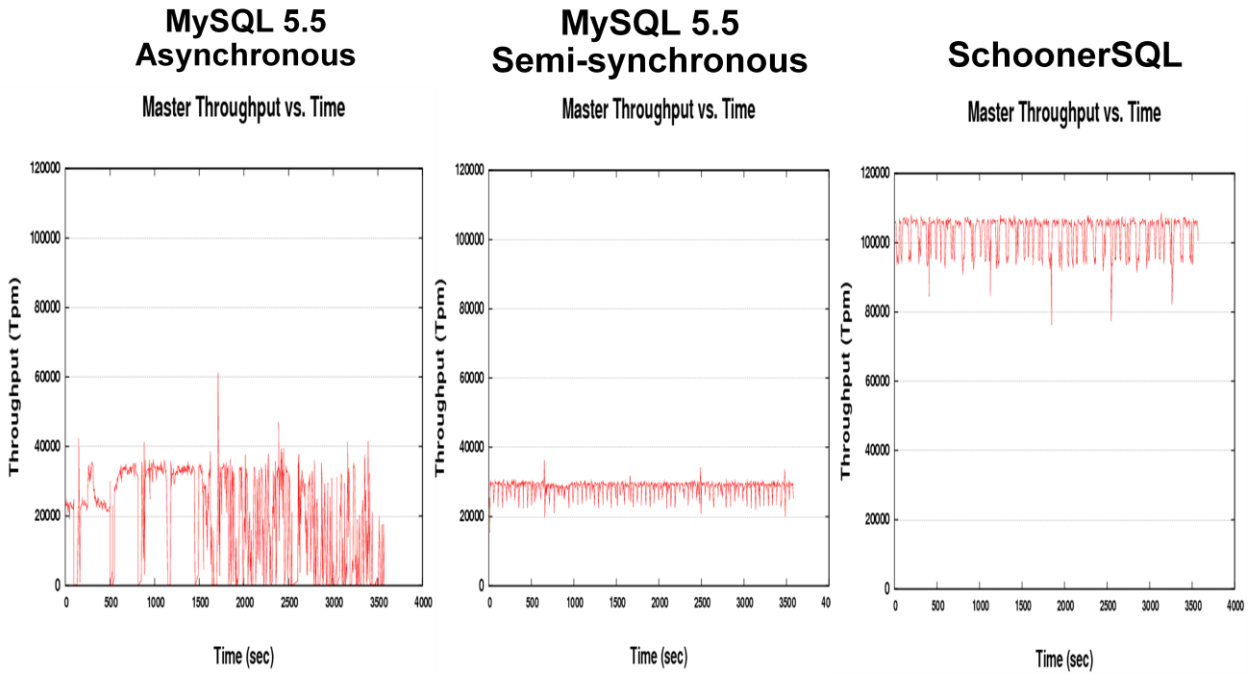




The first chart shows that with the database stored on hard drives SchoonerSQL transaction throughput is over three times higher than that of MySQL 5.5 asynchronous replication and over two times higher than MySQL 5.5 semi-synchronous replication. The second chart shows that with the database stored in flash, SchoonerSQL transaction throughput is four to five times higher

than that of MySQL 5.5 asynchronous or semi-synchronous replication, and 10 times higher throughput than can be achieved on a pure hard disc drive configuration.

SchoonerSQL also gives much more stable throughput with higher sustained rates than MySQL 5.5 asynchronous or semi-synchronous replication. The following charts show throughput for 5.5 asynchronous and semi-synchronous replication, and SchoonerSQL using flash drives (SchoonerSQL delivers equally better performance stability for hard drives):



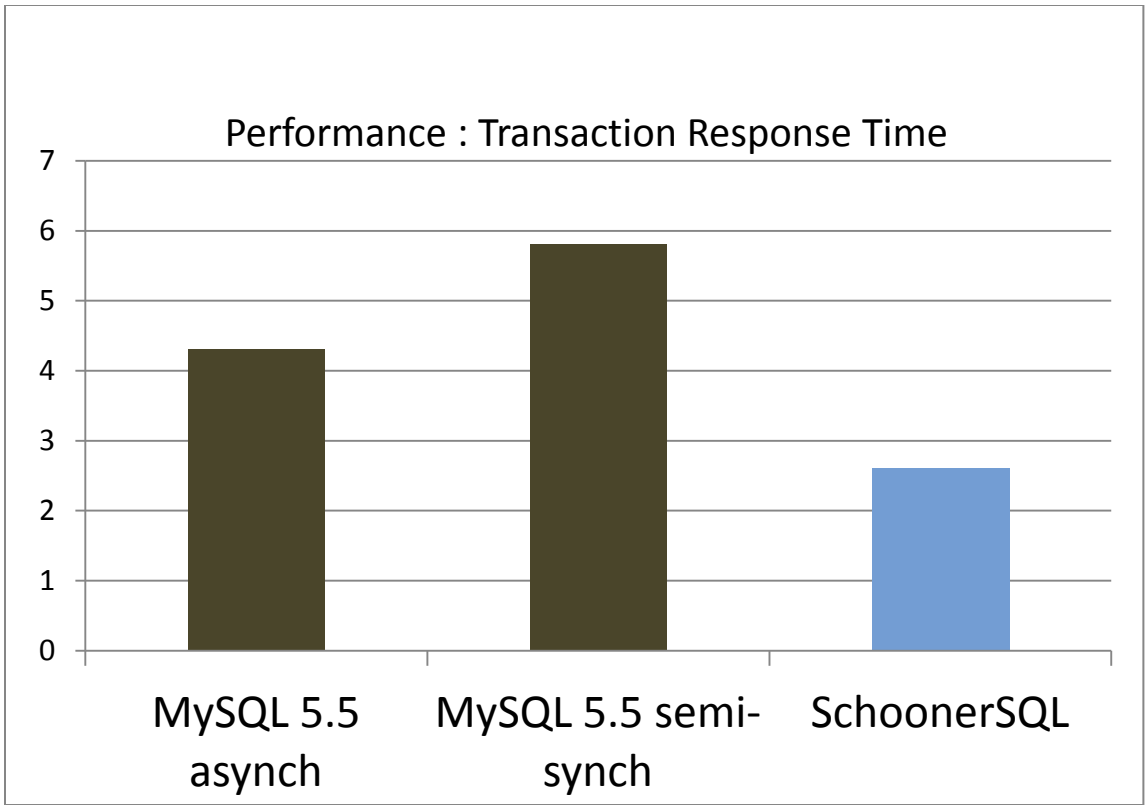
SchoonerSQL Performance Stability with Flash Drives

Using SchoonerSQL synchronous replication, slaves are always current with the master and there is no risk of slave lag driving up recovery time. Master updates can execute at full throughput, allowing high write scalability and reduced sharding. Using SchoonerSQL, reads can be scaled linearly across slaves without server sprawl.

SchoonerSQL provides a large, immediate availability and performance benefit with hard disk drive configurations, and an easy in-place growth path for increasing throughput by an additional 10 times by adding flash memory. You can download SchoonerSQL and immediately evaluate the availability, administrative, and performance benefits with your workloads on your servers.

10. Does SchoonerSQL synchronous replication introduce latency and slow down transaction response time?

No. Schooner's innovative architecture with high thread- and core-level parallelism and granular concurrency control actually improves response time. For example, SchoonerSQL can reduce response time by almost 50% when compared to MySQL 5.5 asynchronous or synchronous replication, as demonstrated in the following chart, which is based on measurements of the same DBT2 benchmark and system configurations described in question 9 with the database on flash.



11. Are multiple masters still needed in a SchoonerSQL environment?

No. Multiple masters are commonly used to provide a stand-by master for rapid recovery in case the master fails. With SchoonerSQL:

- synchronous replication ensures that any slave can become the master if the master fails; and
- the promotion of a slave to the master is automatic, instantaneous, and transparent to the application with no loss in service availability.

SchoonerSQL completely eliminates the need for multiple masters.

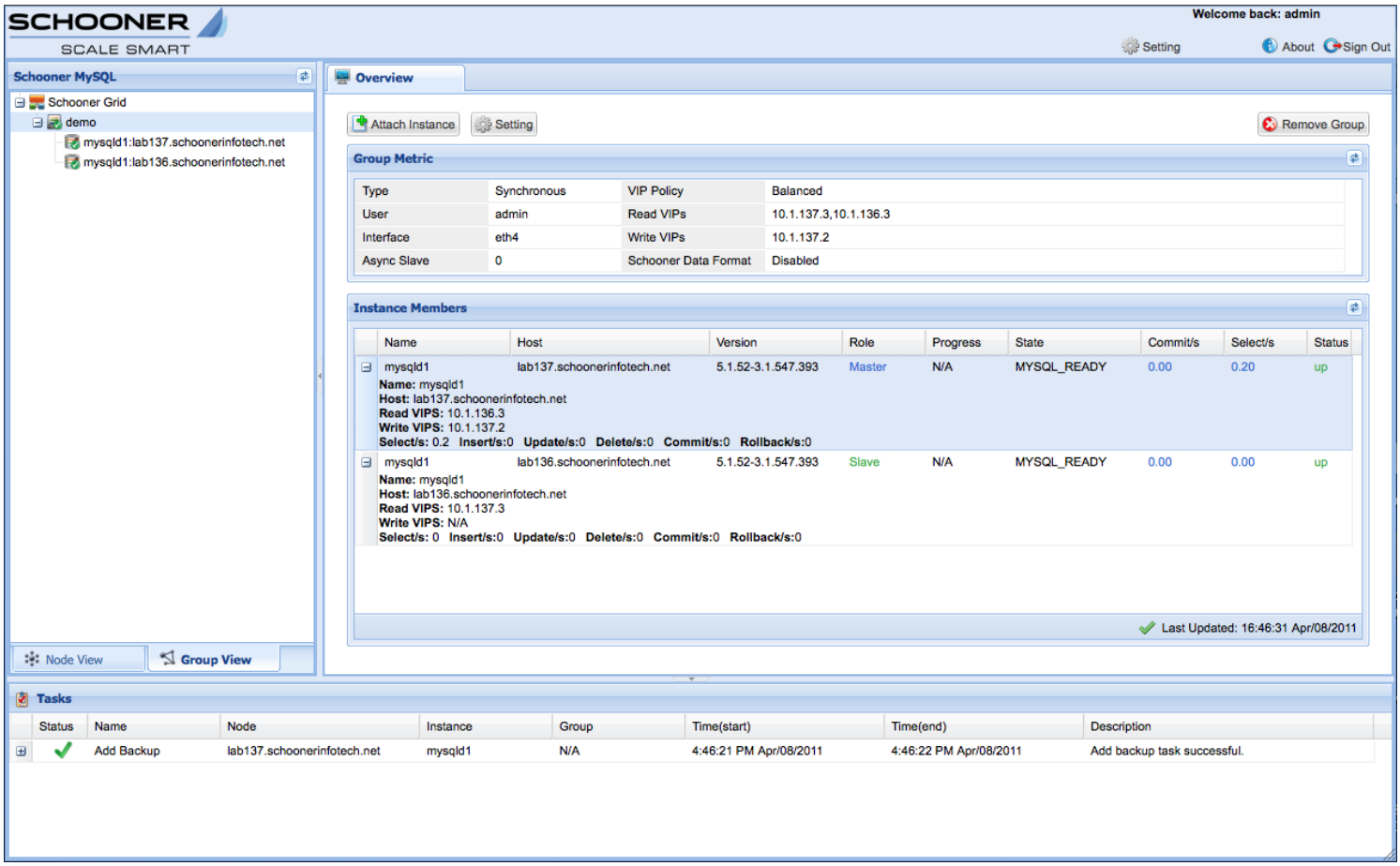
12. What is the impact of SchoonerSQL on data integrity?

SchoonerSQL eliminates the data loss and data inconsistencies of MySQL 5.5. With SchoonerSQL, reads on slaves always provide the latest committed data, resulting in full cluster-wide data consistency. Slaves always have the latest committed data from the master, so there is no data loss if the master fails. Checksums on logs are persisted, so data corruption is detected and corrected.

13. Does SchoonerSQL make administration simpler?

Yes. With SchoonerSQL, fail-over and recovery are completely automatic and instant, requiring no administrator intervention. Also, the SchoonerSQL Administrator GUI and CLI provide a single point for cluster-wide and per-instance management, monitoring, trouble-shooting, and tuning. Simple single-click actions perform hardware and software upgrades, slave migrations,

provisioning, and dynamic schema changes. The SchoonerSQL Administrator GUI (shown in the following screen capture) allows simple yet powerful management of cluster nodes, replication groups, database instances, and activities.



SchoonerSQL Admin GUI: Node, Group, Instance Management

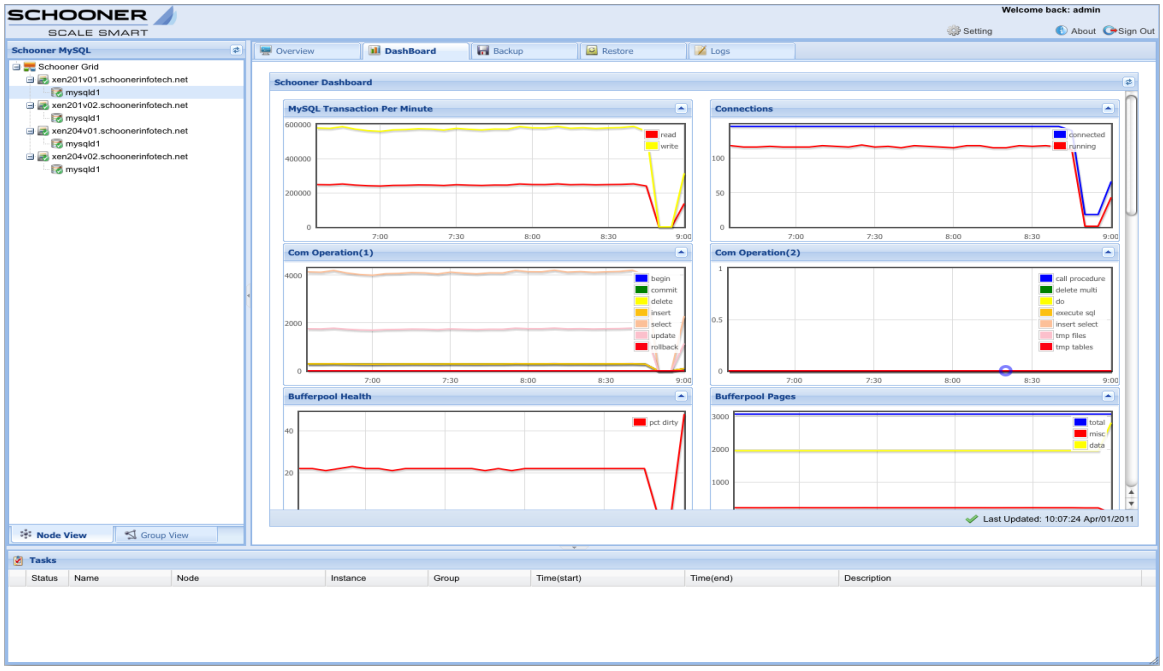
14. How does SchoonerSQL differ from MMM?

MMM is a set of scripts to assist in fail-over and recovery after a system using MySQL asynchronous replication fails. MMM does not help solve with any of the problems inherent in asynchronous replication: poor data integrity, slave lag, data inconsistency, and poor performance and low utilization. SchoonerSQL provides dramatically simpler configuration and management than MMM. When a system fails, SchoonerSQL provides much faster fault detection and fail-over than MMM, and is far more robust than MMM.

15. Does SchoonerSQL provide cluster, node, and database monitoring and tuning information?

Yes. The SchoonerSQL Administrator gathers and displays an extensive set of cluster, node, replication group, and database instance statistics for monitoring and tuning. These can be easily extended and customized.

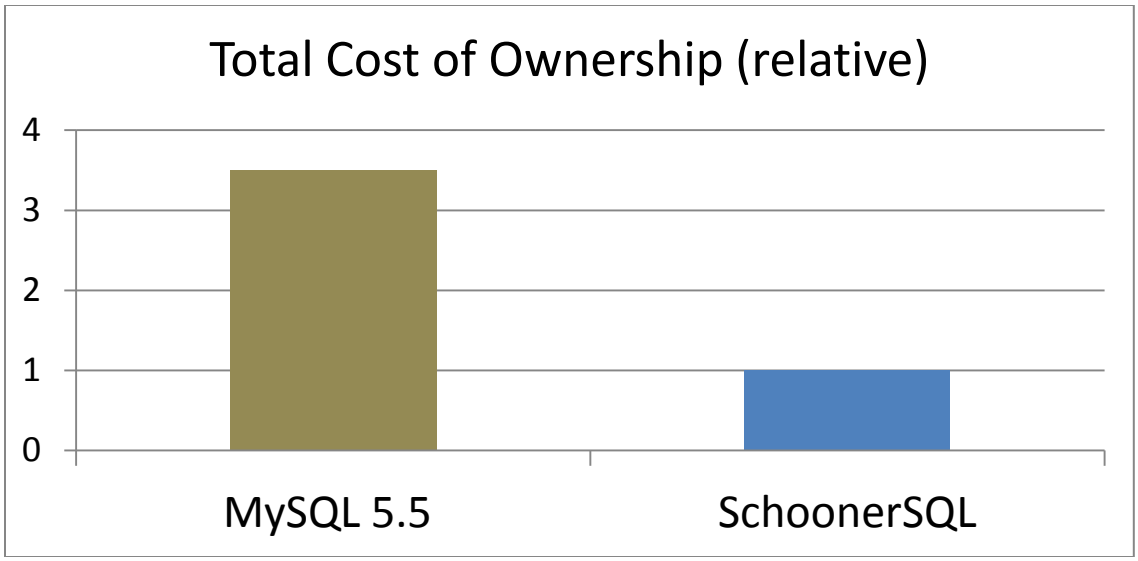
Default system statistics include CPU, DRAM, I/O and Network utilization Default MySQL database instance-specific statistics include time-based charts of transaction throughput, connections, cores, buffers, locks, etc. as shown in the following screen capture of the MySQL instance dashboard of the SchoonerSQL Administrator GUI.



SchoonerSQL Administrator GUI: Instance Dashboard

16. How does the cost of ownership of SchoonerSQL compare with MySQL 5.5 asynchronous or semi-synchronous replication?

SchoonerSQL is 70% cheaper than MySQL 5.5, as shown in the following chart. SchoonerSQL's minimum 3x performance advantage immediately reduces capital and operating expenses based on reductions in servers, rack, power and pipe. Additionally, SchoonerSQL's 90% lower downtime increases revenue, and SchoonerSQL's administrative simplicity further reduces operating expense.



You can work with Schooner's TCO/ROI calculator to estimate your savings.

17. Does SchoonerSQL support WAN replication deployments that span multiple datacenters?

Yes. In addition to synchronous replication for database clusters within a data center, SchoonerSQL also supports high-speed (multi-threaded) replication across a WAN, and immediate automated failover. You can also use traditional MySQL asynchronous replication over a WAN with SchoonerSQL. Schooner synchronous replication groups can be configured as asynchronous masters or slaves, and replicate asynchronously to or from other synchronous replication groups or traditional MySQL instances. SchoonerSQL lets you use these capabilities exactly as you do today across datacenters to provide geographic scalability and disaster recovery.

18. Can SchoonerSQL interoperate in a mixed environment with my existing MySQL synchronous and semi-synchronous masters and slaves?

Yes. SchoonerSQL supports traditional MySQL asynchronous replication, enabling easy interoperability as either an asynchronous slave or asynchronous master in mixed environments with legacy MySQL masters and slaves.

19. How does SchoonerSQL compare with Oracle MySQL Cluster?

Oracle MySQL Cluster is a shared-nothing distributed database targeting real-time telecom applications. Both SchoonerSQL and Oracle's MySQL Cluster provide very high availability, including automated fail-over, on-line addition and upgrades of nodes, etc. However, Oracle's MySQL Cluster does not support InnoDB, and it is not compatible with MySQL Enterprise applications, whereas SchoonerSQL is a full MySQL with InnoDB. SchoonerSQL is also much simpler to administer and provides higher performance than Oracle MySQL Cluster.

20. How does SchoonerSQL compare with Continuent Tungsten?

Continuent Tungsten Replicator converts the MySQL asynchronous bin log to a transaction history log and uses JDBC through a client proxy to access MySQL indirectly. This enables heterogeneous database replication interoperability with PostgreSQL and other MySQL instances. Tungsten also provides a Global Transaction ID, which are used by slaves to point to new master when an old master fails. Tungsten has a SaaS & ISP capability allowing parallel replication across independent, multi-tenant MySQL databases, but has no parallel replication within a single database. Since Tungsten is a very loosely coupled external replication service, its performance is significantly worse than that of MySQL 5.5 for MySQL Master Slave environments. Since Tungsten uses the MySQL asynchronous replication bin log, it suffers from the same issues of reduced service availability, poor data integrity, high administrative complexity, and high cost discussed above for MySQL 5.5 when used in MySQL Master Slave deployments.

21. How does SchoonerSQL compare with Oracle's Golden Gate?

Oracle's Golden Gate converts the MySQL asynchronous bin log to a common log format to provide heterogeneous database replication interoperability with Oracle, IBM DB2, and Microsoft SQL Server, or other MySQL instances. Since Golden Gate is a very loosely coupled external replication service, its performance is significantly worse than that of MySQL 5.5 for MySQL Master Slave environments. Since Golden Gate uses the MySQL asynchronous or semi-synchronous replication bin log, it is even more loosely coupled and suffers from the same issues of reduced service availability, poor data integrity, high administrative complexity, and high cost discussed above for MySQL 5.5 when used in MySQL Master Slave deployments.

22. How does SchoonerSQL compare with Linux DRBD?

Linux DRBD (Distributed Replicated Block Device) provides active-passive mirroring at the block device level. After each commit the stand-by server is guaranteed to have blocks identical to those on the master, so the stand-by storage is kept in lock-step with the master. Linux DRBD does prevent data loss, and limits downtime after master failure since the stand-by master can usually be restarted in minutes. But the stand-by server does not service any load, and all of the slaves are still operating with asynchronous or semi-synchronous replication. Thus the same issues of reduced service availability, poor data integrity, high administrative complexity, high cost, and poor performance discussed above for MySQL 5.5 Master Slave deployments are still present with DRBD.

23. How much does SchoonerSQL cost and how can I buy it?

The quantity-one one-year license fee for SchoonerSQL is $9500/year/server in the U.S., including 24x7/365 support of the Schooner replication and failover features. Quantity discount are offered as well as discounts for multi-year licenses; site, project, and enterprise licenses are also available. SchoonerSQL is available directly from Schooner as well as Schooner's partners world-wide. More comprehensive MySQL Customer Support Service is optionally available

from our support partner, SkySQL AB. Our expert professional service partners, Percona and Pythian, also provide in-depth MySQL consulting.

24. Can I download SchoonerSQL to try it before I buy it?

Yes. You can get started right away by downloading the full SchoonerSQL product for free trial at www.schoonerinfotech.com/free_trials. You can get more complete product information at http://www.schoonerinfotech.com/products/schoonersql. You can get detailed information on supported platforms at www.schoonerinfotech.com/products/supported_platforms.

Schooner Information Technology
501 Macara Ave., Suite 101
Sunnyvale, CA 94085, USA
Tel: 408-773-7500
Fax: 408-736-4212
info@schoonerinfotech.com
www.schoonerinfotech.com
Updated 2012-Mar-08