

Mission-Critical MySQL and InnoDB

*How to achieve the previously impossible
and get 99.999% availability using SchoonerSQL™*

About Schooner Information Technology

Schooner Information Technology provides a high-availability high-performance OLTP database, 100% compatible with standard MySQL and InnoDB, for demanding mission-critical applications. Schooner's flagship product, SchoonerSQL, provides industry-leading 99.999% availability with auto fail-over, guarantees no data loss and no stale data with "synchronous read masters", delivers industry-leading performance and scalability on commodity servers and storage, radically simplifies cluster administration, and minimizes your database Total Cost of Ownership (TCO).

Schooner is headquartered in Sunnyvale, California.

For more information, please visit www.SchoonerInfoTech.com.



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

2011-October-19

1. Introduction

This white paper discusses the requirements, evaluation criteria, and alternatives for datacenter managers who need to deliver mission-critical services using MySQL databases. We begin by discussing the emerging mission-critical imperative, then the requirements and metrics for mission-critical databases. We then examine the limitations of today's MySQL offerings. Next we discuss SchoonerSQL™ and its dramatic impact on enabling truly mission-critical use of MySQL with 99.999% availability. We then compare SchoonerSQL to the alternatives in the market. Finally we discuss best practices and how you can evaluate the options for your data center.

2. The Mission-Critical Imperative

The amount of data used in web-facing businesses, in business processes, and in the software that runs essential systems is growing explosively. The business opportunities that are created from the deployment of these systems are substantial, but the demands they place on the datacenter are daunting. Data is now the most important and valuable component of modern applications and websites and the enabler of revenue generation. Data access downtime or poor performance can cost customers, reputation, and revenue. The business-critical importance of providing high service availability with excellent performance has increased exponentially.

***"Let me tell you the difference between
Facebook and everyone else, we don't crash
EVER! If our service is down for even a minute,
our entire reputation is irreversibly destroyed!"
-- The Social Network***



Maintaining high service availability and low response time is mission-critical for many key classes of businesses, including e-commerce, social media, gaming, finance, telecommunications, and enterprise. Companies like Facebook and Google invest hundreds of millions of dollars every year on custom software and hardware infrastructure to optimize availability, performance, administration, and cost. Most organizations cannot afford to do this and need effective scalable out-of-the-box solutions that can be easily deployed.

3. Requirements and Metrics for Mission-Critical Databases

Mission-critical applications and websites require mission-critical databases to serve their valuable data. These databases are a core element of delivering highly available services with low response time. The primary requirements for a mission-critical database are: high availability; high data integrity; high performance and scalability; simple and powerful administration; cost effectiveness; and application investment preservation through compatible support of standards. Table 1 summarizes the metrics to evaluate these requirements.

Table 1: Requirements and Metrics for Mission-Critical Databases

<u>Requirement</u>	<u>Metrics</u>
High Availability (HA)	Service unavailability (minutes per year) from failures, during planned upgrades and maintenance, or from disasters
High Data Integrity	Probability of data loss or corruption; data consistency levels
High Performance and Scalability	Transaction throughput, response time; performance scalability; performance stability
Simple and Powerful Administration	Ease of cluster administration; fail-over automation; monitoring and optimization tools

Cost Effective	Total cost of ownership (TCO); return on investment (ROI)
Standards and Compatibility	Level of standards compliance

4. Evaluating Alternative MySQL Architectures and Technologies for Mission-Critical Use

Database high-availability and scalability architecture has a profound impact on achieving the mission-critical requirements of high service availability, data integrity, performance and scalability, administrative simplicity, cost effectiveness, and compatibility. In this section, we evaluate MySQL HA and scalability alternatives relative to the mission-critical requirements and metrics.

The fundamental HA database approach for high service availability in the event of database unplanned or planned downtime is to create and maintain a replica of the database so the replica can be used when the master database is down. When a master database is down due to a failure or for routine maintenance, the load is switched to the replica to maintain service availability.

4.1 Legacy MySQL 5.x: Limitations on Mission-Critical Use

Replication in MySQL Enterprise 5.1/5.5/5.6 (and derived forks) is based on a loosely-coupled design using a transaction log (Bin log). The Bin log is generated by MySQL and asynchronously and independently pulled by slaves from the master and later applied by the slaves in a serial manner. This design imposes severe limitations on achievable service availability, data integrity, performance, and scalability. This design also drives up administrative complexity and cost of ownership. Table 2 evaluates Legacy MySQL 5.x relative to database mission-critical requirements and metrics.

Table 2: Evaluation of Legacy MySQL 5.x Relative to Mission-Critical Requirements and Metrics

<u>Impact on Requirement</u>	<u>Impact on Metrics for Mission-Critical Use</u>
Reduced Service Availability	Service is unavailable for minutes to hours when a master database 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 are reconfigured.
Reduced Data Integrity	Lost Data: When the master database fails, asynchronous slaves do not have the latest committed data. Inconsistent Data: Slaves give old ("stale") data in response to read transactions
Poor Performance: Low Throughput and Low Utilization; Poor Scaling	Slaves have low throughput and utilization since relay log application is single-threaded to maintain consistency, creating "server sprawl". Master updates must be limited ("throttled") to match the slaves' slow update performance to ensure that the slaves are not too far out of date (otherwise the recovery time would be very long when the master fails waiting for slaves to apply updates from the log).
High Administrative Complexity	DBAs have tedious, error-prone, and usually manual processes in common tasks such as recovery from a Master failure, promotion of a Slave to a Master, Slave migrations or additions, and hardware or software upgrades.
High Cost of Ownership	Increased capital expense due to server sprawl caused by low utilization of masters and slaves. Increased operating expense due to server sprawl leading to an increase in power,

	space, and administrative costs. Reduced revenue and customer satisfaction due to service unavailability.
High MySQL Compatibility	Fully standards-compliant MySQL and InnoDB

4.2 SchoonerSQL: Truly Mission-Critical MySQL and InnoDB

We discussed above how loosely-coupled MySQL with asynchronous replication imposes inherent and severe limitations on service availability, data integrity, performance, ease of management, and cost effectiveness.

Schooner has taken a completely new approach in its architecture and design to achieve industry-leading MySQL service availability and performance scalability. SchoonerSQL, which is 100% compatible with MySQL and InnoDB, used tight coupling and high parallelism to deliver very high performance and unlimited scalability. And it is based on synchronous replication with immediate, automatic fail-over without service interruption or data loss. SchoonerSQL delivers 99.999% availability with the industry's highest performance and scalability.

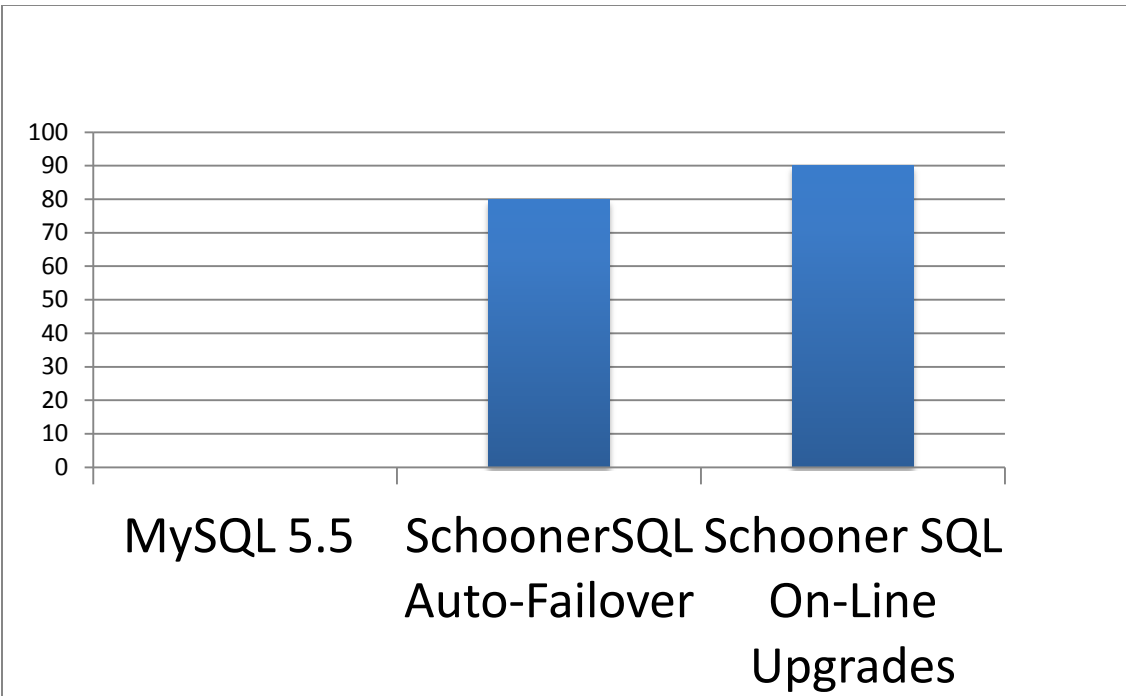
MySQL replication is most often implemented using a Master - Slave configuration (usually with multiple slaves). The term "slave" is traditionally used because the slave servers have to perform every task in copying from the binary log on the master server, then updating their relay logs and finally committing transactions to the slave databases. The master plays no role in replication other than storing the replication events in its binary log. SchoonerSQL uses a unique push mechanism and parallel (multi-threaded) synchronous replication to apply the replication events on every slave node in parallel. ***It's more accurate to refer to SchoonerSQL slaves as "Read Masters", because they are always fully synchronized with the master.*** We therefore "read master" below.

SchoonerSQL synchronous replication provides the highest service availability and data integrity, because synchronous replication is essential to ensure no data loss. With synchronous replication, the master does not complete a transaction commit until all the read masters in the synchronous replication group have committed the transaction. Schooner's synchronous replication architecture is a profound advance in achieving high availability, high data integrity and simplified management. In the event that a master is downed due to failure or maintenance, any one of the read masters in the synchronous replication group instantly takes over as the new master. Service continuity is maintained without administrator intervention. Data integrity is insured by the nature of synchronous replication: no master updates are lost and all reads from all the read masters are fully consistent. SchoonerSQL's Administrative GUI provides a simple and powerful interface for managing replication groups, and for cluster monitoring, tuning, and alerts. SchoonerSQL eliminates service downtime and increases customer satisfaction, resulting in increased revenue

SchoonerSQL's tightly-coupled and highly-parallel, low-overhead resource management algorithms and replication mechanisms deliver exceptional performance and scaling. This reduces sharding and eliminates server sprawl, reducing capital and operating expenses.

4.2.1 Evaluation of SchoonerSQL Relative to Mission-Critical Requirements and Metrics

Highest Service Availability: Instant, automated fail-over for both unplanned and planned downtime reduces service downtime by 95% relative to legacy MySQL 5.x:



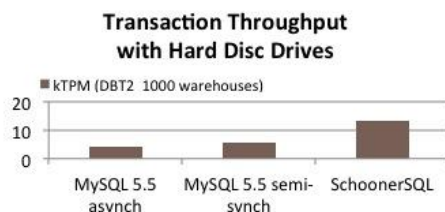
Highest Data Integrity: Schooner synchronous replication guarantees no data loss: when a master database fails, all read masters have the latest committed data. All data is fully consistent across the entire synchronous cluster.

Highest Performance: Schooner multi-threaded (parallel) synchronous replication provides much higher throughput than legacy MySQL 5.x. Schooner also reduces transaction response time and increases performance stability, resulting in industry-leading performance in all server configurations and high vertical scaling with commodity processors and storage.

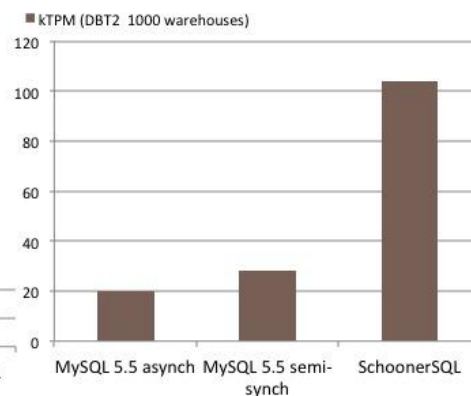
Performance : Transaction Throughput

DBT2 open-source OLTP version of TPC-C
1000 warehouses, 32 connections
0 think-time
Result metric: TPM (new order)

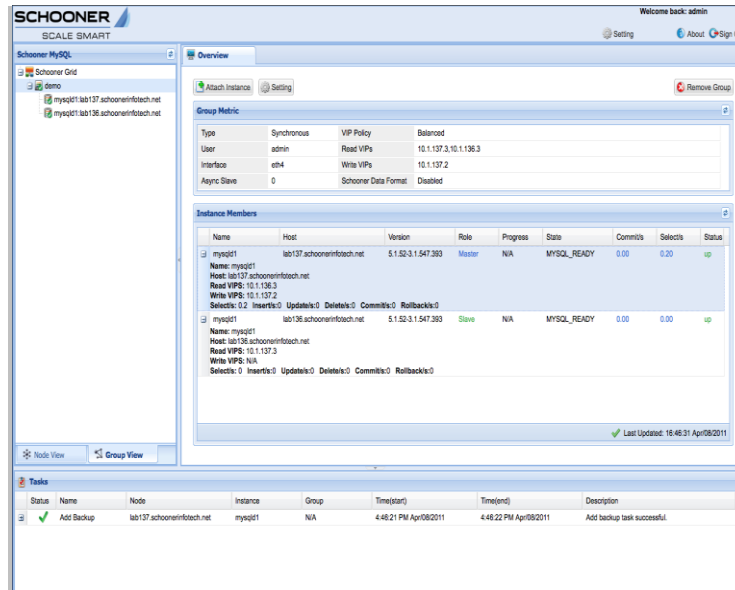
Measurement Configuration
2 node Master-Slave configuration
2 socket Westmere
72GB DRAM



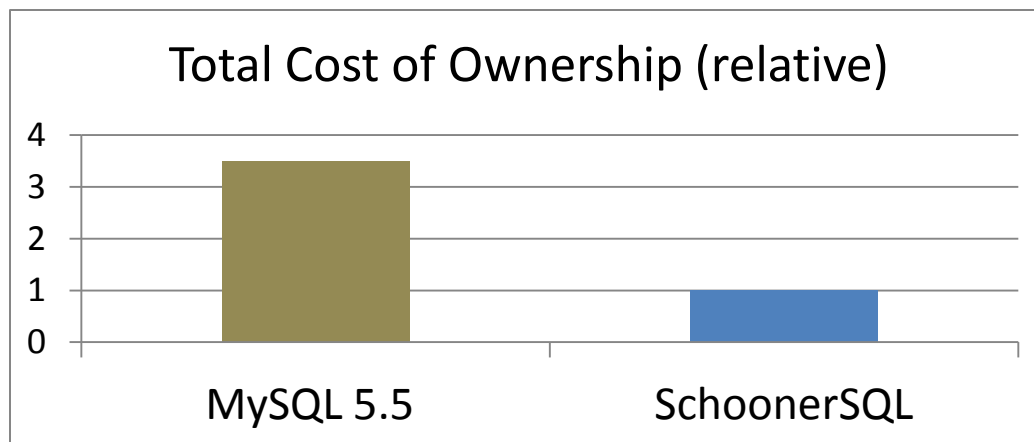
Transaction Throughput with Flash



Simplest and Most Powerful Administration: SchoonerSQL provides completely automatic and instant failover, eliminating service interruptions and the need for administrator intervention. The Schooner GUI and CLI enable easy cluster-wide administration, including point-and-click cluster management, extensive monitoring, troubleshooting, tuning, and alerts. An example of the GUI is shown below:



Lowest Cost of Ownership: SchoonerSQL significantly lowers the total cost of ownership and typically provides a very fast return on investment. These savings are achieved through a combination of reduced capital and operating expenses and increased revenue and customer retention. TCO and ROI models are specific to each customer's application, architecture, and workload. Major variables include throughput per server; server, rack, and network costs; software license and support costs; the cost of space and power; the cost of DBAs to administer the servers; and the value of avoiding downtime. Nonetheless, this data is illustrative of the compelling economics of SchoonerSQL. Schooner will help you compute the TCO and ROI for your organization.



Highest MySQL Compatibility: SchoonerSQL is 100% compatible at the API and schema levels with Oracle MySQL. No application changes are required.

4.2.2 SchoonerSQL WAN High Availability and Unlimited Query/Update Scaling

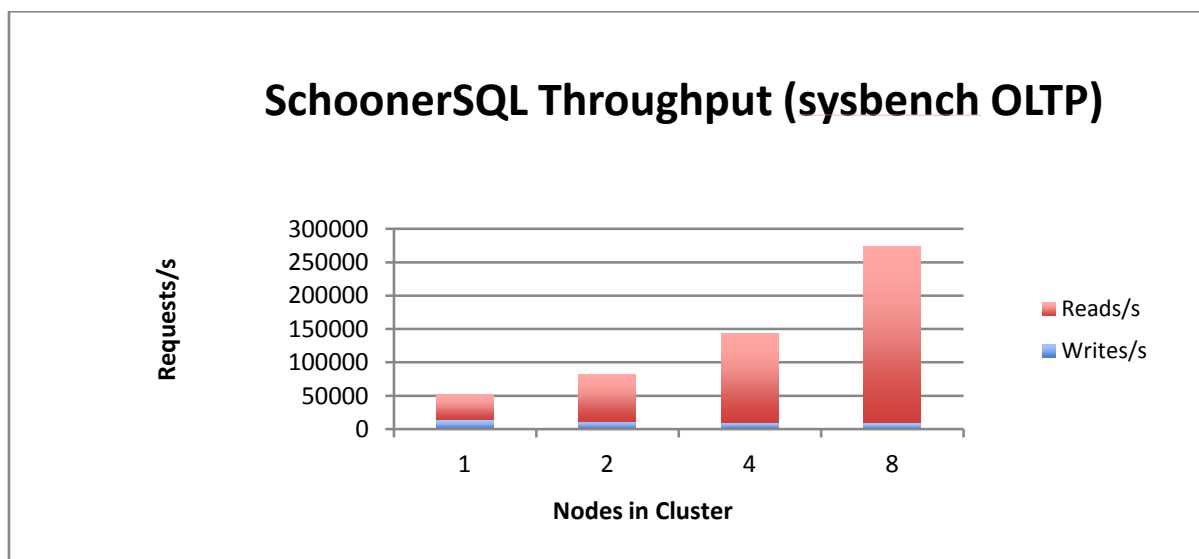
As discussed above, SchoonerSQL synchronous replication groups provide data centers with truly mission-critical MySQL. Enterprise-wide mission-critical deployments often incorporate geographically distributed data centers or require unlimited query/update scalability of a database. We discuss in this section how SchoonerSQL delivers high availability and high performance in these important scenarios through Schooner innovations in tightly coupled parallel asynchronous replication and application transparent sharding.

4.2.2.1 WAN High Data Consistency, Automated Failover, and High Performance

WAN transmission latency is typically tens to hundreds of milliseconds with high variance. It is often impractical to add WAN latency and potential instability to every local query response time, as would be required with geographically distributed synchronous replication. As a result, WAN HA and scaling typically is based on asynchronous replication. But, mission-critical WAN deployments require high WAN data consistency and fast WAN failover, which are not available with legacy MySQL asynchronous replication technologies. SchoonerSQL provides high data consistency and fast, automated fail-over across a WAN. These are based on Schooner innovations in high performance, multi-threaded, integrated asynchronous replication which eliminates slave lag, provides automated fail-over, and delivers high throughput.

4.2.1.2 Unlimited Query Scaling with High Availability and High Performance

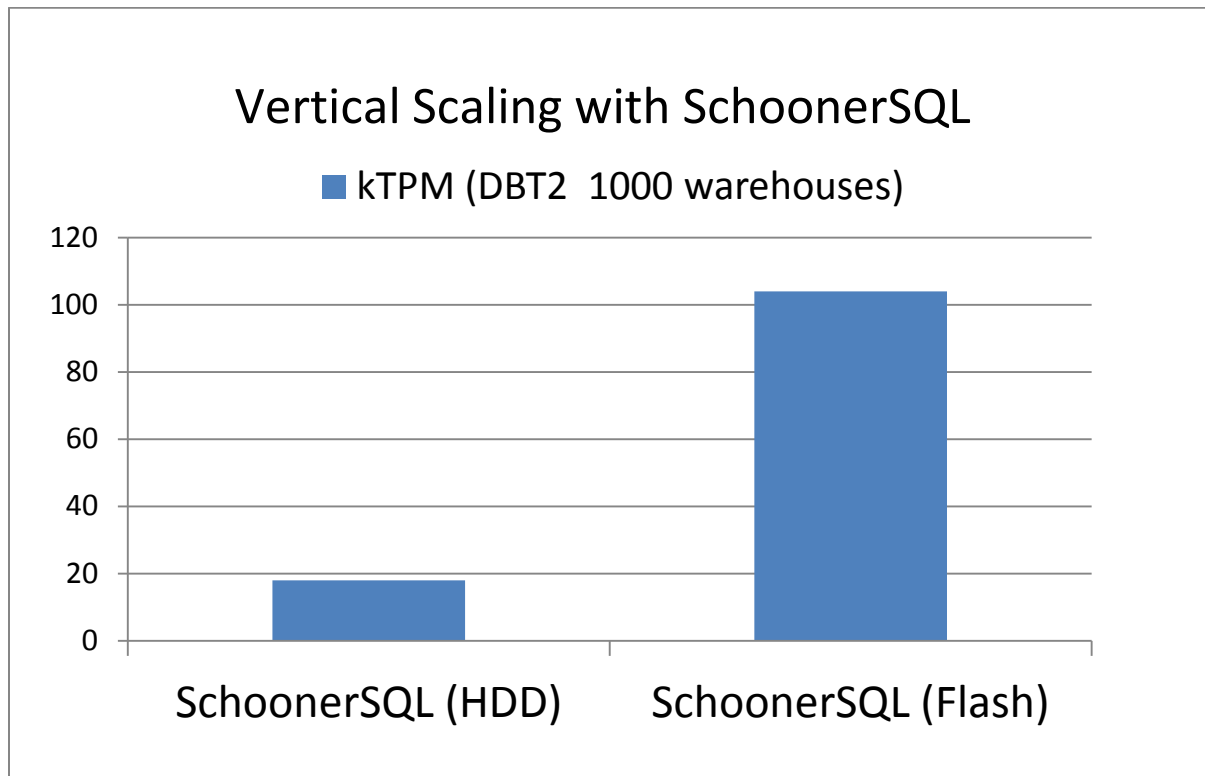
Synchronous replication provides the fundamental solution for eliminating data loss and providing high availability through automated fail-over in the event of a master failure. With SchoonerSQL there is no cluster overhead for a query, and query nodes can be added to a synchronous cluster with near-linear performance as shown below, meeting most workload demands.



Cluster update mechanisms may limit the number of nodes within a synchronous cluster for some workloads. When this limit is reached the cluster is further scaled to meet the load by configuring a base synchronous master/ read master replication group to feed multiple integrated asynchronous replication groups. The Schooner mechanisms discussed above to achieve high performance, highly-available integrated asynchronous replication for WAN are applied to local integrated asynchronous replication groups. Schooner ensures that no data is lost, high service availability is retained with instant automated fail-over, maximal consistency is maintained, and unified cluster-wide administration is provided.

4.2.1.3 Unlimited Database Update Scaling with High Availability and High Performance

Modern commodity multi-core processors and memory technologies offer the potential for high vertical scaling at low cost. SchoonerSQL is uniquely able to fully exploit commodity processor and storage technologies. This chart shows that by adding flash memory to standard dual-socket Intel Westmere servers, SchoonerSQL increases throughput by ten times. Vertically scaling performance by exploiting low cost, high performance commodity technology is compelling. It eliminates the complexity and performance variability encountered with cross-server data partitioning.



For some workloads there is a limit on vertically scaling updates to a master database. After optimal vertical scaling of updates for a master has been done, additional update scaling is accomplished through horizontal sharding. There are two fundamental approaches to horizontal sharding: Application Aware and Application Transparent.

Application-aware sharding partitions the total dataset into multiple databases and applications explicitly select the database to access. For data partitions that are natural and long lived, this can be an appropriate scaling model, with each database implemented with synchronous replication, vertical update scaling, and horizontal read scaling for high availability and high performance.

SchoonerSQL provides application-transparent sharding with tightly integrated dbShards software. Administrator tools are provided for analyzing and optimizing sharding, database layout, and queries.

4.3 Shared-Nothing MySQL Clustering: Oracle MySQL Cluster NDB 7.1 and Clustrix

A shared-nothing database architecture transparently partitions the data among nodes and breaks up a query into sub-queries which are fanned out to the partitions and rejoined to complete a transaction. This approach is taken in Oracle MySQL Cluster NDB 7.1 and Clustrix. These products have several deficiencies relative to deploying mission-critical MySQL. They are not compatible with InnoDB, the standard storage engine used in

MySQL. Query and update performance is highly variable, and very poor when queries cross nodes (such as joins or range operations) or when the data is not in DRAM. They impose hard limits on query scaling, database size, and cluster size. Additionally, Clustrix does not prevent data loss and requires the purchase of custom hardware, which increases the cost and adds the risk of obsolescence.

Table 3 compares SchoonerSQL, Legacy MySQL 5.x, MySQL NDB Cluster, and Clustrix relative to the mission-critical database requirements and metrics.

Table 3: MySQL Alternatives for Mission-Critical Deployments

	Legacy MySQL 5.X	MySQL Cluster NDB 7.1	Clustrix	SchoonerSQL
Fail-Over Downtime	Minutes-hours	seconds	seconds	seconds
Automated Fail-over	No	Yes	Yes	Yes
Data Loss	Yes	No	Yes	No
Data Consistency	No	Yes	Yes	Yes
Performance	Med	Med/Low	Med/Low	High
Scalability	Low	Med/Low	Med/Low	High
Ease of Management	Low	Med/Low	High	High
WAN Scalability and Fail-Over	No	No	No	Yes
InnoDB Compatible	High	Med/Low	Med/Low	High
Custom Hardware	No	No	Yes	No
Cost (TCO)	High	Med	High	Low

4.4 MySQL-Independent Technologies and Products: Linux DRBD, Golden Gate, Tungsten

In addition to MySQL-specific technologies, there are several MySQL-independent technologies sometimes used to improve MySQL service availability.

4.4.1 Linux DRBD (Distributed Replicated Block Device)

Linux DRBD provides operating system block-level replication to a standby server, transparent to the database. Linux DRBD can eliminate data loss, and potentially enable master failover to the standby in minutes. But the standby is not transactionally consistent so it cannot service load. The passive standby must restart with an empty buffer pool (cold re-start), which could require hours to restore full service performance. Also, DRBD can propagate corruptions since log checksums are not being computed. Most importantly, the query slaves are still operating with loosely-coupled asynchronous replication, so the deployment has the same issues as MySQL 5.5/5.6 relative to low consistency, poor performance, low utilization, complex administration, and high TCO.

4.4.2 Very Loosely-Coupled Products: Golden Gate and Tungsten

Golden Gate converts MySQL's asynchronous Bin log to a common log format for heterogeneous database replication interoperability with Oracle, IBM DB2, and Microsoft SQL Server.

Continuent's Tungsten converts the MySQL asynchronous Bin log to a transaction history log. Tungsten uses JDBC through a client proxy to access MySQL indirectly for heterogeneous database replication interoperability with PostgreSQL and to provide automated master fail-over within a MySQL deployment.

These loosely-coupled HA technologies cannot be practically used to scale MySQL database Master - Slave deployments. They have all of the severe mission-critical limitations that result from loosely-coupled asynchronous serial Bin log approaches that we discussed above for legacy MySQL 5.x. Moreover their resulting performance would be significantly worse than that achievable with legacy MySQL 5.x.

5. Best Practices: Evaluating the Options and Trade-offs for Your Datacenter

We have shown the range of MySQL architecture and product alternatives for high availability and performance scalability, and evaluated them relative to the mission-critical database metrics. Table 5 summarizes best practices to maximize mission-critical database metrics. You can contact Schooner to help you evaluate the options and trade-offs for your data center.

Table 5: Mission-Critical Database Best Practices

<u>Goal</u>	<u>Best Practice</u>
High Availability	Local parallel synchronous and WAN parallel asynchronous replication; automation of recovery from failures
High Data Integrity	Synchronous replication to eliminate master data loss and maintain fully consistent data; parallel asynchronous replication with high data consistency levels (near zero lag) for unlimited query scaling and WAN disaster recovery
Excellent Performance and Scalability	Effective vertical and horizontal scaling for high performance update scaling; unlimited horizontal query scaling; multi-threaded parallel asynchronous WAN replication
Simple and Powerful Administration	Centralized management; automation; visibility (statistics); alerts
Cost Effective	Effectively leverage commodity technology with high utilization to minimize capital and operating expense; eliminate service interruptions to maximize revenue
Standards and Compatibility	100% standards compliance

6. The SchoonerSQL Advantage

Schooner has created a truly mission-critical MySQL + InnoDB solution that maximizes all the mission critical database metrics. SchoonerSQL is broadly deployed across industry segments, with customers realizing the business benefits 24/7/365. Let Schooner help your business too!

See the next page for a simple summary of Schooner's advantages in mission-critical MySQL.

The SchoonerSQL Advantage for Your Datacenter

MISSION CRITICAL



Highest Availability

- No service interruption for planned or unplanned database downtime
- Instant automatic fail-over
- On-line upgrade and migration
- 95% less downtime vs. MySQL 5.X
- High performance WAN with auto-failover



Highest Data Integrity

- No lost data
- Synchronous cluster-wide consistency
- Asynchronous cluster consistency
- Reduces slave lag by 95%



Visibility and Control

- Easy cluster administration
- No error-prone manual processes
- Monitoring and Optimization



Highest Performance and Scalability

- 4-20x more throughput/server vs. MySQL 5.5
- High performance synchronous replication
- Transparent sharding (with dbShards)



Out-of-the-box Product

- Full MySQL + InnoDB: not a toolkit
- Free your staff to build your business, not a custom database



Compelling Economics

- Cut server capex (consolidation)
- Cut opex (power, pipe, DBA time)
- TCO 70% cheaper than MySQL 5.5



100% MySQL/InnoDB Compatible

Evaluating the Options and Trade-offs for Your Data Center?

Let Schooner Help!

info@schoonerinfotech.com

www.schoonerinfotech.com

Tel: +1 408-773-7500