Data

- Most important and valuable component of modern applications and websites
- Driving revolutionary changes in computing and the internet
  - New opportunities for generating revenue
  - More efficient use of current business processes and infrastructure
- Data access downtime or poor performance has a major cost to a business’ bottom line
“Let me tell you the difference between Facebook and everyone else, we don't crash EVER! If our service is down for even a day, our entire reputation is irreversibly destroyed!

Facebook and Google invest hundreds of millions of dollars every year on custom software and hardware infrastructure to optimize availability, performance, administration, and cost.
Maintaining data availability and response time is critical for key classes of businesses:
- Web 2.0
- eCommerce
- High-volume websites
- Telecommunications

IT departments and application developers seek architectures and deployments providing:
- high service availability
- resilient performance scalability

Meet rising service demand while controlling capital and operating expenses
Mission-Critical Database Requirements

High Availability

High Performance and Scalability

Simple and Powerful Administration

Data Integrity

Cost Effective

Standards and Compatibility

Mission Critical
## Mission-Critical Database Goals and Metrics

<table>
<thead>
<tr>
<th>Goals</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Availability</td>
<td>Service unavailability (minutes/year) from failures, disaster recovery, or during planned administration</td>
</tr>
<tr>
<td>High Data Integrity</td>
<td>Probability of data loss or corruption; data consistency levels</td>
</tr>
<tr>
<td>High Performance and Scalability</td>
<td>Transaction throughput, response time; performance scalability; performance stability</td>
</tr>
<tr>
<td>Simple and powerful administration</td>
<td>Ease of cluster administration; fail-over automation; monitoring and optimization tools</td>
</tr>
<tr>
<td>Cost effective</td>
<td>Total cost of ownership (TCO); return on investment (ROI)</td>
</tr>
<tr>
<td>Standards and Compatibility</td>
<td>Level of standards compliance and certification</td>
</tr>
</tbody>
</table>
Opportunity for Large Impact on All Mission Critical Dimensions

- Flash
- Multi-core
- Optimized Database Architecture
- Cloud
Tightly-Coupled Flash Optimized Database with Synchronous Replication

- Cluster Admin
- Parallel Execution Threads
- Parallel Replication Threads
- OS
- Standard X86 Server
- MySQL

Concurrently Executing Transactions

Parallel Synchronous Replication
Key Resource Management Algorithm Design
Requirements

• Processor
  • Multi-core scalability
    • fine-grained locking, concurrent data structures
• Storage
  • Log files on HDD with persistent DRAM controller
    • Fast, saves flash for high access data
  • Concurrent DRAM buffer-pool management algorithms
    • Multi-threaded background write of dirty blocks so clean on misses
• Batched commits
• Highly-parallel multi-threaded flash-memory access
  • Utilizes ~150k IOPS for balancing a 2 socket Westmere Server with 64GB DRAM
  • Flash Cache give ~80% throughput if database working set fits in flash: must size

• Network
  • Memory to memory multi-threaded parallel synchronous
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 Hard Disc Drives
Lower Response Times

Response Time (ms)

- 2-node 5.5 async
- 2-node 5.5 semi
- 2-node Schoo
Higher Performance Stability

5.5 Async

Master Throughput vs. Time

5.5 Semi-sync

Master Throughput vs. Time

SAC

Master Throughput vs. Time

© 2011 Schooner Information Technology, p

Thursday, August 11, 2011
Increased Service Availability and Data Integrity

Availability Improvement from Synchronous Replication
(% Cumulative Down Time Reduction)

- MySQL 5.5.8
- Schooner Active Cluster Auto-Failover (Unplanned Downtime)
- Schooner Active Cluster On-Line Upgrades (Planned Downtime)
Lower Cost

- Reduced capital and operating costs through reduction in servers, power, space, admin
- Savings from increased service availability and associated revenue and customer retention

TCO and ROI models are customer and workload specific
- Function (throughput/server; server, rack, and network costs, software license and support costs, admin costs; space and power costs; cost of downtime)
Simplified Administration

- Fail-over can be completely automatic and instant
  - requiring no administrator intervention or service interruption
- Cluster Administrator GUI and CLI can provide a single point for cluster-wide management
  - single click slave creation and database migration; monitoring; trouble-shooting; tuning
## Mission-Critical Database Best Practices

<table>
<thead>
<tr>
<th>Goal</th>
<th>Best Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Availability</td>
<td>Replication (synch local and asynch parallel WAN); automation of failure detection and recovery</td>
</tr>
<tr>
<td>High Data Integrity</td>
<td>Synchronous replication to eliminate data loss and fully consistent data; combined with parallel asynchronous replication for WAN disaster recovery</td>
</tr>
<tr>
<td>Excellent Performance and Scalability</td>
<td>Effective vertical and horizontal scaling for exploiting flash and multi-core</td>
</tr>
<tr>
<td>Simple and powerful administration</td>
<td>Centralized management; automation; visibility (statistics); alerts</td>
</tr>
<tr>
<td>Cost effective</td>
<td>Leverage commodity hardware and software; achieve high hardware utilization</td>
</tr>
<tr>
<td>Standards and Compatibility</td>
<td>100% standards compliance and certification</td>
</tr>
</tbody>
</table>
Cloud Requirements and Challenges for Scaled Enterprise Services

- Cloud providers must deliver:
  - guaranteed service availability, performance, and elastic scale
  - multi-tenant management and security
  - and a net TCO savings vs. dedicated data centers

- Barriers in deploying enterprise class services into the cloud at scale
  - For many classes of applications and services:
    - the realized performance and availability characteristics of cloud deployments are disappointing at scale
    - the large quantity of cloud instances needed to support scaling a deployment drive the cost of cloud deployment to unacceptable levels
  - Opportunity for flash, but innovation is required
Current Cloud Virtualization: Successes and Limitations

- Cloud server-virtualization
  - Provisioning application instances in virtual machines on servers
    - combine existing applications with multi-core systems to increase utilization
    - elasticity of service capacity through dynamic provisioning of more or fewer application instances based on the current workload demand.

- Successes
  - applications that scale horizontally and can run under a VM hypervisor within a server’s DRAM (eg web application tier)
  - works well for low volume apps and services (start-ups, new games, ...)

- Problems: scaled production databases
  - virtualization kills performance if they do not fit in DRAM
Cloud Virtualization Impact on Production Databases

- Databases in production cloud environments:
  - provide additional data partitioning (very small databases)
  - provide additional caching layers to minimize I/O (breaks ACID)
  - provision many more database instances than in a non-virtualized environment

- Net Impact
  - drives up application and management complexity
  - increases cost
  - reduces service availability and data integrity

- Less than 10 percent of production data-tier server workloads are virtualized today.
Fusing Cloud + Flash + Optimized Databases

• Short term
  – virtualized machine instances for the web and application tiers
  – non-virtualized, vertically scaling data-tier solutions
    • Exploit balanced commodity, flash-based, multi-core system configurations
  – custom management APIs and tools to link together in a hybrid cloud
Fusing Cloud + Flash + Optimized Databases

• Longer Term: Innovation Required
  – Need improved virtualization technologies
    • Flash optimized virtualization cutting flash access overhead
    • unified virtual administration model
      – applicable to all tiers in the data center including flash-optimized data tier
      – dynamic provisioning, management, monitoring, and accounting

  – Large potential Quality of Service and TCO Benefits
    • increased performance, scalability, and service availability
    • reduced capital and operating expenses
Thank You!

Schooners, first built in the 1700s, applied an innovative design to the standard cargo sailing ship, enabling stupendous levels of speed and range. They enabled a set of visionary companies to enter new markets on a global basis. Where can a Schooner take your company?
The Schooner MySQL® with Active Cluster™ Advantage

Schooner MySQL with Active Cluster

High Availability
- No service interruption for planned or unplanned database downtime
- Instant automatic fail-over
- On-line upgrade and migration
- 90% less downtime vs. MySQL 5.5
- Full WAN support with master auto-failover

Great Performance and Scalability
- 4–20x more throughput/server vs. MySQL 5.5
- High performance synchronous and asynchronous replication

Compelling Economics
- Cut server capex (consolidation)
- Cut opex (power, pipe, DBA time)
- Increase revenue (eliminate service interruptions)
- TCO 70% cheaper than MySQL 5.5

100% MySQL Enterprise Compatible and Certified

Broad Industry Deployments
- eCommerce, telco, financial,...

Data Integrity
- No lost data
- Cluster-wide data consistency

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

Out-of-the-box Product
- Full MySQL + InnoDB: not a toolkit
- Free your staff to build your business, not a custom database

www.SchoonerInfoTech.com

© 2011 Schooner Information Technology, p

Thursday, August 11, 2011
How Best to Provide WAN Replication and Disaster

• WAN/geographically dispersed data centers
  – Typically requires Asynchronous replication
    – Can’t add additional ~200ms with high potential variance to query response time for synchronous replication

• Goal: WAN async slave should automatically fail-over when synchronous master fail-over occurs
  • Requires WAN async replication to be loosely integrated with synch replication group

• Goal: Limit remote slave lag and recovery to ~ WAN latency
  – Maximize WAN data consistency
  – Minimize disaster recovery time
  • Requires high performance asynchronous replication
    – Need multi-threaded async for parallelizing updates

Thursday, August 11, 2011
Limitations of Synchronous Replication: How Best to Scale Database Queries

Query Scaling in a Synchronous Replication Group

- Fully replicated Master/Slave cluster
  - No cluster overhead for adding queries to a slave
  - Can add synchronous query nodes linearly
  - Update synchronization and cluster management eventually limit
    - workload dependent
- With partitioned databases, scaling is sub-linear with severe cross node query degradation
Limitations of Synchronous Replication: How Best To Scale Queries (ctd)

1 Synch Replication Group + Multiple Integrated Asynchronous Replication Groups

- Can infinitely scale reads
- No data loss, auto-failover
- Near zero slave lag requires asynchronous parallel update slave replication
Limitations of Synchronous Replication: How Best to Scale Updates

- Database Update Scalability
  - Vertically scale with commodity: flash memory, more cores, higher frequency

- Compelling option exploiting low cost, high performance commodity technology
Limitations of Synchronous Replication: How Best to Scale Updates (ctd)

- Database Update Scalability

...After Optimal Vertical Scaling:

Horizontally Scale Through Sharding

- Application Transparent
  - Automated internal partitioning (MySQL NDB Cluster and Clustrix)
    » High query performance sensitivity (very slow cross partition queries)
  - Administrator analysis and set-up tools (Schooner MySQL Active Cluster)
    » DB Shards allows layout and query data access optimization

- Application Aware
<table>
<thead>
<tr>
<th>MySQL Specific Database Alternatives for Mission-Critical Deployments</th>
<th>MySQL 5.5</th>
<th>MySQL NDB Cluster</th>
<th>Clustrix</th>
<th>Schooner MySQL Active Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fail-Over Downtime</td>
<td>Minutes-hours</td>
<td>seconds</td>
<td>seconds</td>
<td>seconds</td>
</tr>
<tr>
<td>Automated Fail-over</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Data Loss</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Data Consistency</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Performance</td>
<td>Med</td>
<td>Med/Low</td>
<td>Med/Low</td>
<td>High</td>
</tr>
<tr>
<td>Scalability</td>
<td>Low</td>
<td>Med/Low</td>
<td>Med/Low</td>
<td>High</td>
</tr>
<tr>
<td>Ease of Management</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>WAN perf and fail-over</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>InnoDB Compatible</td>
<td>High</td>
<td>Med/Low</td>
<td>Med/Low</td>
<td>High</td>
</tr>
<tr>
<td>Custom Hardware</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Cost (TCO)</td>
<td>High</td>
<td>Med</td>
<td>High</td>
<td>Low</td>
</tr>
</tbody>
</table>

Thursday, August 11, 2011
**MySQL-Independent Replication: Linux DRBD Passive Standby Master**

- **Master**
  - MySQL clients
  - MySQL
  - Linux
  - Standard X86 Server

- **Passive Slave**
  - MySQL
  - Linux
  - Standard X86 Server

- **Disc block level sync replication**

- **Eliminates data loss**
- **Master failure–over to standby in minutes**

But not transitionally consistent:
- **Stand–by master cannot service load**
- **No warm re–start => hours for full service**
- **Can propagate corruptions(no log checksums)**

- **Slaves are still operating with asynchronous replication => same issues as MySQL 5.5/5.6**

- **High administrative complexity**
- **Reduced service availability**
- **Inconsistent slave data**
- **Poor performance**
- **High TCO**

---

© 2011 Schooner Information Technology, p

Thursday, August 11, 2011
MySQL–Independent Replication for Heterogeneous Database Interoperability

Very loosely-coupled external replication services based on MySQL’s asynchronous replication Bin log

**Oracle Golden Gate:**
- Converts MySQL Asynchronous Bin log to a common log format
- Heterogeneous database replication interoperability: Oracle, IBM DB2, and Microsoft SQL Server

**Continuent Tungsten Replicator**
- Converts the MySQL asynchronous Bin log to a transaction history log
- Uses JDBC through a client proxy to access MySQL indirectly
- Heterogeneous database replication interoperability: PostgreSQL

**If used in MySQL Master – Slave deployments:**
- Performance is significantly worse than MySQL 5.5/5.6
- Same issues as all loosely coupled async Bin log approaches
  - reduced service availability
  - poor data integrity
  - high administrative complexity
  - high TCO
<table>
<thead>
<tr>
<th></th>
<th>MySQL 5.5</th>
<th>MySQL NDB Cluster</th>
<th>Clustrix</th>
<th>Linux DRDB</th>
<th>Continue Tungsten</th>
<th>Golden Gate</th>
<th>Schooner MySQL Active Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fail-Over Downtime</strong></td>
<td>Minutes–hours</td>
<td>seconds</td>
<td>seconds</td>
<td>minutes</td>
<td>seconds</td>
<td>Minutes–hours</td>
<td>seconds</td>
</tr>
<tr>
<td><strong>Automated Fail-over</strong></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Data Loss</strong></td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>Data Consistency</strong></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>Med</td>
<td>Med/Low</td>
<td>Med/Low</td>
<td>Med</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>Low</td>
<td>Med/Low</td>
<td>Med/Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td><strong>Ease of Management</strong></td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>Med</td>
<td>Med</td>
<td>High</td>
</tr>
<tr>
<td><strong>WAN perf and auto-fail-over</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>InnoDB Compatible</strong></td>
<td>High</td>
<td>Med/Low</td>
<td>Med/Low</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td><strong>Custom Hardware</strong></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>