
ORACLE®

MySQL : 5.6 the Next Generation

Lynn Ferrante

Principal Consultant, Technical Sales Engineering

Northern California Oracle Users Group November 2012

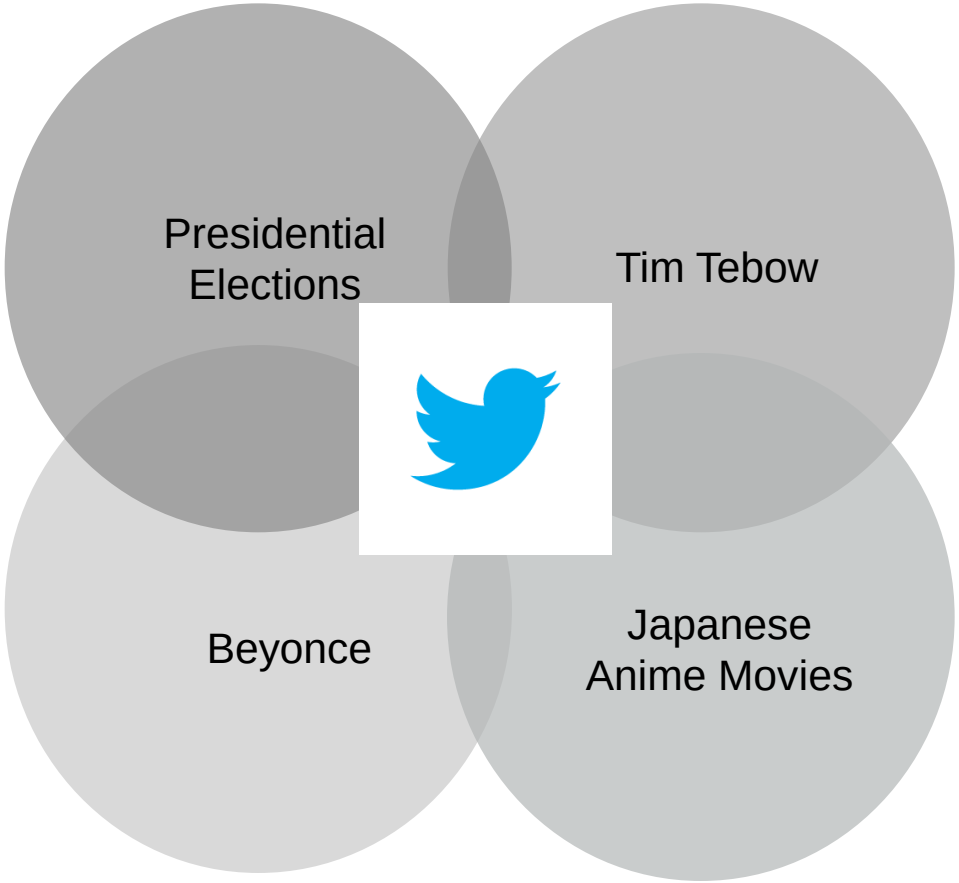
Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decision. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda

- Oracle's MySQL Strategy
- MySQL Overview
- What's New in MySQL 5.6
- Extensions and Tools
- Q&A

Intersection?



Tweets

- Anime: 25,000 per second
- Beyonce:327,452 per minute
- Election: 9,965 per second 8-9 pm
- Tebow: 15,107 8:20 pm

Random Server at Twitter

- 212 days

  uptime of random MySQL server at twitter

- 127 billion

  number of queries executed on single server

- 24.9 trillion

  innodb_rows_read, 1.36M per second

source: Jeremy Cole, Twitter DBA, MySQL UC 2011

Why MySQL Makes Sense for Oracle

- Complete Solutions
- Best of Breed at Every Level
- On Premise and in the Cloud
- MySQL: Web, Mobile & Embedded

Industry Leaders Rely on MySQL

Driving MySQL Innovation

MySQL Enterprise Monitor 2.2
MySQL Cluster 7.1
MySQL Cluster Manager 1.0
MySQL Workbench 5.2
MySQL Database 5.5
MySQL Enterprise Backup 3.5
MySQL Enterprise Monitor 2.3
MySQL Cluster Manager 1.1

All GA!

2010

MySQL Enterprise Backup 3.7
Oracle VM Template for MySQL Enterprise Edition
MySQL Enterprise Oracle Certifications
MySQL Windows Installer
New MySQL Enterprise Commercial Extensions

All GA!

MySQL Database 5.6 DMR*
MySQL Cluster 7.2 DMR
MySQL Labs!
("early and often")

2011

MySQL Cluster 7.2
MySQL Utilities 1.0.6
Database Migration Wizard
New Windows Tools/Features
New MySQL Enterprise Commercial Extensions

MySQL Database 5.6 RC
MySQL Cluster 7.3 DMR
MySQL Cluster features in MySQL Labs

All GA!

2012

*Development Milestone Release

MySQL Database Architecture

Performance, Reliability, Ease of Use

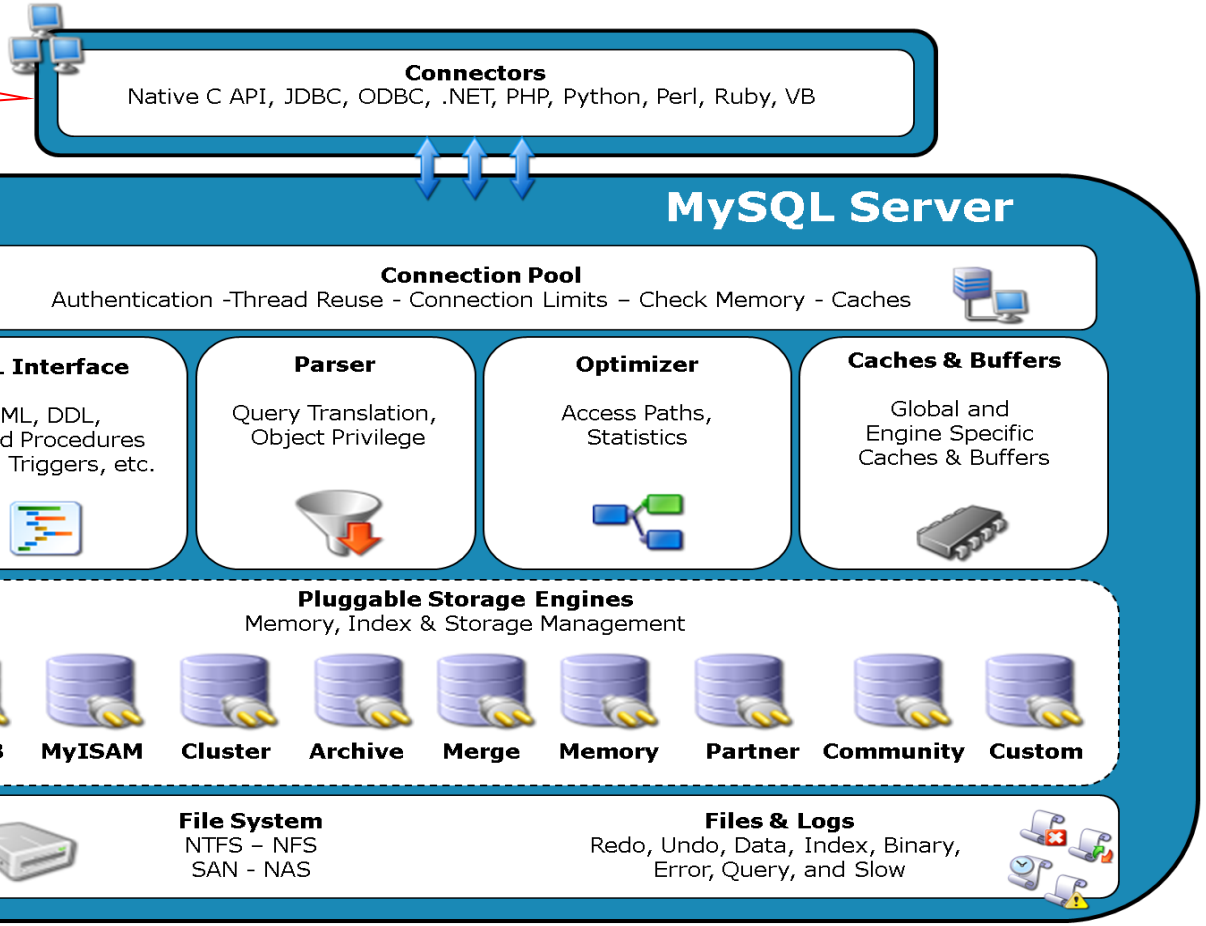
Support for common development languages/platforms

Efficient multi-threaded session handling

Full DML, DDL parsing, cost based optimizer, caching of queries and result sets

Flexible Storage Engine options for application specific storage needs

Flexible logging and physical storage options



MySQL 5.6

#1 Replication

MySQL Replication

- Replication
  - Simple
  - Robust
  - Proven

K.I.S.S

MySQL Replication

Tao of YouTube

"Choose the simplest solution possible with the loosest guarantees that are practical"

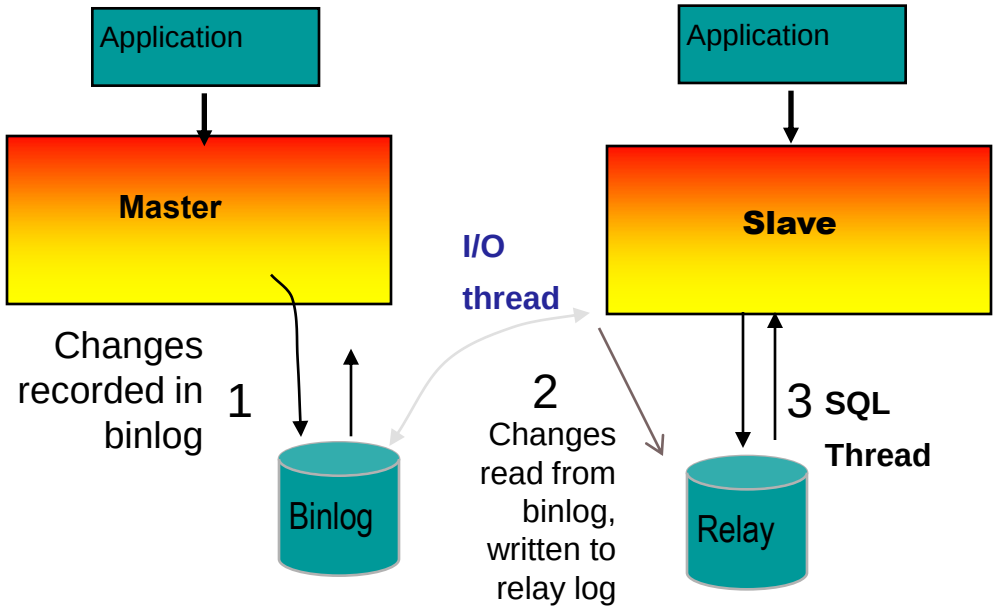
To solve a problem: One word - simple.

"Look for the most simple thing that will address the problem space…"

http://highscalability.com/blog/2012/3/26/7-years-of-youtube-scalability-lessons-in-30-minutes.html

Replication Basics: the Big Picture

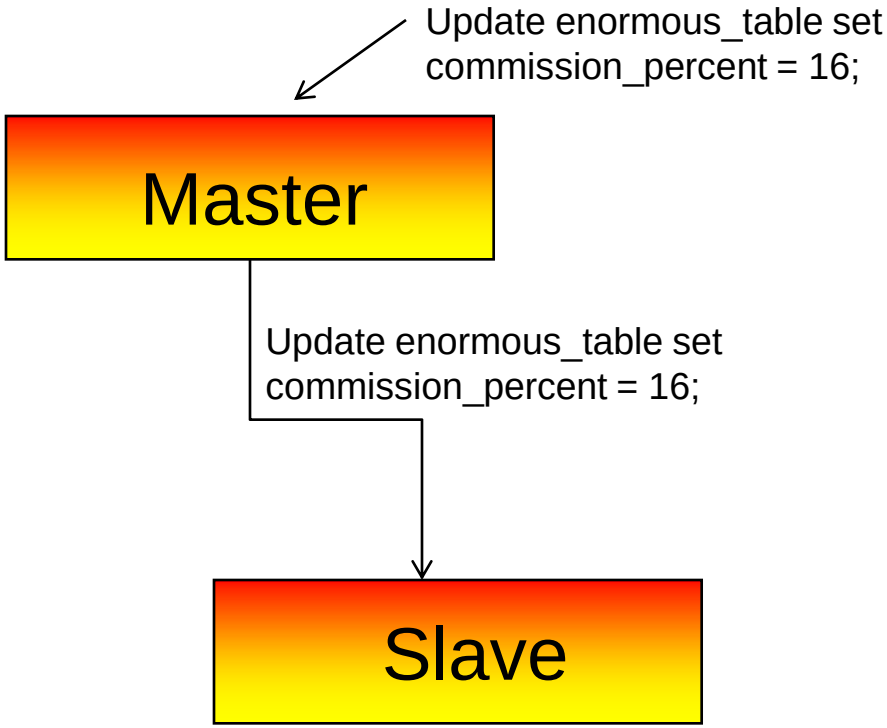
- Native in MySQL
- Each slave adds minimal load on master



Replication Basics: Statement and Row Based Replication

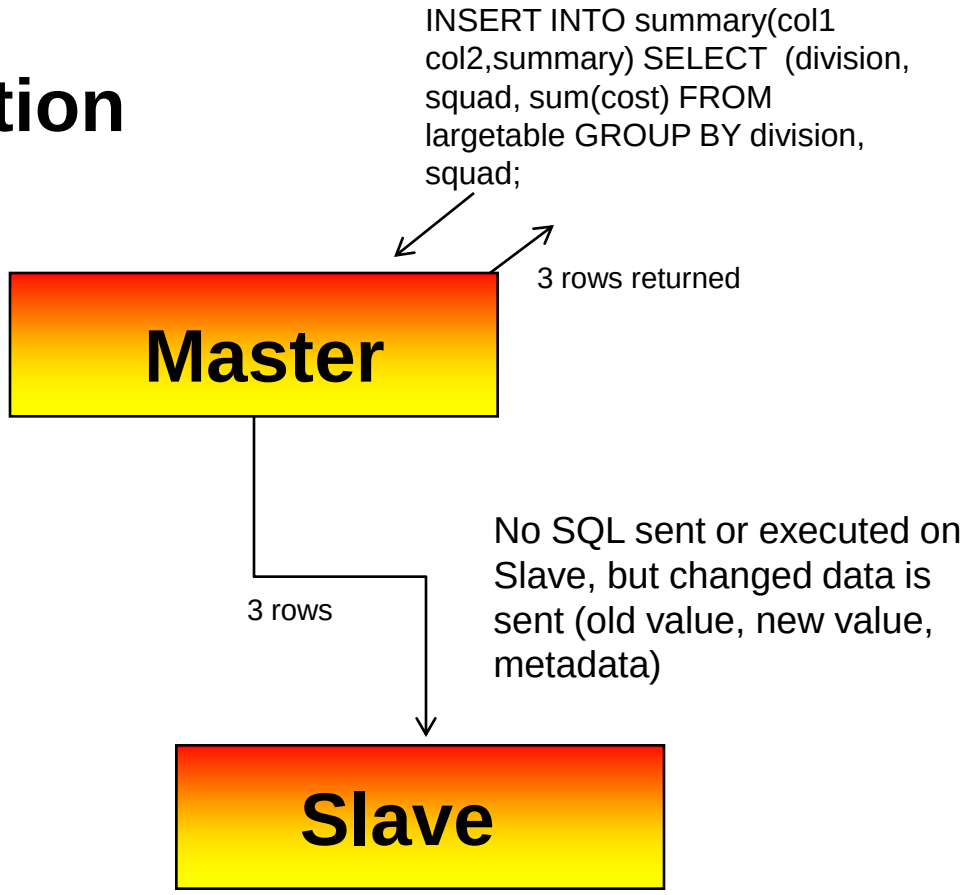
Statement based replication

- Actual SQL from Master is executed on Slave
- Binary log usually compact
- Default

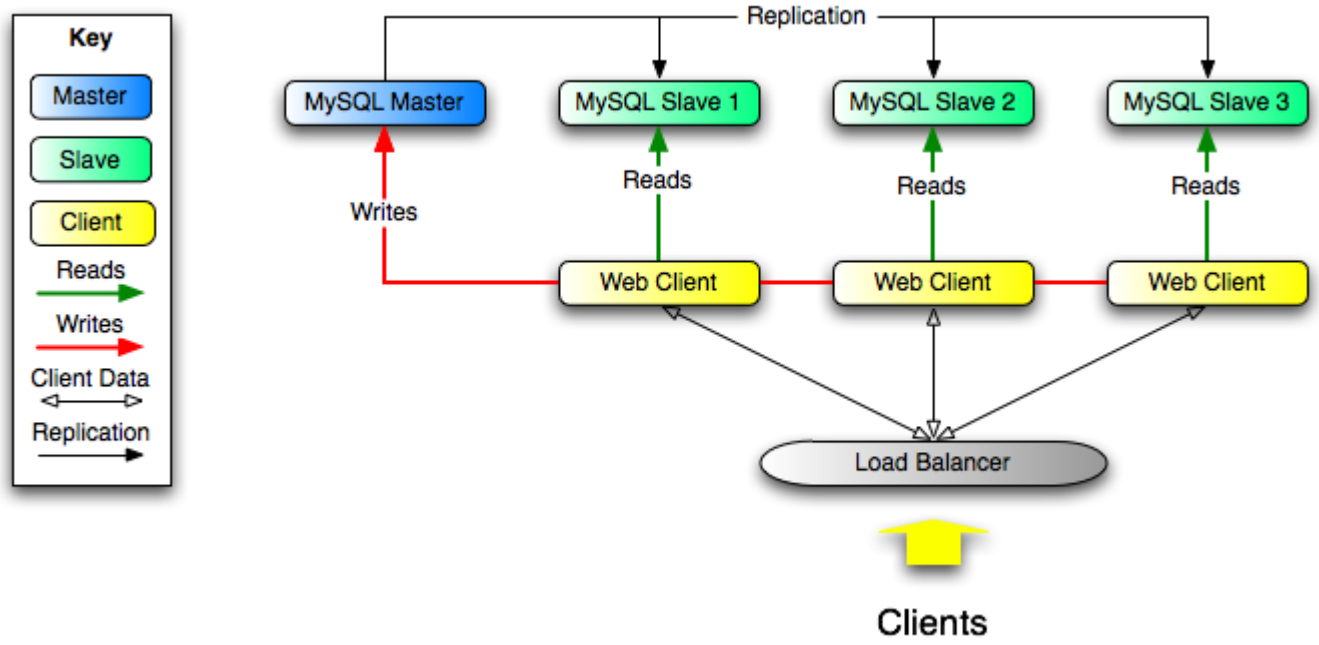


Replication Basics: Row Based Replication

- Replaying some queries can be expensive
- Some statements cannot be replicated with statement based
- Actual changed data is recorded

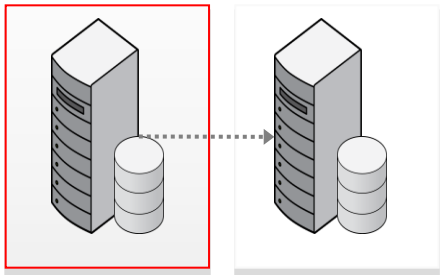


Replication Basics: Scale Out Example

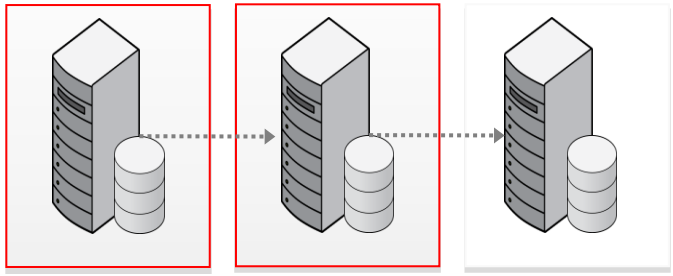


Replication Basics: Topologies

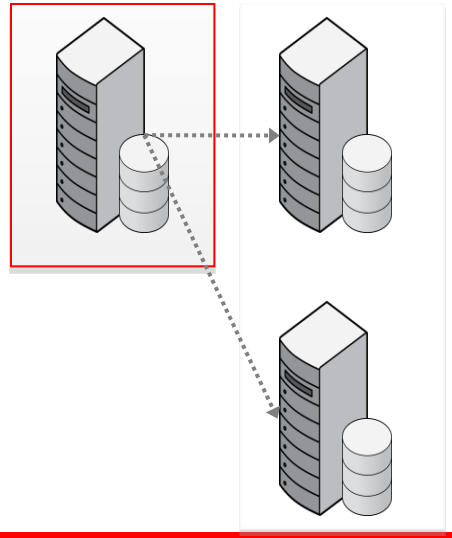
Single



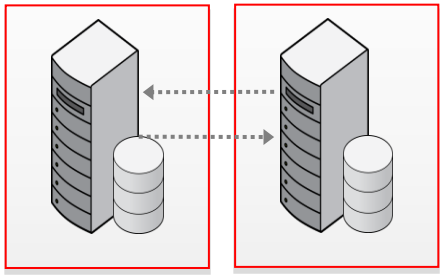
Chain



Multiple



Dual Master/Circular

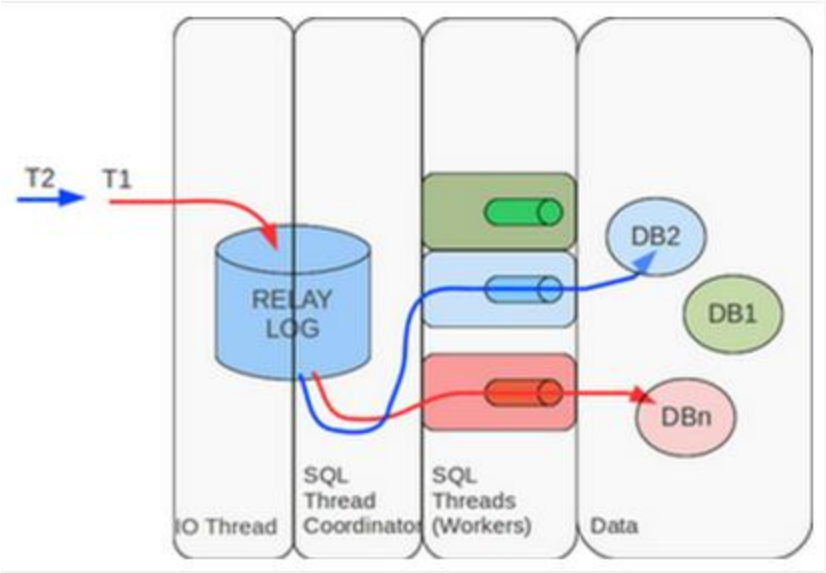


Master

Slave

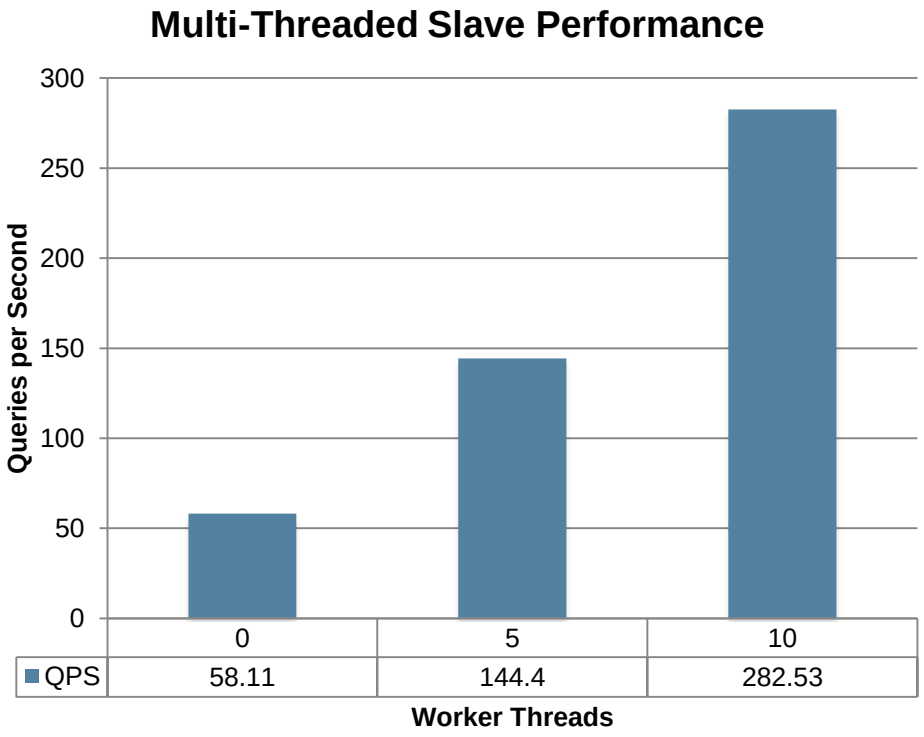
5.6 Replication Multi Threaded Slaves

- Improves replication performance by using multiple threads to apply events
- Slave SQL thread acts as the coordinator for slave worker threads
- Threads split based on schema



5.6 Multi-Threaded Slaves

- 5x Performance Gain
- Reduce Slave Latency
- Per Schema
  - Threads per Schema
  - Multi-tenancy

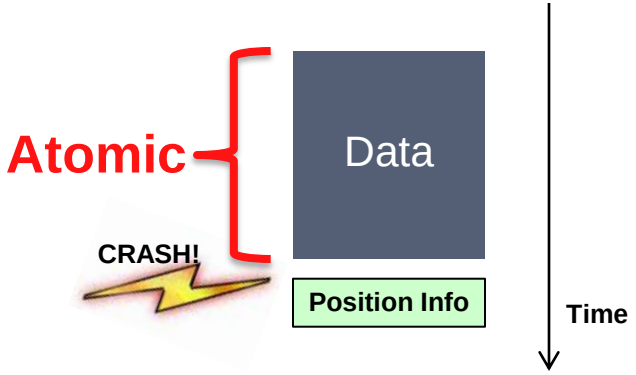


Oracle Linux 6.1, Oracle Sun Fire x4150 m2 Server

MySQL 5.6 Crash-Safe Slaves

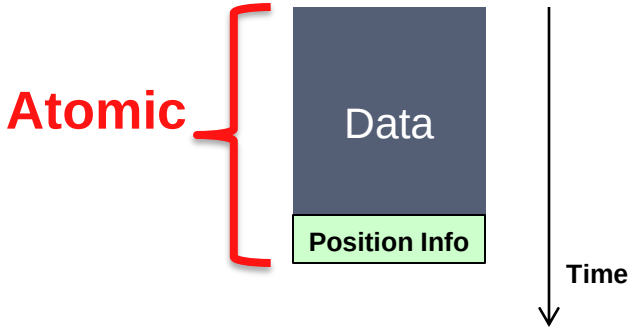
Before:

- Transaction Data: **in tables**
- Replication Info: **in files**



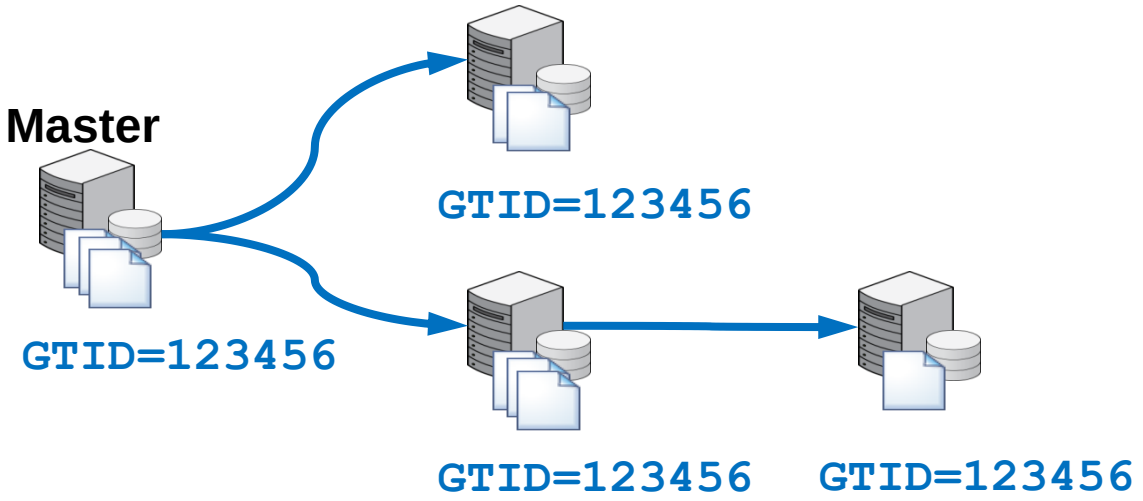
MySQL 5.6

- Transaction Data: **in tables**
- Replication Info: **in tables**



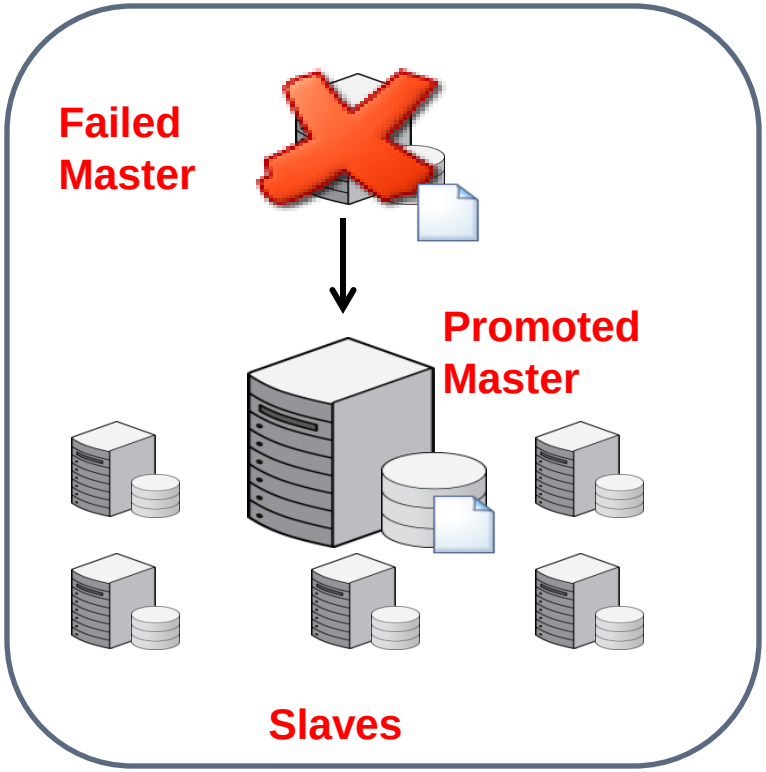
MySQL 5.6 Global Transaction Identifier

- Unique ID for Binlog
- Locate and Track Transactions
- Automate Failover



MySQL 5.6 High Availability Utilities

- failover
  - automatically promote slave on failure
- switchover
  - Automatically promotes slave on switchover
- mysqlreplfailover
  - automatically promote slave on failure
- Workbench download



MySQL 5.6 Binary Log Group Commit

- Significantly Reduce Replication Overhead
- Multiple Transactions
- One Commit

Rock Solid Highly Available Infrastructure

MySQL Replication

- Self healing recovery mechanisms
- Preventing unplanned downtime
  - Automatic failure detection and recovery
- Preventing planned downtime
  - Controlled switchover
- Proactive monitoring to identify issues before they cause an outage

#2 Online Operations

Online DDL for InnoDB Tables

ADD INDEX

DROP INDEX

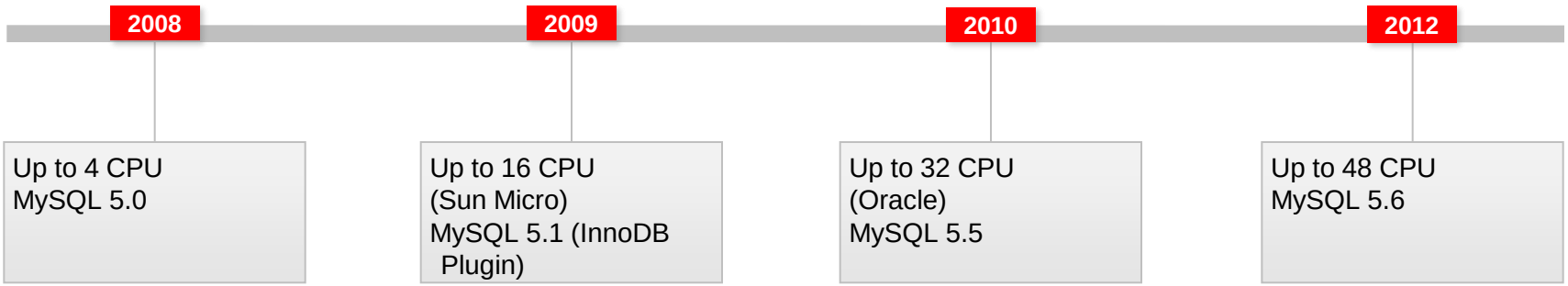
ADD COLUMN

DROP COLUMN

RENAME COLUMN

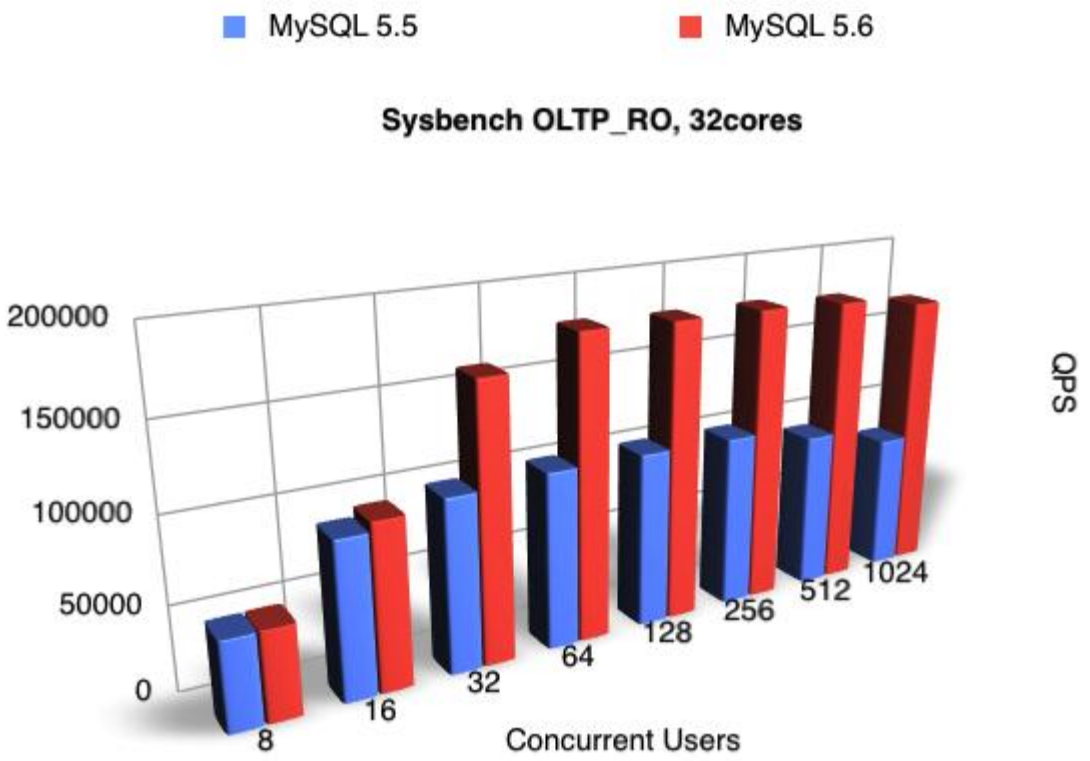
#3 Performance

MySQL Scalability



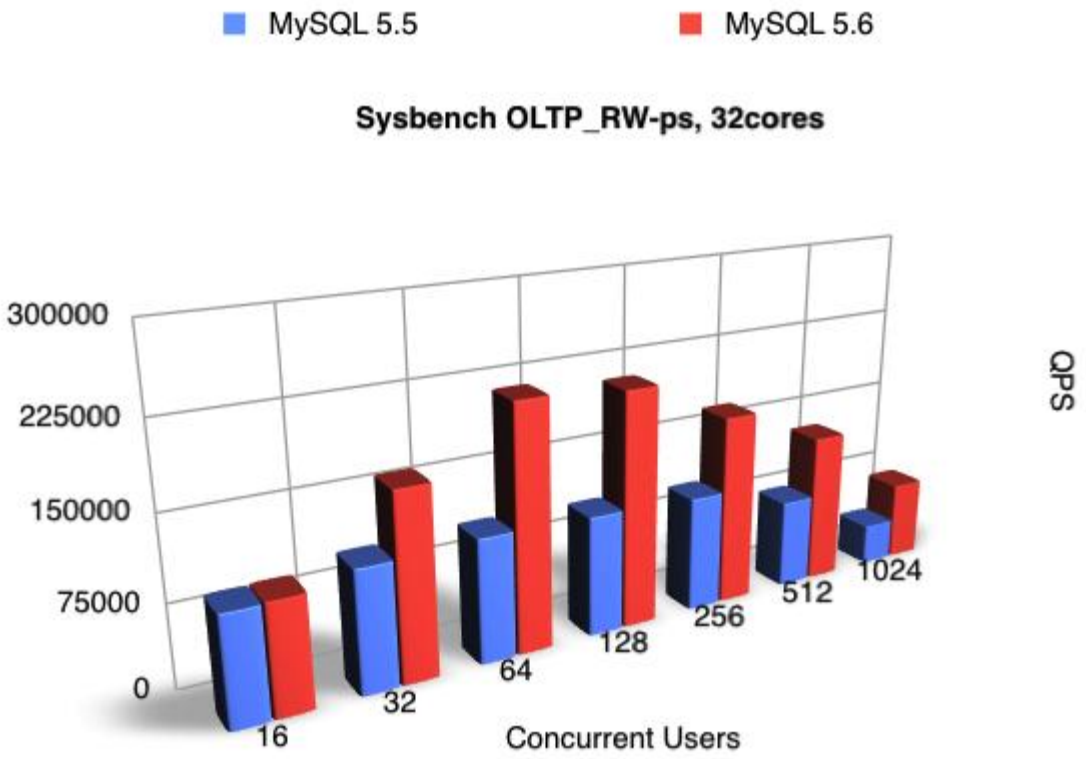
Benchmarks MySQL 5.5 vs. 5.6

- Sysbench
- 32-core
- Same Hardware
- Optimal Tuning



Benchmarks MySQL 5.5 vs. 5.6

- Sysbench
- 32-core
- Same Hardware
- Optimal Tuning



Performance Improvements

InnoDB and MySQL

- Refactored InnoDB
  - Split kernel mutex
  - Separate thread for flushing operations
  - Multi-threaded purge
  - Reduced buffer pool contention
  - New adaptive hashing algorithm
- MySQL
  - Memory Allocation
    - Switch from malloc to better memory allocators for multi-threaded concurrency
  - Lock_open contention (bottleneck when opening tables)

#4 Optimizer

Materialized Subqueries

- DBT Benchmark Query #17
  - ~40 days to execute
  - Now – 6.8 seconds
- Executes Once
- Hash Index

QUERY #17

```
SELECT c_name, c_custkey, o_orderkey, o_orderdate,
 o_totalprice, sum(l_quantity)
  FROM customer, orders, lineitem
    WHERE o_orderkey in (
     SELECT l_orderkey
      FROM lineitem
       GROUP BY l_orderkey
          HAVING sum(l_quantity) > 313)
    AND c_custkey = o_custkey
    AND o_orderkey = l_orderkey
      GROUP BY c_name, c_custkey, o_orderkey,
            o_orderdate, o_totalprice
        ORDER BY o_totalprice desc, o_orderdate
          LIMIT 100;
```

Optimizer Summary

- Additional Optimizations for Complex Queries
  - Optimize Many Tables in Join 25+
  - Postpone Materialization of Views/Subqueries in FROM
  - Indexes for Derived Tables
- Continued Improvements for Online Apps
  - Optimize “IN” clause
  - Optimized SELECT col1, … FROM t1 .. ORDER BY name LIMIT 10
  - Index Condition Pushdown
  - Better Optimizer Diagnostics
    - EXPLAIN for INSERT, UPDATE and DELETE
    - EXPLAIN output in JSON

http://mysqloptimizerteam.blogspot.co.uk/

#5 Performance Schema

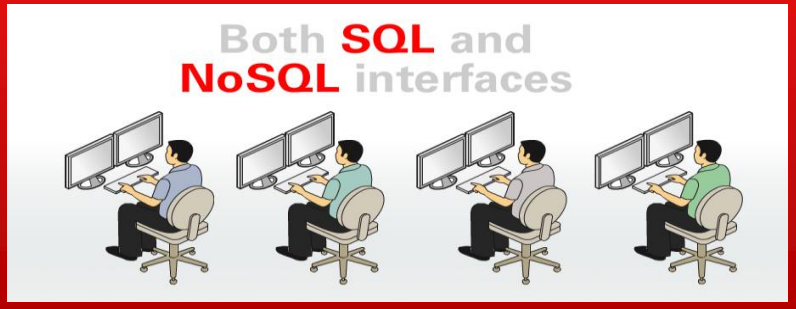
MySQL 5.6 – Performance Schema

- Resource-intensive queries
- Tables/indexes with most load
- Users consuming the most resources
- Network load
- Aggregated statistics by
  - thread
  - user
  - host
  - object

Honorable Mentions

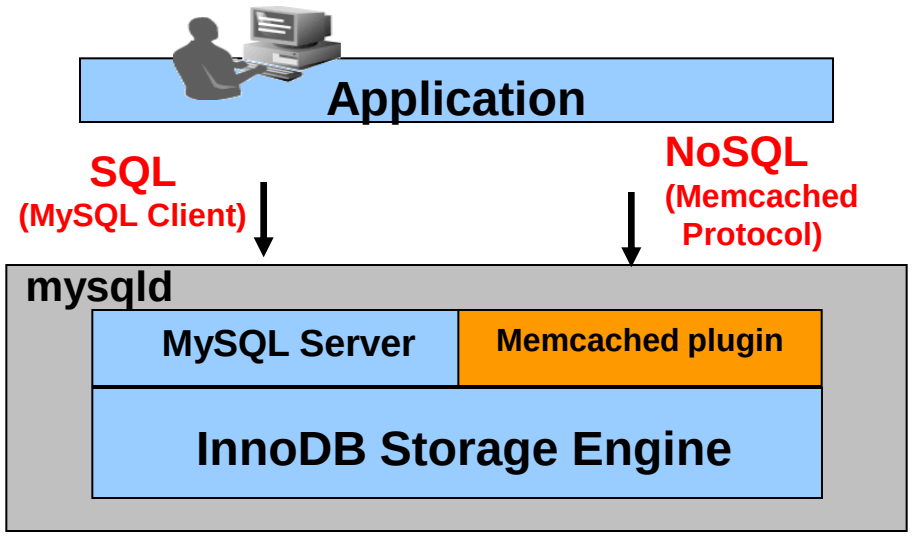
The Best of Both Words - No Trade-Offs
SQL AND NoSQL

- Mix Key Value & Complex Queries
- Transactional Integrity
- Standards & Skillsets



MySQL 5.6: InnoDB

Best of Both Worlds



- **Key-value access to InnoDB**
  - Via industry standard Memcached API
  - Use existing Memcached clients
  - Bypasses SQL parsing
- **NotOnlySQL access**
  - For key-value operations
  - SQL for rich queries, JOINs, FKs, etc.
- **Implemented via:**
  - Memcached plug-in to mysqld
  - Memcached mapped to native InnoDB API
  - Shared process for ultra-low latency

Full Text Search in InnoDB

- FULLTEXT indexes on InnoDB tables
- Keys on text-based content
- Speeds up searches for words, phrases
- Fully transactional, fast look up
- Natural language/Boolean modes, proximity search, relevance ranking

```
create table quotes
( id int unsigned
auto_increment primary
key
,author varchar(64)
, quote varchar(4000)
, source varchar(64)
, fulltext(quote)
) engine=innodb;
```

```
select author as "Apple" from quotes
    where match(quote) against ('apple' in natural language mode);
```

Transportable Tablespaces

Export:

```
CREATE TABLE t(c1 INT) engine=InnoDB;
FLUSH TABLE t FOR EXPORT;
$innodb_data_home_dir/test/t.cfg
UNLOCK TABLES;
```

Import:

```
CREATE TABLE t(c1 INT) engine=InnoDB;
ALTER TABLE t DISCARD TABLESPACE;
ALTER TABLE t IMPORT TABLESPACE;
```

5.5 Oracle Product Integrations

Oracle Integrations

- GoldenGate
- Database Firewall
- Oracle Secure Backup
- WebLogic Server
- Database Adapter for Oracle SOA Suite **
- Oracle Business Process Management **
- Oracle Virtual Directory
- Oracle Data Integrator
- Planned MySQL Enterprise Monitor integration with OEM

5.5 Additional Extensions

5.5 Enterprise Extensions and Tools

- External Authentication
- Auditing
- Thread Pool
- Windows Failover
- DRBD
- Solaris Clustering
- Oracle VM Template
- Monitor
- Hot, Online Backup

MySQL Development Priorities

MySQL Database Development Priorities

- Optimized for Web, Cloud-based, Embedded use cases
- Simplified, Pluggable architecture
  - Maintainability, more extensible
  - More NoSQL options (HTTP, JSON, JavaScript, etc.)
- InnoDB
  - Optimized for mass use cases (read only, fast recovery), GIS
- Easy HA, Replication and Sharding

Hardware and Software

ORACLE

Engineered to Work Together


ORACLE®