

# Towards Improved MySQL Reliability and Scalability

*Ryan Huddleston [rhuddleston@rightnow.com](mailto:rhuddleston@rightnow.com)*

*Manager, Core Infrastructure Services*



# The challenge

- ~70,000 qps in MySQL
- ~75 million page views per day (~2 billion per month)
- 4600 schemas with unique data
- Customers can modify and customize their schema ad hoc
- All dynamic queries generated by our customers, designed in our Product
- ~450 production mysql instances
  - 100,000+ tables on many servers
- Hundreds of TB of data
- High security requirements

# Goals

- We strive for and achieve 99.99% reliability on our mysql tier
- No scheduled downtime
- How do we do that?
  - Dual Master replication
    - Failover per customers to isolate load issues
    - Auto failover during machine failure
  - Natural Sharding, customers have separate schema's
    - Shift load by moving customers between separate DB clusters
    - Spread risk out across hundreds of servers
  - Commodity Servers
    - Share nothing architecture
    - 4 copies of data
    - Local disk allow for use PCI-E SSD for high performance
- Allows for unlimited scalability at DB layer as new customers are added

# MySQL Challenges

- Mutex locking
  - Each drop table locks mysql “table cache” mutex
  - At 1000 qps doesn't take long to overwhelm connections
- Large complex queries, optimizer deficiencies
- Schema evolution during version upgrades or on demand by customers
- Write Transactions Per Second too high

# MySQL Forks

- RightNow customized MySQL Fork - Percona Server 5.1.58 branch available here:
  - <https://code.launchpad.net/~percona-dev/percona-server/rnt-5.1>
- RightNow has been a major sponsor of NRE work on mysql since 2001. e.g in 2002/2003 we sponsored Innodb File per table feature with Innobase Oy.
- Rightnow has sponsored dozens of significant mysql features and submitted hundreds of bugs



# Mutex locking

- Table operations such ALTER/DROP TABLE must acquire “table cache” mutex
- Was a major cause of downtime on our systems
- Switched to XFS from ext3. Problem much better. Mutex lock time no longer increased with table size
- Still problematic, Percona investigated the issue and reduced the time we hold table cache mutex by up to 10x (Yasufumi)
- Even better now Percona has released “innodb\_lazy\_drop\_table=1” which eliminates the DROP TABLE problem

# Complex Queries

- Ongoing battle to prevent large queries from overloading servers
- Queries taking longer than customers expect
- Optimizer deficiencies slow down many large complex queries
  - Have sponsored many improvements to optimizer over the years
- Percona innodb\_stats.patch
  - innodb\_stats\_method=nulls\_equal, fix rows per key estimates
  - innodb\_use\_sys\_stats\_table cache statistics after mysql restarts, also greatly reduces downtime incase of a crash
- Many others...
- MariaDB 5.3/5.5 has rewritten optimizer

# Schema Evolution

Upgrades happen in three parts

- Pre - add tables/columns, move data in batches
  - Cutover - rename columns and tables
  - Post - drop tables and columns
- 
- Failover queue, rolling alters between slaves without customer impact
    - `replication_slave_skip_columns` you can run with RBR and drop columns on slave, not possible with stock mysql
    - Allow mysql to rename a field instantly without rebuilding the table, increase varchar without a table rebuild (`innodb_fast_alter_column`). Not possible with stock innodb
  - `innodb_expand_fast_index_creation.patch` – Speed up Alters and mysqldump loads using fast index creation

# High TPS

- DQA – delayed query aggregation, log and bulk loads stats into database
- PCI-E SSD, flashcache
- Memcache for caching expensive queries

# Future

- Plan to get custom features part of Maria DB 5.5
  - Index merge intersect support for range access. This work was done by MontyProgram AB. (<http://askmonty.org/worklog/Server-Sprint/?tid=21>)
  - Fair choice of index merge optimization (<http://askmonty.org/worklog/Server-Sprint/?tid=24>)
  - optimizer rewritten, subquery optimization, batch key access, multi-read range, hash join...

## Other Technology and Features:

- Galera - synchronous mysql replication <http://codership.com/>
- Multi-threaded replication in MySQL 5.6
- Group commit issues fixed in MariaDB 5.3
- Memcache handler in MySQL 5.6
- HandlerSocket – NoSQL interface into MySQL

# Other MySQL patches

- Other patches included in RightNow Percona Server branch
  - Mysqlbinlog rewrite option to pipe ROW based binlogs to other schema names (--rewrite-db, <http://askmonty.org/worklog/Server-Sprint/?tid=436>)
  - create timestamp with no default "TIMESTAMP NOT NULL DEFAULT NULL"
  - Annotated row based binlog statements for security to log query syntax and comments of data changing statements (<http://askmonty.org/worklog/Server-Sprint/?tid=47>)
  - Row based replication uses most appropriate index instead of first index in table (row\_based\_replication\_without\_primary\_key.patch)
  - Option to allow mysql to skip a single statement and not a whole transaction (replication\_skip\_single\_statement.patch)
  - New alter syntax to allow mysql to rename an index instantly (rename\_index.patch)
  - Pipe ROW based binlogs into a schema that has only subset of rows (--slave-exec-idempotent)
  - Change replication so non-transaction tables are not logged as part of the transaction and are logged immediately (binlog\_direct\_non\_transactional\_updates)
  - Option to prevent innodb from crashing when it encounters corrupted data
  - max\_binlog\_packet.patch adds --max-binlog-packet mysqlbinlog option which fixes issues where you can't roll forward binlogs as there is a single RBR statement over 8MB (if a 7MB row gets an update it results in a 14MB RBR statement), not in HMS yet
  - support for 128 indexes per table
- Many others...

Questions?