3 March 2012

MySQL Developments

Narayan Newton

Lead Sysadmin Drupal.org

Performance Engineer Tag1 Consulting

MySQL Fragmentation

- MySQL used to be just MySQL, there might have been a discussion between 5.0 vs 5.1
- Now MySQL Means:
  - MariaDB
  - PerconaDB
  - Percona Cluster
  - Drizzle
  - Oracle MySQL (5.0, 5.1, 5.5, 5.6-dev)

Presentations Like This Are Terrible.

Focusing On Solutions To Problems....

04

Vertical Scaling, Locking And Reliability

Oracle MySQL 5.5

- InnoDB Plugin Merged
- Vastly Improved Locking (Vertical Scalability)
- Introduction of a PERFORMANCE_SCHEMA for statistics gathering
- Countless performance and scalability fixes, optimizations and features added
- Generally “better” than the release before, focusing on performance.
- This is currently the MySQL of choice for many people

Percona Server (XtraDB)

- Focus on instrumentation, configurability and vertical scalability
- Many very specialized features targeting specific problems you will have when scaling a MySQL instance massively. For example:
  - InnoDB Data Dictionary Size, which in vanilla MySQL is not configurable.
  - Percona Server has the ability to dump and restore the buffer pool, removing a very large pain point with cold starts when you have a large MySQL instance
- An endless list of performance fixes and little features that are really helpful when you are trying to run a vertically scaled MySQL instance or set of instances

07

New Features, Optimizer Improvements And Replication

Oracle MySQL 5.6-dev

- Large-scale improvements to the query optimizer and join algorithms
  - Batched Key Access (and Multi-Range Read)
  - Index Condition Pushdown
  - Subquery Optimizations
- NoSQL Interface (Memcache)
- Multi-Threaded Slaves
- Replication Checksums Built-In
- Global Transactions IDs and Server UUIDs

MariaDB

- Enhanced Testing
- Optimizer Enhancements
  - BKA, Hash Joins, Table Elimination..etc
- Engine Changes - Percona XtraDB, Aria
- Lots of smaller changes
  - Pluggable authentication, segmented key cache, microsecond resolution, HandlerSocket support, binary log group commit...and much more.
- Something rather unique: Virtual Columns and Dynamic Columns

MariaDB - Virtual Columns

```
MariaDB [test]> describe table1;
+-------+-------------+------+-----+---------+------------+
| Field | Type        | Null | Key | Default | Extra      |
+-------+-------------+------+-----+---------+------------+
| a     | int(11)     | NO   |     | NULL    |            |
| b     | varchar(32) | YES  |     | NULL    |            |
| c     | int(11)     | YES  |     | NULL    | VIRTUAL    |
| d     | varchar(5)  | YES  |     | NULL    | PERSISTENT |
+-------+-------------+------+-----+---------+------------+
4 rows in set (0.00 sec)
```

```
MariaDB [test]> show create table table1;

| table1 | CREATE TABLE `table1` (
  `a` int(11) NOT NULL,
  `b` varchar(32) DEFAULT NULL,
  `c` int(11) AS (a mod 10) VIRTUAL,
  `d` varchar(5) AS (left(b,5)) PERSISTENT
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

```
MariaDB [test]> select * from table1;
+-----+----------------+------+-------+
| a   | b              | c    | d     |
+-----+----------------+------+-------+
|   1 | some text      |    1 | some  |
|   2 | more text      |    2 | more  |
| 123 | even more text |    3 | even  |
+-----+----------------+------+-------+
3 rows in set (0.00 sec)
```

Examples From: http://kb.askmonty.org/en/virtual-columns/

MariaDB - Dynamic Columns

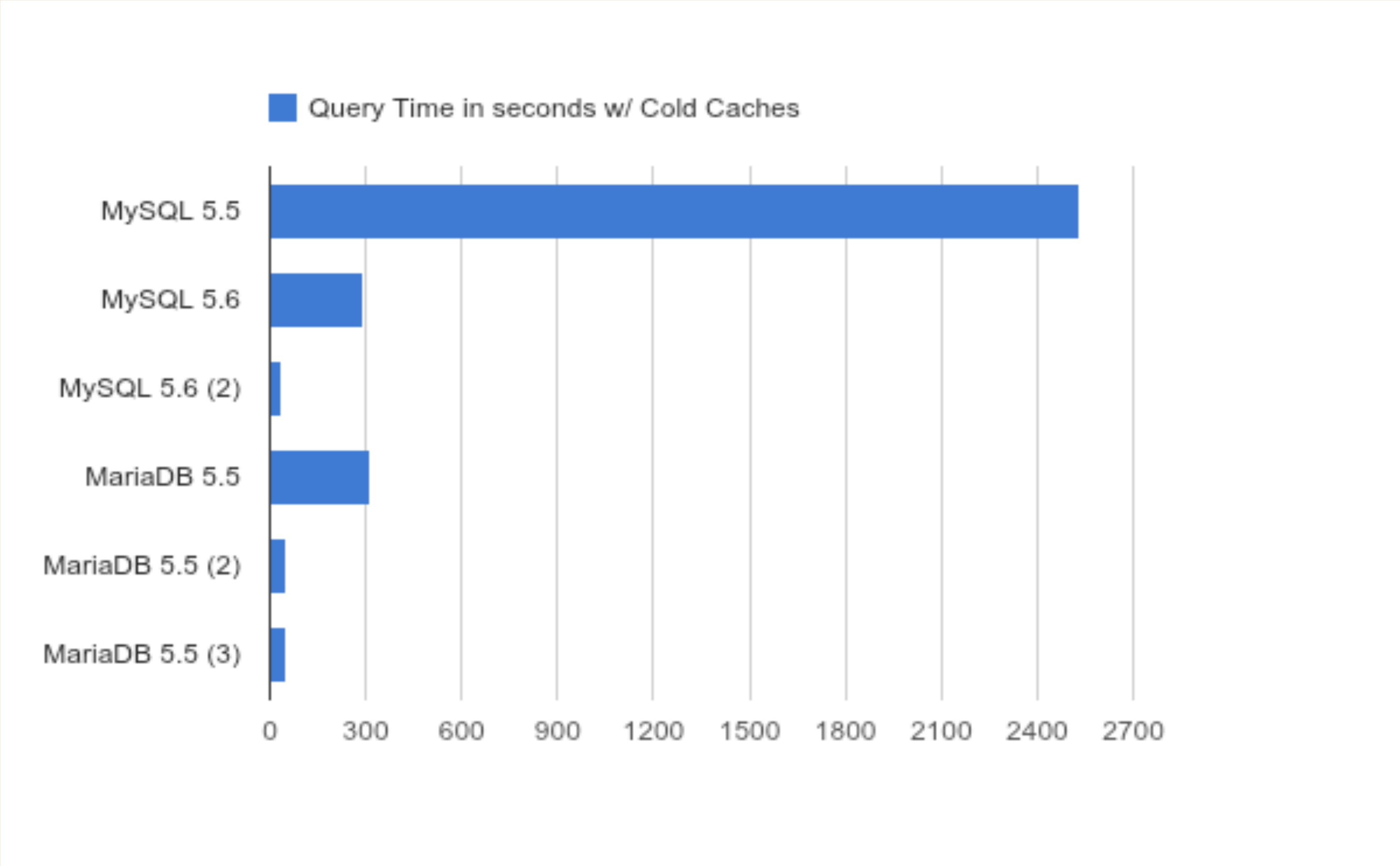
```
create table t1 (id int auto_increment primary key,
          name varchar(40),
          type enum ("shirt", "phone", "computer"),
          price decimal(10,2),
          dynstr mediumblob);
```

```
insert into t1 (name, type, price, dynstr) values
("Funny shirt", "shirt", 10.0, COLUMN_CREATE(1, "blue", 10, "XL")),
("nokia", "phone", 649, COLUMN_CREATE(1, "black", 2, "touchscreen")),
("htc Desire hd", "phone", 579, COLUMN_CREATE(1, "black", 3, "Android")),
("BM/Lenovo Thinkpad X60s", "computer", 419, COLUMN_CREATE(1, "black", 3,
"Linux"));
```

```
SELECT name FROM t1 WHERE COLUMN_GET(dynstr, 1 as char(10)) = "black";
+-------------------------+
| name                    |
+-------------------------+
| nokia                   |
| htc Desire hd           |
| BM/Lenovo Thinkpad X60s |
+-------------------------+
```

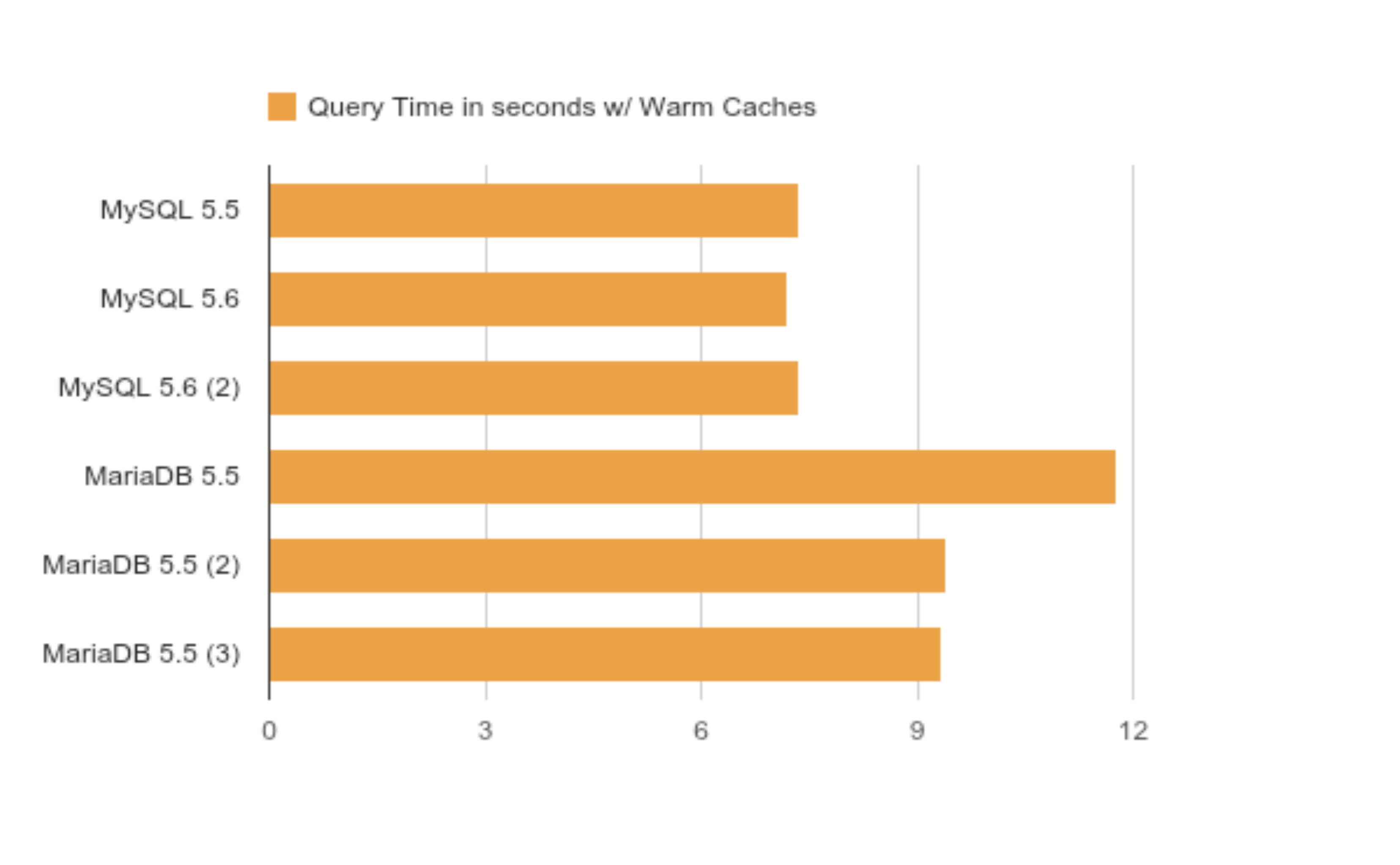
Examples From: http://kb.askmonty.org/en/dynamic-columns/

Optimizer Improvements



http://www.mysqlperformanceblog.com/2012/04/04/join-optimizations-in-mysql-5-6-and-mariadb-5-5/

Optimizer Improvements



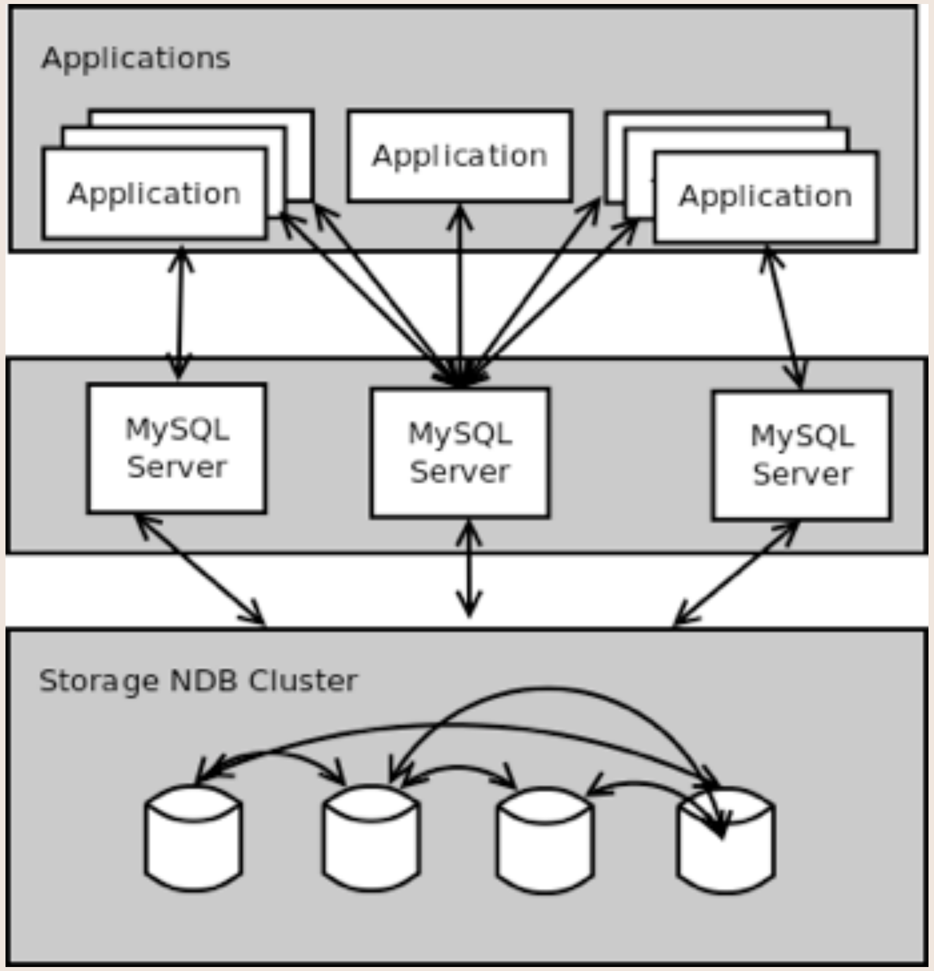
http://www.mysqlperformanceblog.com/2012/04/04/join-optimizations-in-mysql-5-6-and-mariadb-5-5/

Clustering

PerconaCluster/Galera

- “True” Master-Master Replication
- Transactional heuristics used to replicate a transaction and “acknowledge” it before it actually runs
- Can write to any node
- Dependent on network throughput and slowest node
- COMMIT time transaction checking
- Not a write-scaling solution

MySQL Cluster



- The MySQL Server instances are often called "SQL Nodes" and have no data themselves
- The Storage NDB Cluster or "Data Nodes" hold the actual data.
- Data is sharded across the NDB Cluster with a configurable number of replicas
- Massive amount of network traffic

Ref: http://anandafit.info/2011/03/29/introduction-to-mysql-clustering/

MySQL Cluster 7.2

- “Carrier Grade MySQL” is becoming more suitable for the generic case
- Lots of improvements
  - Multi-threading improvements to data nodes
  - Locking improvements on data nodes
  - Query planning improvements
  - Cross-DC replication improvements
- The big ticket items though are Adaptive Query Localization and Extended index Information
  - Pushes what it can to the data nodes themselves, reducing the amount of data needing to be pulled across the network to the SQL nodes.
  - Data Nodes now send more index information to the SQL Nodes, reducing the number of index hints required

And Then There Was Drizzle

- Drizzle 7, the first GA release, was pushed last year
- Drizzle 7.2 is on the horizon
- Not exactly a drop-in replacement
- Pluggable
- UTF-8
- IPv6
- Multi-Master Replication based on Google Protobuffers

MySQL Tools

- Percona Toolkit
  - Every MySQL DBA should have this installed. Allows you to checksum slaves, produce excellent slow log reports, kill problem queries automatically, sync tables and even online schema changes.
- Percona Playback
  - Allows you to replay the load represented in a slow query log or even a TCPdump of the MySQL protocol exchange.
- XtraBackup
  - Allows you to take a mostly non-blocking binary backup of InnoDB tables. An open source/free version of InnoDB's premium backup tool
- OpenArk Kit
  - A really massively random set of MySQL utilities. Automatically kill slow queries, clear master logs based on slave lag, repeat a query until a condition is true...etc.

Questions...

Session Evaluation

Please fill out the session evaluation available online:
http://munich2012.drupal.org/node/add/session-evaluation/1808