MariaDB - A MySQL replacement?

Boston Metup 2012

Max Mether
max@skysql.com
@maxmether

What is MariaDB?

SkySQL

- A branch of MySQL
  - MariaDB is a backward compatible, drop-in replacement for the MySQL Database Server
- Open Source
  - The source code for MariaDB is publicly available from Launchpad
  - All code in MariaDB is open source
    - No closed source modules
  - Open bugs database

A Brief History

- First version of what would become MySQL created by Monty Widenius 1983
- MySQL AB founded in 1995
- MySQL AB acquired by Sun Microsystems early 2008 for $1bn
- Monty Program founded in late 2008
- Oracle acquire Sun April 2009
- MariaDB 5.1 released in February 2010

Who's Behind MariaDB

- Monty Program
  - ~30 core MySQL developers
  - Located around the world
- Community

Community and MariaDB

- The goal of MariaDB is to provide a community developed, stable and free database
- MariaDB takes community contributions
- Many MariaDB Captains outside of Monty Program (64% MP, 36% outside)
- Open development model

MariaDB 5.1 – Feb 2010

MariaDB 5.1 = MySQL 5.1 + the following:

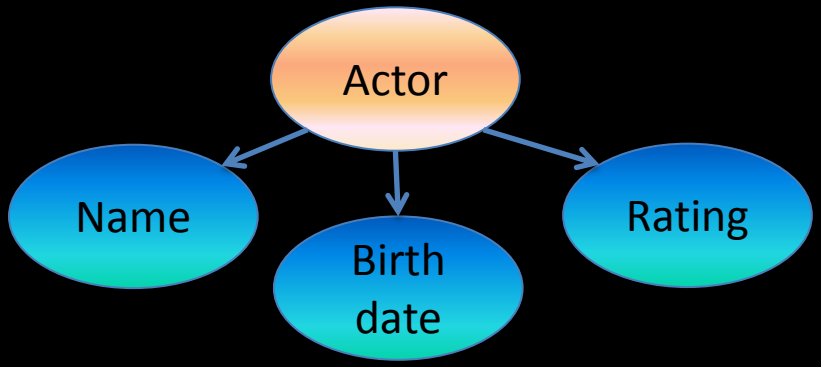
- Storage Engines
  - PBXT*
  - XtraDB
  - FederatedX
  - Aria
- Bugfixes
- Removal of Mutexes

MariaDB 5.1

```
MariaDB [(none)]> select id, time, time_ms, command, state
 -> from information_schema.processlist, (select sleep(2)) t;
+----+------+----------+---------+-----------+
| id | time | time_ms  | command | state     |
+----+------+----------+---------+-----------+
| 37 | 2    | 2000.493 | Query   | executing |
+----+------+----------+---------+-----------+
1 row in set (2.00 sec)
```

- Microsecond support
  - Slow query log, SHOW PROCESSLIST etc
- Thread pool*
  - Pool of threads instead of one thread /connection

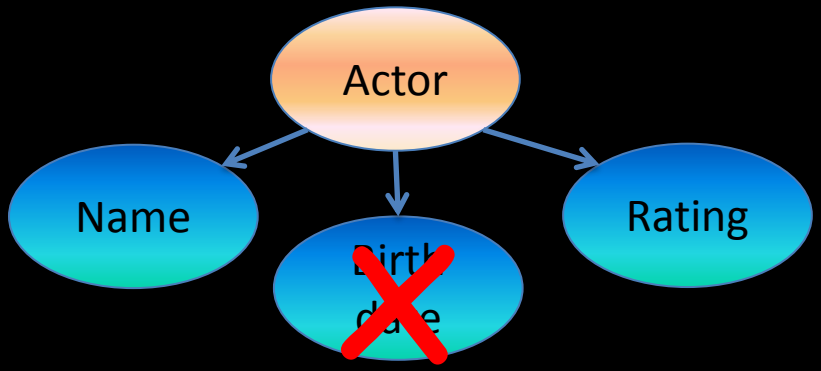
Table Elimination




```
create view actors as
select  ACNAM_Name, ACDOB_birthdate, ACRAT_rating
from ac_anchor
left join ac_name on ac_anchor.AC_ID=ac_name.AC_ID
left join ac_birthdate on ac_anchor.AC_ID=ac_birthdate.AC_ID
left join ac_rating on (ac_anchor.AC_ID=ac_rating.AC_ID and
                        ac_rating.ACRAT_fromdate =
                        (select max(sub.ACRAT_fromdate)
                      from ac_rating sub where sub.AC_ID = ac_rating.AC_ID))


select ACRAT_rating from actors where ACNAM_name='Gary Oldman'
```

Table Elimination




```
create view actors as
select  ACNAM_Name, ACDOB_birthdate, ACRAT_rating
from ac_anchor
left join ac_name on ac_anchor.AC_ID=ac_name.AC_ID
left join ac_birthdate on ac_anchor.AC_ID=ac_birthdate.AC_ID
left join ac_rating on (ac_anchor.AC_ID=ac_rating.AC_ID and
                        ac_rating.ACRAT_fromdate =
                        (select max(sub.ACRAT_fromdate)
                    from ac_rating sub where sub.AC_ID = ac_rating.AC_ID))


select ACRAT_rating from actors where ACNAM_name='Gary Oldman'
```

MariaDB 5.2 – Nov 2010

SkySQL

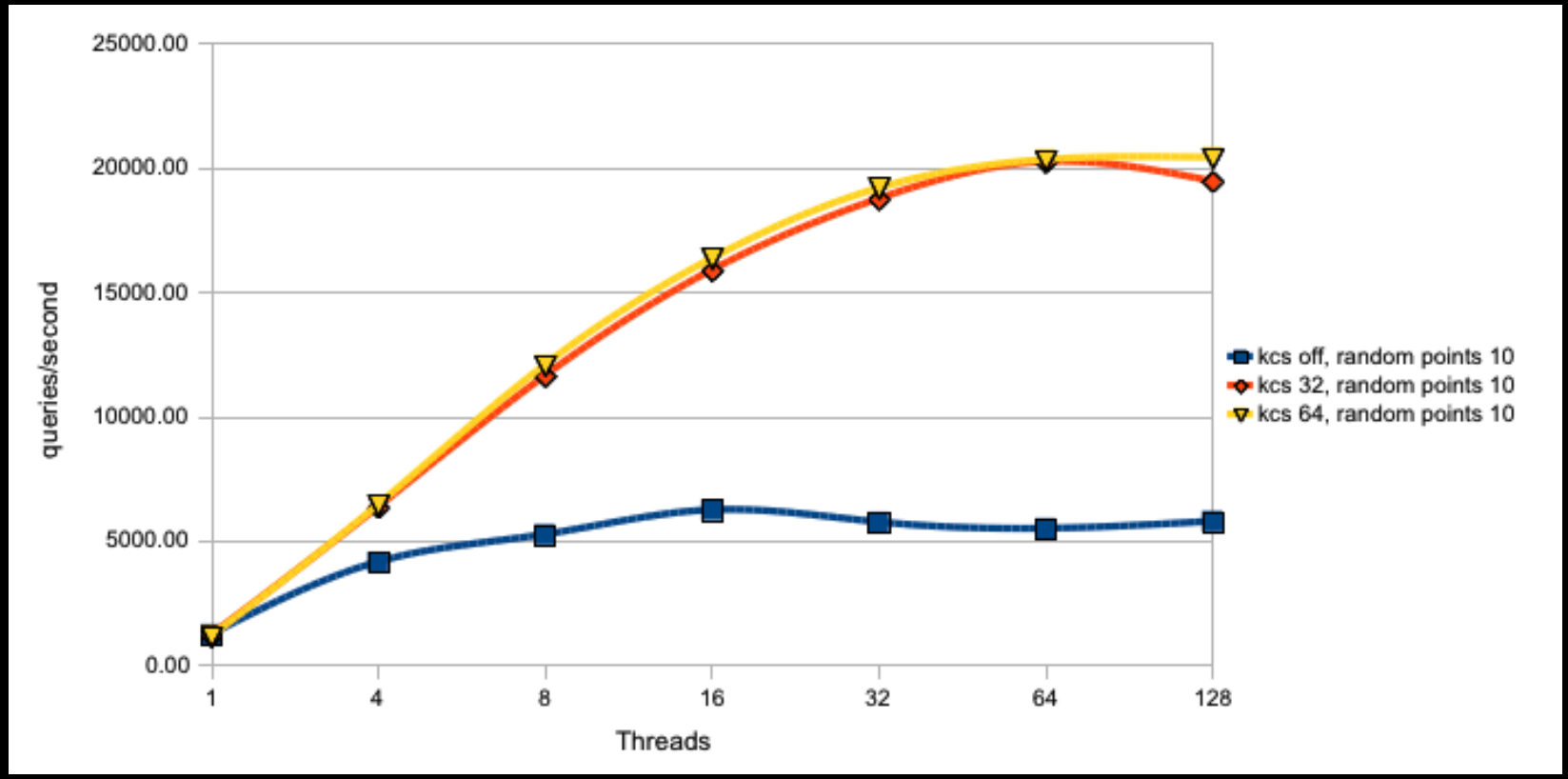
MariaDB 5.2 = MariaDB 5.1 + the following:

- Pluggable authentication
  - Authentication handled by plugins
  - PAM plugin included
- User statistics
  - CLIENT_STATISTICS
  - USER_STATISTICS
  - INDEX_STATISTICS
  - TABLE_STATISTICS

MariaDB 5.2

- Virtual Columns
  - **`PERSISTENT`** or **`VIRTUAL`**
- Sphinx Storage Engine
  - Allows access to Sphinx through MySQL
- Segmented MyISAM key cache
  - Key cache divided into different segments
  - Allows for better key cache concurrency
  - Between 1 and 64 segments

Segmented Key Cache



MariaDB 5.3 – Jan 2012

SkySQL

MariaDB 5.3 = MariaDB 5.2 + the following:

- Handler socket
  - Direct access to InnoDB/XtraDB storage layer
  - No SQL statements
    - Simple CRUD operations on tables
  - Can be match faster for large batch operations
- Dynamic columns
  - Allows you to create columns with dynamic content
  - Basically a blob with handling functions

Dynamic Columns


```
MariaDB [test]> create table t1 (id int auto_increment primary key,
    ->                  name varchar(40),
    ->                  type enum ("shirt", "phone", "computer"),
    ->                  price decimal(10,2),
    ->                  dynstr mediumblob);
Query OK, 0 rows affected (0.11 sec)

MariaDB [test]> 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"));
Query OK, 4 rows affected (0.04 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [test]> select id, name, type, price, length(dynstr) as len, column_list(dynstr) as
list from t1;
+----+-------------------------+----------+--------+------+------+
| id | name                    | type     | price  | len  | list |
+----+-------------------------+----------+--------+------+------+
|  1 | Funny shirt             | shirt    |  10.00 |   17 | 1,10 |
|  2 | nokia                   | phone    | 649.00 |   27 | 1,2  |
|  3 | htc Desire hd           | phone    | 579.00 |   23 | 1,3  |
|  4 | BM/Lenovo Thinkpad X60s | computer | 419.00 |   21 | 1,3  |
+----+-------------------------+----------+--------+------+------+
4 rows in set (0.03 sec)
```

Dynamic Columns


```
MariaDB [test]> select name from t1 where COLUMN_GET(dynstr, 1 as char(10)) = "black";
+-------------------------+
| name                    |
+-------------------------+
| nokia                   |
| htc Desire hd           |
| BM/Lenovo Thinkpad X60s |
+-------------------------+
3 rows in set (0.01 sec)

MariaDB [test]>
MariaDB [test]> select name, COLUMN_GET(dynstr, 1 as char(10)) from t1 where
COLUMN_EXISTS(dynstr, 1);
+-------------------------+-----------------------------------+
| name                    | COLUMN_GET(dynstr, 1 as char(10)) |
+-------------------------+-----------------------------------+
| Funny shirt             | blue                              |
| nokia                   | black                             |
| htc Desire hd           | black                             |
| BM/Lenovo Thinkpad X60s | black                             |
+-------------------------+-----------------------------------+
4 rows in set (0.00 sec)
```

MariaDB 5.3

MariaDB 5.3 = MariaDB 5.2 + the following:

- Replication enhancements
  - Original statement logged with RBR events
  - Checksum for binlog events
  - RBR fixed for tables with no PK
  - Consistent snapshot between storage engines
- User feedback plugin
- Extended OpenGIS SFS

Progress Report

- Progress report for ALTER TABLE

```
MariaDB [employees]> alter table salaries engine = maria;
Stage: 1 of 2 'copy to tmp table' 17.55% of stage done


MariaDB [employees]> select id, user, db, command, state,
-> time_ms, progress from information_schema.processlist;
+---------+--------------------+----------+----------+
| command | state              | time_ms  | progress |
+---------+--------------------+----------+----------+
| Query   | copy to tmp table | 23407.131 |   17.551 |
+---------+--------------------+----------+----------+
1 row in set (0.47 sec)
```

Optimizer Enhancements

1. Sub-query optimizations
2. Join additions
3. Optimizations for derived tables and views
4. Disk access optimization
5. Optimizer control

Optimizer Enhancements

Subqueries are finally usable in practice!

1. Sub-query optimizations
   - Semi-join subquery optimization
   - Materialization for non-correlated IN-queries
   - Sub-query cache

It is no longer necessary to rewrite subqueries manually into joins or into separate queries

Optimizer Enhancements

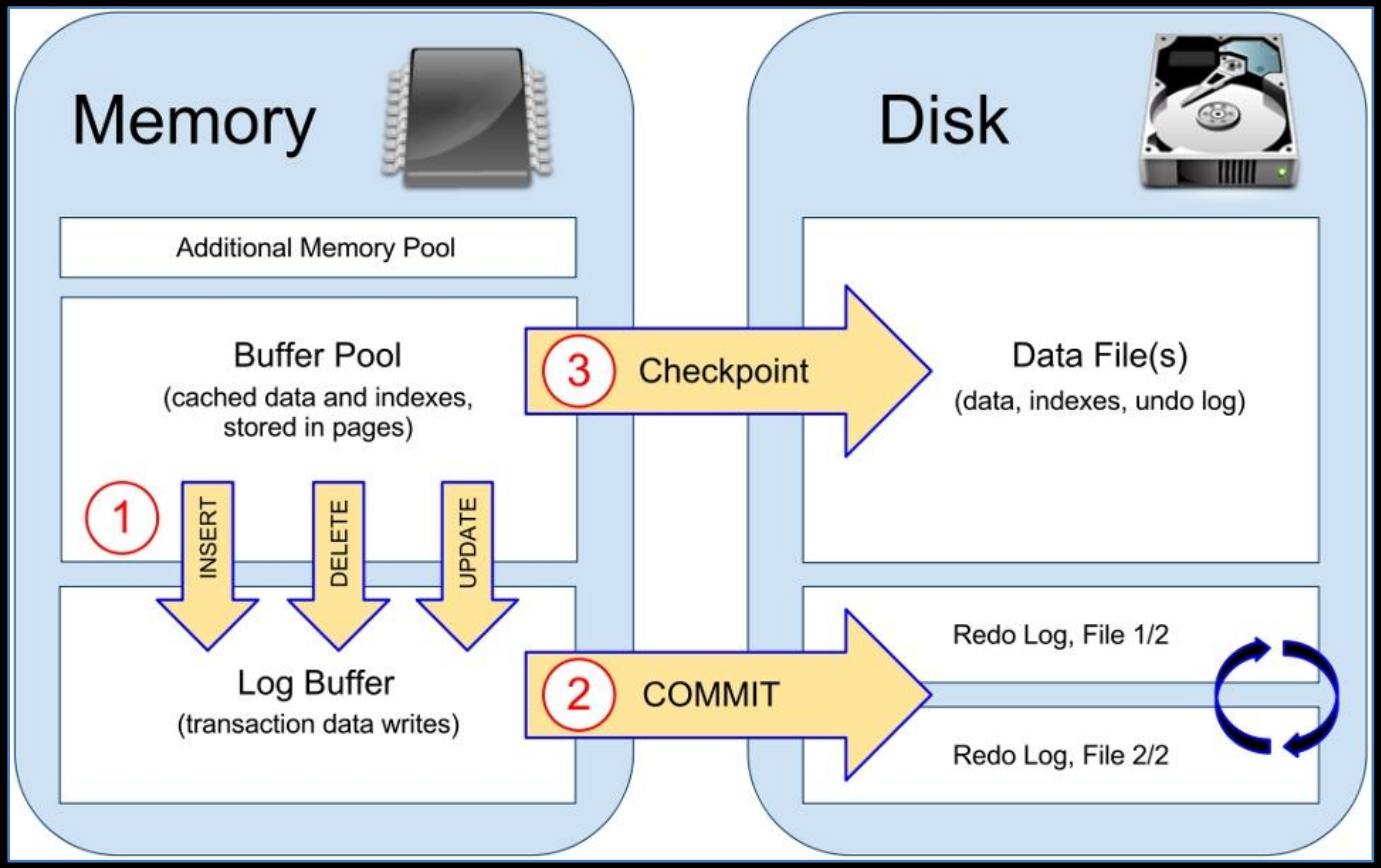
Before MariaDB 5.3 only Nested-Loop-Joins available

2. Join additions
   - Block Nested-Loop-Joins for outer joins
   - Block Hash-Joins
   - Batch-Key-Access

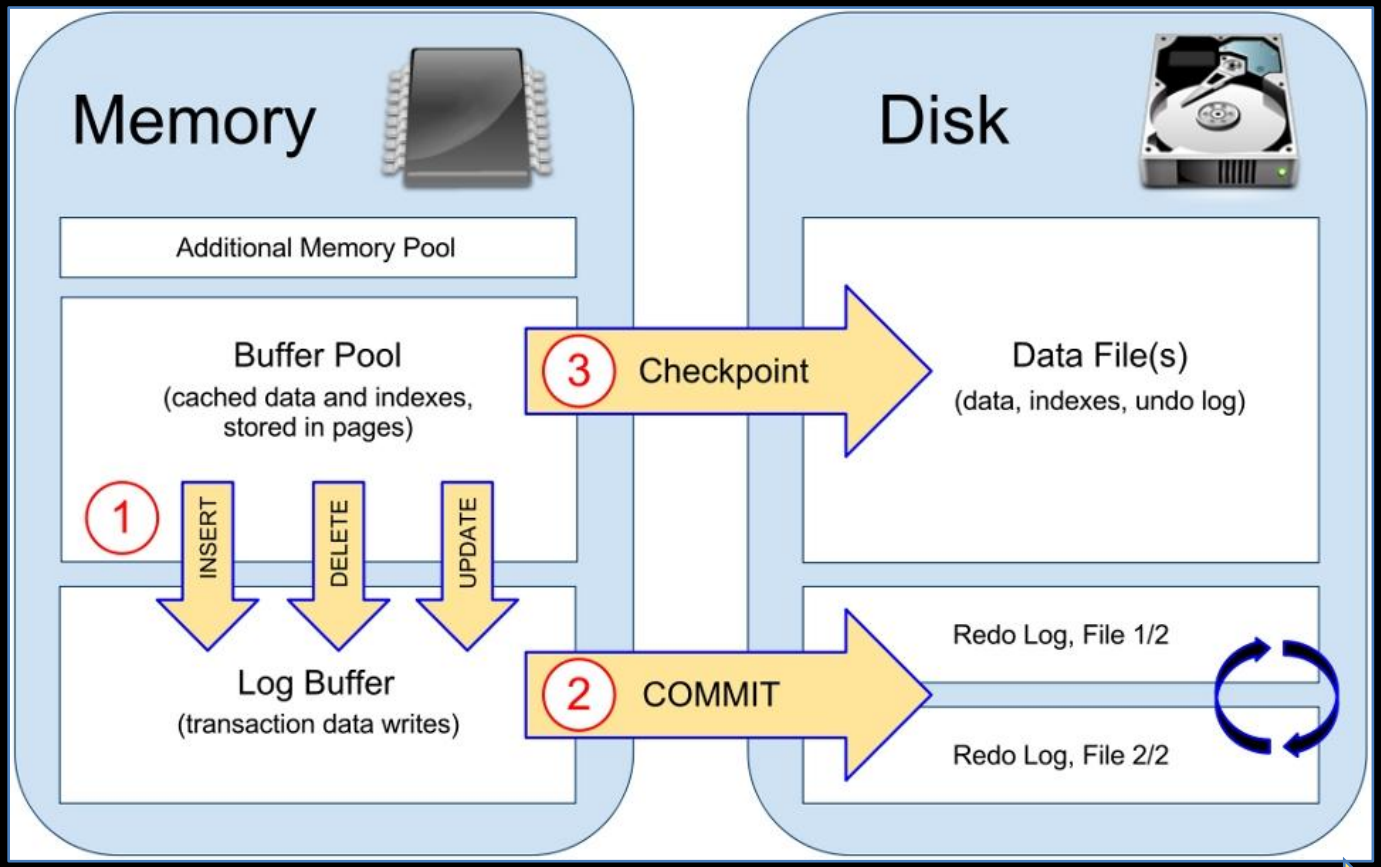
Optimizer Enhancements

3. Optimizations for derived tables and views
   - Mergeable derived tables processed like VIEWs
   - Optimizer can create indexes over materialized derived tables
4. Disk access optimization
   - Index Condition Pushdown
   - Multi-Range-Read optimization (MRR)
5. Optimizer control
   - @@optimizer_switch for all new options

Group Commit

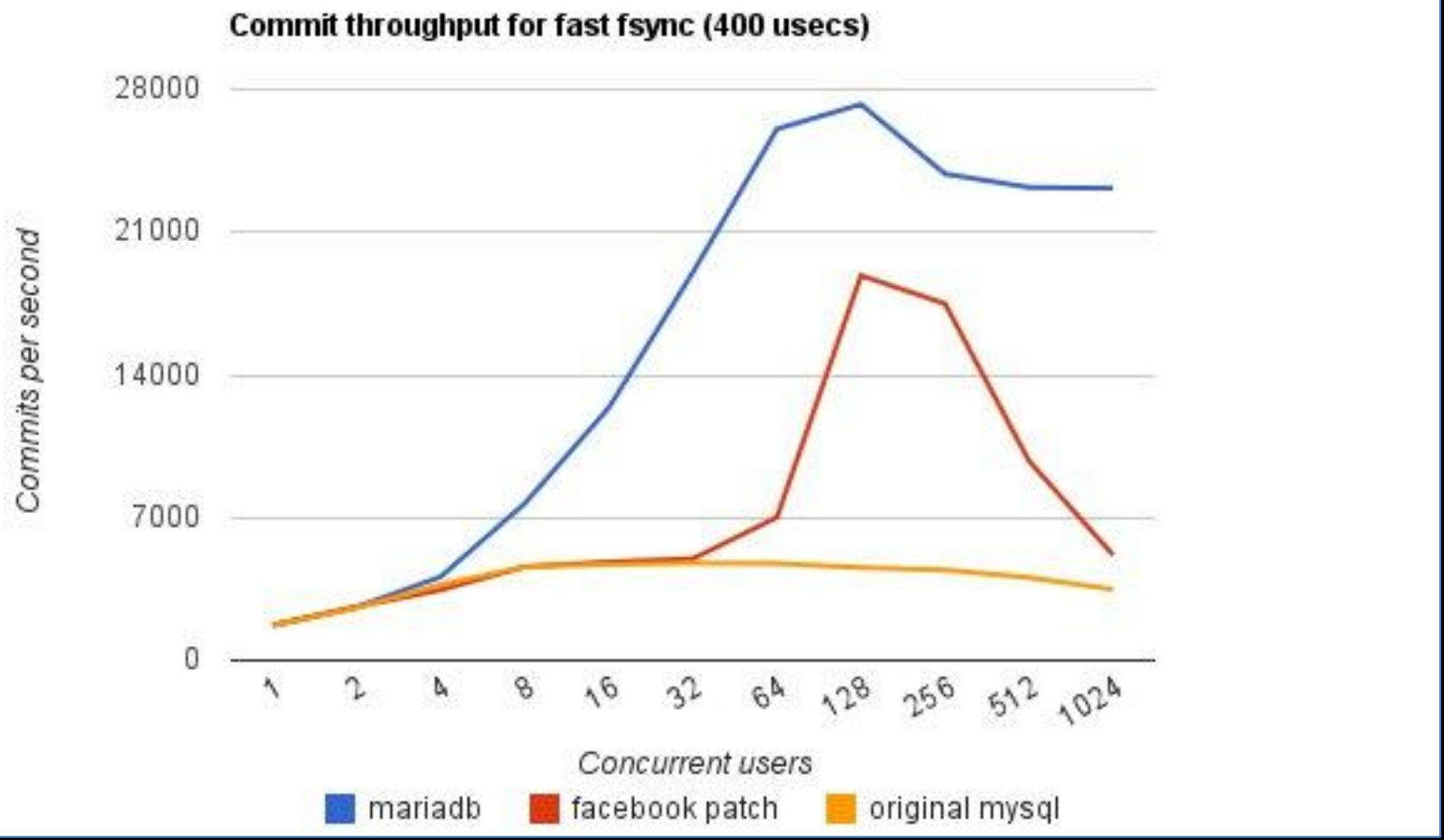


Group Commit

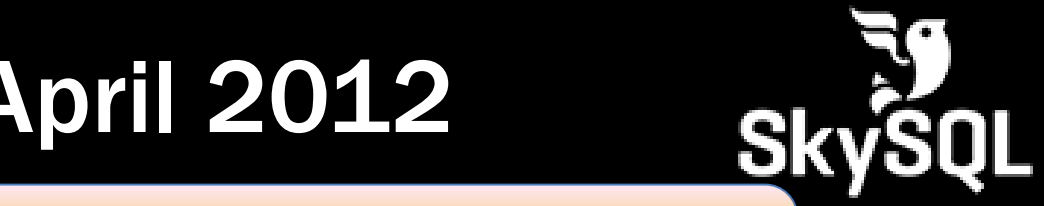


3 fsyncs / transaction

Group Commit



MariaDB 5.5 – April 2012



MariaDB 5.5 = **MySQL 5.5** + MariaDB 5.3 features
+ the following:

- New optimized thread pool implementation
  - Pool size dynamic
  - Different implementation on Linux and Windows
    - Fine grained tuning possible on Linux
- @@skip_replication
- LIMIT ROWS EXAMINED

```
MariaDB [employees]> SELECT * from t1, t2
-> LIMIT 10 ROWS EXAMINED 1000;
```

Next MariaDB

SkySQL

All highly speculative

- Global transaction Id (from MySQL 5.6)
- Multi-source replication
- New improved InnoDB (from MySQL 5.6)
- Cassandra storage engine

Vote for features on
http://www.skysql.com/content/new-server-functionality-have-your-say

A branch or a fork?

- MySQL 5.1 CE = MySQL EE <-> MariaDB 5.1
- MariaDB 5.2 and 5.3 <-> MySQL 5.1 CE & EE
- MySQL 5.5 CE != MySQL 5.5 EE
  - Commercial extensions
  - MariaDB 5.5 implements same features
    - Merged nonetheless
- 5.6 ?

Getting MariaDB

- http://mariadb.org is the main place
- Available via OpenSUSE build services
- Also available via
  - Gentoo
  - FreeBSD
  - Homebrew
  - Slackware
  - ArchLinux
- yum and apt repos available from MP

Support & Services

- Monty Program does NRE
- Enterprise level support available from SkySQL
  - Monty Program providing L3
- Training, consulting etc available from SkySQL
- Others providing MariaDB services:

  http://mariadb.org/service-providers/

More Information

- Downloads: http://mariadb.org/
- Mailing lists on launchpad
- *#maria* on freenode
- Knowledgbase: http://kb.askmonty.org/
- Support: http://www.skysql.com/products/skysql-enterprise
- Training & consulting: http://www.skysql.com/services/mysql/overview
- MariaDB book: *MariaDB Crash Course* by Ben Forta

THANK YOU!

Max Mether
max@skysql.com
@maxmether

SkySQL