The Complete MariaDB Server Tutorial

Colin Charles
colin@mariadb.org
http://bytebot.net/blog/ | @bytebot on twitter
http://mariadb.org/ | http://mariadb.com/

Percona Live Santa Clara, California, USA
13 April 2015

License

-

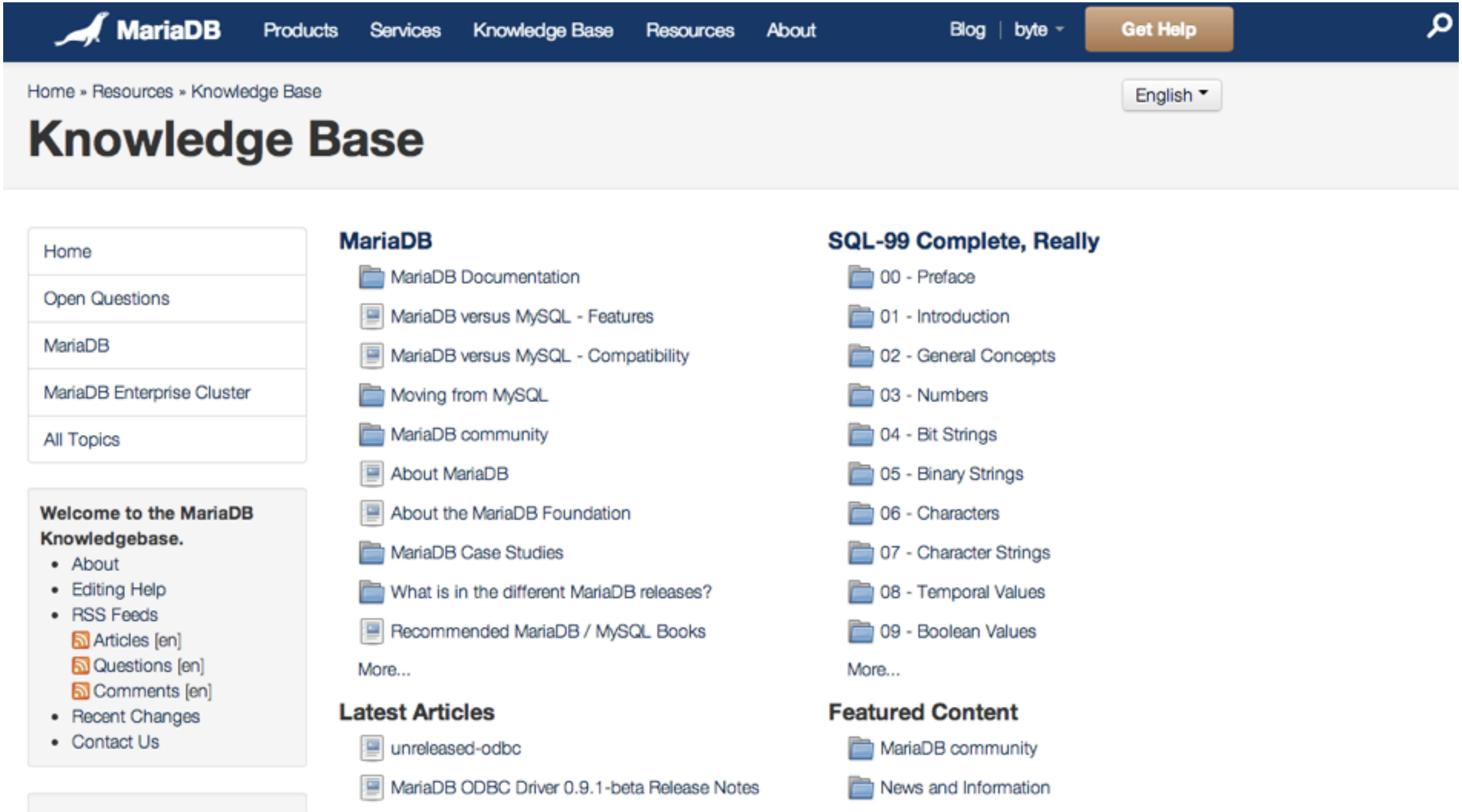
whoami

- Work on MariaDB at MariaDB Corporation (~~SkySQL Ab~~)
    - Merged with Monty Program Ab, makers of MariaDB
- Formerly MySQL AB (exit: Sun Microsystems)
- Past lives include Fedora Project (FESCO), OpenOffice.org
- Been a MySQL user since 2000

MariaDB is very social

- facebook: fb.com/MariaDB.dbms
- twitter: @MariaDB
- g+: plus.google.com/+mariadb/
- Tweet things you learn with #mariadb

Knowledgebase



Knowledge Base

Welcome to the MariaDB Knowledgebase.

- About
- Editing Help
- RSS Feeds
  - Articles [en]
  - Questions [en]
  - Comments [en]
- Recent Changes
- Contact Us

MariaDB

- MariaDB Documentation
- MariaDB versus MySQL - Features
- MariaDB versus MySQL - Compatibility
- Moving from MySQL
- MariaDB community
- About MariaDB
- About the MariaDB Foundation
- MariaDB Case Studies
- What is in the different MariaDB releases?
- Recommended MariaDB / MySQL Books

More...

SQL-99 Complete, Really

- 00 - Preface
- 01 - Introduction
- 02 - General Concepts
- 03 - Numbers
- 04 - Bit Strings
- 05 - Binary Strings
- 06 - Characters
- 07 - Character Strings
- 08 - Temporal Values
- 09 - Boolean Values

More...

Latest Articles

- unreleased-odbc
- MariaDB ODBC Driver 0.9.1-beta Release Notes

Featured Content

- MariaDB community
- News and Information

Who are you?

- Database Administrator (DBA)
- Developer
- Database Architect (DA)

Database Experience

- MySQL
  - 4.1 or earlier?
  - 5.0
  - 5.1
  - 5.5
  - 5.6

- MariaDB
  - 5.1
  - 5.2
  - 5.3
  - 5.5
  - 10.0

Understanding releases

- Alpha
- Beta
- Release Candidate
- Stable (GA)
- Labs
- Developer Milestone Releases (DMR)

What is MariaDB?

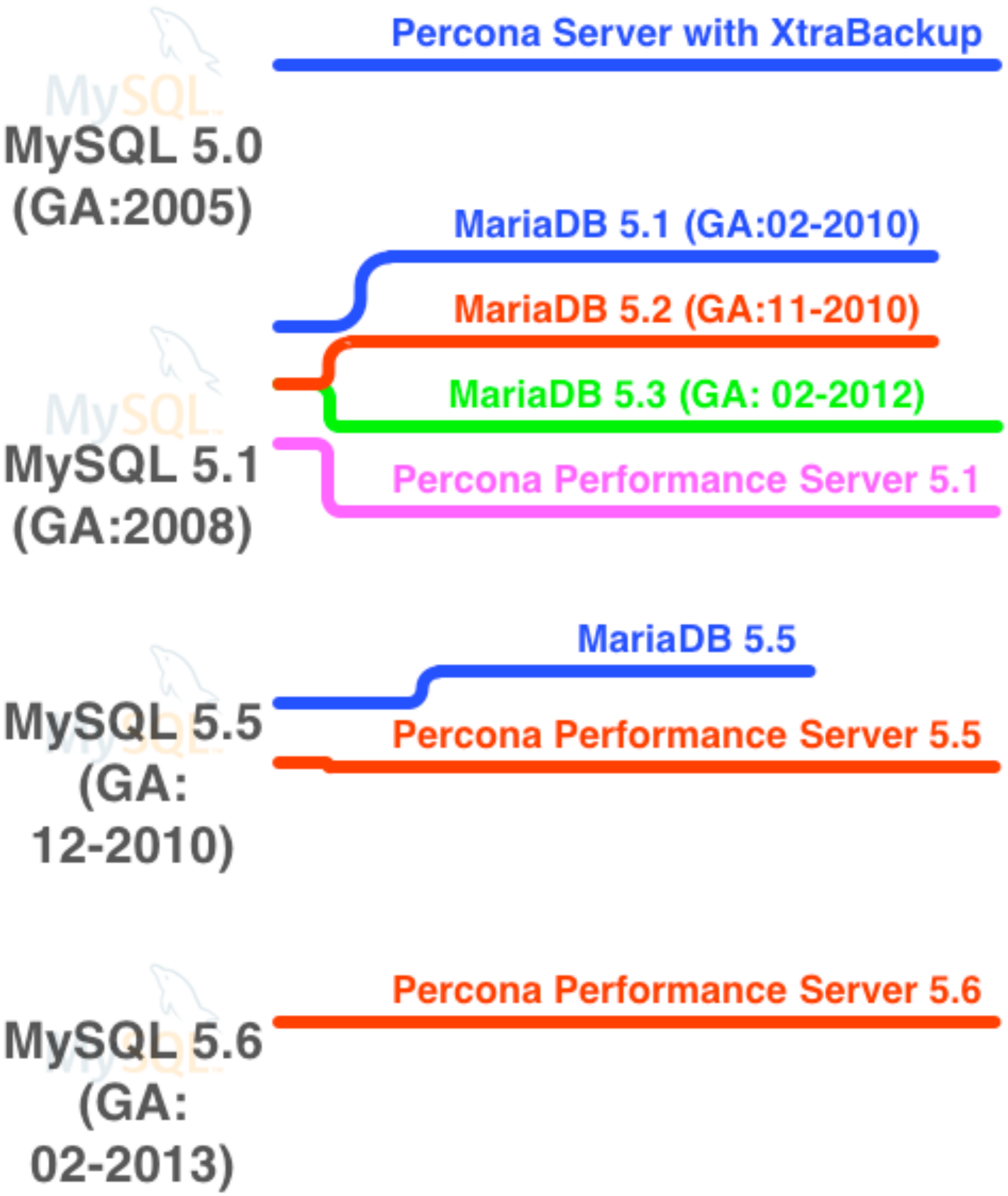
- Community developed
    - maria-captains: 42% Team MariaDB, 58% community including Sphinxsearch, Twitter, SkySQL,Taobao, Facebook, Percona, Codership & more
- Feature enhanced
    - MariaDB doesn’t depend on MySQL for development - many features are developed independently of MySQL
- Backwards compatible with MySQL
    - feature complete
    - replication supported for easy migration

Aims of MariaDB

- Compatible, drop-in replacement to MySQL
  - your application shouldn’t care that its running MariaDB, easy upgrade (uninstall mysql, install mariadb, continue ops!)
- Stable (bug-free) releases with no regressions
- GPLv2

5 years, many server releases

- MariaDB 5.1, GA February 2010
- MariaDB 5.2, GA November 2010
- MariaDB 5.3, GA February 2012
- MariaDB 5.5, GA April 2012
- MariaDB Galera Cluster, GA March 2013
- MariaDB 10.0.10 (March 2014)



MySQL 5.0
(GA:2005)

Percona Server with XtraBackup

MySQL 5.1
(GA:2008)

MariaDB 5.1 (GA:02-2010)

MariaDB 5.2 (GA:11-2010)

MariaDB 5.3 (GA: 02-2012)

Percona Performance Server 5.1

MySQL 5.5
(GA:
12-2010)

MariaDB 5.5

Percona Performance Server 5.5

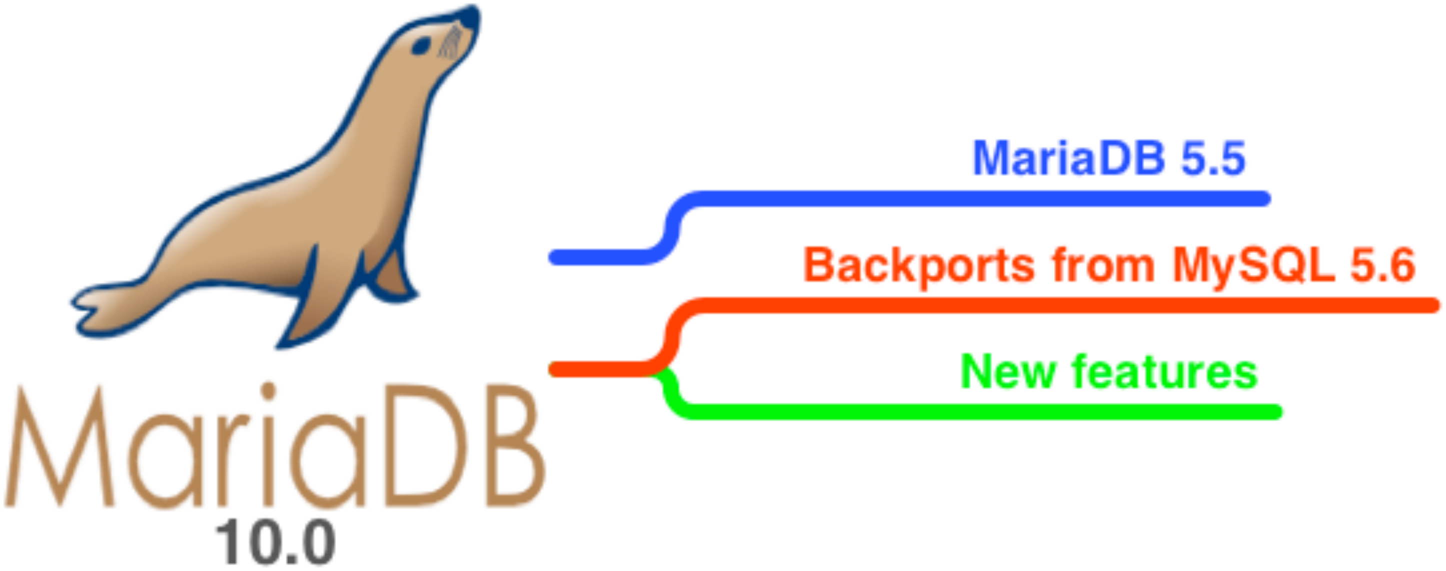
MySQL 5.6
(GA:
02-2013)

Percona Performance Server 5.6

MariaDB 5.6?

- There will never be a MariaDB 5.6 - the numbers have changed
- Many companies will still continue to support both MySQL and MariaDB releases

MariaDB 10.0

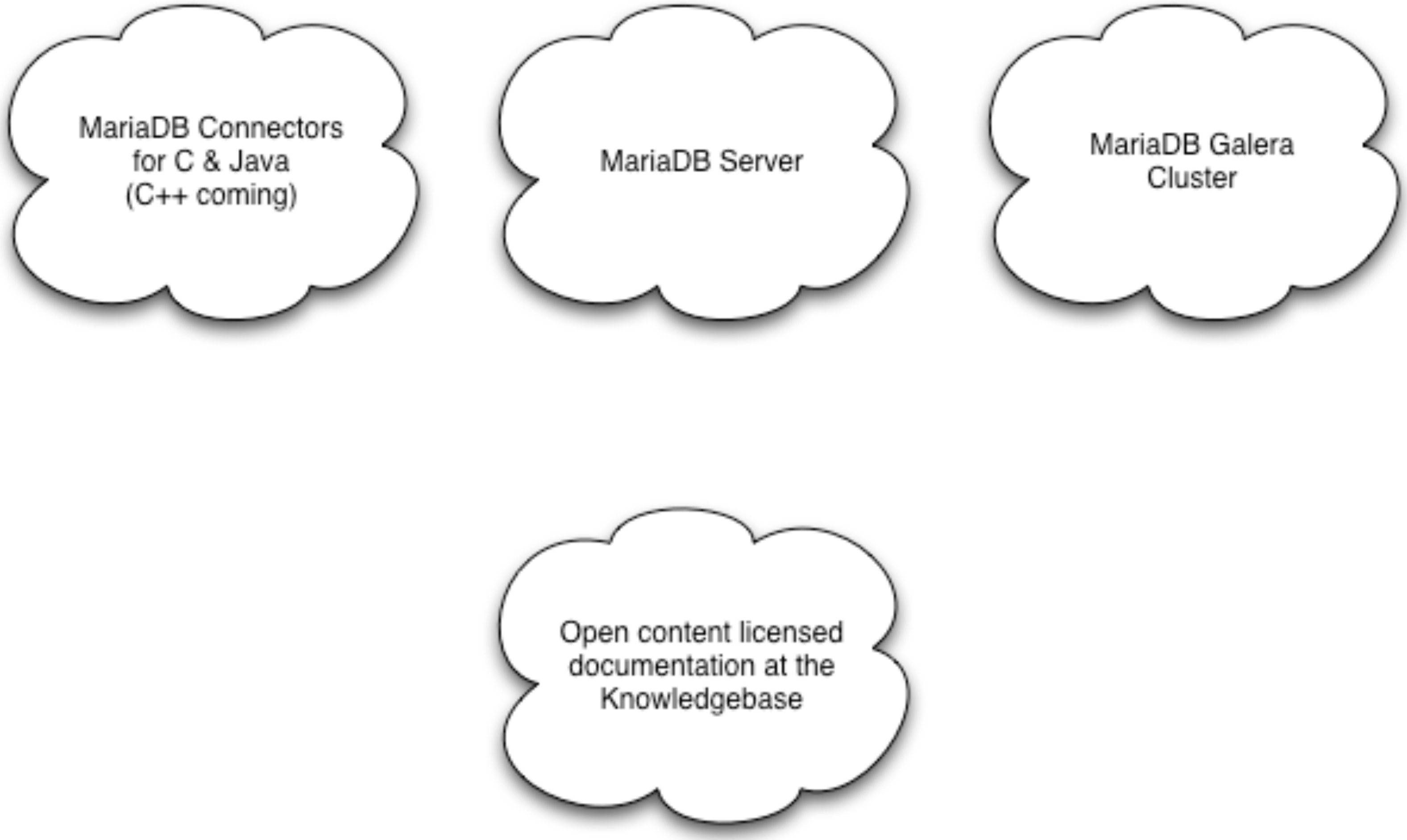


MariaDB 5.5

Backports from MySQL 5.6

New features

The MariaDB Ecosystem



MariaDB Connectors for C & Java (C++ coming)

MariaDB Server

MariaDB Galera Cluster

Open content licensed documentation at the Knowledgebase

Importance of understanding MariaDB (and MySQL)

- Generics are inefficient
- Since you have chosen MariaDB:
  - maximise its strengths
  - minimise its weaknesses

Sample databases to play with

- http://dev.mysql.com/doc/index-other.html
    - `sakila` sample database, `world` database (used in MySQL training), `menagerie` database (used in book: Beginning MySQL), `employees` database (large dataset, comes with data - best to play with)

Picking hardware

- Just use 64-bit hardware
- VM's are improving to use 64-bit OSes & software
- Physical > virtual
- Disk: battery backed storage, plan for RAID usage
- MariaDB 10.1 has optimisations for flash/SSD/FusionIO

Testing MySQL

- Use MySQL Sandbox
  - http://mysqlsandbox.net/
- Express one-click MySQL installs
- make_sandbox foo.tar.gz
- Does not require root privileges

Installation

- Binaries (tarballs) are available at http://mariadb.org/ (source too)
  - Built, tested by MariaDB
  - Graphical installer & configuration for Microsoft Windows, with HeidiSQL GUI
  - Up-to-date predictable release schedule
- RPM, DEB packages are provided, including YUM & APT repositories
  - Use the repository configuration tool
- Inside Linux/*BSD distributions
  - Easy to install, basic defaults, may be older than upstream

Upgrades

- Review changelogs carefully, even for minor versions
- Make backups (using `xtrabackup`)
- Don’t forget to run `mysql_upgrade`
  - by default with a distribution package
- Replication can reduce downtime by upgrading the slave, promoting it, then upgrading the master

Upgrading from MySQL 5.0

- MariaDB 5.1 fixes upgrades better than MySQL 5.1 handles this
- InnoDB + Archive tables upgraded properly
- mysql_upgrade, mysqlcheck have more options to see what's going on
- mysqlcheck wrong warnings removed
- MySQL 5.1 -> MariaDB 5.1 is "drop-in"
- https://mariadb.com/kb/en/upgrading-to-mariadb-from-mysql/

Packages (RPM example)

- MariaDB-common
  - character sets
- MariaDB-shared
  - latest libmysqlclient, some plugins, /etc/my.cnf.d/
- MariaDB-compat
  - compatible client libraries that replace mysql-libs
- MariaDB-server
  - The actual server!
- MariaDB-client
  - mysql client utilities
- MariaDB-test
  - test suite
- MariaDB-devel
  - development headers enabling you to build your own package
- MariaDB-CassandraSE
  - plugin for CassandraSE

What is libmysqlclient?

- Client libraries
- Many applications are compiled against libmysqlclient
- It is an Application Binary Interface (ABI)
- When you write in Java & use Connector/J, Connector/J is compiled against libmysqlclient
- Similarly with PHP & Connector/PHP

MySQL utilities

- Why does MariaDB use all the same client utilities?
  - Because the aim is to be a compatible drop-in replacement
- There are very few non-MySQL utilities shipped: `aria_chk`, `aria_dump_log`, `aria_ftdump`, `aria_pack`, `aria_read_log`
- `xtstat` (PBXT) is deprecated in MariaDB 5.5 onwards
- `mytop` as it adds features not-present upstream

Finding help

- How do you know what CLI options to use?
- Use the `man`(ual) pages!
- `man mysql`
- `man <command_name>` usually works for any CLI command
- `HELP SELECT` in mysql works too (help tables) - `HELP <operator>`

Error messages

- `perror` is a great tool
- MariaDB-specific errors are 1900 and above
- https://mariadb.com/kb/en/mariadb-error-codes/#mariadb-specific-error-codes

mysql_secure_installation

- Now that you’ve tested MySQL works fine (by default, no root password), it’s time to ensure you secure the installation
- Set root password
- Remove anonymous users
- Disallow remote root login
- Remove test database
- Reload privileges

mysqladmin

- Manage user accounts, passwords, permissions
- Display mysqld settings & status counters
- Shutdown, create, drop databases

SHOW PROCESSLIST

```
shell> mysqladmin processlist
```

SHOW STATUS

```
shell> mysqladmin extended-status

shell> mysqladmin -uroot -p create wordpress
```

mysqldump

- Dump schema and/or data to .sql file, tab
- Useful for backups, transferring data
- Locks for consistency, so troublesome

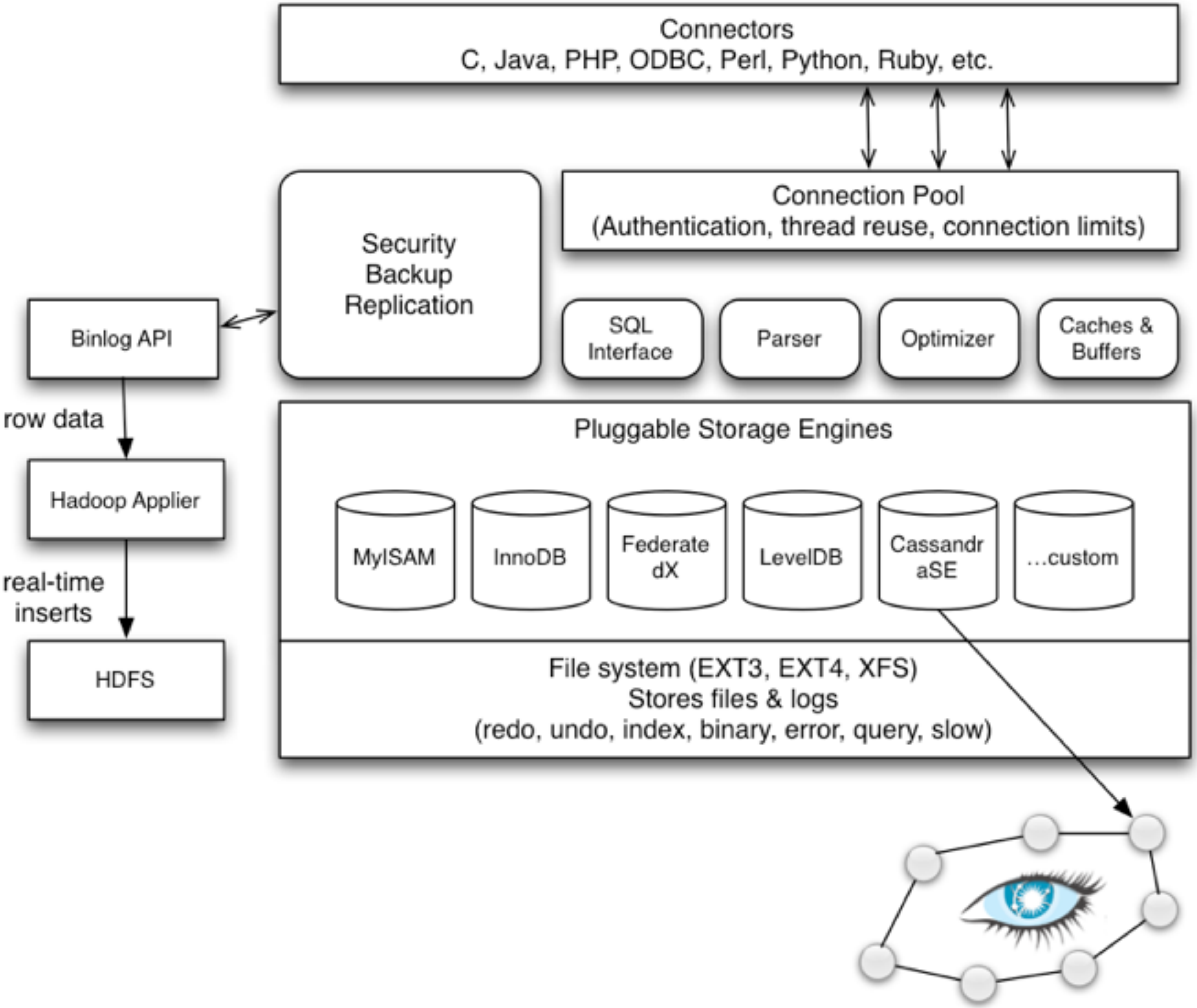
```
shell> mysqldump --all-databases > backup.sql

shell> mysqldump --all-databases --single-transaction >
backup.sql
```

Other tools

- There are many other command line tools...
- Front-ends
  - HeidiSQL
  - Webyog’s SQLyog
  - Sequel Pro (OSX)

MariaDB Architecture



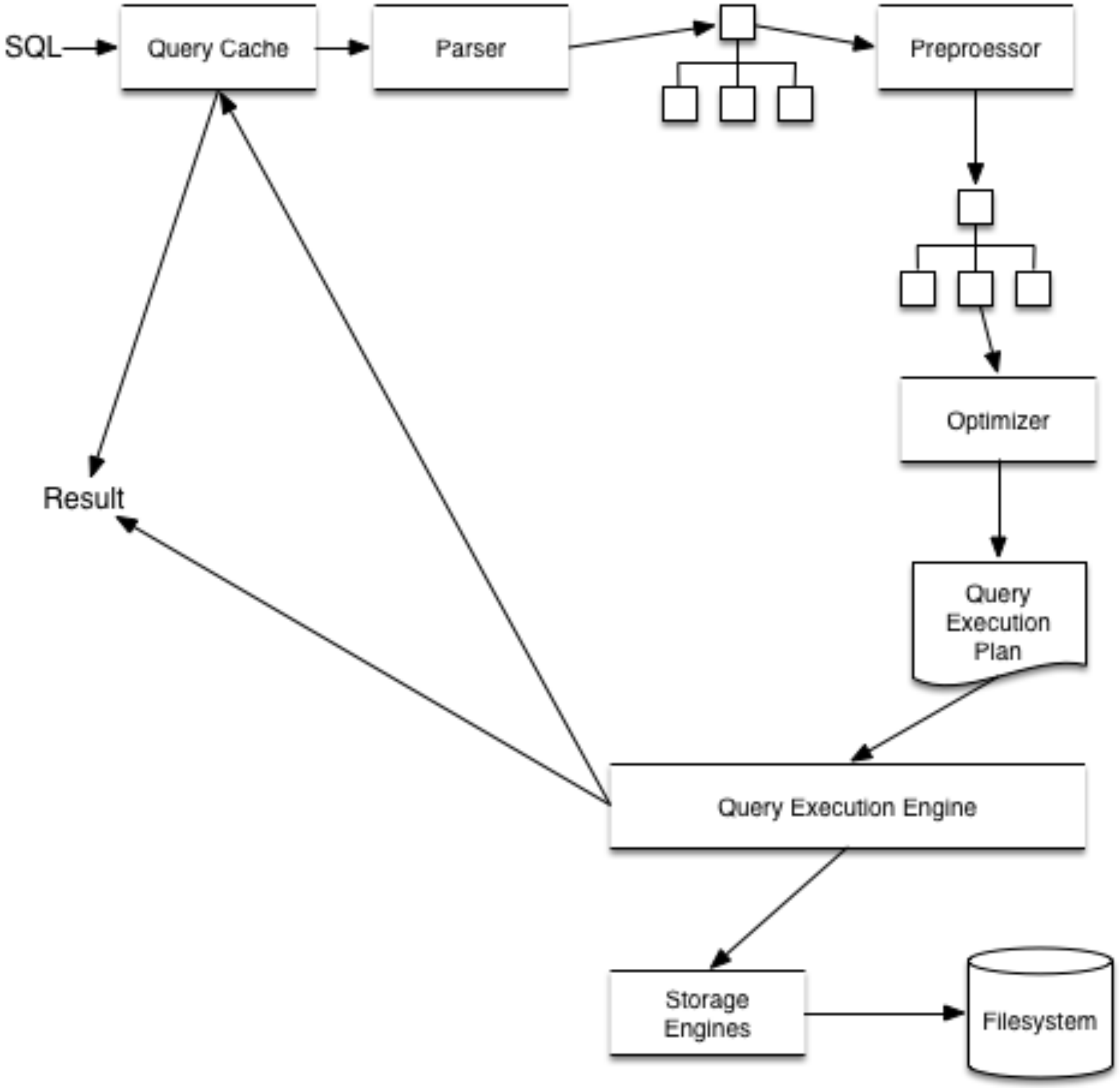
Making client connections

- TCP/IP is available on all platforms
  - disable using `--skip-networking`
- Socket files available on Unix
  - fastest communication path
- MySQL connections are generally cheap
  - If you have many short running queries (typical web app), you benefit from MariaDB threadpool
- Set connection limit `max_connections=n` in my.cnf

Connection Pool

- mysqld’s main thread listens for connections
- Each connection has a thread assigned to it
- Threads can be:
  - created newly
  - reused from thread cache
- User authentication processed based on hostname, username & password
- Client specific buffers for session variables are allocated

How a query is processed



Query Cache

- Stores SELECT queries and their result sets
- Subquery cache exists only in MariaDB
- Frequently changed table data will cause queries to naturally be missed
- Regularly replaced these days with memcached, et al.

SQL Parser

- Lexical scanner & grammar rules
- Parse SQL into tokens
- Apply grammar rules to check statement validity
- Construct a parse tree for Optimizer to use

Optimizer

- Reads the parse tree and calculates the best query execution plan (QEP) to handle query
  - find indexes
  - determine JOIN order
  - eliminate unnecessary tables
  - etc.
- The optimizer is smart, don’t try to force query plans

Pluggable Storage Engines

- MariaDB can use many storage engines with different features
- Installed/removed on fly with `INSTALL/ UNINSTALL PLUGIN`
- Mixing & matching on same server, even on same query

Value proposition

- Unmatched flexibility + customisation potential
- MEMORY engine for performance/routine lookup data
- Right storage engine can improve performance in many applications
- ARCHIVE compresses data, up to 80%
- Partners & community benefit from this

What makes engines different?

- Storage: how the data is stored on disk
  - Or in NDB (memory+disk), CassandraSE (access a Cassandra Cluster), SphinxSE (access the Sphinx daemon)
- Indexes: improves search operations
- Memory usage: improves data access for speed
- Transactions: protects the integrity of your data (Atomic-Consistent-Isolated-Durable - ACID)
- Locking level: MyISAM (table locks), InnoDB (row locks), old BDB (page locks)
- Data types: Data types may be converted, MEMORY doesn’t support TEXT, etc.
- Caching: InnoDB caches data & indexes, MyISAM caches indexes only (relying on OS disk cache for data)
- Full-text search capability: MyISAM has this, InnoDB 5.6 got this
- GIS: MyISAM & Aria work (R-tree indexes exist), InnoDB 5.7 has this too

INFORMATION_SCHEMA

- Holds metadata (data about the data) on all other databases & tables, exposed as regular tables
- Generated on the fly
- Has extensions in MariaDB as we expose more data than native MySQL

PERFORMANCE_SCHEMA

- Allows for monitoring execution at a low level
- This is a storage engine, monitoring server events (anything that takes time & can be instrumented)
- Tables are views or temporary tables that use no on-disk storage

Transactional vs. non-transactional

- Transaction-safe tables (InnoDB) have advantages over non-transaction safe tables (MyISAM):
  - server crash? Automatic recovery, or a backup +transaction log
  - ROLLBACK can be executed to ignore changes
  - Update fails? Changes reverted
  - Concurrency - tables w/ many update + concurrent reads
- Disadvantages in today's environments (transaction overhead = slower), more disk space requirements, more memory to perform updates don't seem like they apply any longer

Indexes

- Tree Indexes
  - B-Trees
  - B+Trees (InnoDB)
  - T-Trees (NDB)
  - Red-black binary trees (MEMORY)
  - R-Trees (MyISAM for spatial indexes)

- Hash Indexes (MEMORY, NDB, InnoDB)
  - If table fits entirely in memory, fastest way to perform queries is a hash index
  - InnoDB has an internal adaptive hash index. InnoDB monitors index searches, and if it notices that it will benefit from a hash index, InnoDB automatically builds one. (5.1.24 and greater)

MyISAM

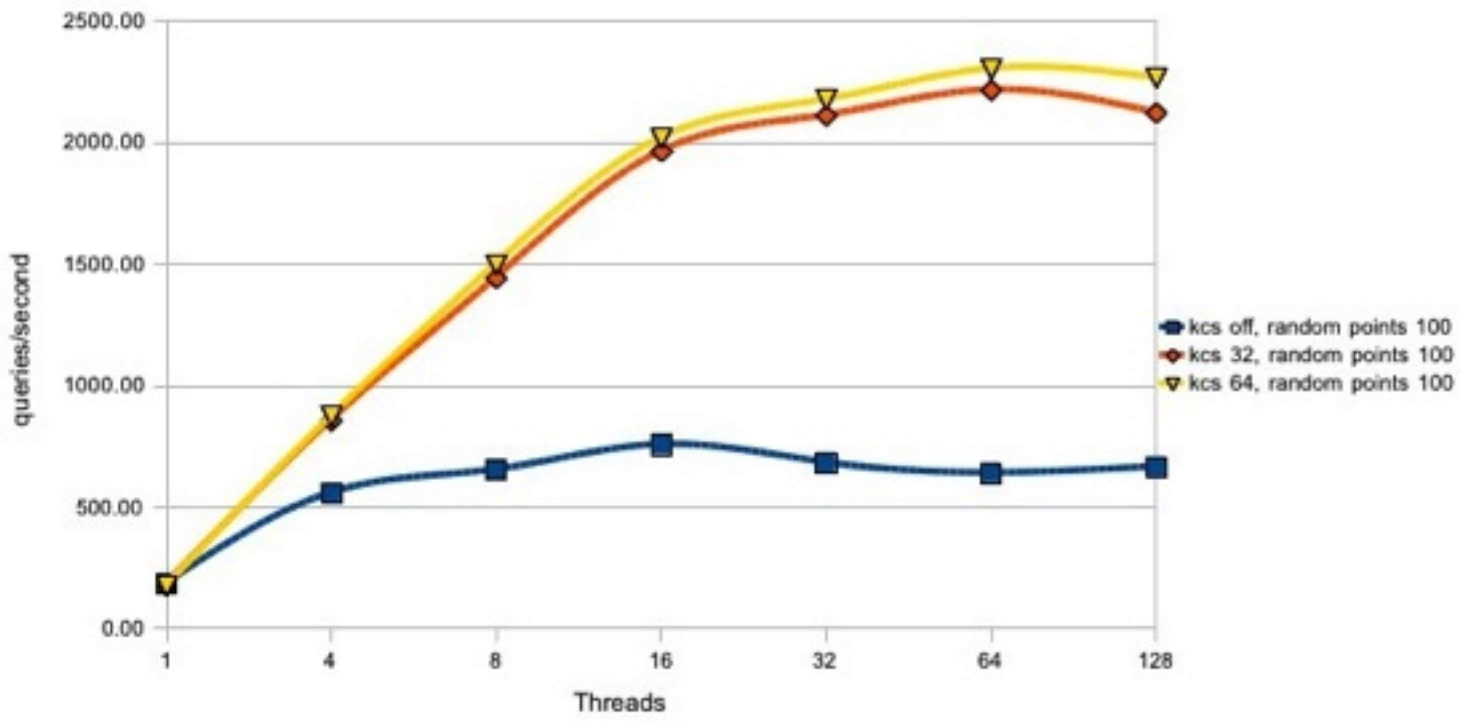
- Pros?
  - excellent INSERT performance
  - small footprint
  - supports full-text search (FTS)
- Cons?
  - no transactions
  - no foreign key support
- Typical uses
  - logging
  - auditing
  - data warehousing

MyISAM II

- In my.cnf, remember to set the key_buffer_size. This is memory*0.40, as MyISAM uses the OS cache for tables
- myisam_use_mmap enables MyISAM to use memory mapping (7-40% speed improvement)
- key_cache_segments = 1 enables segmented key caches in MariaDB - ~250% improvements, as it mitigates thread contention for key cache lock

MyISAM segmented key caches

- Mitigates thread contention for key cache lock, with notable performance improvements
- Key caches divided into different segments, allowing for better key cache concurrency
- 1-64 segments



InnoDB

- Maintains its own buffer pool (does aggressive memory caching)
- Uses tablespaces (several files on disk, raw disk support)
- Typically used for OLTP operations

ARCHIVE

- Store large amounts of data without indexes, in small disk footprint
- SELECT and INSERT operations only
- Good for data audit use
- Uses AZIO (zlib) compression

FederatedX

- Create logical pointers to tables that exist on other MySQL servers; these can then be linked together to form one logical database
- A federated table pointing to an InnoDB table on another server, will have transaction support (in 5.1)
  - Capabilities limited to underlying engine on remote server
- CREATE TABLE t1 (...) ENGINE=FEDERATED CONNECTION='mysql://username:pwd@myhost:3306/db_name/tbl_name
- Can also be used for synchronous replication
  - Federated table on master server pointing to slave; triggers on master table to write all changes to remote table once applied to the master

Memory

- Previously known as HEAP tables
- In-memory engine
- Hash index used by default (changes in 5.2, enable much better INSERT performance), B-Tree available too
- https://mariadb.com/kb/en/performance-of-memory-tables/

Aria

- Based off the 5.1 code
    - 1.0 – crash-safe MyISAM, with cacheable row format
    - 1.5 – concurrent INSERT/SELECT
    - Soon to be merged into 6.0, then...
        - 2.0: transactional + ACID compliance
        - 3.0: high concurrency, online backup
- Goal: ACID compliant, MVCC transactional storage engine, based on MyISAM
- Target? Data warehousing
- Uses big log files (1GB by default)
- 8K pages used by default (MyISAM uses 1K pages)
- Has group commit (MariaDB 5.2) to speed up inserts

PBXT (deprecated)

- MVCC, transactional, ACID compliant, foreign key support
- row-level locking for updates, so maximum concurrency
- immediate notification if client processes are deadlocked
- write-once, as it uses a log-based architecture (write data to DB without first writing to transaction log)
- support for BLOB streaming with Blob Streaming engine

Storage Engine API

- http://dev.mysql.com/tech-resources/articles/creating-new-storage-engine.html
- SHOW PLUGINS;
- https://kb.askmonty.org/v/extending-create-table
- storage/example/ha_example.cc and storage/example/ha_example.h

Writing your own

- Find the plugin path - show variables like “%plugin%”;

```
+-----------------+-----------------------------------------------+
| Variable_name   | Value                                         |
+-----------------+-----------------------------------------------+
| plugin_dir      | /usr/local/Cellar/mariadb/10.0.15/lib/plugin/ |
| plugin_maturity | unknown                                       |
+-----------------+-----------------------------------------------+
```

  - note that this is also where you store UDFs
- Copy the relevant engine (eg. myengine.so)
- INSTALL PLUGIN myengine SONAME 'myengine.so';
- Server registers plugin to mysql.plugin table, and now ENGINE=myengine will work

Things to think about

- Backup is not engine-independent
    - MyISAM, InnoDB, TokuDB
    - LVM/ZFS snapshots mitigate this
- Different engines have different monitoring options
- Mix and match; use summary tables

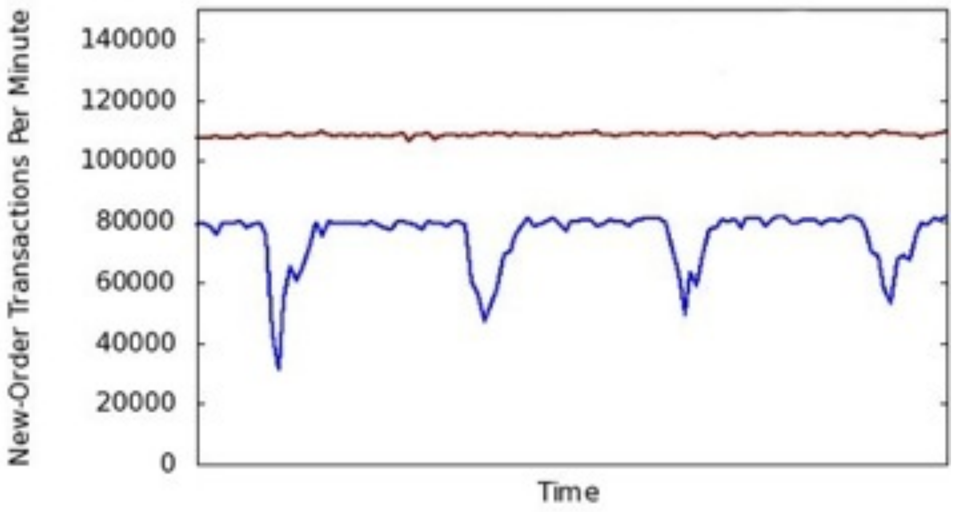
Survey of popular OSS tools - what they use

- Wordpress (blog): uses default engine, MyISAM is fine
- MediaWiki (wiki): prefers InnoDB, except for "searchindex" table, which is MyISAM
    - http://svn.wikimedia.org/viewvc/mediawiki/trunk/phase3/maintenance/tables.sql?view=markup
- vBulletin (forum): MyISAM
- SugarCRM (CRM): MyISAM (with conversion script to InnoDB provided)
- Zimbra Collaboration Suite: InnoDB

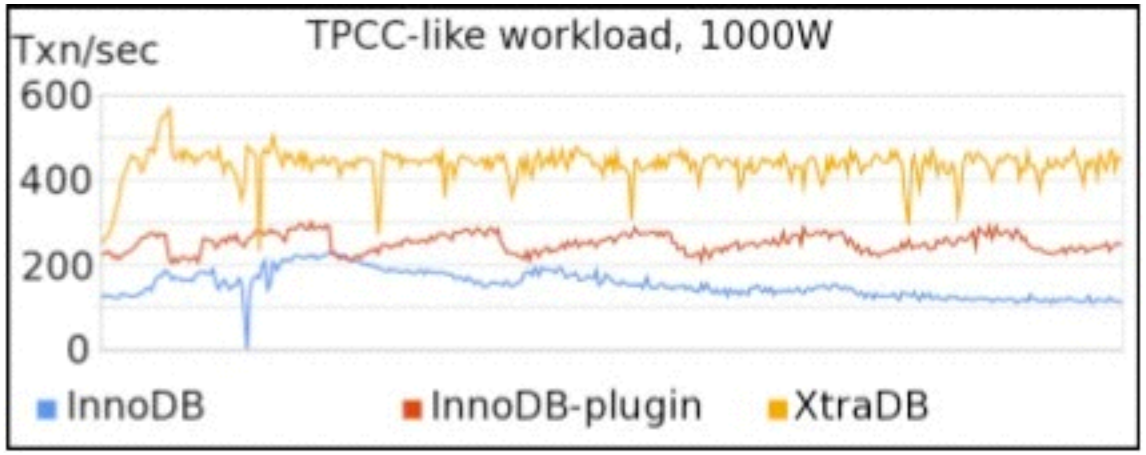
Now, let’s focus on MariaDB features

We start with...

- What came in 5.1, 5.2, 5.3, 5.5 (jump around appropriately)
- What comes in 10.0 series
- I won’t talk about deprecated features like PBXT in-depth, or even the old MariaDB 5.1 pool of threads (5.5 threadpool is better + improvements in 10.1)

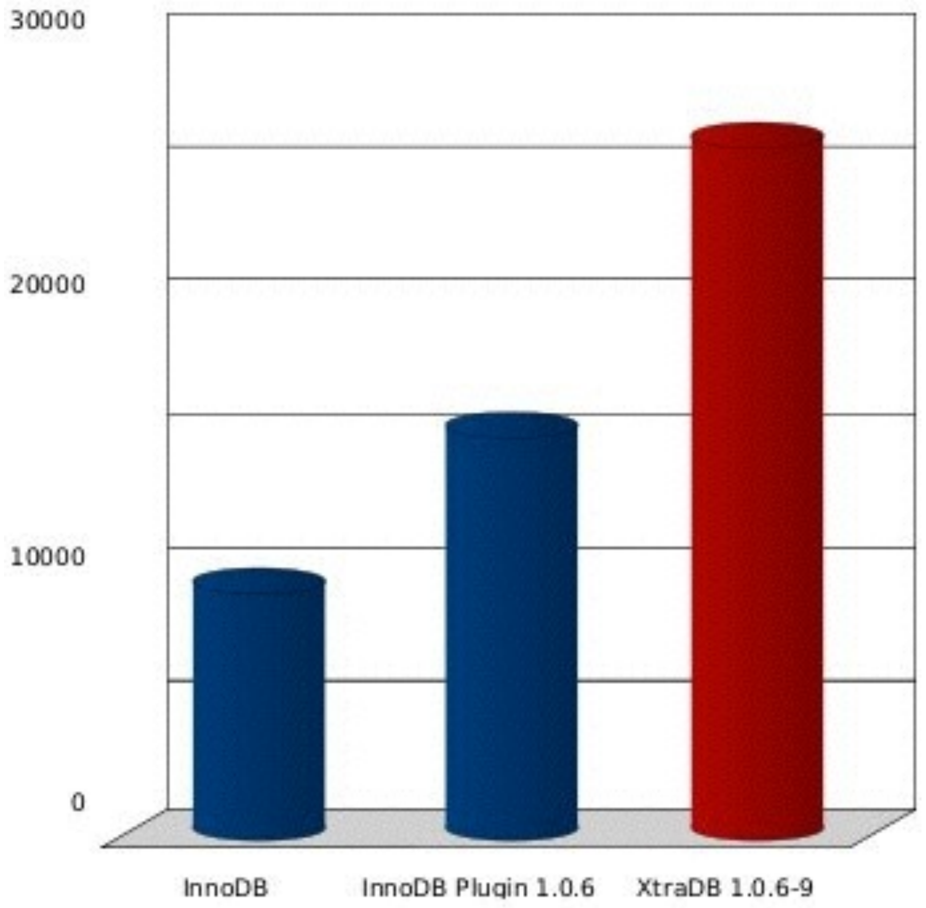


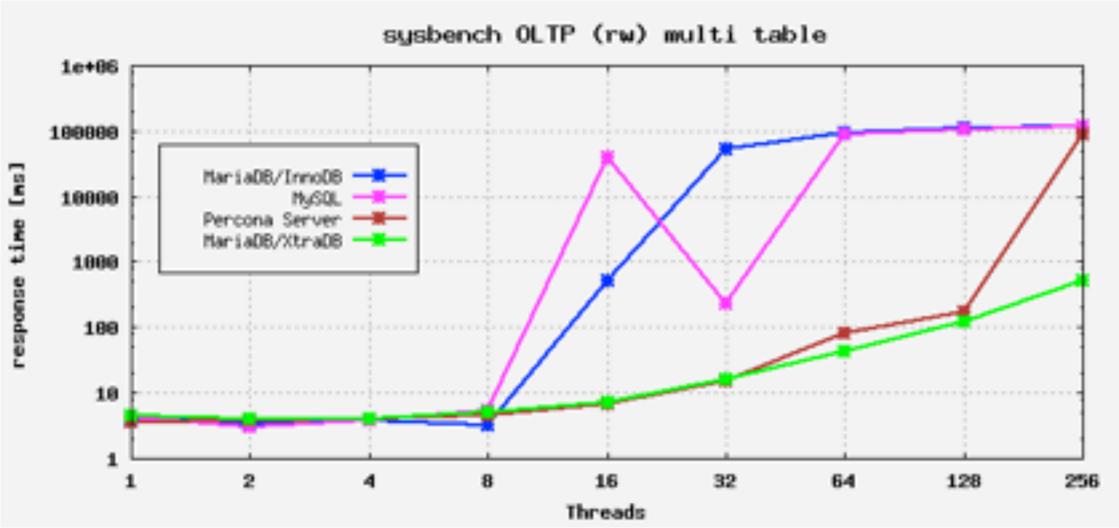
XtraDB



- A more performant InnoDB designed to scale on modern hardware

- Less checkpointing (smoother), less flushing to disk





Switching between XtraDB & InnoDB

```
mysqld --ignore-builtin-innodb --plugin-load=innodb=ha_innodb.so --plugin_dir=/usr/local/mysql/lib/mysql/plugin
```

Or in my.cnf

```
[mysqld]

ignore-builtin-innodb

plugin-load=innodb=ha_innodb.so

plugin_dir=/usr/local/mysql/lib/mysql/plugin
```

Usernames

- Usernames in MariaDB > 5.5.31? 80 character limit (which you have to reload manually)

```
create user
'long12345678901234567890'@'localhost'
identified by 'pass';
```

```
Query OK, 0 rows affected (0.01 sec)
```

vs

```
ERROR 1470 (HY000): String
'long12345678901234567890' is too long for
user name (should be no longer than 16)
```

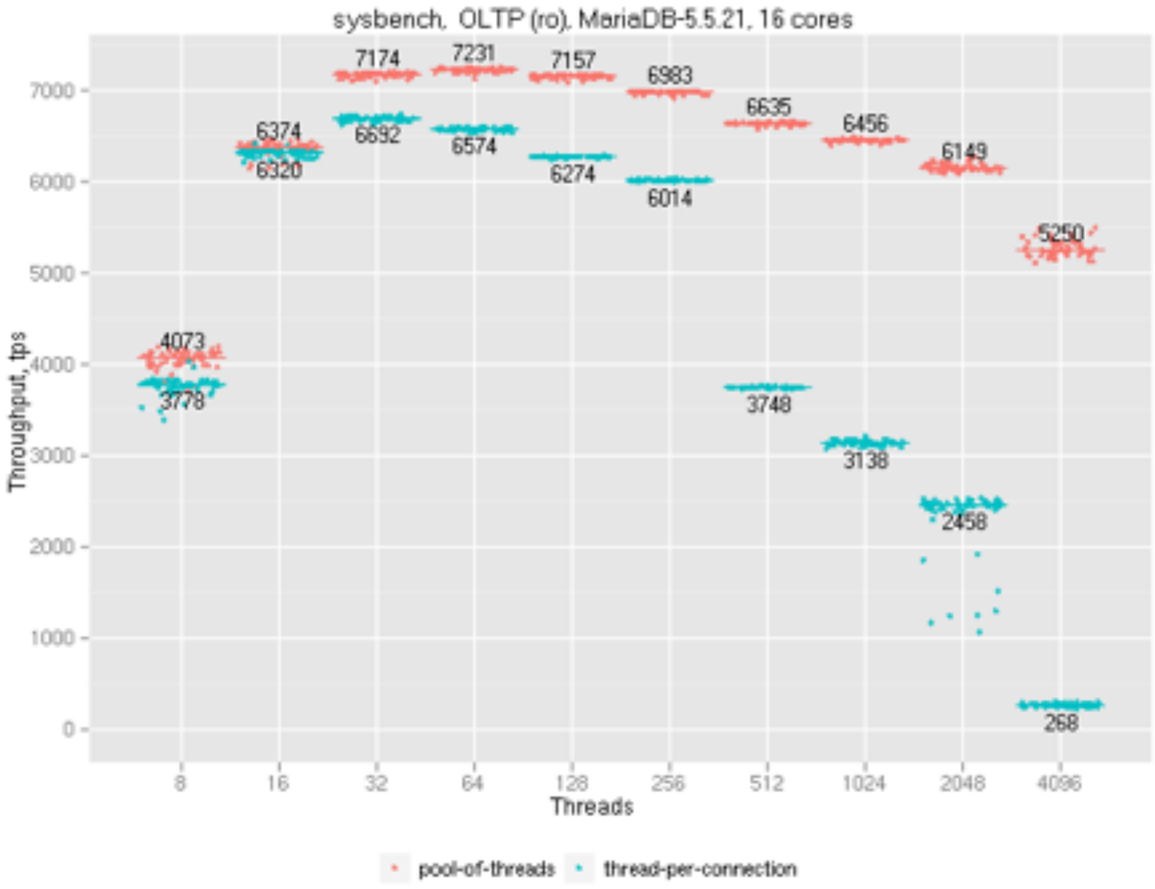
MariaDB 5.5: an opensource threadpool

- Modified from 5.1 (libevent based), great for CPU bound loads and short running queries

- No minimization of concurrent transactions with dynamic pool size

- `thread_handling=pool-of-threads`

- https://mariadb.com/kb/en/mariadb/thread-pool-in-mariadb/



Improved threadpool

- You can limit resources used by threads: thread_pool_max_threads
- SHOW GLOBAL VARIABLES LIKE '%thread_%';

Better for DBAs: async client library

- start operation, do work in thread, operation processed, result travels back
  - use cases: multiple queries against single server (utilize more CPUs); queries against multiple servers (SHOW STATUS on many machines)

- https://kb.askmonty.org/en/about-non-blocking-operation-in-the-client-library/
- fast node.js driver available: mariasql
  - https://kb.askmonty.org/en/mariasql-for-nodejs/

LIMIT ROWS EXAMINED

- The purpose of this optimization is to provide the means to terminate the execution of SELECT statements which examine too many rows, and thus use too many resources.

- `SELECT * from t1, t2 LIMIT 10 ROWS EXAMINED 1000;`

- https://kb.askmonty.org/en/limit-rows-examined/

SHOW STATUS

- `SHOW STATUS` provides server status information. It is like mysqladmin extended-status.
- `SHOW STATUS LIKE ‘Key%’;`
- https://kb.askmonty.org/en/show-status/
- https://kb.askmonty.org/en/server-status-variables/
- MariaDB has opened_views, executed_triggers, executed_events, feature_* as new options

SQL Error Logging Plugin

- Log errors sent to clients in a log file that can be analysed later. Log file can be rotated (recommended)
- a MYSQL_AUDIT_PLUGIN

```
install plugin SQL_ERROR_LOG
soname 'sql_errlog.so';
```

Audit Plugin

- Log server activity - who connects to the server, what queries run, what tables touched - rotating log file or syslogd
- MariaDB has extended the audit API, so user filtering is possible
- a MYSQL_AUDIT_PLUGIN

```
INSTALL PLUGIN server_audit SONAME
‘server_audit.so’;
```

Replication: selective skipping

- All changes that are logged as events in the binlog are replicated to all slaves
- However, sometimes you want all to be logged to binlog but skipped replication to slaves
- `@@skip_replication` (session only)
- `replicate_events_marked_for_skip = replicate|filter_on_slave|filter_on_master` (dynamic)

Replication: dynamic variables

- The variables replicate_do_*, replicate_ignore_*, and replicate_wild_* have been made dynamic, so they can be changed without requiring a server restart.
- https://kb.askmonty.org/en/dynamic-replication-variables/

Replication: Annotation of RBR events

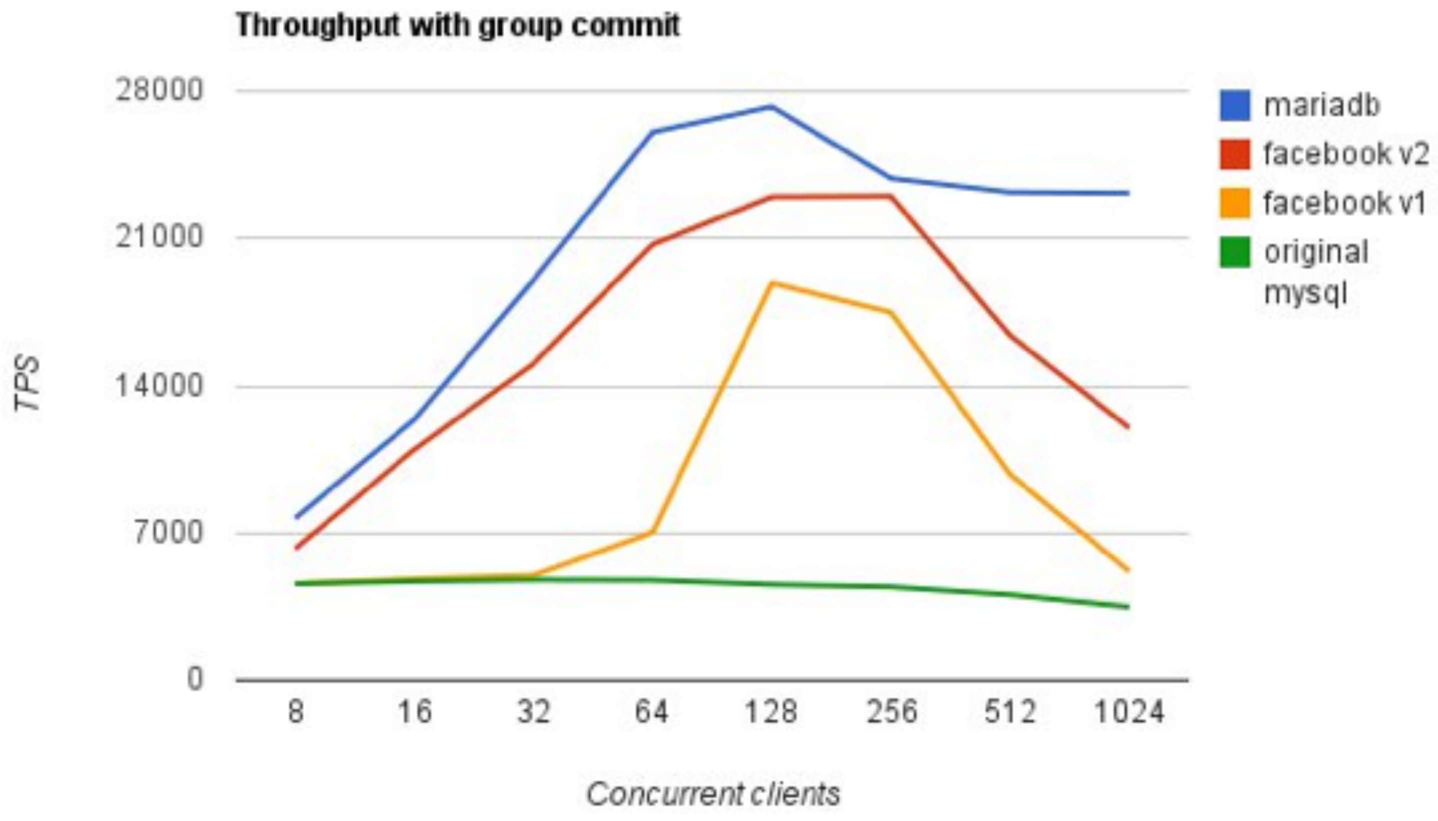
- MariaDB supports statement & row based replication (RBR)
- In RBR, the binlog has no SQL statements, only events are logged (INSERT, DELETE, etc)
- Option to include original SQL statement (default OFF)
- https://kb.askmonty.org/en/annotate_rows_log_event/

Replication: binlog event checksums

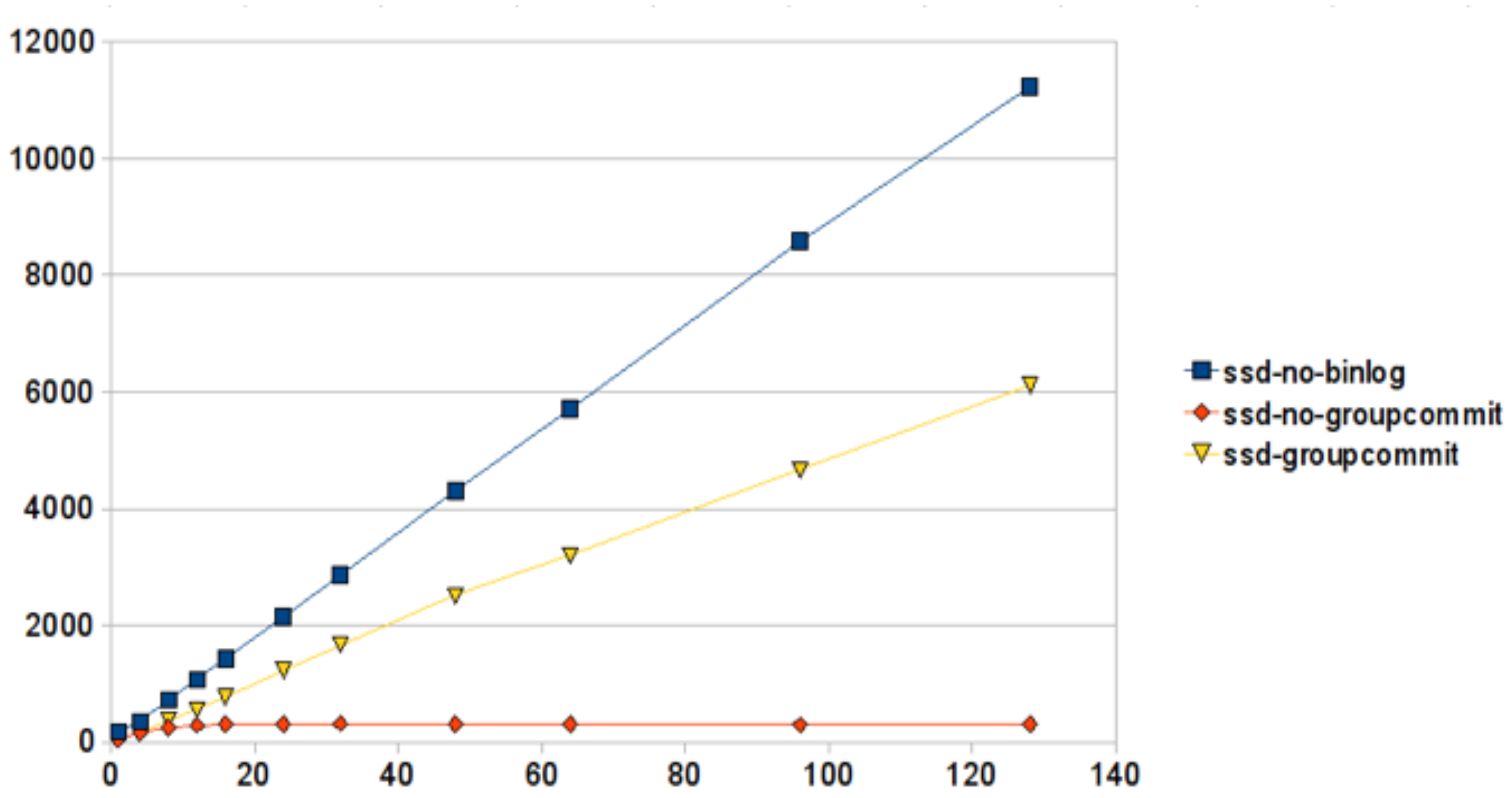
- Backport from MySQL 5.6 (in MariaDB 5.3+)
- `binlog_checksum` option
- Slaves perform checksums on events received & will stop if there is corruption
- https://kb.askmonty.org/en/binlog-event-checksums/

Replication: group commit in the binary log

- `sync_binlog=1, innodb_flush_log_at_trx_commit=1`
- https://www.facebook.com/note.php?note_id=10150261692455933
- http://kb.askmonty.org/en/group-commit-for-the-binary-log
- SHOW STATUS LIKE 'binlog_%commits';



MySQL 5.5 sync_binlog=1



Group commit in MariaDB 5.3 onwards

- Do slow part of `prepare()` in parallel in InnoDB (first `fsync()`, InnoDB group commit)
- Put transaction in queue, decide commit order

- First in queue runs serial part for all, rest wait
  - Wait for access to the binlog
  - Write transactions into binlog, in order, then sync (second `fsync()`)
  - Run the fast part of `commit()` for all transactions in order

- Finally, run the slow part of `commit()` in parallel (third `fsync()`, InnoDB group commit)
- Only 2 context switches per thread (one sleep, one wakeup)
- Note: MySQL 5.6, MariaDB 10 only does 2 fsyncs/group commit

Group commit in MariaDB 10

- Remove commit in slow part of InnoDB commit (stage 4)
- Reduce cost of crash-safe binlog
  - A binlog checkpoint is a point in the binlog where no crash recovery is needed before it. In InnoDB you wait for flush + fsync its redo log for commit

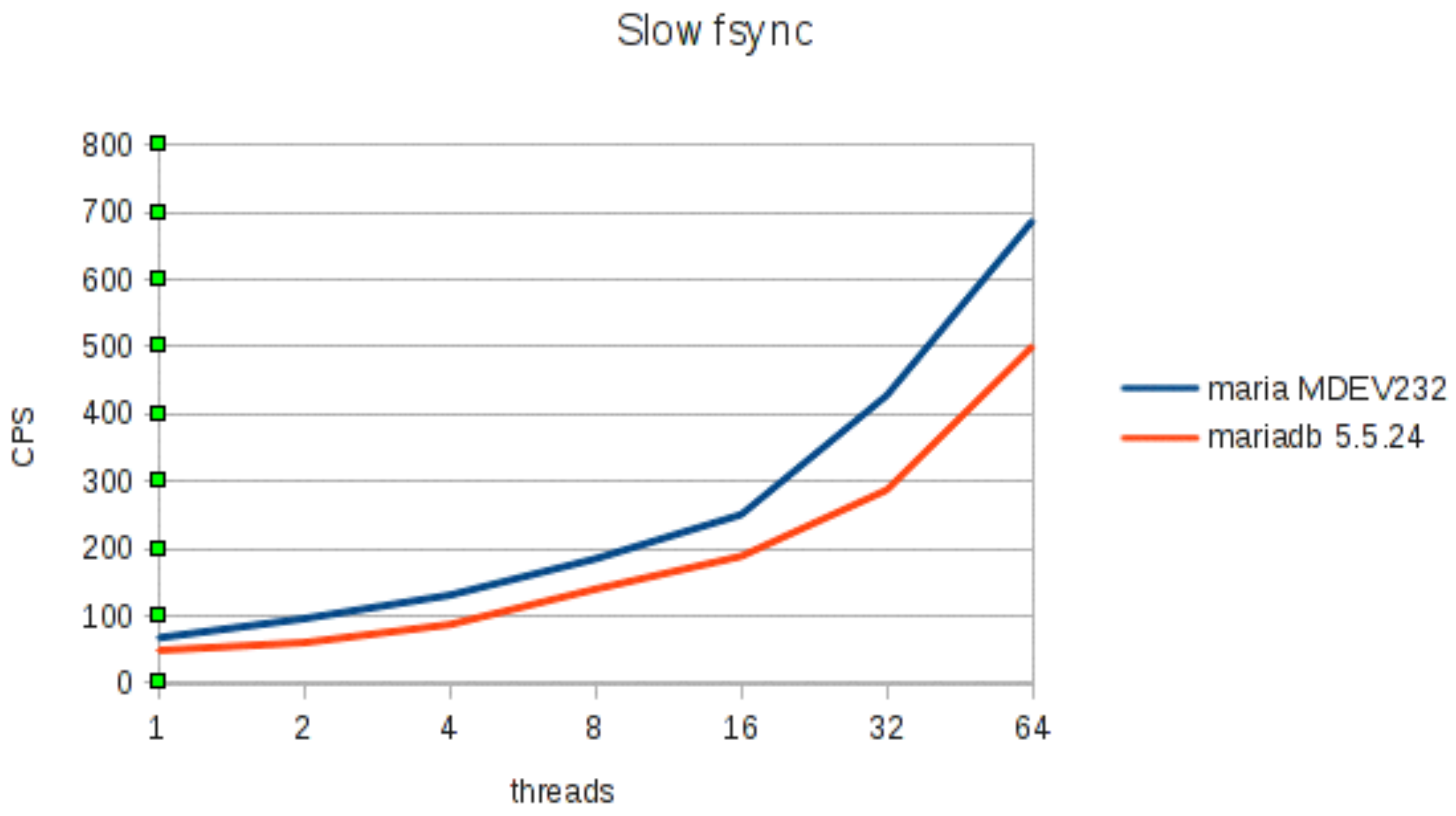
crash-safe binlog

- MariaDB 5.5 checkpoints after every commit —> quite expensive!
- 5.5/5.6 stalls commits around binlog rotate, waiting for all prepared transactions to commit (since crash recovery can only scan latest binlog file)

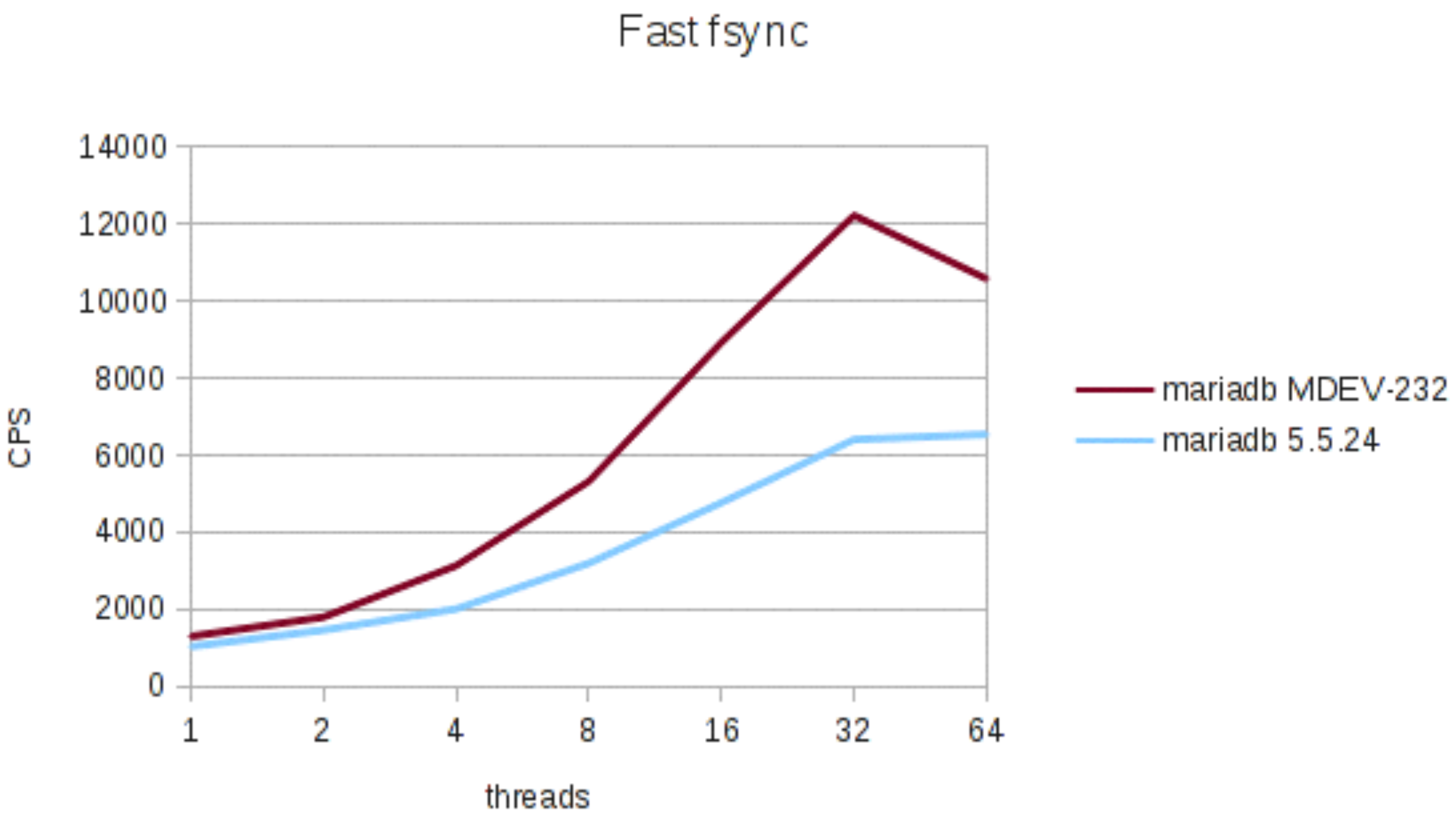
crash-safe binlog 10.0

- 10.0 makes binlog checkpoints asynchronous
  - A binlog can have no checkpoints at all
  - Ability to scan multiple binlogs during crash recovery
  - Remove stalls around binlog rotates

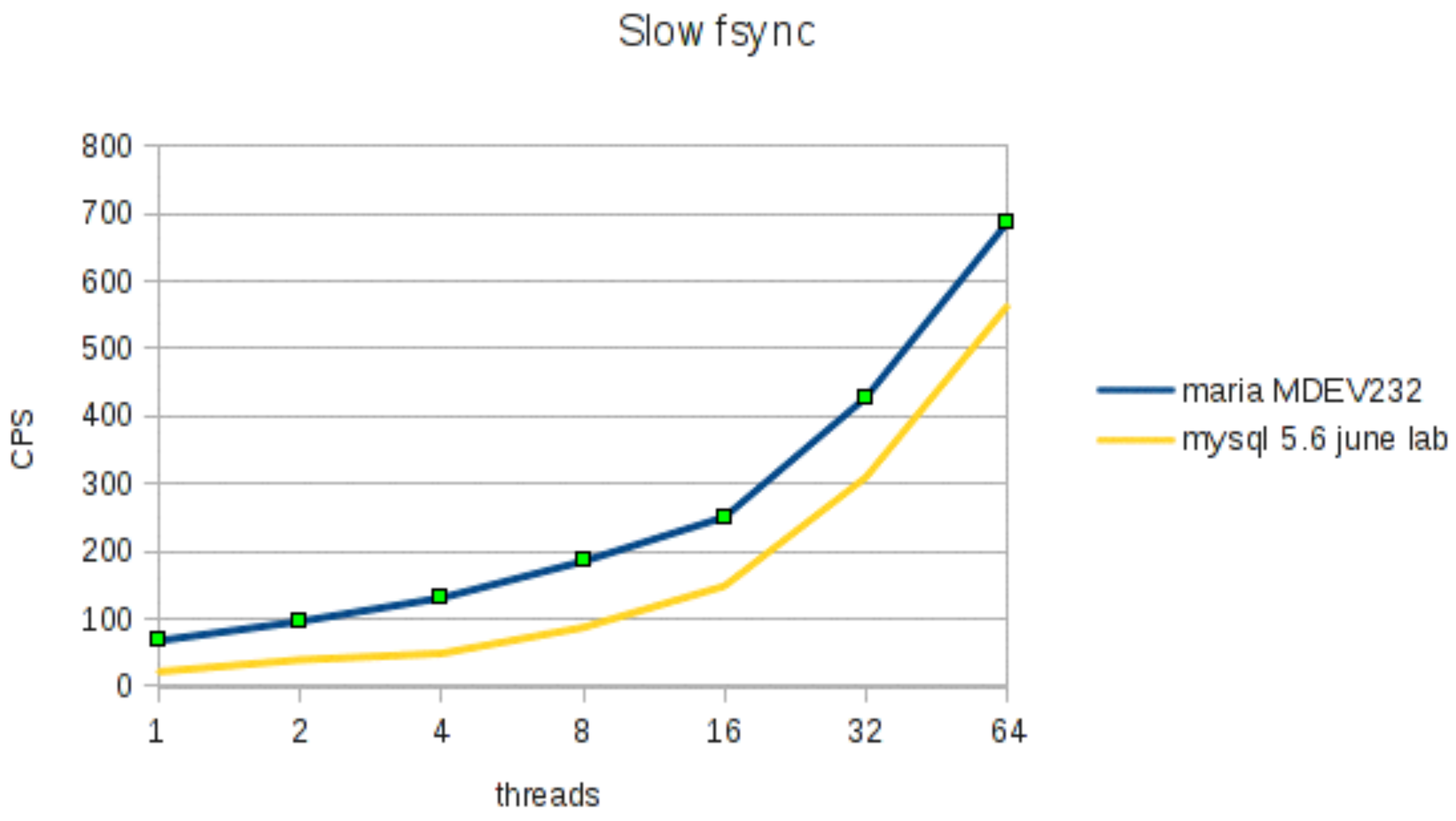
Slow fsync()


Slow fsync
800
700
600
500
400
300
200
100
0
CPS
1
2
4
8
16
32
64
threads
maria MDEV232
mariadb 5.5.24

Fast fsync()


Fast fsync
14000
12000
10000
8000
6000
4000
2000
0
CPS
1
2
4
8
16
32
64
threads
mariadb MDEV-232
mariadb 5.5.24

10.0 vs 5.6 group commit



Extensions to the SE API

- `prepare()` - write prepared trx in parallel w/group commit
- `prepare_ordered()` - called serially, in commit order
- `commit_ordered()` - called serially, in commit order; fast commit to memory
- `commit()` - commit to disk in parallel,

group commit in 10.1

- Tricky locking issues hard to change without getting deadlocks sometimes
  - mysql#68251, mysql#68569
- New code? **Binlog rotate in background thread (further reducing stalls)**. Split transactions across binlogs, so big transactions do not lead to big binlog files
- **Enhanced semi-sync replication** (wait for slave before commit on the master rather than after commit)

Replication: START TRANSACTION WITH CONSISTENT SNAPSHOT

- Works with the binlog, possible to obtain the binlog position corresponding to a transactional snapshot of the database without blocking any other queries.
  - by-product of group commit in the binlog to view commit ordering
- Used by the command `mysqldump--single-transaction --master-data` to do a fully non-blocking backup which can be used to provision a new slave
- Works consistently between transactions involving more than one storage engine
- https://kb.askmonty.org/en/enhancements-for-start-transaction-with-consistent/

GIS support!

- MySQL has OpenGIS SFS (Simple feature access, SQL access method)
  - Now, SQL with full geometry types
  - ST_ prefix (incl. ST_RELATE, ST_BOUNDARY, etc.)
- MyISAM, Aria for SPATIAL & non-spatial indexes
  - When 5.7-InnoDB is merged, it will get support too

GIS II

- Spatial reference systems support (REF_SYSTEM_ID) can be specified as a column attribute
- INFORMATION_SCHEMA.GEOMETRY_COLUMNS for queries of references
- Use Osmosis, you can load all OpenStreetMap data into MariaDB now
- https://mariadb.com/kb/en/gis-features-in-533/
- https://mariadb.com/kb/en/openstreetmap-dataset/

Progress reporting

- ALTER TABLE & LOAD DATA INFILE

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

TIME_MS in I_S.PROCESSLIST

- Extra column 'TIME_MS' has been added to the INFORMATION_SCHEMA.PROCESSLIST table
- Units of milliseconds with microsecond precision (the unit and precision of the 'TIME' column is one second).

New KILL syntax

- HARD | SOFT & USER USERNAME are MariaDB-specific (5.3.2)
- KILL QUERY ID query_id (10.0.5) - kill by query id, rather than thread id
- SOFT ensures things that may leave a table in an inconsistent state aren't interrupted (like REPAIR or INDEX creation for MyISAM or Aria)

```
KILL [HARD | SOFT] [CONNECTION | QUERY]
[thread_id | USER user_name]
```

SphinxSE

The old days

- Download MySQL, including sources
- Download SphinxSE for compiling
- Download Sphinx to compile with MySQL support
- Documented: http://www.howtoforge.com/sphinx-as-mysql-storage-engine-sphinxse

Today

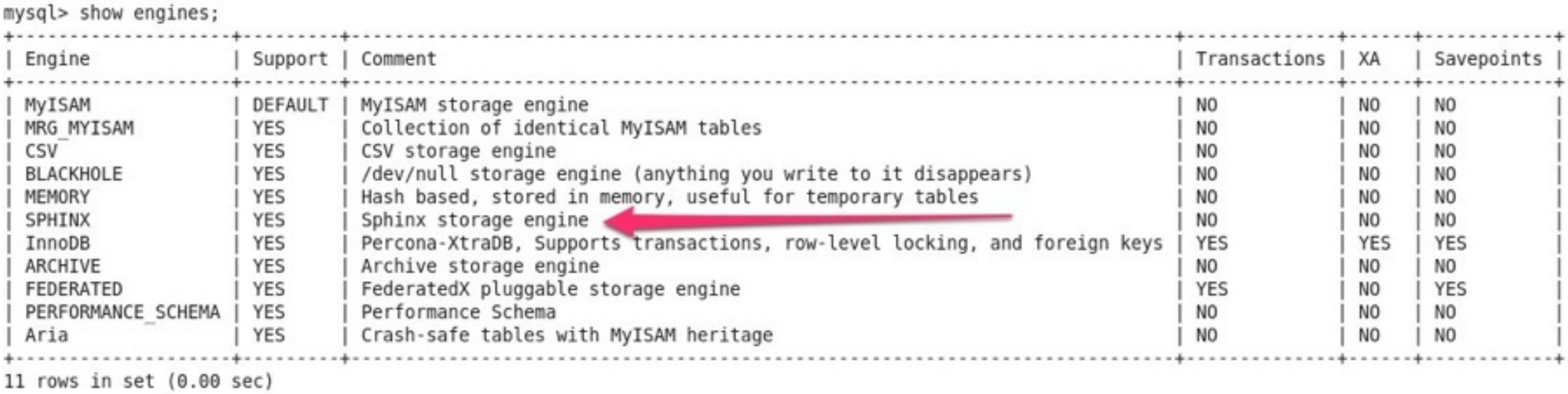
- Install sphinx from your distribution
- Install MariaDB 5.5 from your distribution or from http://mariadb.org/
- Get started!

Getting started

```
mysql> INSTALL PLUGIN sphinx
SONAME 'ha_sphinx.so';

Query OK, 0 rows affected
(0.01 sec)
```

Another engine appears



```
mysql> show engines;
+--------------------+---------+----------------------------------------------------------------------------+--------------+------+------------+
| Engine             | Support | Comment                                                                    | Transactions | XA   | Savepoints |
+--------------------+---------+----------------------------------------------------------------------------+--------------+------+------------+
| MyISAM             | DEFAULT | MyISAM storage engine                                                      | NO           | NO   | NO         |
| MRG_MYISAM         | YES     | Collection of identical MyISAM tables                                      | NO           | NO   | NO         |
| CSV                | YES     | CSV storage engine                                                         | NO           | NO   | NO         |
| BLACKHOLE          | YES     | /dev/null storage engine (anything you write to it disappears)             | NO           | NO   | NO         |
| MEMORY             | YES     | Hash based, stored in memory, useful for temporary tables                  | NO           | NO   | NO         |
| SPHINX             | YES     | Sphinx storage engine                                                      | NO           | NO   | NO         |
| InnoDB             | YES     | Percona-XtraDB, Supports transactions, row-level locking, and foreign keys | YES          | YES  | YES        |
| ARCHIVE            | YES     | Archive storage engine                                                     | NO           | NO   | NO         |
| FEDERATED          | YES     | FederatedX pluggable storage engine                                        | YES          | NO   | YES        |
| PERFORMANCE_SCHEMA | YES     | Performance Schema                                                         | NO           | NO   | NO         |
| Aria               | YES     | Crash-safe tables with MyISAM heritage                                     | NO           | NO   | NO         |
+--------------------+---------+----------------------------------------------------------------------------+--------------+------+------------+
11 rows in set (0.00 sec)
```

What is SphinxSE?

- SphinxSE is just the storage engine that still depends on the Sphinx daemon
- It doesn’t store any data itself
- Its just a built-in client to allow MariaDB to talk to Sphinx searchd, run queries, obtain results
- Indexing, searching is performed on Sphinx

Configure sphinx!

- /usr/local/sphinx/sphinx.conf
- Source (multiple, include mysql, with connection info)
- Setup indexer (esp. if its on localhost) - mem_limit, max_iops, max_iosize
- Setup searchd (where to listen to, query log, etc.)

Use case scenarios

- Already have an existing application that makes use of full-text-search in MyISAM? Porting should be easier
- Have a programming language without a native API for Sphinx? Surely there’s a connector for MariaDB ;-)

Use case scenarios

- Results from Sphinx itself almost always require additional work involving MariaDB
  - Say to pull out text column that Sphinx index doesn’t store
  - JOIN with another table (using a different engine)

An example

```
CREATE TABLE t1
(
    id              INTEGER UNSIGNED NOT NULL,
    weight          INTEGER NOT NULL,
    query           VARCHAR(3072) NOT NULL,
    group_id        INTEGER,
    INDEX(query)
) ENGINE=SPHINX CONNECTION="sphinx://localhost:9312/test";

SELECT * FROM t1 WHERE query='test it;mode=any';
```

Sphinx search tables

- 1st column: INTEGER UNSIGNED or BIGINT (document ID)
- 2nd column: match weight
- 3rd column: VARCHAR or TEXT (your query)
- Query column needs indexing, no other column needs to be

What actually happens

- SELECT passes a Sphinx query as the query column in the WHERE clause
- searchd returns the results
- SphinxSE translates and returns the results to MariaDB

SHOW ENGINE SPHINX STATUS

- Per-query & per-word statistics that searchd returns are accessible via SHOW STATUS

```
mysql> SHOW ENGINE SPHINX STATUS;
+--------+-------+-------------------------------------------------+
| Type   | Name  | Status                                          |
+--------+-------+-------------------------------------------------+
| SPHINX | stats | total: 25, total found: 25, time: 126, words: 2 |
| SPHINX | words | sphinx:591:1256 soft:11076:15945                |
+--------+-------+-------------------------------------------------+
2 rows in set (0.00 sec)
```

What queries are supported?

- Most of the Sphinx API is exposed to SphinxSE
- query, mode, sort, offset, limit, index, minid, maxid, weights, filter, !filter, range, !range, maxmatches, groupby, groupsort, indexweights, comment, select
- Sphinx search modes can also be supported via _sph attributes
  - obtain value of @groupby? use '_sph_groupby'

Efficiency

- Allow Sphinx to perform sorting, filtering, and slicing of result set
  - ... as opposed to using WHERE, ORDER BY, LIMIT clauses on MariaDB
- Why?
  - Sphinx optimises and performs better on these tasks
  - Less data packed by searchd, and transferred and unpacked by SphinxSE

JOINs

- Perform JOINs on a SphinxSE search table using tables from other engines

```
SELECT content, date_added FROM test.documents docs
-> JOIN t1 ON (docs.id=t1.id)
-> WHERE query="one document;mode=any";
+-------------------------------------+---------------------+
| content                             | docdate             |
+-------------------------------------+---------------------+
| this is my test document number two | 2006-06-17 14:04:28 |
| this is my test document number one | 2006-06-17 14:04:28 |
+-------------------------------------+---------------------+
2 rows in set (0.00 sec)
```

Statistics

- Understand server activity better to understand database loads
- `SET GLOBAL userstat=1;`
  - `SHOW CLIENT_STATISTICS; SHOW USER_STATISTICS;`
    - # of connections, CPU usage, bytes received/sent, row statistics
  - `SHOW INDEX_STATISTICS; SHOW TABLE_STATISTICS;`
    - # rows read, changed, indexes

MariaDB 10.0+

- `INFORMATION_SCHEMA.PROCESSLIST` has `MEMORY_USAGE`, `EXAMINED_ROWS` (similar with `SHOW STATUS` output)

Table Elimination

- Resolve a query without accessing some tables query refers to
  - Great for querying highly normalised data
- Basis of "anchor modelling"
  - http://www.anchormodeling.com/
- SQL Server 2005/2008, Oracle 11g have it

Virtual columns

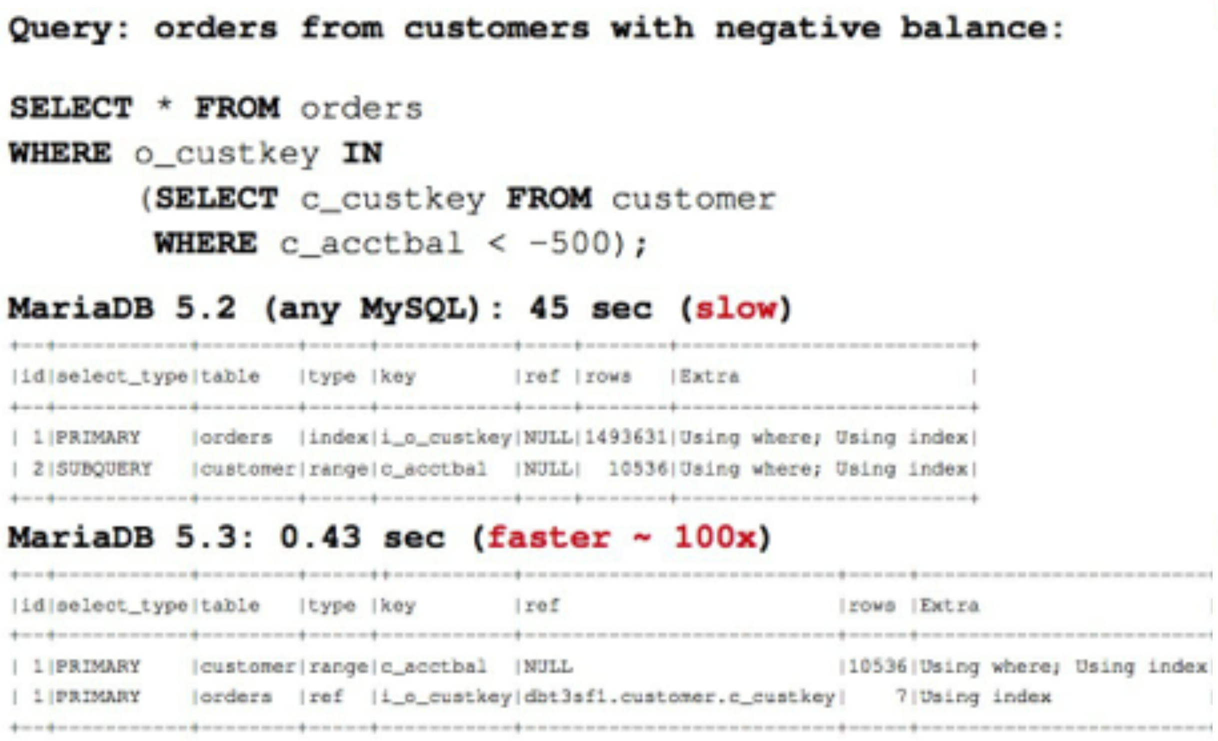
- A column in a table that has its value automatically calculated either with a pre-calculated/deterministic expression or values of other fields in the table
- PERSISTENT (computed when data is inserted or stored in a table) or VIRTUAL (like a VIEW)
- Similar to MS SQL or Oracle
- https://kb.askmonty.org/en/virtual-columns/

Optimizer enchancements

- Join additions
  - block nested loop joins for outer-joins, block hash joins, Batched Key Access (BKA)
- Optimization for derived tables & views
  - mergeable derived tables processed like VIEWs + optimizer creates indexes over materialized derived tables
- Disk access optimization
  - Index Condition Pushdown (ICP), Multi-Range Read (MRR)

Subquery optimizations

- Semi-join optimization, materialization for non-correlated IN queries, subquery cache
- Goodbye rewriting as JOINs or separate queries



Query: orders from customers with negative balance:

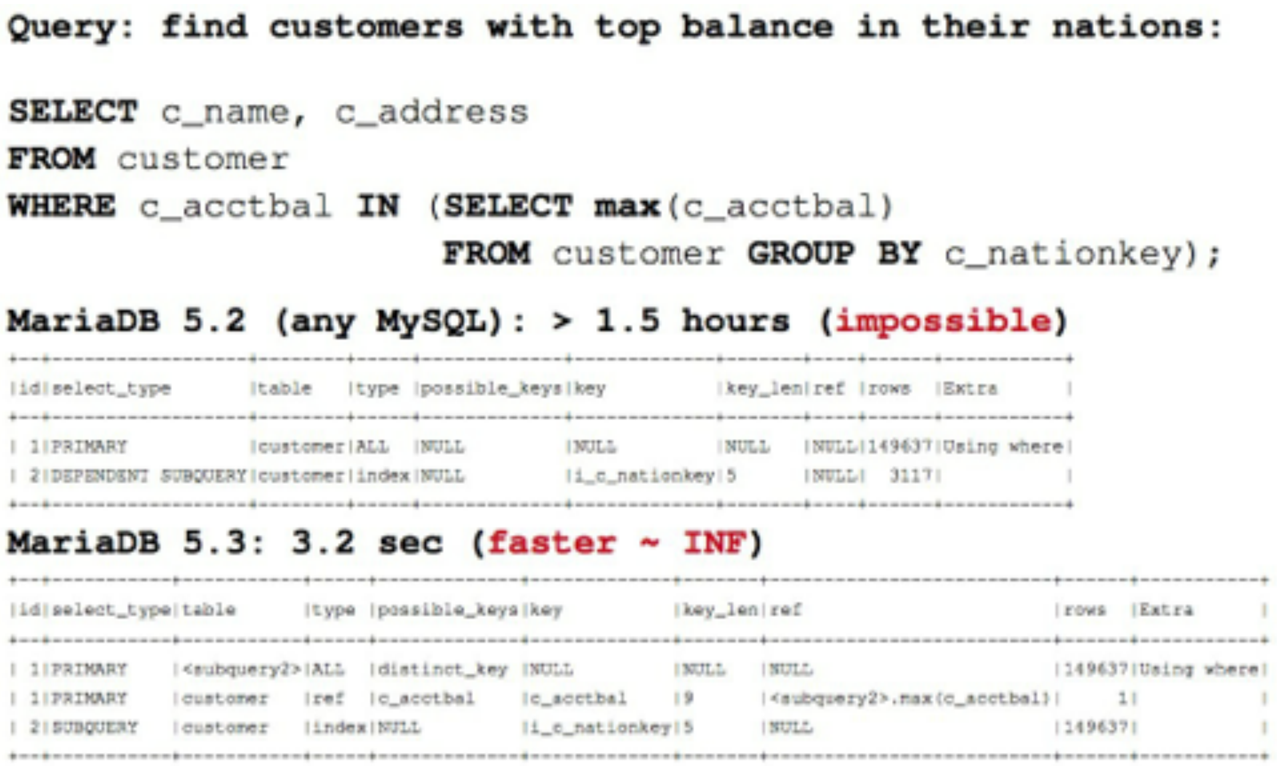
```
SELECT * FROM orders
WHERE o_custkey IN
       (SELECT c_custkey FROM customer
        WHERE c_acctbal < -500);
```

MariaDB 5.2 (any MySQL): 45 sec (slow)

```
+--+-----------+--------+-----+-----------+----+-------+-------------------------+
|id|select_type|table   |type |key        |ref |rows   |Extra                    |
+--+-----------+--------+-----+-----------+----+-------+-------------------------+
| 1|PRIMARY    |orders  |index|i_o_custkey|NULL|1493631|Using where; Using index|
| 2|SUBQUERY   |customer|range|c_acctbal  |NULL|  10536|Using where; Using index|
+--+-----------+--------+-----+-----------+----+-------+-------------------------+
```

MariaDB 5.3: 0.43 sec (faster ~ 100x)

```
+--+-----------+--------+-----+-----------+----------------------------+-----+-----------------------
|id|select_type|table   |type |key        |ref                         |rows |Extra
+--+-----------+--------+-----+-----------+----------------------------+-----+-----------------------
| 1|PRIMARY    |customer|range|c_acctbal  |NULL                        |10536|Using where; Using index
| 1|PRIMARY    |orders  |ref  |i_o_custkey|dbt3sf1.customer.c_custkey  |    7|Using index
+--+-----------+--------+-----+-----------+----------------------------+-----+-----------------------
```



Query: find customers with top balance in their nations:

```
SELECT c_name, c_address
FROM customer
WHERE c_acctbal IN (SELECT max(c_acctbal)
                    FROM customer GROUP BY c_nationkey);
```

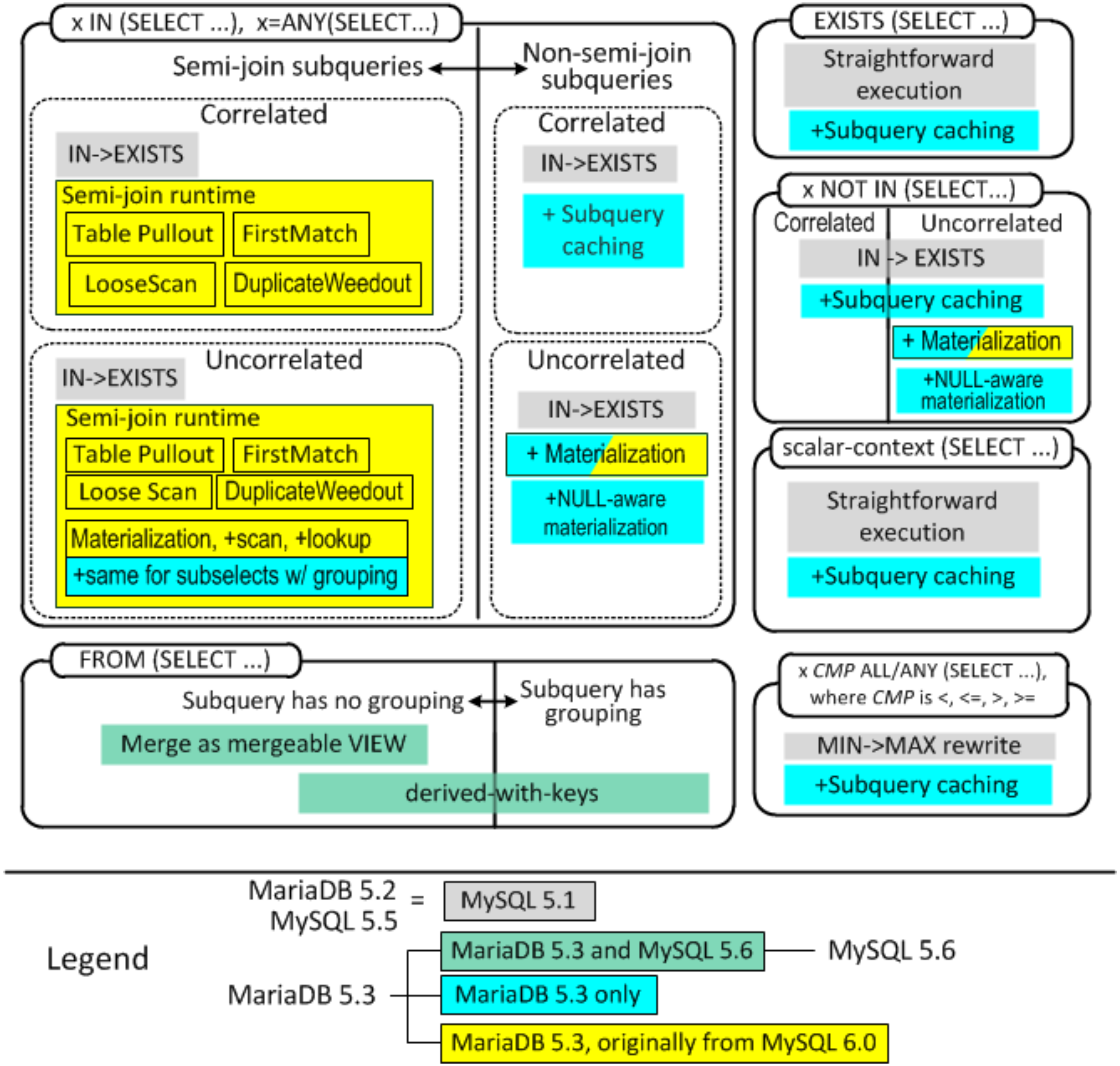
MariaDB 5.2 (any MySQL): > 1.5 hours (impossible)

```
+--+------------------+--------+-----+-------------+------------+-------+----+------+------------+
|id|select_type       |table   |type |possible_keys|key         |key_len|ref |rows  |Extra       |
+--+------------------+--------+-----+-------------+------------+-------+----+------+------------+
| 1|PRIMARY           |customer|ALL  |NULL         |NULL        |NULL   |NULL|149637|Using where|
| 2|DEPENDENT SUBQUERY|customer|index|NULL         |i_c_nationkey|5     |NULL|  3117|            |
+--+------------------+--------+-----+-------------+------------+-------+----+------+------------+
```

MariaDB 5.3: 3.2 sec (faster ~ INF)

```
+--+-----------+-------------+-----+-------------+-------------+-------+--------------------------+------+-----------+
|id|select_type|table        |type |possible_keys|key          |key_len|ref                       |rows  |Extra      |
+--+-----------+-------------+-----+-------------+-------------+-------+--------------------------+------+-----------+
| 1|PRIMARY    |<subquery2>  |ALL  |distinct_key |NULL         |NULL   |NULL                      |149637|Using where|
| 1|PRIMARY    |customer     |ref  |c_acctbal    |c_acctbal    |9      |<subquery2>.max(c_acctbal)|     1|           |
| 2|SUBQUERY   |customer     |index|NULL         |i_c_nationkey|5      |NULL                      |149637|           |
+--+-----------+-------------+-----+-------------+-------------+-------+--------------------------+------+-----------+
```

DBT-3, 60M rows, 29GB XtraDB



x IN (SELECT ...), x=ANY(SELECT...)

Semi-join subqueries ↔ Non-semi-join subqueries

Correlated

IN->EXISTS

Semi-join runtime

Table Pullout | FirstMatch

LooseScan | DuplicateWeedout

Uncorrelated

IN->EXISTS

Semi-join runtime

Table Pullout | FirstMatch

Loose Scan | DuplicateWeedout

Materialization, +scan, +lookup

+same for subselects w/ grouping

Correlated

IN->EXISTS

+ Subquery caching

Uncorrelated

IN->EXISTS

+ Materialization

+NULL-aware materialization

EXISTS (SELECT ...)

Straightforward execution

+Subquery caching

x NOT IN (SELECT...)

Correlated | Uncorrelated

IN -> EXISTS

+Subquery caching

+ Materialization

+NULL-aware materialization

scalar-context (SELECT ...)

Straightforward execution

+Subquery caching

FROM (SELECT ...)

Subquery has no grouping ↔ Subquery has grouping

Merge as mergeable VIEW

derived-with-keys

x *CMP* ALL/ANY (SELECT ...), where *CMP* is <, <=, >, >=

MIN->MAX rewrite

+Subquery caching

Legend

MariaDB 5.2 MySQL 5.5 = MySQL 5.1

MariaDB 5.3 and MySQL 5.6 — MySQL 5.6

MariaDB 5.3 — MariaDB 5.3 only

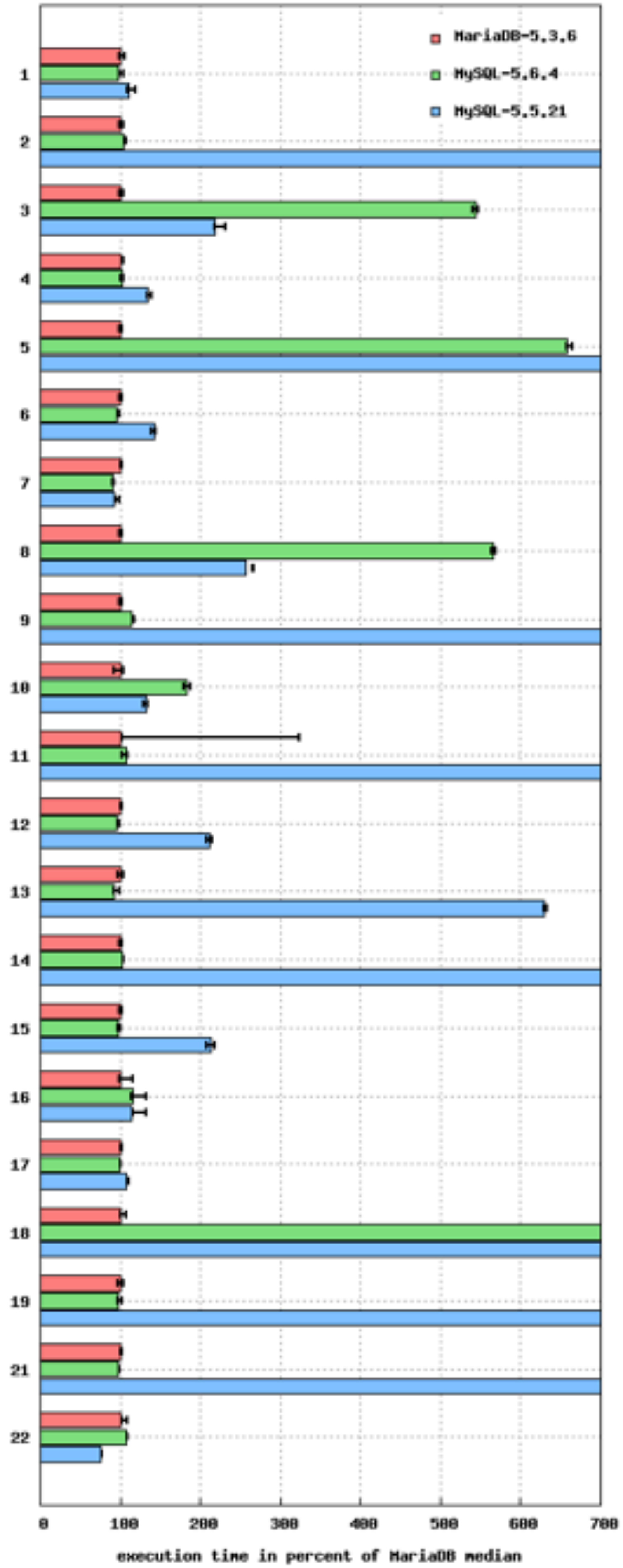
MariaDB 5.3, originally from MySQL 6.0

EXPLAIN

- INSTANT EXPLAIN
- EXPLAIN Analyzer - https://mariadb.org/explain_analyzer/analyze/
- Optimizer feature comparison matrix: https://kb.askmonty.org/en/optimizer-feature-comparison-matrix/
- optimizer_switch meanings: https://mariadb.com/kb/en/mariadb-53-optimizer_switch/

DBT-3

- https://mariadb.com/kb/en/dbt-3-dataset/
- https://blog.mariadb.org/mariadb-5-3-optimizer-benchmark/



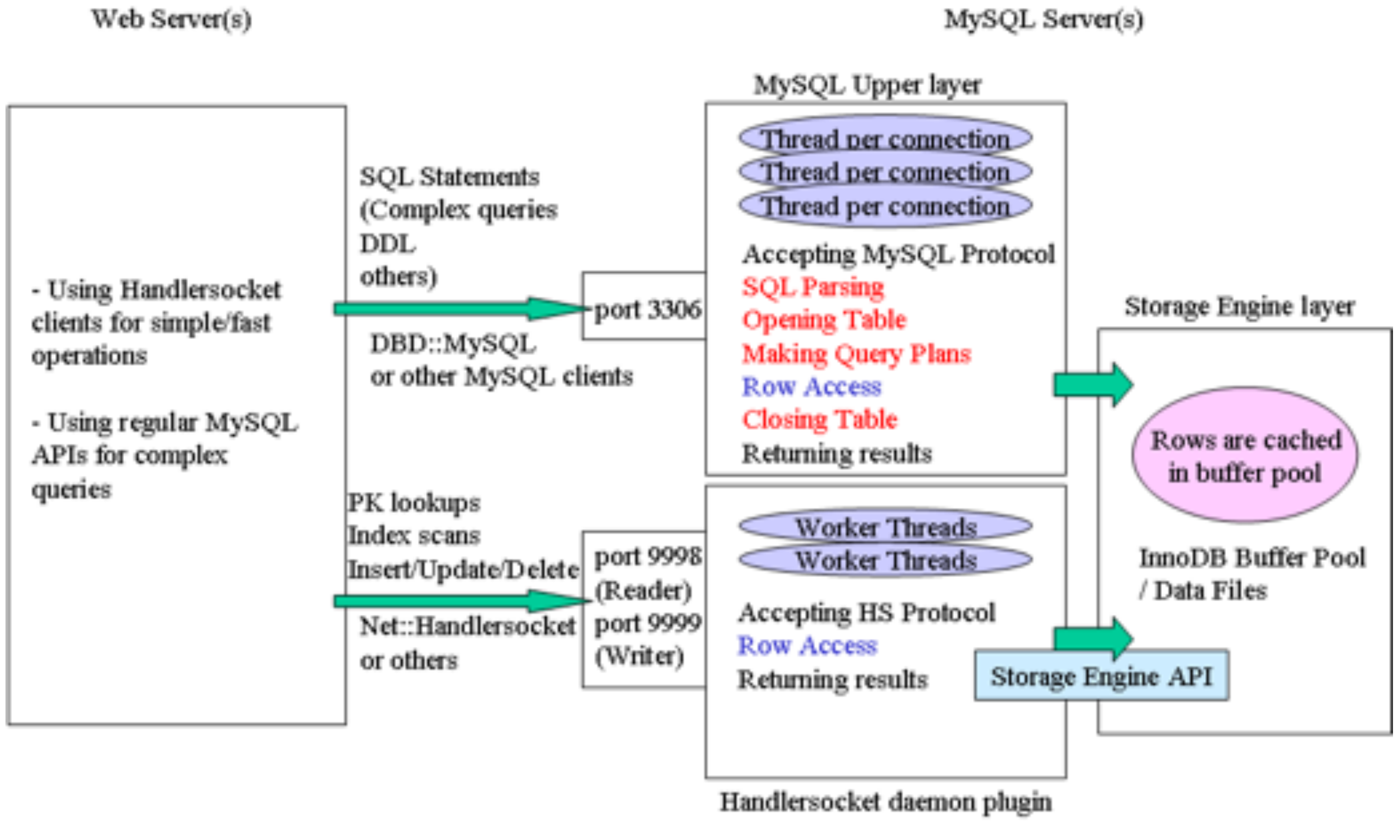
Extended keys

- Default is `extended_keys=on`
- Extended Keys, introduced in MariaDB 5.5, is an optimization which makes use of existing components of InnoDB/XtraDB keys to generate more efficient execution plans. Using these components in many cases allows the server to generate execution plans which employ index-only look-ups.
- https://mariadb.com/kb/en/extended-keys/

NoSQL: HandlerSocket

- Comes with HandlerSocket
  - direct access to XtraDB/ InnoDB for CRUD operations
  - INSTALL PLUGIN handlersocket SONAME 'handlersocket.so';

- SQL: 105,000 qps (60% usr, 28% sys)
- memcached : 420,000 qps (8% usr, 88% sys)
- HandlerSocket: 750,000 qps (45% usr, 53% sys)



NoSQL: dynamic columns

- Allows you to create virtual columns with dynamic content for each row in table
- Basically a BLOB with handling functions
- Store different attributes for each item (like a web store). Hard to do relationally
- In MariaDB 10.0: name support (instead of referring to columns by numbers, name it), convert all dynamic column content to JSON array, interface with Cassandra
- INSERT INTO tbl SET dyncol_blob=COLUMN_CREATE("column_name", "value");
- https://kb.askmonty.org/en/dynamic-columns/

Pluggable authentication

- MariaDB & MySQL now uses password authentication via pluggable auth
- Unix sockets
- PAM

auth_socket

- Authenticates against the Unix socket file
- Uses `so_peercred` socket option to obtain information about user running client
- `CREATE USER ‘monty’@‘localhost’ IDENTIFIED with auth_socket;`

PAM Authentication

- MySQL PAM
- Percona PAM (auth_pam & auth_pam_compat)
- MariaDB PAM (pam)

Let’s get somethings out of the way

- PAM = Pluggable Authentication Module
- Use pam_ldap to to authenticate credentials against LDAP server — configure /etc/pam_ldap.conf (you also obviously need /etc/ldap.conf)
- Simplest way is of course /etc/shadow auth

MariaDB

```
INSTALL SONAME ‘auth_pam’;

CREATE USER byte IDENTIFIED via pam
USING ‘mariadb’;

Edit /etc/pam.d/mariadb:

auth                required
pam_unix.so

account             required
pam_unix.so
```

For MySQL compatibility

- Just use –`pam-use-cleartext-plugin` for MySQL to use mysql_cleartext_password instead of dialog plugin

Possible errors

- Connectors don’t support it:
  - Client does not support authentication protocol requested by server; consider upgrading MySQL client.
  - You really have to re-compile connector using libmysqlclient to have said support

MariaDB 10.0

Why MariaDB 10.0?

- The 5.5 merge took about a year (!)
- We (MariaDB-5.5) have over 1.5 million lines of extra code with a ~61MB diff
- We didn’t want to repeat this for 5.6
- Also, MySQL 5.6 has a lot of re-factoring, thus loosing commit history

In a nutshell

- Built on MariaDB 5.5
- Backported features from MySQL 5.6
- New features

What about tools?

- SELECT VERSION() will return 10.0.1-MariaDB
- Oops, we found a bug in MySQL: https://mariadb.atlassian.net/browse/MDEV-4088 & http://bugs.mysql.com/bug.php?id=68187
- Still deciding:

- Use 9.0 for a name
- Lie to clients (no)
- Disallow replication (no)
- Use handshake packet 5.5.30-mysql-10.0.2-MariaDB without affecting VERSION() / @@global.version

What about tools II?

- Tools really should recognise MariaDB version as there are already many new features that MySQL doesn’t have
- eg. HeidiSQL supports virtual columns (http://www.heidisql.com/forum.php?t=8671)

Backported features

(i.e. these are from MySQL 5.6)

InnoDB & XtraDB

- MariaDB 10.0 ships InnoDB from MySQL 5.6
- MariaDB 10 ships Percona XtraDB as default
  - minimal performance improvements expected, just functionality & features
  - bitmap changed page tracking so xtrabackup can do incremental backups without scanning all InnoDB files
- SHOW GLOBAL VARIABLES LIKE 'innodb_ver%';

More from MySQL 5.6

- PERFORMANCE_SCHEMA
- InnoDB read-only transactions (TRANSACTION READ ONLY)
- Optimizer:
  - EXISTS-TO-IN optimization
  - ORDER BY...LIMIT optimization (show only few rows of a result set)
- CURRENT_TIMESTAMP as DEFAULT for DATETIME columns (this is a re-implementation in MariaDB)

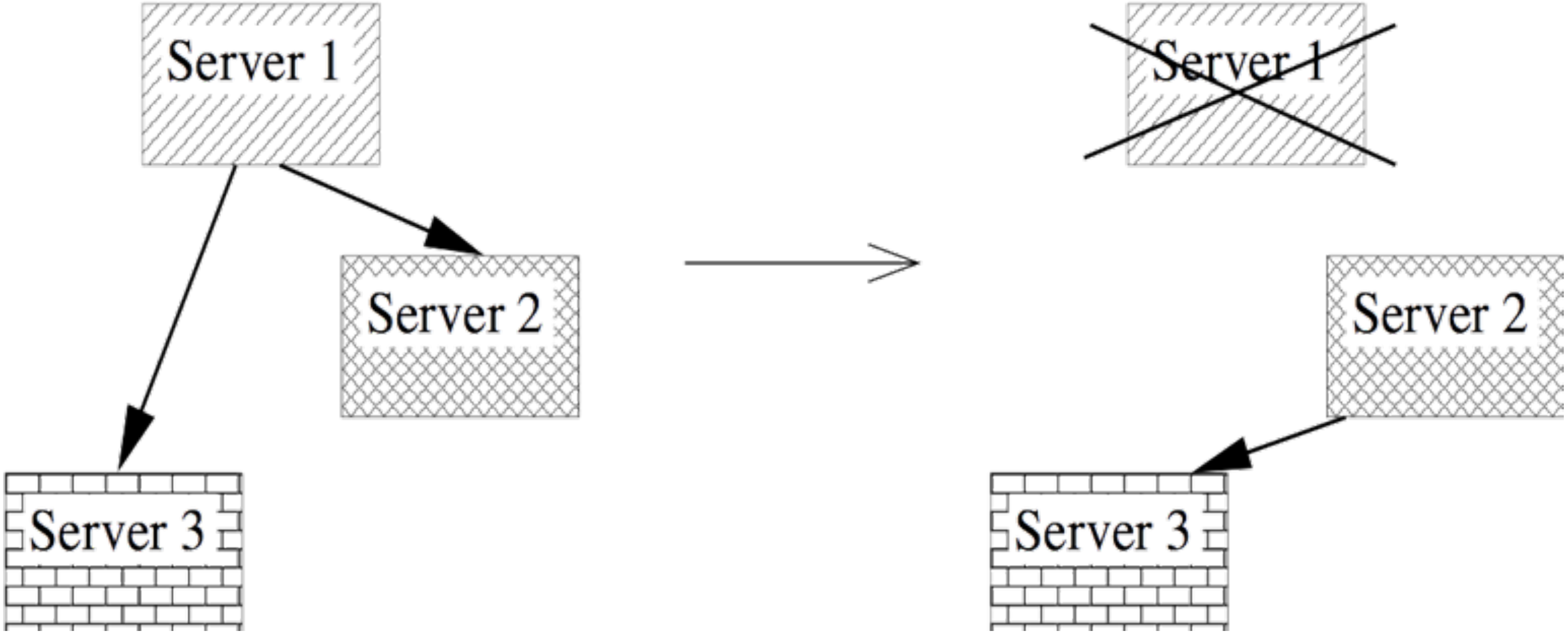
Only in MariaDB 10.0

the new stuff!

MariaDB 10 replication

- Global Transaction ID
  - have complex replication topologies; simple failover & slave promotion
  - doesn’t require restarts!
  - new slave provisioning: SET GLOBAL GTID_SLAVE_POS = BINLOG_GTID_POS("masterbin.00045", 600); CHANGE MASTER TO master_host="192.168.2.4", master_use_gtid=slave_pos; START SLAVE;
  - turning on GTID for slaves: STOP SLAVE
    CHANGE MASTER TO master_use_gtid=current_pos; START SLAVE;
  - change masters: STOP SLAVE
    CHANGE MASTER TO master_host="10.2.3.5"; START SLAVE;
- Crash-safe slaves - GTID position stored in InnoDB table

Automatic binlog position for master failover



- On Server2: CHANGE MASTER TO master_host='server2', master_use_gtid=1;

Why different GTID compared to 5.6?

- MySQL 5.6 GTID does not support multi-source replication
- Supports —log-slave-updates=0 for efficiency
- Enabled by default, with self-healing capabilities

Binlog (size matters!)

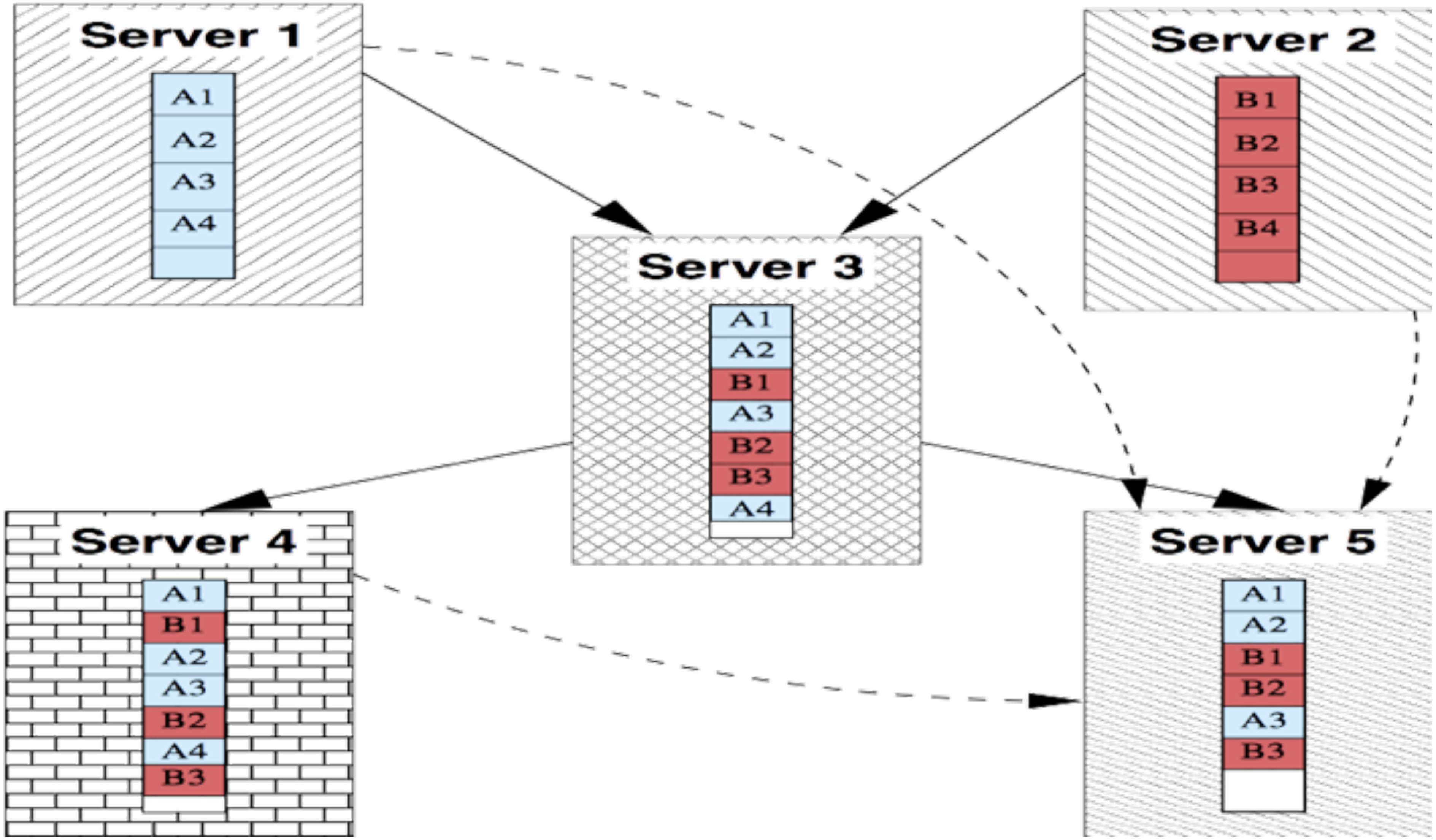
- Example query: INSERT INTO t1 VALUES (10, “foo”);
- MySQL 5.6… 265 bytes
- MariaDB 10.0… 161 bytes
- Do you want a 60% larger binlog size?

Crash-safe slave (w/ InnoDB DML)

- Replace non-transactional file relay_log.info with transactional `mysql.rpl_slave_state`
- Changes to `rpl_slave_state` are transactionally recovered after crash along with user data.

Replication domains

- Keep central concept that replication is just applying events in-order from a serial binlog stream.
- Allow multi-source replication with multiple active masters
- Let's the DBA configure multiple independent binlog streams (one per active master: `mysqld --git-domain-`


Server 1
A1
A2
A3
A4
Server 2
B1
B2
B3
B4
Server 3
A1
A2
B1
A3
B2
B3
A4
Server 4
A1
B1
A2
A3
B2
A4
B3
Server 5
A1
A2
B1
B2
A3
B3

Parallel replication

- Multi-source replication from different masters executed in parallel
  - Queries from different domains are executed in parallel
- Queries that are run in parallel on the master are run in parallel on the slave (based on group commit).
  - Transactions modifying the same table can be updated in parallel on the slave!
- Supports both statement based and row based replication.

Multi-source replication

- Work from Taobao
- Many users partition data across many masters... now you can replicate many masters to a single slave
- Great for analytical queries, complete backups, etc.
- @@default_master_connection contains current connection name (used if connection name is not given)
- All master/slave commands take a connection name now (like CHANGE MASTER "connection_name", SHOW SLAVE "connection_name" STATUS, etc.)
- https://kb.askmonty.org/en/multi-source-replication/

Only in 10.0

- SHOW EXPLAIN for <thread_id> (https://mariadb.com/kb/en/show-explain/) gets the query plan of a running statement

- EXPLAIN ANALYZE equivalent

- Faster ALTER TABLE with unique keys for Aria & MyISAM

  - Segmented MyISAM keycaches (up to 64) since MariaDB 5.2 exist too

- Per-thread memory usage (Taobao)

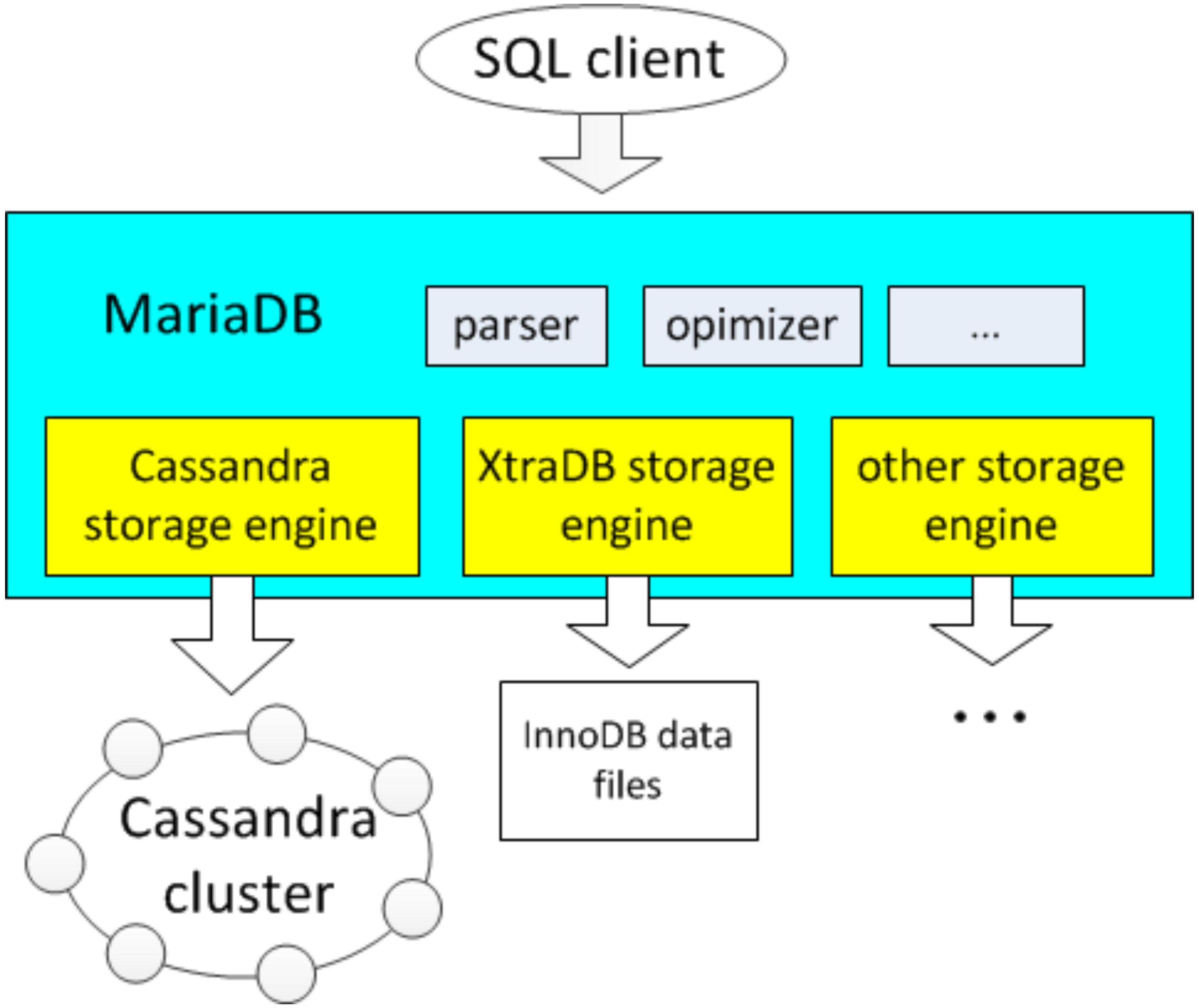
  - I_S.PROCESSLIST has MEMORY_USAGE & EXAMINED_ROWS

  - SHOW STATUS has memory usage too

SHUTDOWN

- shuts down the server; requires GRANTs similar to mysqladmin shutdown command
- you can create an event that does a SHUTDOWN of the server as an example…
- https://mariadb.com/kb/en/shutdown/

CassandraSE



CassandraSE

- Integration with NoSQL/Big Data DB, Apache Cassandra cluster, seen as a storage engine to MariaDB
- Combine (join) data between Cassandra & MariaDB
- Write to Cassandra from SQL (SELECT, INSERT, UPDATE, DELETE)
- CQL is great, but the goal is for you to just work with SQL, not switch between CQL & SQL
- Data is mapped: rowkey, static columns, dynamic columns
  - super columns aren't supported
- No 1-1 direct map for data types (ref: https://kb.askmonty.org/en/cassandra-storage-engine/)

TokuDB

- Opensource - separate MariaDB 5.5+TokuDB/integrated in 10.0.5
- Improved insert (10-20x faster) & query speed, compression (up to 90% space reduction), replication performance and online schema flexibility
- Uses Fractal Tree Indexes instead of B-Tree
- Tests & builds of TokuDB on multiple platforms (think greater distribution)

CONNECT

- CONNECT will speak XML or even grab data over an ODBC connection
- You can CONNECT to Oracle (via ODBC), join results from Cassandra (via CassandraSE) and have all your results sit in InnoDB
- Turn on engine condition pushdown

SPIDER

- Spider has built-in sharding features
- Partitioning & XA transaction capable
- Different MariaDB instance tables handled like it is the same instance

Engine-independent persistent statistics

- InnoDB has persistent statistics in MySQL 5.6; we have an engine-independent version
- These statistics aren't limited by the SE API, and are used by query optimizer to choose best execution plan for each statement
- Statistics collected for non-indexed columns too (unlike InnoDB's)

MariaDB 10.0.2

- Support for atomic writes on FusionIO DirectFS
- Optimizer collects & can use histogram-based statistics for non-indexed columns
- Better table discovery, so FederatedX has assisted discovery, Sequence engine (creates ascending/descending sequences, useful in joins)
- SHOW PLUGINS SONAME;

MariaDB 10.0.4

- SPIDER storage engine for database sharding merged
- Audit plugin
- complete PERFORMANCE_SCHEMA
    - INFORMATION_SCHEMA with upstream defaults too
- Online ALTER for InnoDB and thread information for in-place operations

MariaDB 10.0.5

- Parallel replication - https://mariadb.com/kb/en/parallel-replication/
  - automatically detect independent transactions, parallel within same table, adapts to master load, and preserves commit ordering
- EXPLAIN in the slow query log

MariaDB 10.0.6

- Serious incompatibility and data corruption of DATETIME and DATE types due to get_innobase_type_from_mysql_type refactor combined with InnoDB Online DDL
- https://mariadb.atlassian.net/browse/MDEV-5248
- Fixed upgrades from MySQL 5.1 -> MariaDB
- Parallel replication improvements

MariaDB 10.0.7

- Mostly bug fixes, to stabilise the code
- XtraDB 5.6 merged (InnoDB still default)
- OQGraph v3 - stores data on disk, persistent, larger graph support
- INFORMATION_SCHEMA.METADATA_LOCK_INFO plugin to see active metadata locks

PCRE Regular Expressions

- Powerful REGEXP/RLIKE operator
- New operators:
  - REGEXP_REPLACE(sub,pattern,replace)
  - REGEXP_INSTR(sub,pattern)
  - REGEXP_SUBSTR(sub,pattern)
- Works with multi-byte character sets that MariaDB supports, including East-Asian sets

Roles

- Bundles users together, with similar privileges - follows the SQL standard

```
CREATE ROLE audit_bean_counters;

GRANT SELECT ON accounts.* to
audit_bean_counters;

GRANT audit_bean_counters to
ceo;
```

MariaDB 10.0.9 (RC)

- InnoDB 5.6.15 (XtraDB default; InnoDB plugin)
- Extended keys optimization on by default
- MASTER_GTID_WAIT( ) + @@last_gtid
- TIME casted to DATETIME, date is CURRENT_DATE not 0000-00-00 - SQL standards compliant
  - @@old_mode=ZERO_DATE_TIME_CAST

MariaDB 10.0.10 (GA)

- audit plugin now ships
- XtraDB performance fixed incorrect calculation of flushed pages
- TokuDB compression is now TOKUDB_ZLIB
- Engine independent table statistics improved

What about MySQL 5.6?

- We love the fact that many features we’ve worked on for a long time are now in 5.6
- Optimizer enhancements
- Microseconds
- Binary log annotations
- Binary log group commit (10.0 has a newer faster version now)
- Precise GIS
- Threadpool

What are we missing from 5.6 currently?

- EXPLAIN output in JSON
- InnoDB memcached interface

today what do we have…

- 30 Jun 2014 - MariaDB 10.1.0
- 17 Oct 2014 - MariaDB 10.1.1
- 7 Dec 2014 - MariaDB 10.1.2
- 2 March 2015 - MariaDB 10.1.3
- 13 April 2015 - MariaDB 10.1.4 — late ;-)

Galera Cluster integrated

- Full integration of Galera Cluster 4 into MariaDB 10.1 — it won’t be a separate download
  - no lost transactions
  - optimisations for WAN replication
  - non-blocking DDL
  - no limits on transaction size
- `Server version: 10.1.3-MariaDB-wsrep MariaDB Server, wsrep_25.10.r4144`
- Granular monitoring in INFORMATION_SCHEMA — WSREP_MEMBERSHIP, WSREP_STATUS

Encryption

- Encryption: tablespace and table level encryption with support for rolling keys using the AES algorithm
  - table encryption — `PAGE_ENCRYPTION=1`
  - tablespace encryption — encrypts everything including log files (not the binlog)
- Overhead of ~10%
- XtraDB/InnoDB only; Aria for temporary tables
- New `file_key_management` now
- Pushed & documented — https://mariadb.com/kb/en/mariadb/documentation/managing-mariadb/securing-mariadb/encryption/table-and-tablespace-encryption/

Encryption II

- Table level encryption (Eperi)
  - must choose an `encryption_algorithm=aes_ctr` (or equivalents)
  - Have to use the key management plugin
- loading from filesystem? Insecure. You need a key management server (Eperi has one commercially)
- don’t forget to create keys!
  - eg. `openssl enc -aes-256-ctr -k mypass -P -md sha1`

Encryption III

```
CREATE TABLE customer (
    customer_id bigint not null primary key,
    customer_name varchar(80),
    customer_creditcard varchar(20))
ENGINE=InnoDB
page_encryption=1
page_encryption_key=1;
```

Encryption IV

- Tablespace encryption (Google)
  - again, you need to pick an encryption algorithm
  - specify what to encrypt: `innodb-encrypt-tables`, `aria`, `aria-encrypt-tables`, `encrypt-tmp-disk-tables`, `innodb-encrypt-log`
  - don’t forget key rotation:
    - `innodb-encryption-threads=4`
    - `innodb-encryption-rotate-key-age=1800`

Encryption V

- we also have tablespace scrubbing
  - background process that regularly scans through the tables and upgrades the encryption keys
  - specify in seconds when to scrub data — https://mariadb.com/kb/en/mariadb/xtradb-innodb-data-scrubbing/

Encryption VI

- 10.1.3 vs 10.1.4 have changes (incompatible)
  - The distinction between "tablespace encryption" and "page encryption" was removed, now there is only one single encryption feature.
  - Per table PAGE_ENCRYPTION_KEY was renamed to ENCRYPTION_KEY_ID.
  - Global variable innodb_default_page_encryption_key become a session innodb_default_encryption_key_id.
- Eperi code is mostly torn out. Per-table encryption implemented via Google's patches
- https://mariadb.com/kb/en/mariadb/table-encryption/

Optimistic parallel replication

- Before, transactions committed in parallel on the master could be run in parallel
- Now, more than one transaction will be considered to be run in parallel giving another performance boost in master-to-slave replication
- We have to check if this only will work with a 10.1 master
- Isn't fully pushed (or documented) yet — see: https://mariadb.atlassian.net/browse/MDEV-6676

Threadpool

- thread_pool_high_prio_mode
- thread_pool_high_prio_tickets
- MDEV-5533
- This brings in improvements made in Percona Server to the threadpool

InnoDB improvements

- Multi-threaded flush (also in 5.7, different implementation + we're first)
- Page compression (optimised for Flash, SSD, FusionIO)
- 64KB pages in InnoDB (old limit = 16KB).
- Defragementation (FB, backported by DaumKakao)
- Forced primary key
  - If option is true, create table without primary key or unique key where all keyparts are NOT NULL is not accepted. Instead an error message is printed.

InnoDB WebScaleSQL

- MDEV-6936: Buffer pool list scan optimisation
- MDEV-6929: Port Facebook Prefix Index Queries Optimization
- MDEV-6932: Enable Lazy Flushing
- MDEV-6931: Page cleaner should do LRU flushing regardless of server activity
  - fixes mysql#71988, mysql#70500
- DB-746 merge clustering key is covering key for mariadb 10 (TokuDB)
- MDEV-6933: Spurious lock_wait_timeout_thread wakeup in lock_wait_suspend_thread()
  - fixes mysql#72123

Per query variables

- Long history (http://www.bytebot.net/blog/archives/2014/05/04/per-query-variable-settings-in-mysqlpercona-serverwebscalesql)

-
  ```
  SET STATEMENT
  max_statement_time=1000 FOR
  SELECT name FROM name ORDER
  BY name;
  ```

Statement timeouts

- from Twitter patch; re-written by monty
- `MAX_STATEMENT_TIME` to abort long running queries
- We call it "query timeouts" + have a different syntax
- https://mariadb.atlassian.net/browse/MDEV-4427

Optimiser enhancements

- UNION ALL without temporary tables (5.7)
- Improve ORDER BY in optimiser
- Mostly there is EXPLAIN JSON (like 5.6)*
- EXPLAIN ANALYZE with FORMAT=JSON
  - includes data from the query execution itself — this is MariaDB only
  - https://mariadb.com/kb/en/mariadb/analyze-formatjson-examples/

EXPLAIN ANALYZE

```
MariaDB [information_schema]> explain format=json select * from
all_plugins\G
*************************** 1. row ***************************
EXPLAIN: {
  "query_block": {
    "select_id": 1,
    "table": {
      "table_name": "all_plugins",
      "access_type": "ALL"
    }
  }
}
1 row in set (0.01 sec)
```

WebScaleSQL

- WebScaleSQL improvements
  - https://mariadb.atlassian.net/browse/MDEV-6039
- Lots of running thru AddressSanitizer (ASan)
- Many of these also get backported to 10.0.13

Passwords

- Password validation plugin exists now
  - https://mariadb.com/kb/en/mariadb/development/mariadb-internals-documentation/password-validation/
- simple_password_check password validation plugin
  - can enforce a minimum password length and guarantee that a password contains at least a specified number of uppercase and lowercase letters, digits, and punctuation characters.
- cracklib_password_check password validation plugin
  - Allows passwords that are strong enough to pass CrackLib test. This is the same test that pam_cracklib.so does

Audit plugin improvements

- Monitor access, locate errors, etc.
- Connection — connect/disconnect/failed; Query — DDL/DML+TCL/DCL; Object — Database/Tables
- Passwords in 1.2 replaced by a placeholder (filtered, i.e. not in audit log)

CONNECT

- CONNECT having full JSON/BSON support
- Can read filename.json files with ease
- Writing — INSERT, UPDATE, DELETE is supported — however, if you have added/modified values for objects or arrays, there can be complications (similar like the XML type object issue)
- Works with Sveta's JSON UDFs as well

Other bits

- Slaves can execute triggers now
- Dump thread enhancements (remove binlog lock LOCK_log) from 5.7 included (Google)
- CREATE or REPLACE for most database objects minus indexes
- SET DEFAULT ROLE (there is a default role now for current user)

Other bits

- FRM files are now not created for temporary tables
- INFORMATION_SCHEMA.SYSTEM_VARIABLES - information for system variables
- Compiled with security hardening options (fortify source - https://mariadb.atlassian.net/browse/MDEV-5730)
- @@sql_log_slow can now be controlled on a session basis (not just globally)

GIS

- Full compliance for the OGC standards around GIS.
  - yes, we are missing a few functions, but its likely to improve
  - MDEV-4045 Missing OGC Spatial functions.
  - MDEV-60 Support for Spatial Reference systems for the GIS data.
  - MDEV-12 OpenGIS: create required tables: GeometryColumns, related views.
  - Speaking shortly, the MariaDB GIS part is now OpenGIS compliant, and passes all the OpenGIS conformance tests

Likely

- Kerberos authentication plugin
- Audit plugin to track password changes
- IPv6/IPv4 datatype (pending review)
- Additional character sets (GB18030) for Chinese govt mandate (pending review)

Compatibility

- Temporary tables are stored in Aria but now there is a —default-tmp-storage-engine option
- engine_condition_pushdown flag removed (its always on for engines that support it)
- --mysql56-temporal-format option to use the MySQL-5.6 low level formats to store TIME, DATETIME and TIMESTAMP types
- PERFORMANCE_SCHEMA disabled by default like in 10.0

Caveats (for 100% compatibility)

- GTID implemented differently from 5.6
- InnoDB memcached
- IS_IPV4(), IS_IPV6() + related functions
- validate_password plugin
- EXPLAIN FORMAT=JSON
- Optimizer trace
- Optimizer eq_ref
- mysqlbinlog streaming server

Support

- Five years from every release
- MariaDB 5.5 in Red Hat Enterprise Linux 7
- MariaDB 10 in SUSE Enterprise Linux 12

Benchmarks

- “Lies, damned lies, and statistics” - Mark Twain
  - http://blog.mariadb.org/sysbench-oltp-mysql-5-6-vs-mariadb-10-0/
  - http://dimitrik.free.fr/blog/archives/2013/02/mysql-performance-mysql-56-vs-mysql-55-vs-mariadb-55.html
- Yes, we’ve gotten Oracle to notice MariaDB :-)

Continued commitments

- Security
  - Since about two+ years now, we’re the go-to people for security - good track record
- We don’t like regressions
  - http://www.skysql.com/blogs/hartmut/nasty-innodb-regression-mysql-5525
  - http://www.skysql.com/blogs/kolbe/heads-no-more-query-cache-partitioned-tables-mysql-5523
- We care about backward compatibility & introduce features carefully
  - XtraDB innodb_adaptive_checkpoint=none|reflex|estimate|keep_average (no more reflex...)

We really care about quality

- Automated test suite run upon every push
- Better QA & code coverage
- MySQL test cases: 1,765
- Percona Server test cases: 1,837
- MariaDB test cases: 2,180

Powered By MariaDB

MariaDB deployed

happy users: **pap.fr**, **Paybox Services**, OLX, Jelastic, **Web of Trust**, Wikipedia, Craigslist, etc.

“MariaDB had these same bugs that we ran into with MySQL. However the big difference was that when we reported these bugs, they were quickly resolved within 48 hours!” -- Dreas van Donselaar, Chief Technology Officer, SpamExperts
B.V. after migrating over 300 servers from MySQL 5.0 to MariaDB 5.1.

“We made the switch on Saturday -- and we’re seeing benefits already -- our daily optimization time is down from 24 minutes to just 4 minutes” -- Ali Watters, CEO, travelblog.org

We upgraded the support.mozilla.org databases from Percona 5.1 to MariaDB 5.5. One of the engineers and I had a conversation where he mentioned that “one of our worst performing views on SUMO is doing waaaayyy better with the upgraded databases”, that it “seems more stable” and that “I stopped receiving ‘MySQL went away or disconnected emails’ which came in once in a while.” - Sheeri Cabral, Mozilla IT

“@nginxorg & @mariadb have helped me save $12000/year in infrastructure cost. I love it! Do more with less!” - Ewdison Then, CEO, Slashgear

“Migrating from MySQL 5.1 to MariaDB 5.2 was as simple as removing MySQL RPMs and installing the MariaDB packages, then running mysql_upgrade.” - Panayot Belchev, proprietor, Host Bulgaria on providing
MariaDB to over 7,000 of their web hosting customers.

Resources

- http://mariadb.org/
- http://kb.askmonty.org/ (or https://mariadb.com/kb/en/)
- http://planet.mysql.com/

Books!

1. MariaDB Crash Course, Ben Forta (September 2011)
2. Getting Started with MariaDB, Daniel Bartholomew (October 2013)
3. MariaDB Cookbook, Daniel Bartholomew (March 2014)
4. Real MariaDB, Matt Lee (April 2014)
5. Building a Web Application with PHP & MariaDB: A Reference Guide, Sai Srinivas Sriparasa (June 2014)
6. MariaDB: Beginners Guide, Rodrigo Ribeiro (August 2014)
7. Mastering MariaDB, Federico Razzioli (September 2014)
8. MariaDB High Performance, Pierre Mavro (September 2014)
9. Learning MySQL & MariaDB, Russell Dyer (April 2015)

Q&A / Enjoy your evening

colin@mariadb.org

slides: slideshare.net/bytebot

http://bytebot.net/blog/ | @bytebot on twitter

http://mariadb.org/ | http://mariadb.com/