



## The MySQL-MariaDB story

Haaga-Helia, September 4, 2013  
Michael “Monty” Widenius  
Entrepreneur, MariaDB hacker  
[monty@mariadb.org](mailto:monty@mariadb.org)  
<http://mariadb.org/>

# Questions addressed in talk



- 1) Brief MySQL & MariaDB history
- 2) What challenges did you face in the beginning and how did you overcome them? Eg Convincing early employees to join you, raising seed & venture capital etc.
- 3) How did you handle the growth of your venture?
- 4) Liquidity event.
- 5) End of story (or is it?)
- 6) Where is MariaDB today
- 7) Features of the different MariaDB releases
- 8) Challenges faced while creating MariaDB
- 9) Reasons for the success

# The origin of My (SQL)



At start: Lots of traveling and meeting interesting people

# The origin of My (SQL)



Combined with working from home

# The origin of My (SQL)



Things were good

# The origin of Max (DB)



We also made a MaxDB (based on SAP DB) and MySQL-max

# The origin of My (SQL)



Even if there were some growing pains

# The origin of My (SQL)



Then we came into strange company



# The origin of My (SQL)



Which scared some of us a bit...

# The origin of My (SQL) and Maria (DB)



Fortunately there is someone else that can continue

# The origin of My



While My continues to travel

# The origin of My



Free as a bird

# The origin of Maria (DB)



But is the next generation up to it?

# The origin of Maria (DB)



It's a hard job taking over a success

# The origin of Maria (DB)



Like pulling out a rabbit from a hat

# The origin of Maria (DB)



But we are confident we can pull it off



# The origin of Maria (DB)



So lets be happy that...

# The origin of Maria (DB)



We still can put our trust in the creators of MySQL

# (Very brief) MySQL history



- Unireg (base of MySQL code) was started 1981.
- MySQL released December 1995 under dual licensing.
- MySQL Finland Ab took in investment and hired a new CEO, Mårten Mickos, in 2001.
- Made an agreement with SAP and released MaxDB 2003
- Oracle bought the InnoDB engine 2005.
- MySQL Ab was sold to Sun in March 2008 for 1 billion \$.
- Monty & others left Sun in Feb 2009 to work on Maria engine.
- Oracle started to acquire Sun (including MySQL) in April 2009.
- Original MySQL developers starts focusing on MariaDB.
- MariaDB foundation was created in 2012.
- MariaDB is replacing MySQL in most distributions in 2013.

# Why MySQL was released as Free Software



- David Axmark and Monty had been using Free software for 10 + years and wanted to give something back
- MySQL was our first program suitable for wider usage
- We earned money mainly by doing software development and consulting:
  - **Releasing MySQL under open source would not harm our income**
- We choose to do **dual licensing** to be able to work full time on MySQL
  - Second project with dual licensing (ghostscript was the first)
  - After 2 months we where profitable and could spend all time on developing and spreading MySQL.
- Nowadays I am advocating **Business Source** to companies who wants do create open source products but can't do dual licensing. See <http://monty-says.blogspot.com/>

# The challenges of growing



- During 1995 – 2000 we grow from 2 to 15 people
  - All developers; No sales people, marketing people or lawyers
  - MySQL Ab was a virtual company (no offices) from day 1
- Getting the MySQL name out
  - We were very lucky; Web was emerging and everyone needed a free database
  - PHP & Perl developers were spreading the news about MySQL
  - Monty wrote more than 30,000 emails helping people with MySQL
  - David was visiting conferences; Spent probably more time on airplanes than home
- After the 3'rd year we started to be approached by investors
  - Best early offer was 50M USD in 1999
  - However, we didn't want to sell the whole company and “lose **all** control” at once
- We spent 5 years developing MySQL until “good enough”

# Why the MySQL founders took in investors



- Internal challenges to manage everything with 15 people
  - We needed more people to handle support, documentation, administration, HR and sales
- Customers and users were demanding more features
  - We needed to hire more developers
- Increased competition from databases like PostgreSQL

Three choices:

- Continue as before and grow very slowly
- Sell MySQL (we had already said no to 50M USD)
- Take in investors to:
  - Hire a CEO, Mårten Mickos, to drive the company
  - Get more resources to do more development
  - Get more users and more customers

We decided the last one was the best for MySQL long term

# The Sun deal

## How was it made and what happened at Sun



- When you take in investors, you get money but loose control
  - Company will go either public or get sold
- The internal decision was to go public in 2008
- MySQL Ab had started to fall apart internally since 2005
  - Big chasm between managers in Cupertino and “the rest”
  - Monty + group of developers was about to leave in Dec 2007
- MySQL Ab got several offers and Mårten decided we should sell to Sun instead of going public. (I got to know this in Dec 2007)
- Sun deal was, in my opinion, much better than going public:
  - Sun did understand development
    - Should be able to fix MySQL development organization
  - Sun did understand and promote Open Source
    - No risk of making part of MySQL closed source

# Why MariaDB was created



## “Save the People, Save the Product”

- To keep the MySQL talent together
- To ensure that a free version of MySQL always exists
- To get one community developed and maintained branch
- Work with other MySQL forks/branches to share knowhow and code

After Oracle announced it wanting to buy Sun & MySQL this got to be even more important.



# Monty Program Ab



- Started in February 2009 after Monty's exit from MySQL/SUN
- Shifted focus from (M)aria storage engine to MariaDB (A branch of MySQL) after Oracle acquired Sun.
- Did drive (but doesn't own) the MariaDB development
- Founding member of the Open Database Alliance (ODBA)
- “Virtual company” (no offices) with about 20 employees all over the world. All original 'core' developers of MySQL are employed. (Full optimizer team, 3 of 4 MySQL architects, etc)
- Very technical company (only development, open source consulting and L3 (bug fix & advanced) support)
- Used the Hacking Business Model ('Company is owned by the employees')
- Merged with SkySQL in April 2013

# Role of SkySQL Ab



- Gives 24/7 enterprise level support for MariaDB and MySQL
- Driver of new MariaDB development and infrastructure.
  - Provides paid for development on MariaDB
- Home for most of the core MariaDB developers and some of the best from the original MySQL support organization.
- Will create tools for better monitoring and manage MariaDB in the cloud.

# MariaDB Foundation Overview



The Foundation is the new driver of the MariaDB project

Custodian of the code, Guardian of the community

Foundation can never to be controlled by a single entity or person

# MariaDB Foundation Goals



That MariaDB be actively developed in the community and to:

- Increase adoption of MariaDB
- Ensure sustainable high-quality efforts to build, test and distribute MariaDB
- Ensure that community patches are reviewed and adopted
- Guarantee a community voice
- Keep MariaDB compatible with MySQL
- Maintain mariadb.org.

The MariaDB foundation now employes:

- Management team: CEO, CTO, lawyer
- 3 full time MariaDB developers (including the CTO)
- 1 documentation writer



**More members and sponsors are  
welcome!**

**If you care about the future of the MySQL  
ecosystem, please contact us and  
ask how you can get involved!**

**Simon Phipps, CEO, [simon@mariadb.org](mailto:simon@mariadb.org)**

**Michael Widenius, CTO, [monty@mariadb.org](mailto:monty@mariadb.org)**

# *MariaDB server is a branch of MySQL*



- User level (data, API, replication..) compatible with MySQL
  - Drop in replacement
  - More plugins, more features, faster, better code quality.
- GPL-only server license.
- LGPL C and Java connectors.
- More open development
  - Source in public repository on launchpad
  - Active external contributors
  - All development plans public on [mariadb.com](http://mariadb.com)

# *MariaDB server releases*



- MariaDB 5.1 was released as stable in February 2010
- MariaDB 5.2 was released as stable in November 2010
- MariaDB 5.3 was released as stable in April 2012
- MariaDB 5.5 was released as stable in April 2012
- MariaDB 10.0 was released as alpha in November 2012
  - Goal is to be released as beta in September
- MariaDB-Galera (multi-master) was released as stable (GA) in February 2013 after a lot of testing.

# *The MariaDB releases*



- MariaDB 5.1 (based on MySQL 5.1)
  - Better build & test system, code cleanups, community patches, new storage engines, table elimination.
- MariaDB 5.2 (based on MariaDB 5.1)
  - Community features that did not go into 5.1:
    - Virtual columns
    - Extended User Statistics
    - Segmented MyISAM key cache (faster multi user!)
- MariaDB 5.3 (based on MariaDB 5.2)
  - Optimizer features (faster subquerier, joins etc)
  - Microsecond, dynamic columns, faster HANDLER etc.
  - Better replication (group commit, more options)
- MariaDB 5.5 (based on MariaDB 5.3 and MySQL 5.5)



# Feedback plugin



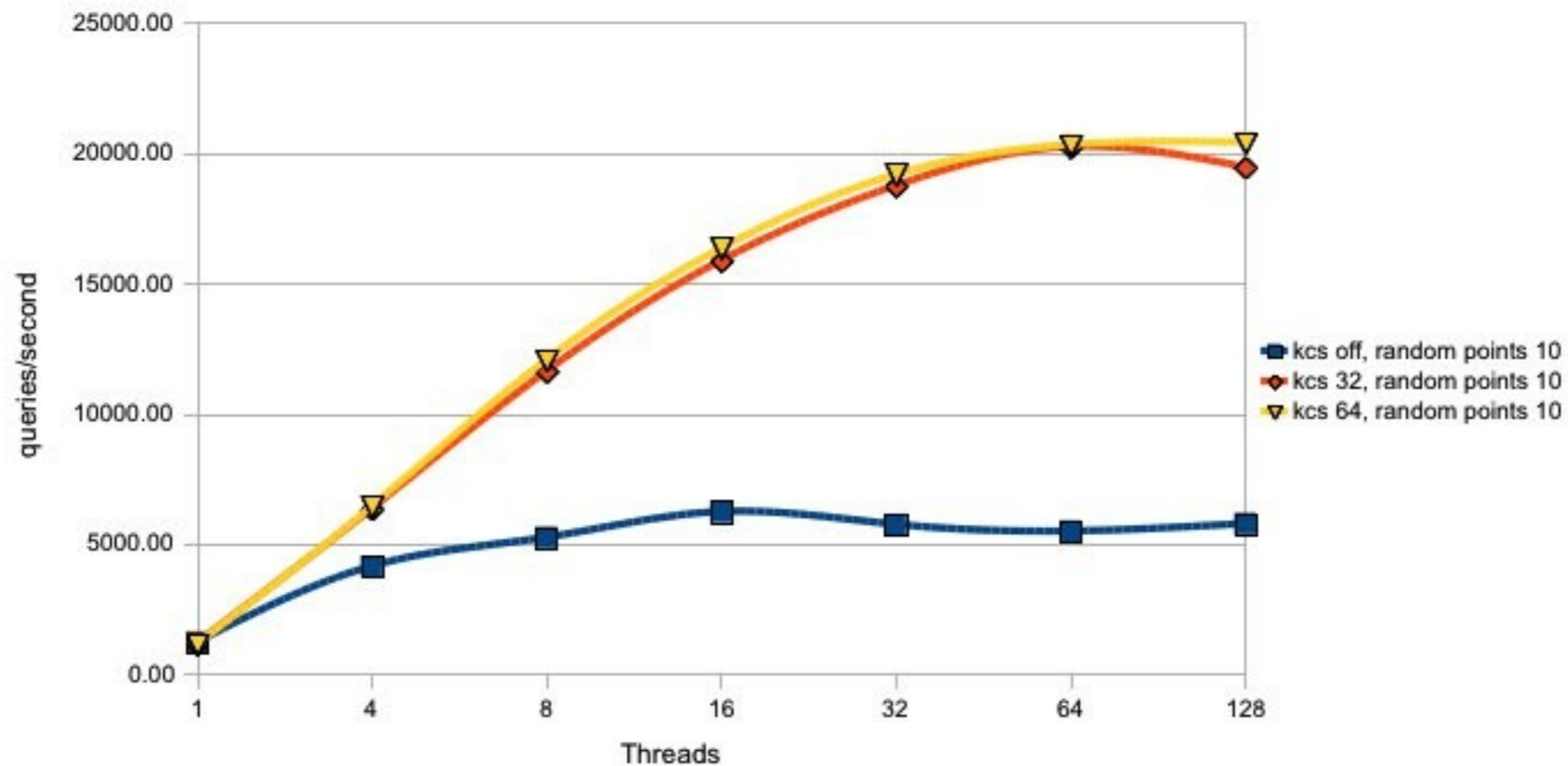
- All recent MariaDB versions has the feedback plugin
- Enable by adding “plugin-load=feedback.so” and “enable-feedback” to the [mysqld] section in my.cnf.
- Feedback plugin will automatically send a report (basically SHOW STATUS)a few minutes after a startup and once a week
- This information is used to decide what features should be developed/expanded upon
- For more information see <https://mariadb.com/kb/en/feedback-plugin/>
- For statistics see [http://mariadb.org/feedback\\_plugin/](http://mariadb.org/feedback_plugin/)
- In August 2013 some 3000 people has it enabled

# Major new features in MariaDB 5.2



- **SphinxSE**: Text search within MariaDB
  - Built-in Sphinx client which allows MariaDB to talk to searchd, run search queries, and obtain search results.
- **Virtual columns**
  - Columns that are an expression and are calculated on retrieval.
- **Extended User Statistics**
  - Client, User, Index and Table statistics.
- **Segmented MyISAM key cache** (see separate slide)
- **Pluggable Authentication**
- Storage-engine-specific CREATE TABLE
- Very fast 'copying to temp table' phase (speeds up GROUP BY and other complex queries).
- **Group commit** & better recovery for the **Aria** engine.
  - Speeds up multi-user inserts.

# MyISAM Segmented key cache



- Blue line is without segmented key cache.
- Solves one of the major read bottlenecks for MyISAM
- We see up to 250% performance gain depending on the amount of concurrent users.
- Fix applies to all MyISAM usage with many readers!

# What's new in MariaDB 5.3



This is the biggest redesign of the MariaDB optimizer in 10 years and it will finally makes subqueries usable in MariaDB.

- Faster subqueries
  - Back porting and extending subquery optimization from MySQL 6.0
  - No materialization for many kinds of subqueries or VIEW's in the FROM clause. `SELECT * from (SELECT ....)`
  - Caching of subquery results

In applicable cases, you can get 10x – 100x speedups.

- Faster joins (of big tables) thanks to
  - Multi-Read-Range (MRR) access (better than in MySQL 5.6)
  - Batch key access (BKA)
  - Index condition pushdown
  - Classic Hash joins

# What's new in MariaDB 5.3



- Microsecond support for NOW(), CAST() and timestamp, time, and datetime columns.
- Windows speed improvements
- Asynchronous IO in XtraDB is redesigned and is now faster, due to the use of IO completion ports.
- A new Windows MSI installer. Includes a GUI-tool, HeidiSQL.
- Lots of new status variables that helps finds out what's wrong
- Progress reports for ALTER TABLE and LOAD DATA INFILE (and some other admin commands)

# What's new in MariaDB 5.3



Some common sub queries that are now significantly faster:

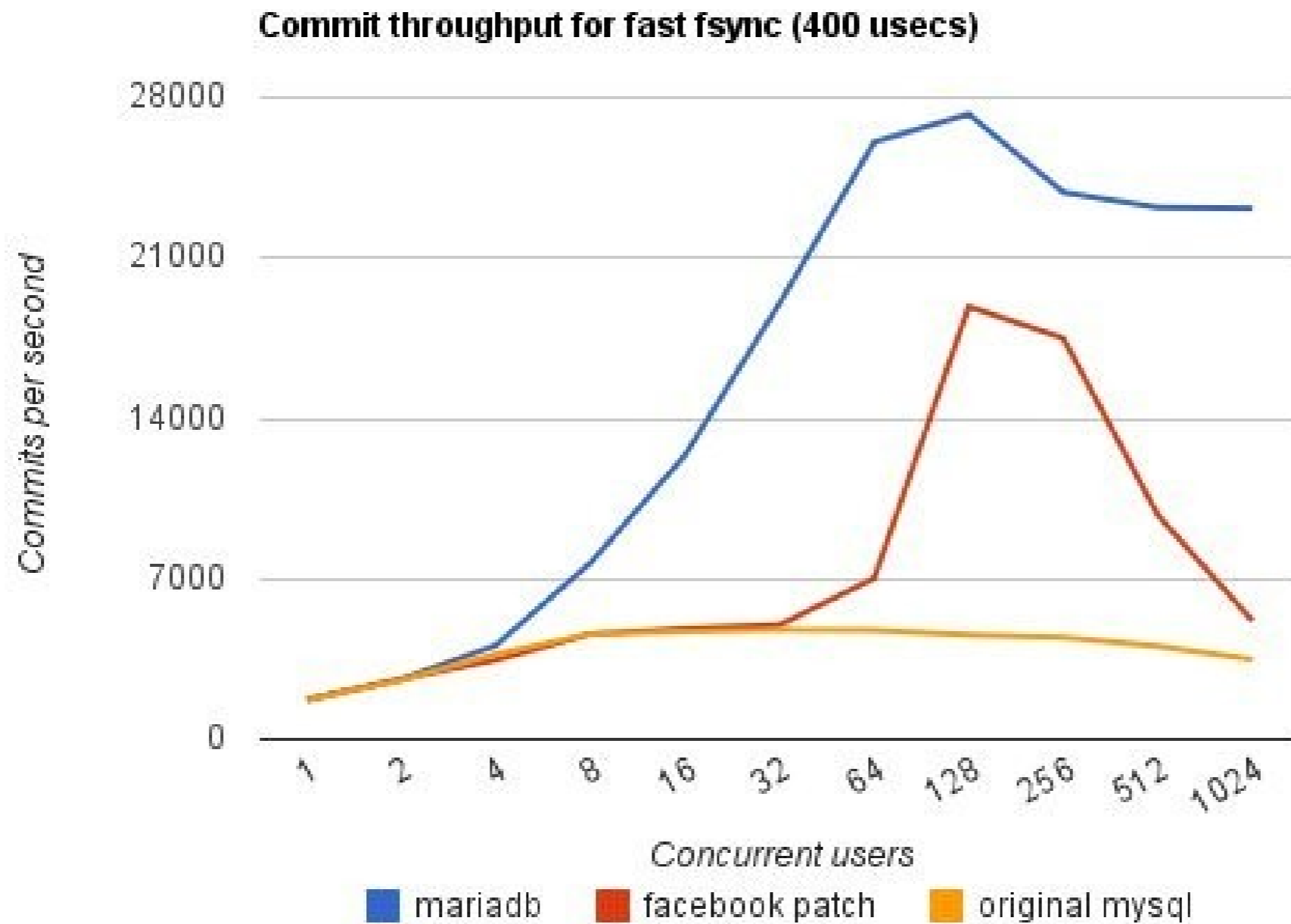
- No materialization or materialization with keys:
  - `SELECT anything FROM (SELECT ....) AS a WHERE a=...`
- Caching of common values (Good if `outer_ref` has a few values)
  - `SELECT (SELECT ... WHERE outer_ref=xxx) FROM ...`
- Transformations
  - `SELECT * FROM big_table WHERE big_table.col IN (SELECT anything FROM small_table) ->`  
Reorders `SELECT`:s to use sub query as driving table
- Materialization with keys in temporary table also for `WHERE`
  - `SELECT ... WHERE a [NOT] IN (SELECT not-a-key ...)`

# What's new in MariaDB 5.3



- Full **microsecond** support. This includes `TIMESTAMP`, `TIME`, `DATETIME` types, `NOW()` and all `CAST` and `TIME` related functions, replication etc.
- **Group commit** between binary log and storage engines
  - **FASTER** and safer replication
- Progress report (with `PUSH` method) for `ALTER TABLE`, `LOAD DATA INFILE`, `REPAIR`, `OPTIMIZE & ANALYZE`.
- Precise GIS operations.
- Windows installer that includes Windows GUI-tool (HeidiSQL)
- Lots of small optimizations, code cleanups, better error messages and bug fixes.
- For full list, see <http://kb.askmonty.org/en/what-is-mariadb-53>

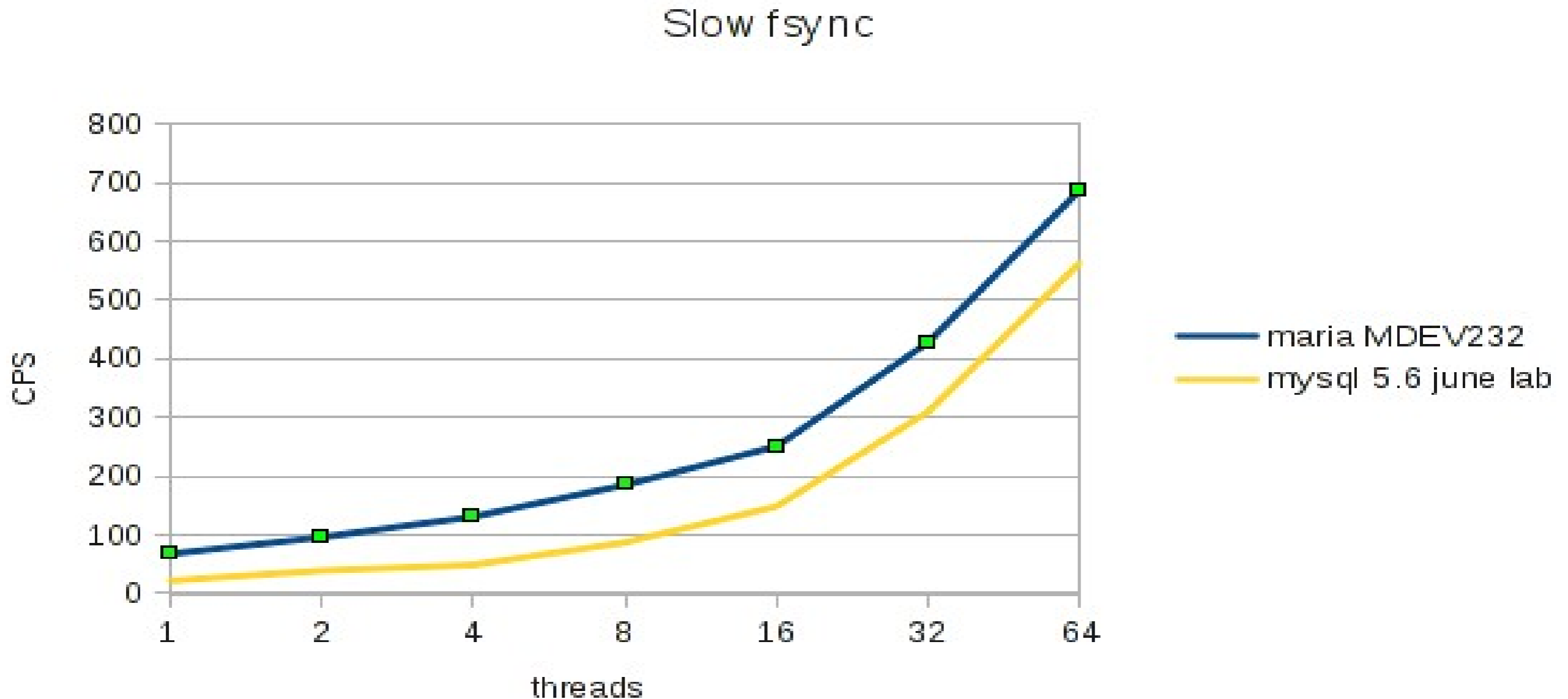
# Group commit, verified



Source: Marc Callaghan's facebook blog for a server with 400 microsecond fsync latency



# Benchmark of new group commit Comparison with MySQL 5.6



- MySQL 5.6 also removes checkpoint-per-commit
- More complex thread scheduling, less grouping of commits
- Does not remove stall on LOCK\_log, less grouping of commits

# MariaDB 5.3 and NoSQL



The main reasons for using NoSQL are:

- Handling of unstructured data (not everything is table and fixed number of columns)
- Faster replication (usually with 'unconventional' shortcuts)
- The same way MySQL with it's storage engine interface can handle both transactional and datawarehousing , we are extending MariaDB to be a bridge between SQL and NoSQL.
- MariaDB 5.3 has now even better “NoSQL” support:
  - 50 % Faster HANDLER commands; HANDLER READ now also work with prepared statements.
  - HandlerSocket compiled in (Direct access to InnoDB)
  - Dynamic columns (each row can have different set of columns)

# SQL doesn't solve all common problems



The (web) store problem:

All items needs: ID, Type, Price, Country, Manufacturer)

A T-Shirt has the following additional properties:

Size, color...

A computer has the following additional properties:

CPU, MHz, memory, Watt...

There is no easy way to store many different types into a relational database!

(It will not work by having one table/types as joins becomes impossible to manage).

# Dynamic columns in MariaDB 5.3



- With dynamic columns all extra columns are stored in one or many packed blobs, maintained by the database.
- You can instantly add more columns, remove or query them for a row.
- You can access columns in the server or retrieve the full blob to the client and manipulate it there.
- You can use virtual columns to create indexes on some values.
  - True indexes for dynamic columns is planned for later.
- Implemented through functions to enable use by ODBC etc.
- First implementation uses an integer to access columns.
  - In MariaDB 10.0 columns can be accessed by name.

# Dynamic columns in MariaDB 5.3



- Simple set of functions (available in server and client):
  - `COLUMN_CREATE(column_nr, value, [column_nr,...])`
  - `COLUMN_ADD(blob, column_nr, value, [column_nr,...])`
  - `COLUMN_DELETE(blob, column_nr, column_nr...);`
  - `COLUMN_EXISTS(blob, column_nr);`
  - `COLUMN_LIST(blob, column_nr);`
  - `COLUMN_GET(blob, column_nr, type);`

As a proof of concept we have in MariaDB 10.0 created storage engines for **Cassandra** and **LevelDB** where we use dynamic columns as a bridge.

We are working on doing the same with MongoDB.

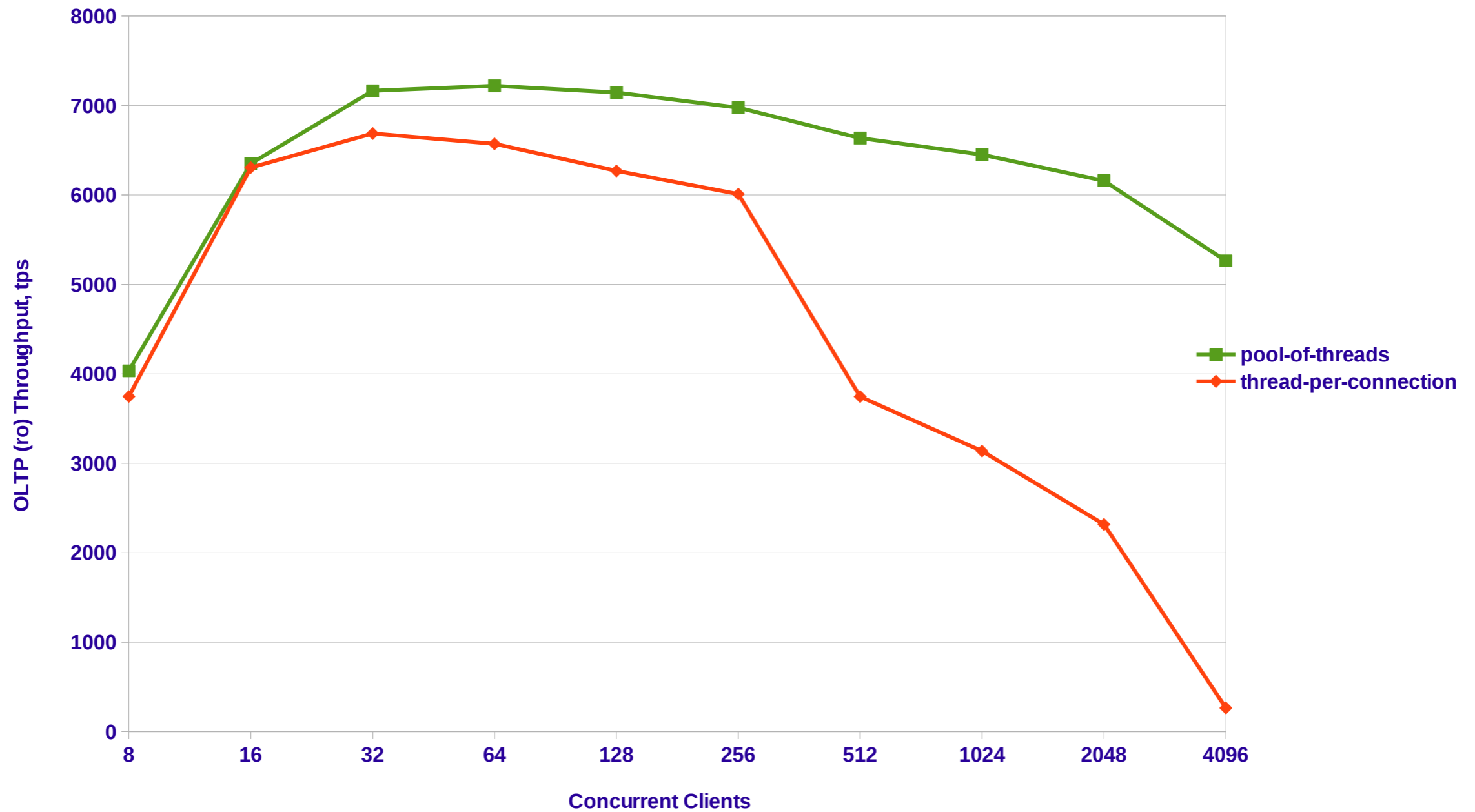
# New thread pool for 5.5



Thread pools solves a couple of problems:

- Allows you to limit the number of worker threads at your machines peak performance.
- More fair scheduler; Less query time distribution
- If too many queries, machine can run at 1% of peak performance
  
- New tread pool for 5.5: (Tested with 24 CPU's):
  - Always better on Windows
  - Better one Linux than thread-per-connection after 1024 connections
  - Much less performance degradation when more connections (60% performance instead of 1%)

# New thread pool for 5.5



# Optimizations comparison



Feature	MariaDB 5.3/5.5	MySQL 5.5	MySQL 5.6
Index Condition Pushdown (ICP)	Yes		Yes
Disk-sweep Multi-range read (DS-MRR)	Yes		Yes
DS-MRR with Key-ordered retrieval	Yes		
Index_merge / Sort_intersection	Yes		
Cost-based choice of range vs. index_merge	Yes		
ORDER BY ... LIMIT <small_limit>	(In 10.0)		Yes
Use extended (hidden) primary keys for innodb/xtradb	5.5		
Batched key access (BKA)	Yes		Yes
Block hash join	Yes		
User-set memory limits on join buffers	Yes		
Apply early outer table ON conditions	Yes		
Null-rejecting conditions tested early for NULLs	Yes		



# Optimizations comparison



Feature	MariaDB 5.3/5.5	MySQL 5.5	MySQL 5.6
Subquery: In-to-exists	Yes	Yes	Yes
Subquery: Semi-join	Yes		Yes
Subquery: Materialization	Yes		Yes
Subquery: NULL-aware Materialization	Yes		
Subquery: Cost choice of materialization vs. in-to-exists	Yes		
Subquery: Cache	Yes		
Subquery: Fast explain with subqueries	Yes		
Delayed materialization of derived tables / materialized views	Yes		Yes
Instant EXPLAIN for derived tables	Yes		Yes
Derived Table with Keys optimization	Yes		Yes
Fields of merge-able views and derived tables used in equality optimizations	Yes		

# Optimizations comparison



Feature	MariaDB 5.3/5.5	MySQL 5.5	MySQL 5.6
LIMIT ROWS EXAMINED rows_limit	5.5		
Systematic control of all optimizer strategies	Yes		Partial
Explain for DELETE, INSERT, REPLACE, and UPDATE			Yes
EXPLAIN in JSON format			Yes
More detailed and consistent EXPLAIN for subqueries	Yes		

# Why MariaDB 10.0



- MariaDB 5.5 already have most (+ a lot more) of the optimizer features of MySQL 5.6
- MariaDB 5.5 is already a superset of MySQL 5.5. MySQL 5.6 has only a fraction of the MariaDB 5.5 new features.
- A full merge of MySQL 5.6 into MariaDB 5.6 is a one year project as a lot of the code has to be completely rewritten.
  - Features and usable code are removed, either intentionally or by mistake
  - New code is way to complex (you can do same thing much simpler)
  - Lots of new introduced bugs we have to get rid of.
  - It's clear that some of the new MySQL programmers doesn't understand the current code (see Kristian Nielsen's blog)
  - A lot of the new code is re-factoring we don't want to have.
- Better to do the merge in 2 steps into 10.0, and 10.1
- MariaDB 10.1 will have all important features of MySQL 5.6

# MariaDB 10.0



MariaDB 10.0 is MariaDB 5.5 + some features from MySQL 5.6 + some new features

Features back ported from MySQL 5.6:

- All InnoDB changes (done)
- Performance schema changes (done)
- Read only transaction (significant InnoDB optimization) (done)
- Online ALTER TABLE (done)

Features from MySQL 5.6 that are reimplemented:

- Better error message (with system error string) (done)
- NOW() as default value for datetime (done)
- Global transaction ID for replication (done)
- Parallel replication (Base implementation exists)

# MariaDB 10.0



## New features:

- `SHOW EXPLAIN` (see what other thread is doing) (done)
- Multi source (one slave can have many masters) (done)
- Faster `ALTER TABLE` with `UNIQUE` index (done)
- `DELETE ... RETURNING` (done)
- Even faster group commit (done)
- Storage engine for **Cassandra** (done)
- Storage engine for **Leveldb** (done)
- Storage engine Connect (done)
- Storage engine TokuDB (In 10.0.5 and 5.5.33)
- Per thread memory usage (done)
- Much better GIS support (done)

For full list, see <http://kb.askmonty.org/v/plans-for-10x>

# What are planned for MariaDB 10.x ?



This list of proposed features is still work in progress, and depends on the community and customers.

- GIS
  - OpenGIS compliance (largely done). MariaDB can already work with OpenGIS data!
  - Deeper integration of GIS with optimizer
- More online operations
  - Analyze table
- Compatibility & usability
  - IPv6 type
  - Query logging and summary per query
  - Audit for specific users

# What is planned for MariaDB 10.x ?



- Replication
  - Extend group commit to have on sync per group commit
  - ALTER TABLE's can be applied in parallel
- Statistics and monitoring
  - Better EXPLAIN
  - Persistent table statistics
  - Log all SQL errors
  - Progress indicator for SELECT

# What is planned for MariaDB 10.x ?



- Optimizer
  - Implement UNION ALL without usage of a temporary table
  - Grace HASH join and Sort merge join (**Need sponsors**)
- Performance
  - Better multi CPU performance above 16 cores
  - More scalable query cache under higher concurrency
  - Faster VIEW (don't open & parse view for every query)
- Easy of use
  - VARCHAR and BLOB support for memory tables
  - Table functions
- GET DIAGNOSTIC

For full list, see <https://mariadb.com/kb/en/plans-for-10x>



# New LGPL client libraries



- LGPL client libraries for C and Java
  - C is based on the LGPL library from MySQL 3.23
  - API compatible with latest MySQL client libraries.
  - Java is based on the drizzle driver.
  - LGPL ODBC driver is under development.
- Works with MariaDB, Percona server, MySQL and drizzle
- Developed by Monty Program Ab and SkySQL.
- Announced and released 2012-10-29

You can download these from  
<https://downloads.mariadb.org/>

# MariaDB and TokuDB



MariaDB and Tokutek have agreed to make TokuDB a native plugin in MariaDB 5.5.33 and MariaDB 10.0.5.

This means that the official MariaDB binary will be able to dynamically load the TokuDB storage engine directly (no patches needed for MariaDB).

TokuDB will be added to the MariaDB buildbot test suite to ensure that the combination is properly tested on all supported platforms.

Note that the TokuDB version for MariaDB 5.5 will not have all online features that the original TokuDB version from [www.tokutek.com](http://www.tokutek.com) as we didn't want to do big changes in 5.5. The online features will be in 10.0.

## About TokuDB

- TokuDB uses Fractal Tree® indexing to improve insert and query speed, compression, replication performance, and online schema flexibility.
- TokuDB is created by Tokutek Inc. See [www.tokutek.com](http://www.tokutek.com) for details.

# Connect storage engine



MariaDB 10.0.2 includes the Connect storage engine by Olivier Bertrand.

With the connect storage engine you can read, write and update files in a lot of different storage formats:

- Various fixed and dynamic text formats
- .DBF (dBASE format)
- .CSV
- .INI
- .XML
- ODBC ; Table extracted from an application accessible with ODBC; You can for example connect MariaDB to an Oracle database with this.

# Challenges with forking MySQL (1)



- Creating a team that can continue and take over MySQL
- Creating free documentation & forums (askmonty.org)
- Creating a free build & test environment (buildbot)
- Competing against a well know trademark (MySQL)
  - Visiting most open source trade shows
  - Working with OS distributions to get MariaDB
- Keeping up with MySQL development
  - Lots of bugs found while doing monthly merges
  - Merging MariaDB 5.3 and MySQL 5.5 took 6 months
  - Adding new “must have” features (in MariaDB 5.3/5.5)
- Creating a developer community
  - Relatively easy as Oracle is not working with the community to get in their patches or handle their bugs

# Challenges with forking MySQL (2)



- Finding a business model not based on licensing
  - Developer support (for advanced MySQL users)
  - Third level support via partners like SkySQL
  - Getting paid for adding features to MariaDB/MySQL
- No paying customers for the first 3 years
  - All major paying customers bought 3-5 year contracts to protect against price increases from Oracle
- Things changed after the MariaDB foundation was created and Monty Program merged with SkySQL.

# Free MariaDB/MySQL/SQL99 documentation



The screenshot shows the MariaDB Knowledge Base website. At the top, there is a navigation bar with links for 'KB Home', 'Downloads', 'Open Questions', 'MariaDB', and 'All Topics'. A search bar and language selector are also present. The main content area is divided into several sections:

- MariaDB**: A list of links including 'MariaDB Documentation', 'About MariaDB', 'About the MariaDB Foundation', 'MariaDB community', 'MariaDB versus MySQL - Features', 'MariaDB versus MySQL - Compatibility', 'MariaDB Case Studies', 'What is in the different MariaDB releases?', 'Recommended MariaDB / MySQL Books', 'Development', and 'More...'
- SQL-99 Complete, Really**: A list of links for SQL-99 topics, including '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', and 'More...'
- Latest Articles**: A list of article titles such as 'sql99-banner', 'ST\_BUFFER', 'ST\_SYMDIFFERENCE', 'ST\_INTERSECTION', 'ASWKT', 'ST\_UNION', 'ST\_OVERLAPS', 'ST\_TOUCHES', 'ST\_EQUALS', and 'ST\_INTERSECTS'.
- Featured Content**: A list of links including 'MariaDB community', 'News and Information', 'About the AskMonty.org Knowledgebase', 'Contributing to the AskMonty Knowledgebase', 'Licensing FAQ', 'MariaDB Case Studies', 'Distributions Which Include MariaDB', and 'Recommended MariaDB / MySQL Books'.
- Welcome to the MariaDB Knowledgebase**: A section with links for 'About', 'Editing Help', 'RSS Feeds', 'Articles [en]', 'Questions [en]', and 'Comments [en]'.
- Current Releases**: Two buttons for 'Download 10.0.4 Alpha' (released on 2013-08-16) and 'Download 5.5.32 Stable' (released on 2013-07-18).
- What's happening now?**: A section with statistics for the past week, past month, and since launch, including metrics like 'Added/Changed Articles', 'Freenode #maria' activity, 'Launchpad Activity', and 'List Emails'.
- Social Media**: Buttons for 'Tweet', '+1', and 'Like' with zero counts.

At the bottom, there is a footer with links for 'Recent Changes', 'Contact Us', 'About', 'Legal', and copyright information: '©2010 - 2013 Monty Program Ab'.

The knowledgebase allows you to:

- Find answers to your problems
- Ask questions and get answers from others
- Add your own documentation or help with translations

# There are a lot of others involved

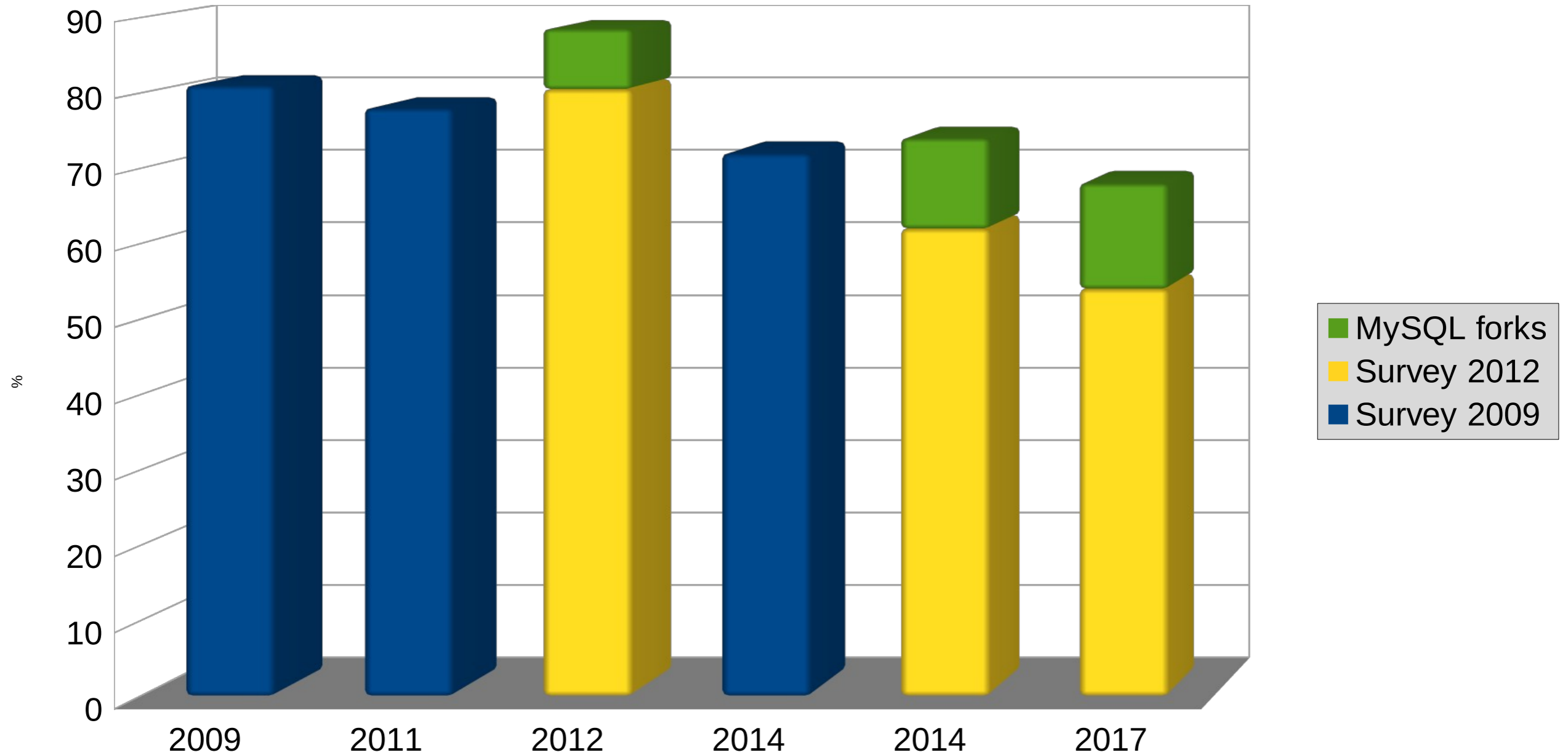


- Most features in MariaDB 5.2 were contributed by the community!
- Many of the advanced features in MariaDB 5.3 are sponsored features
- In the mariadb.com knowledge base (free MariaDB and MySQL documentation) we have now 3000 (mostly English) articles

Statistics from the past month (August 2013):

- Added/Changed Articles: 513
- On Freenode #maria, 487 people wrote 7904lines
- Launchpad Activity:
  - 24 active branches
  - 343 commits
- Hundreds of thousands of downloads of MariaDB. Probably >> 1M users
- We have seen companies converting hundreds of machines to MariaDB in a few days without any problems.
  - See <http://mariadb.com/kb/en/mariadb-case-studies>
- Distributions are moving from having included MariaDB to make it default.

# Predicted MySQL usage



Based on 451 Research surveys  
Sample 285/205 users



# MariaDB popularity is increasing



- In December 2012
  - Wikipedia announced they are moving to MariaDB.
- In January-March 2013
  - DB at Mozilla blogged they have moved to MariaDB
  - Fedora voted 7-0 to make MariaDB the default MySQL database on Fedora.
  - OpenSuse 12.3 now includes MariaDB as default.
  - Slackware, Chakra Linux and Arch Linux have announced that they have made MariaDB default.
- In April 2013
  - Google is basing their new SQL offerings on MariaDB
  - FusionIO is showing benchmarks with MariaDB.
- June 2013
  - RedHat will include MariaDB in RedHat Enterprise

# Reasons to switch to MariaDB today



- MariaDB has 20 man years of more development than MySQL (and the gap will continue growing).
- MariaDB is maintained by the people that originally created MySQL and has the best knowledge of the MySQL code.
- MariaDB is binary compatible (data and API) with MySQL, so its trivial to replace MySQL with MariaDB (minutes).
- Reasons to switch to MariaDB
  - Faster queries thanks to XtraDB (InnoDB plugin fork from Percona), TokuDB, a much better optimizer, better replication and better code.
  - Open source development: Anyone can be part of the development at all stages. Developer meetings are public!
  - More features, including critical ones like microseconds, better statistics, dynamic columns and Cassandra storage engine.
  - Less risk as MariaDB will not remove features like MySQL is doing (thread pool, storage engines, safemalloc (developer feature), older OS, No RedHat 6.0 support etc)

# Summary: What made MySQL successful?



- We were using it (for data warehousing and web)
- Internet was new and everyone needed a web-optimized database
- “Virtual company” made it easy to find good people
- New “free” license scheme (this was before Open Source)
  - Free for most, a few have to pay
  - Second program (ghostscript was first) to use dual licensing, MySQL first to do it with GPL.
- Very easy to install and use (15 minute rule)
  - Released source and tested binaries for most platforms
- Friendly and helpful towards community
  - I personally wrote 30,000+ emails during the first 5 years to help people with using MySQL
- Waited with investments until product was “good enough”
  - Needed, stable and easy to use product with right price

# Questions ?



For questions later, use the public MariaDB email list at [maria-discuss@lists.launchpad.net](mailto:maria-discuss@lists.launchpad.net) or [#maria](#) on [Freenode](#).

For questions regarding the MariaDB foundation:  
[monty@mariadb.org](mailto:monty@mariadb.org)