About FromDual GmbH

Enterprise Support
codership

MariaDB

MySQL

Training

Consulting

GALERA CLUSTER

remote-DBA

/CH/Open

OSB Business Alliance

Oracle Silver Partner
MariaDB 10.4 – New Features

- Branches and Forks
- MariaDB and Linux Distributions
- Authentication
- InnoDB
- MariaDB Optimizer
- Application-Time Period Tables
- General Stuff
- Backup Stage
- Galera 4
- Outlook MariaDB 10.5
Branches and Forks

• MariaDB and MySQL are Open Source (GPL v2)
  • This means everybody is allowed to DiY
  • → Branches and Forks

• Who uses a MariaDB/MySQL in here?
  • Chaos will happen! :-(
  • See Sybase ASE vs. Microsoft SQL Server (1995(v6.0)-2005)
  • Examples: GTID, Protocol X, MariaDB CS, Virtual Columns, JSON, User Management, Group Replication, PL/SQL, etc.
MariaDB and Linux Distros

• Redhat/CentOS:
  • 6 → MySQL 5.1
  • 7 → MariaDB 5.5
  • 8 → MariaDB 10.3, MySQL 8.0

• Ubuntu:
  • 16.04 → MySQL 5.7
  • 18.04 → MySQL 5.7 (MariaDB 10.1)
  • 20.04 → ? (Ubuntu 19.10: MySQL 8.0 (MariaDB 10.3))

• Debian:
  • 8 → MySQL 5.5
  • 9 → MariaDB 10.1
  • 10 → MariaDB 10.3

• SuSE SLE / OpenSuSE:
  • 12 → MariaDB 10.0
  • 15 → MariaDB 10.2
  • Leap 42 → MariaDB 10.0, Leap 15, → MariaDB 10.2
Michael Howard: Embrace of open source is destroying 'artificial definitions' of legacy vendors

MariaDB boss says IPO is part of his 3-year plan

By Rebecca Hill 13 Nov 2018 at 09:04

Interview Michael Howard, Berkley grad and alumnus of Oracle and EMC, took the helm at open-source biz MariaDB almost three years ago. Reflecting on how things have changed, he reckons the biggest shift is in how both investors and enterprise have embrace open-source. Now, he has an IPO on his mind.

In an interview with *El Reg*, Howard – who, as noted at the time of his appointment, has worked for a number of companies who were slurped up by bigger businesses – said the end of 2018 will see the end of the first year of a three-year plan he devised for the firm.
Retrospect MariaDB 10.3

- GA May 2018
- Invisible Columns
- System-versioned Tables
- Instant ADD COLUMN
- Storage Engine independent Column Compression
- Semi-synchronous Replication Built-in (before Plug-in)
- PROXY Protocol Support (Galera/HAproxy)
- Optimizer Improvements (SQL Performance)
- Aggregate Stored Functions (DWH, MariaDB Column Store)
- Oracle Compatibility (sql_mode = ORACLE)
- Oracle PL/SQL Packages, Oracle Style Sequences
- and many, many more...
MariaDB 10.4 - Overview

- GA June 2019 (10.4.6)
  - → wait 6 – 12 months for production (mid 2020)!
  - Still very poor quality (especially Galera 4!)
  - Regression in 10.4.9 (5. 11. 2019)
  - Remember MySQL IPO plan 2005: MySQL 5.0 was “worst release ever”
- Cloud… (IPO?)
- Standard compliant (IPO?)
- More feature complete
Authentication

- unix_socket authentication is default!
  - Access if O/S user = DB user
  - New DB user: mysql
- User Password Expiry:
  - `ALTER USER 'oli'@'localhost'
    PASSWORD EXPIRE INTERVAL 90 DAY;`
- Account Locking:
  - `ALTER USER 'oli'@'localhost'
    ACCOUNT LOCK;`
- Table `mysql.user` is retired!!!
  - → Can break Admin Applications...
  - New: `mysql.global_priv` Table
- More than 1 authentication plugin possible
  - → slowly migrate users to more secure authentication
  - `CREATE USER admin@localhost IDENTIFIED VIA unix_socket OR
    mysql_native_password USING 'secret';`
InnoDB

● Instant DROP COLUMN operation
  ● Changing of column order
  ● More Instant operations (VARCHAR, collation and character set)
● Improvements in Index DDL
  ● RENAME INDEX
● Merge InnoDB changes from “upstream”
● InnoDB row length count fixed (10.4.7)
  ● Leads to errors:
    [Warning] InnoDB: Cannot add field `thumbnails` in table `test`.`products` because after adding it, the row size is 8702 which is greater than maximum allowed size (8126) for a record on index leaf page.
ALTER TABLE test
ADD COLUMN d BIGINT
/*!100400 , ALGORITHM=INSTANT */;
Query OK, 0 rows affected (0.247 sec)

ALTER TABLE test MODIFY COLUMN
c BIGINT(20) DEFAULT NULL AFTER d
/*!100400 , ALGORITHM=INSTANT */;
Query OK, 0 rows affected (0.072 sec)
Optimizer

- **Optimizer Trace:**
  - `SET SESSION optimizer_trace='enabled=on';`
  - `I_S.optimizer_trace`

- **SE Independent Table Statistics**
  - Histogram collection by default

- **Improved Condition Pushdown Optimization**
  - `SELECT ... WHERE XXX AND ... IN (<subquery>)`

- **Automatic optimized use of Join Buffer**
  - Exists since 5.3.0 but was disabled :-(

- **Rowid Filtering Optimization**
  - `WHERE a.date BETWEEN '2018-01-01' AND '2018-01-31' AND b.price between 200000 and 230000;`

- But...
## Optimizer – Query Runtime

<table>
<thead>
<tr>
<th></th>
<th>5.5.41</th>
<th>10.0.28</th>
<th>10.1.29</th>
<th>10.2.14</th>
<th>10.3.18</th>
<th>10.4.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1</td>
<td>3.03</td>
<td>3.17</td>
<td>3.36</td>
<td>3.29</td>
<td>3.38</td>
<td>3.57</td>
</tr>
<tr>
<td>Query 2</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.19</td>
<td>0.19</td>
<td>0.21</td>
</tr>
<tr>
<td>Query 3</td>
<td>2.41</td>
<td>2.40</td>
<td>2.51</td>
<td>3.27</td>
<td>3.92</td>
<td>4.44</td>
</tr>
</tbody>
</table>

### MariaDB Optimizer - Over time

![Graph showing the performance of queries over different versions of MariaDB.](graph.png)

- **5.5.41**: Blue bars
- **10.0.28**: Red bars
- **10.1.29**: Orange bars
- **10.2.14**: Green bars
- **10.3.18**: Yellow bars
- **10.4.8**: Gray bars
SQL> SET SESSION optimizer_trace='enabled=on';
SQL> EXPLAIN SELECT * FROM test WHERE id <10;
SQL> SELECT trace
FROM information_schema.optimizer_trace;

"steps": [
  {
    "condition_processing": {
      "condition": "WHERE",
      "original_condition": "test.`id` < 10",
      "steps": [
        {
          "transformation": "equality_propagation",
          "resulting_condition": "test.`id` < 10"
        },
        {
          "transformation": "constant_propagation",
          "resulting_condition": "test.`id` < 10"
        },
        {
          "transformation": "trivial_condition_removal",
          "resulting_condition": "test.`id` < 10"
        }
      ]
    }
  },
  ...
]
Application-Time Period Tables

- ISO/IEC 9075, SQL:2011 Part 2
- MariaDB 10.3: System Versioned Tables
- Journal/Tracking of an Item.

```
CREATE TABLE employee (
    ID         INT UNSIGNED NOT NULL,
    Start      DATE,
    End        DATE,
    Department VARCHAR(32),
    Position   VARCHAR(32),
    PRIMARY KEY (ID, Start, End),
    PERIOD FOR Period (Start, End)
);
```

- AUTO_INCREMENT ID is NO good plan!
Lend Employee

UPDATE employee
    FOR PORTION OF Period
    FROM '2018-03-15' TO '2018-07-15'
    SET Department = 'Development'
    , Position = 'DBA'
    WHERE ID = 12345
;
Query OK, 1 row affected (0.000 sec)
Rows matched: 1  Changed: 1  Inserted: 2  Warnings: 0

SELECT * FROM employee ORDER BY Start;

<table>
<thead>
<tr>
<th>ID</th>
<th>Start</th>
<th>End</th>
<th>Department</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>12345</td>
<td>2017-01-01</td>
<td>2018-03-15</td>
<td>Engineering</td>
<td>Junior DBA</td>
</tr>
<tr>
<td>12345</td>
<td>2018-03-15</td>
<td>2018-07-15</td>
<td>Development</td>
<td>DBA</td>
</tr>
<tr>
<td>12345</td>
<td>2018-07-15</td>
<td>9999-12-31</td>
<td>Engineering</td>
<td>Junior DBA</td>
</tr>
</tbody>
</table>
## Upgrade Employee

```sql
UPDATE employee
  FOR PORTION OF Period
FROM '2018-07-15'
  TO '9999-12-31'
SET Position = 'DBA'
WHERE ID = 12345
;
```

<table>
<thead>
<tr>
<th>ID</th>
<th>Start</th>
<th>End</th>
<th>Department</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>12345</td>
<td>2017-01-01</td>
<td>2018-03-15</td>
<td>Engineering</td>
<td>Junior DBA</td>
</tr>
<tr>
<td>12345</td>
<td>2018-03-15</td>
<td>2018-07-15</td>
<td>Development</td>
<td>DBA</td>
</tr>
<tr>
<td>12345</td>
<td>2018-07-15</td>
<td>9999-12-31</td>
<td>Engineering</td>
<td>DBA</td>
</tr>
</tbody>
</table>
1 year vacation (unpaid)

DELETE FROM employee
    FOR PORTION OF Period
    FROM '2018-09-30'   -- !?!
    TO '2019-10-01'    -- !?!
    WHERE ID = 12345
;

+-------+------------+------------+-------------+------------+
| ID    | Start      | End        | Department  | Position   |
+-------+------------+------------+-------------+------------+
| 12345 | 2017-01-01 | 2018-03-15 | Engineering | Junior DBA |
| 12345 | 2018-03-15 | 2018-07-15 | Development | DBA        |
| 12345 | 2018-07-15 | 2018-09-30 | Engineering | DBA        |
| 12345 | 2019-10-01 | 9999-12-31 | Engineering | DBA        |
Querying

What is valid now?

SELECT *
  FROM employee
  WHERE Start <= CURRENT_DATE()
    AND End > CURRENT_DATE()
;

What was valid during last year?

SELECT *
  FROM employee
  WHERE End >= '2018-01-01'
    AND Start <= '2018-12-31'
;
Syntax – Variables – Replication

- Bitemporal Tables is also possible
  - Combination of system versioned and application-time periods
- `SQL> FLUSH SSL;`
  - Dynamically exchange servers TLS certificates
- `SQL> INSTALL/UNINSTALL IF [NOT] EXISTS PLUGIN ...`

- Variables: Some minor changes...

- Replication:
  - GTID clean-up (`gtid_cleanup_batch_size`)
  - Binary Log Rotation speed up
  - `SQL> SHUTDOWN WAIT FOR ALL SLAVES;`
General

- System Tables (`mysql.*`) Crash-safe Aria!
- Commands `mysql* → mariadb*`
  - Expected but breaks with many applications! :-(
- Performance improvements for Unicode collations
- User data type plugin (work in progress)
  - Oracle `TYPE ... TABLE OF, ... AS OBJECT OF`
- Much faster privilege checks (Cloud)
  - Many users accounts or database grants
- MS SQL Server compatibility: `sql_mode = MSSQL`
  - “For the moment MSSQL mode only has limited functionality, but we plan to add more later according to demand.”
- JSON: `JSON_MERGE_PATCH` and `JSON_MERGE_PRESERVE`
BACKUP STAGE

- Differences between MariaDB Community and Enterprise Server :-(
- More efficient Backup Locks for Storage Snapshots:

```
SQL> BACKUP STAGE START;
SQL> BACKUP STAGE BLOCK_COMMIT;

SQL> system lvcreate --size 1G --snapshot --name snapshot /dev/vg/snapshots

SQL> BACKUP STAGE END;
```

- Now officially supported (thanks to Cloud)!
- Better than FLUSH TABLES WITH READ LOCK;
Galera 4

- Galera ready by default since MariaDB 10.1
- New Galera Tables
  - `mysql.wsrep_{cluster,cluster_members,streaming_log}`
- Streaming Replication
  - Transactions of unlimited size
  - Replicates gradually in small fragments
  - Dynamically per session: `wsrep_trx_fragment_size = <n>`
  - Usefull size ~10k rows
  - Degrades transaction throughput!!!
  - Conflicts with `LOAD DATA` splitting (`wsrep_load_data_splitting`)
- Rolling Upgrade from Galera 3 to 4 is supported
- Completely not mature yet (10.4.10) also w/o Streaming Repl!
Outlook MariaDB 10.5

- Last week: 10.5.0 (alpha!!)
- INSERT/REPLACE ... RETURNING
- S3 Storage Engine (Archive in the Cloud)
- Aria SE improvements (for S3 SE?)
- Thread Pool Statistics (Cloud?)
- InnoDB clean-up and refactoring (BP Instances?)
- MySQL extended Binlog Metadata from Upstream
- Optimizer improvements
- INFORMATION_SCHEMA improvements
- Galera 4 – Inconsistency Voting
- Perl Scripts from DBD::mysql → DBD::MariaDB
Q & A

Questions ?
Discussion?

We have time for some face-to-face talks...

- FromDual provides neutral and independent:
  - Consulting
  - Training
  - Remote-DBA
  - Support for MariaDB, Galera Cluster and MySQL