MySQL 5.7

New Features

CeBIT 2016, Hannover

Oli Sennhauser

Senior MySQL Consultant, FromDual GmbH
oli.sennhauser@fromdual.com
About FromDual GmbH

Support

MySQL

Consulting

GALERA CLUSTER

remote-DBA

Training

SOUO

MariaDB

PERCONA SERVER

DOAG
MySQL (Release) History

- InnoDB + MySQL: 2001
- Innobase Black Friday: 2005-10-07
- MySQL 5.0.15: 2005-10-19
- Sun Buys MySQL: 2008-01-16
- MySQL 5.1.30: 2008-11-14
- Oracle buys Sun: 2009-04-20
- MySQL 5.5.8: 2010-12-03
- MySQL 5.6.10: 2013-02-05
- MySQL 5.7.9: 2015-10-21
Which release?

- MySQL: 4.x, 5.0, 5.1, 5.5, 5.6, 5.7, 5.8
- Now (2016) we should be on MySQL 5.6 and think about 5.7 upgrade!!!
- New projects with MySQL 5.7!

<table>
<thead>
<tr>
<th>RELEASE</th>
<th>GA DATE</th>
<th>SUPPORT END</th>
<th>EXT. SUPPORT END</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL Database 5.0</td>
<td>Oct 2005</td>
<td>Dec 2011</td>
<td>n.a.</td>
</tr>
<tr>
<td>MySQL Database 5.1</td>
<td>Dec 2008</td>
<td>Dec 2013</td>
<td>n.a.</td>
</tr>
<tr>
<td>MySQL Database 5.5</td>
<td>Dec 2010</td>
<td><strong>Dec 2015</strong></td>
<td>Dec 2018</td>
</tr>
<tr>
<td>MySQL Database 5.6</td>
<td>Feb 2013</td>
<td>Feb 2018</td>
<td>Feb 2021</td>
</tr>
<tr>
<td>MySQL Database 5.7</td>
<td>Oct 2015</td>
<td>Oct 2020</td>
<td>Oct 2023</td>
</tr>
<tr>
<td>MySQL Database 5.8</td>
<td>expected 2017</td>
<td>unknown</td>
<td>unknown</td>
</tr>
</tbody>
</table>
Why MySQL 5.7?

- Scalability + Performance
- Security improvements
- Features
  - JSON support
  - GIS (spatial) features
  - Generated columns
  - Optimizer
  - Multi-source replication
- Operations
- Clean-up
Scalability

- Who runs 16 – 32 or more parallel queries?
- 100 – 200k Qps?
- Who here is from FB or similar?

Single query performance gets worse!!!
Performance

- TCP timeout 120s?
- 400 – 800 conn/s
- 32,000 ports
- SSL new default!
- Impact on connection rate?
Security improvements

- Secure by default!
  - We all know it means less KISS!
- SSL by default (slower and more expensive)
- Tablespace encryption (does it make sense?)
- Password expiration policy
- Password strength policy
  - Dropped column in `mysql.user` table :-(
    - Old application might not work any more?
- `sql_mode` more strict (good!)
InnoDB tablespace encryption

```sql
SELECT plugin_name, plugin_status FROM information_schema.plugins
  WHERE plugin_name='keyring_file';
```

<table>
<thead>
<tr>
<th>PLUGIN_NAME</th>
<th>PLUGIN_STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>keyring_file</td>
<td>ACTIVE</td>
</tr>
</tbody>
</table>

```
+-----------------------+-------+
| Variable_name         | Value |
+-----------------------+-------+
| innodb_file_per_table | ON    |
```

```sql
SHOW GLOBAL VARIABLES LIKE 'keyring%';
```

<table>
<thead>
<tr>
<th>Variable_name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>keyring_file_data</td>
<td>/usr/local/mysql/keyring/keyring</td>
</tr>
</tbody>
</table>

ERROR 3185 (HY000): Can't find master key from keyring, please check keyring plugin is loaded.
InnoDB tablespace encryption

strings test.ibd
infimum
supremum
secret dataV
secret dataV
secret dataV
secret dataV

ALTER TABLE TEST ENCRYPTION = 'y';

strings test.ibd
VBw1?"sh
Pk-{
`dIO
...

CREATE TABLE product (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    product_id VARCHAR(40),
    product_id_norm INT AS (SUBSTR(product_id,3)) VIRTUAL,
    PRIMARY KEY (id),
    INDEX (product_id_norm)
);

SELECT * FROM product WHERE product_id_norm = 1;

<table>
<thead>
<tr>
<th>id</th>
<th>product_id</th>
<th>product_id_norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5500000001</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>table</th>
<th>type</th>
<th>possible_keys</th>
<th>key</th>
<th>ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>product</td>
<td>ref</td>
<td>product_id_norm</td>
<td>product_id_norm</td>
<td>const</td>
</tr>
</tbody>
</table>
JSON support

- JSON data type
  - Automatic validation of JSON documents
  - Optimized storage format for quick read
  - Up to `max_allowed_packet` size

```sql
CREATE TABLE t1 (  
  id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
  jdoc JSON);

INSERT INTO t1 VALUES(NULL, '{"key1": "value1", "key2": "value2"}');

SELECT * FROM t1;
+----+--------------------------------------+
| id | jdoc                                 |
+----+--------------------------------------+
|  1 |{"key1": "value1", "key2": "value2"} |
+----+--------------------------------------+
```
JSON functions

- JSON_xxx
  - APPEND, ARRAY, ARRAY_APPEND, ARRAY_INSERT, CONTAINS, DEPTH, EXTRACT, INSERT, KEYS, LENGTH, MERGE, OBJECT, QUOTE, REMOVE, REPLACE, SEARCH, SET, TYPE, UNQUOTE, VALID

- Indexing: Generated column

```sql
CREATE TABLE jsongen (  
c JSON ,  
g INT GENERATED ALWAYS AS (JSON_EXTRACT(c, '$.id')) ,  
INDEX (g)  
);
```
GIS (spatial) indices

CREATE TABLE gis (  
id INT UNSIGNED NOT NULL  ,  
g GEOMETRY NOT NULL  ,  
PRIMARY KEY (id)  ,  
SPATIAL INDEX(g)  ) ENGINE = InnoDB;

- Spatial Viewer in MySQL workbench
- Customer in Germany in less than 1 hour!
- Combine with JSON
Multi-source replication

- Fan-out replication (we have since long)
- Fan-in replication (many Master one Slave)
  - For aggregating/collecting
  - Replication Channel

```
master_info_repository    = TABLE
relay_log_info_repository = TABLE

CHANGE MASTER
TO MASTER_HOST='master1', MASTER_USER='replication'
    , MASTER_PASSWORD='secret'
    , MASTER_LOG_FILE='master1-bin.000042', MASTER_LOG_POS=466918
FOR CHANNEL 'master1';

START SLAVE FOR CHANNEL 'master1';
```
Multi-Source replication use cases

- World-wide manufacturing-data-distribution
- Fleet-Management
Flexible (general) tablespaces

- Hoster/SaaS: multi-tenant applications
  - 10'000 customers with 200 tables/schema
  - `innodb_file_per_table = 0` or `1` ???
  - 2 Mio tables!!!
- File Handles
  - Linux is not happy
  - MySQL is not happy
Flexible (general) tablespaces

CREATE TABLESPACE customer0001
    ADD DATAFILE 'customer0001.ibd' ENGINE = InnoDB;

ALTER TABLE customer0001.invoices TABLESPACE = customer0001;

SELECT ts.space AS ts_id, ts.name AS ts_name, d.path, SUBSTRING_INDEX(t.name, '/', 1) AS t_schema, SUBSTRING_INDEX(t.name, '/', -1) AS t_name
FROM I_S.innodb_sys_tablespaces AS ts
JOIN I_S.innodb_sys_tables AS t ON t.space = ts.space
JOIN I_S.innodb_sys_datafiles AS d ON t.space = d.space
WHERE ts.space_type = 'General';

+-------+--------------+----------+---------+--------------------+
| ts_id | ts_name      | t_schema | t_name  | path               |
+-------+--------------+----------+---------+--------------------+
|    30 | my_ts        | mysql    | test2   | ./my_ts.ibd        |
|   186 | customer0001 | test     | jsongen | ./customer0001.ibd |
Optimizer improvements

EXPLAIN FOR CONNECTION 42;

• New optimizer hints:
  
  SELECT /*+ hint(<table> <index>) */ * FROM table;

  • NO_RANGE_OPTIMIZATION, NO_MRR, MAX_EXECUTION_TIME, NO_ICP

• New cost model
  
  • mysql.engine_cost, mysql.server_cost
sys Schema

- MySQL 5.7 by default (mysql_upgrade)
- Easy access to PERFORMANCE_SCHEMA
- Topics:
  - host_* → Activities grouped by host
  - innodb_* → InnoDB Information
  - io_* → I/O consumers grouped by file, bytes, latency
  - memory_* → Memory usage grouped by host, thread, user, type
  - schema_* → Various information about schema
  - statement_* → Statistics about statements
  - user_* → Information per user
  - waits_* → Wait event informations
- Use cases: http://fromdual.com/mysql-performance-schema-hints
Operational aspects

- InnoDB buffer pool online change:
  - \texttt{SET} \texttt{GLOBAL innodb_buffer_pool_size} = \texttt{16 * CAST(POW(1024, 3) AS SIGNED)};
- \texttt{mysql_install_db} obsolete:
  - \texttt{mysqld --initialize / --initialize-insecure}
- Implicit create user not allowed:
  - \texttt{GRANT ALL ON *.* to 'nonexist'@'localhost';}
    \footnotesize{ERROR 1133 (42000): Can't find any matching row in the user table}
  - → First: \texttt{CREATE USER} ... then \texttt{GRANT} ...
- Upgrade Master/Slave replication from 5.5 directly to 5.7 is NOT possible (Server UUID)!
- Error logging
  - Syslog native support
    - \texttt{SET GLOBAL log_verbosity = 1 - 3} (errors, warnings, notes)
- \texttt{super_read_only = 1}
- \texttt{SET GLOBAL offline_mode=1}
- \texttt{disabled_storage_engines="MyISAM"}
## Deprecated and Removed

- **Deprecated features**
  - `innodb_install_db` → `mysqld -initialize`
  - Old variables like `innodb_file_format`, `innodb_large_prefix`
  - `GRANT USER` to create a user → `CREATE USER`
  - Implicit `GROUP BY` sorting → `ORDER BY`
  - `--log-warnings` → `log_error_verbosity`
  - etc.

- **Removed features**
  - Old passwords!
  - `YEAR(2)` → Convert to `YEAR(4)` (expensive!)
  - `storage_engine` → `default_storage_engine`
  - `thread_concurrency`
  - `--key-buffer` → `--key-buffer-size`
  - `innodb_addtitional_mem_pool_size`
  - `INSERT DELAYED`
  - InnoDB Monitor and InnoDB Lock Monitor tables
  - MySQL utilities: `mysqlhotcopy`, etc.

- Test upgrade carefully!!!
Q & A

Questions?

Discussions?

Halle 3, Stand D36 / 630

Slides are on-line:

www.fromdual.com/presentations