Many Non Easily Repeatable Issues in Complex Systems
Resolving and Preventing Issues Requires Internals Understanding
Data Capture

ONGOING DATA CAPTURE (MONITORING)

TEMPORARY DATA CAPTURE (DEBUGGING)
Comprehensive View Needed

You Can’t just use at MariaDB Alone

OS Issues, Hardware Issues are often root cause

Application issues can’t be ignored

Background Load, Noisy Neighbors
MariaDB
MariaDB 10.4 Key Data Sources

- SHOW STATUS
- INFORMATION SCHEMA
- PERFORMANCE SCHEMA
- LOGS
- EXPLAIN
- OPTIMIZER TRACE
SHOW [GLOBAL] STATUS

- Existed Forever
- Shows 500+ status variables
- Most are counters, some are gauges, some text
- Session and Global Scope
- Also available as Information Schema Table
Global And Session

MariaDB [(none)]> select * from information_schema.global_status
where variable_name='Questions';

+---------------+----------------+
| VARIABLE_NAME | VARIABLE_VALUE |
+---------------+----------------+
| QUESTIONS     | 82457893       |
+---------------+----------------+
1 row in set (0.0001 sec)

MariaDB [(none)]> select * from information_schema.session_status
where variable_name='Questions';

+---------------+----------------+
| VARIABLE_NAME | VARIABLE_VALUE |
+---------------+----------------+
| QUESTIONS     | 153            |
+---------------+----------------+
1 row in set (0.0000 sec)
MariaDB [(none)]> select * from information_schema.session_status where variable_name like "%rows%";

<table>
<thead>
<tr>
<th>VARIABLE_NAME</th>
<th>VARIABLE_VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>INNODB_ROWS_DELETED</td>
<td>14553828</td>
</tr>
<tr>
<td>INNODB_ROWS_INSERTED</td>
<td>15840851</td>
</tr>
<tr>
<td>INNODB_ROWS_READ</td>
<td>1927758552</td>
</tr>
<tr>
<td>INNODB_ROWS_UPDATED</td>
<td>29290781</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

14 rows in set (0.0000 sec)
```
root@mariadb104:~# mariadb-admin extended -i1 -r | grep Questions
| Questions | 82525506
| Questions | 378
| Questions | 519
| Questions | 591
```
70+ Tables

Some are Schema Related

Others are Performance Statistics
<table>
<thead>
<tr>
<th>TABLE_CATALOG</th>
<th>TABLE_SCHEMA</th>
<th>TABLE_NAME</th>
<th>TABLE_TYPE</th>
<th>ENGINE</th>
<th>VERSION</th>
<th>ROW_FORMAT</th>
<th>TABLE_ROWS</th>
<th>AVG_ROW_LENGTH</th>
<th>DATA_LENGTH</th>
<th>MAX_DATA_LENGTH</th>
<th>INDEX_LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>def</td>
<td>information_schema</td>
<td>ALL_PLUGINS</td>
<td>SYSTEM VIEW</td>
<td>Aria</td>
<td>11</td>
<td>Page</td>
<td>NULL</td>
<td>0</td>
<td>8192</td>
<td>4503599627288576</td>
<td>8192</td>
</tr>
</tbody>
</table>

**Data**
- **TABLE_COMMENT**: MAX_INDEX_LENGTH: 137438945280
- **TABLE_COLLATION**: utf8_general_ci
- **CREATE_OPTIONS**: max_rows=18882
- **CHECKSUM**: NULL
- **CREATE_TIME**: 2020-02-01 21:30:36
- **UPDATE_TIME**: 2020-02-01 21:30:36
- **CHECK_TIME**: NULL
- **TEMPORARY**: Y

1 row in set (0.007 sec)
Innodb Metrics

MariaDB [information_schema]>

```sql
SELECT *
FROM innodb_metrics
WHERE status='enabled'
LIMIT 1
```

```
*************************** 1. row ***************************
   NAME: lock_deadlocks
  SUBSYSTEM: lock
     COUNT: 0
   MAX_COUNT: NULL
  MIN_COUNT: NULL
  AVG_COUNT: 0
COUNTER_RESET: 0
   MAX_COUNT_RESET: NULL
 MIN_COUNT_RESET: NULL
 AVG_COUNT_RESET: NULL
 TIME_ENABLED: 2020-02-01 20:34:32
TIME_DISABLED: NULL
TIME_ELAPSED: 3606
TIME_RESET: NULL
   STATUS: enabled
     TYPE: counter
   COMMENT: Number of deadlocks
```

1 row in set (0.001 sec)
While innodb_metrics look similar to SHOW STATUS many of them are **NOT** enabled by default.

`innodb_monitor_enable=all`
MariaDB [information_schema]>

```
| NAME  | CREATE_FILE | CREATE_LINE | OS_WAITS |
|-------+-------------+-------------+----------|
|      | log0log.cc  |         578 |        1 |
|      | btr0sea.cc  |         243 |      232 |
```

2 rows in set (0.002 sec)
<table>
<thead>
<tr>
<th>ID: 107</th>
<th>USER: root</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOST: localhost</td>
<td>DB: information_schema</td>
</tr>
<tr>
<td>COMMAND: Query</td>
<td>TIME: 0</td>
</tr>
<tr>
<td>STATE: Filling schema table</td>
<td>INFO: select * from processlist limit 1</td>
</tr>
<tr>
<td>TIME_MS: 0.295</td>
<td>STAGE: 0</td>
</tr>
<tr>
<td>MAX_STAGE: 0</td>
<td>PROGRESS: 0.000</td>
</tr>
<tr>
<td>MEMORY_USED: 106240</td>
<td>MAX_MEMORY_USED: 2214408</td>
</tr>
<tr>
<td>EXAMINED_ROWS: 0</td>
<td>QUERY_ID: 17799397</td>
</tr>
<tr>
<td>INFO_BINARY: select * from processlist limit 1</td>
<td>TID: 5398</td>
</tr>
<tr>
<td>1 row in set (0.001 sec)</td>
<td></td>
</tr>
</tbody>
</table>
Some of the Most Advanced Instrumentation Available

50+ Tables

Disabled by Default in MariaDB

Can enable more instrumentation permanently or temporarily

Overhead can be high
Enabling Performance Schema

`performance_schema=1`

(Restart Required)
### Performance Schema Configuration

```sql
MariaDB [performance_schema]> show tables like "setup%";

+---------------------------------------+
| Tables_in_performance_schema (setup%) |
+---------------------------------------+
| setup_actors                          |
| setup_consumers                       |
| setup_instruments                     |
| setup_objects                         |
| setup_timers                          |
+---------------------------------------+

5 rows in set (0.001 sec)
```
MariaDB [performance_schema]>

```sql
select * from setup_actors;
```

```
+------+------+------+
| HOST | USER | ROLE |
+------+------+------+
| %    | %    | %    |
+------+------+------+

1 row in set (0.000 sec)
Consumers – What Summaries are going to be built

```sql
MariaDB [performance_schema]> select * from setup_consumers;
+--------------------------------+---------+
| NAME                           | ENABLED |
|--------------------------------+---------+
| events_stages_current          | NO      |
| events_stages_history          | NO      |
| events_stages_history_long     | NO      |
| events_statements_current      | YES     |
| events_statements_history      | NO      |
| events_statements_history_long | NO      |
| events_waits_current           | NO      |
| events_waits_history           | NO      |
| events_waits_history_long      | NO      |
| global_instrumentation         | YES     |
| thread_instrumentation         | YES     |
| statements_digest              | YES     |
+--------------------------------+---------+
12 rows in set (0.000 sec)
```
• **700+ Instrumentation Points**

• **250+ Enabled and Timed when Performance Schema is Enabled**

MariaDB [performance_schema]> select * from setup_instruments where enabled='yes' limit 3 \G

*************************** 1. row ***************************
NAME: wait/io/file/sql/map
ENABLED: YES
TIMED: YES

*************************** 2. row ***************************
NAME: wait/io/file/sql/binlog
ENABLED: YES
TIMED: YES

*************************** 3. row ***************************
NAME: wait/io/file/sql/binlog_index
ENABLED: YES
TIMED: YES

3 rows in set (0.000 sec)
What Objects do we want to Instrument

```
MariaDB [performance_schema]> select * from setup_objects;
+-------------+--------------------+-------------+---------+-------+
| OBJECT_TYPE | OBJECT_SCHEMA      | OBJECT_NAME | ENABLED | TIMED |
+-------------+--------------------+-------------+---------+-------+
| TABLE       | mysql              | %           | NO      | NO    |
| TABLE       | performance_schema | %           | NO      | NO    |
| TABLE       | information_schema | %           | NO      | NO    |
| TABLE       | %                  | %           | YES     | YES   |
+-------------+--------------------+-------------+---------+-------+
4 rows in set (0.000 sec)
```
Example of Performance Schema

```
MariaDB [performance_schema]> select * from events_statements_current limit 1

*************************** 1. row ***************************
  THREAD_ID: 53
  EVENT_ID: 226
  END_EVENT_ID: NULL
  EVENT_NAME: statement/sql/select
  SOURCE: mysqld.cc:1179
  TIMER_START: 1484013787233000
  TIMER_END: 1484014492287000
  TIMER_WAIT: 705054000
  LOCK_TIME: 145000000
  SQL_TEXT: select * from events_statements_current limit 1
  DIGEST: NULL
  DIGEST_TEXT: NULL
  CURRENT_SCHEMA: performance_schema
  OBJECT_TYPE: NULL
  OBJECT_SCHEMA: NULL
  OBJECT_NAME: NULL
  OBJECT_INSTANCE_BEGIN: NULL
  MYSQL_ERRNO: 0
  RETURNED_SQLSTATE: NULL
  MESSAGE_TEXT: NULL
  ERRORS: 0
  WARNINGS: 0
  ROWS_AFFECTED: 0
  ROWS_SENT: 0
  ROWS_EXAMINED: 0
  CREATED_TMP_DISK_TABLES: 0
  CREATED_TMP_TABLES: 0
  SELECT_FULL_JOIN: 0
  SELECT_FULL_RANGE_JOIN: 0
  SELECT_RANGE: 0
  SELECT_RANGE_CHECK: 0
  SELECT_SCAN: 1
  SORT_MERGE_PASSES: 0
  SORT_RANGE: 0
  SORT_ROWS: 0
  SORT_SCAN: 0
  NO_INDEX_USED: 1
  NO_GOOD_INDEX_USED: 0
  NESTING_EVENT_ID: NULL
  NESTING_EVENT_TYPE: NULL

1 row in set (0.001 sec)
```
MariaDB [performance_schema]> select * from file_summary_by_instance limit 5,1 \G

*************************** 1. row
***************************

FILE_NAME: /var/lib/mysql/ibdata1
EVENT_NAME: wait/io/file/innodb/innodb_data_file
OBJECT_INSTANCE_BEGIN: 140586480369856

COUNT_STAR: 161
SUM_TIMER_WAIT: 34773929190
MIN_TIMER_WAIT: 13463450
AVG_TIMER_WAIT: 215987135
MAX_TIMER_WAIT: 2412890480
COUNT_READ: 158
SUM_TIMER_READ: 34702257590
MIN_TIMER_READ: 76386310
AVG_TIMER_READ: 219634415
MAX_TIMER_READ: 2412890480
SUM_NUMBER_OF_BYTES_READ: 4702208
COUNT_WRITE: 0
SUM_TIMER_WRITE: 0
MIN_TIMER_WRITE: 0
AVG_TIMER_WRITE: 0
MAX_TIMER_WRITE: 0
SUM_NUMBER_OF_BYTES_WRITE: 0
COUNT_MISC: 3
SUM_TIMER_MISC: 71671600
MIN_TIMER_MISC: 13463450
AVG_TIMER_MISC: 23890230
MAX_TIMER_MISC: 30040010

1 row in set (0.000 sec)
Logs

- Error Log
- General Query Log
- Slow Query Log
Slow Query Log [Default]

# User@Host: root[root] @ localhost []
# Thread_id: 111  Schema: sbtest  QC_hit: No
# Query_time: 0.000356  Lock_time: 0.000149  Rows_sent: 1  Rows_examined: 1
# Rows_affected: 0  Bytes_sent: 190
SET timestamp=1580596196;
SELECT c FROM sbtest1 WHERE id=767650;
Slow Query Log

[With Explain]

# Time: 200201 22:32:37
# User@Host: root[root] @ localhost []
# Thread_id: 113  Schema: sbtest  QC_hit: No
# Query_time: 0.000220  Lock_time: 0.000091  Rows_sent: 1  Rows_examined: 1
# Rows_affected: 0  Bytes_sent: 190

# explain:
# id   select_type  table   type    possible_keys   key     key_len ref
# rows    r_rowsfiltered r_filtered      Extra
# explain: 1    SIMPLE  sbtest1 const   PRIMARY PRIMARY 4       const   1       NULL
# 100.00  NULL

SET timestamp=1580596357;

SELECT c FROM sbtest1 WHERE id=101985;
EXPLAIN

- Understand Query Execution Plan
- Essential Skill for Developers and DBAs
- Multiple Output Formats
- Can get EXPLAIN plan for actual running query
<table>
<thead>
<tr>
<th>id</th>
<th>select_type</th>
<th>table</th>
<th>type</th>
<th>possible_keys</th>
<th>key</th>
<th>key_len</th>
<th>ref</th>
<th>rows</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SIMPLE</td>
<td>s1</td>
<td>index</td>
<td>NULL</td>
<td>k_1</td>
<td>4</td>
<td>NULL</td>
<td>986499</td>
<td>Using index</td>
</tr>
<tr>
<td>1</td>
<td>SIMPLE</td>
<td>s2</td>
<td>index</td>
<td>NULL</td>
<td>k_1</td>
<td>4</td>
<td>NULL</td>
<td>986499</td>
<td>Using index; Using join buffer (flat, BNL join)</td>
</tr>
</tbody>
</table>
Advanced Explain

- EXPLAIN FORMAT=JSON
- SHOW EXPLAIN FOR <CONNECTION_ID>
- ANALYZE
<table>
<thead>
<tr>
<th>id</th>
<th>select_type: SIMPLE</th>
<th>table: s1</th>
<th>type: range</th>
<th>possible_keys: PRIMARY</th>
<th>key: PRIMARY</th>
<th>key_len: 4</th>
<th>ref: NULL</th>
<th>rows: 99</th>
<th>r_rows: 99.00</th>
<th>filtered: 100.00</th>
<th>r_filtered: 100.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Using where; Using index</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>id</th>
<th>select_type: SIMPLE</th>
<th>table: s2</th>
<th>type: index</th>
<th>possible_keys: NULL</th>
<th>key: k_1</th>
<th>key_len: 4</th>
<th>ref: NULL</th>
<th>rows: 986499</th>
<th>r_rows: 1000000.00</th>
<th>filtered: 100.00</th>
<th>r_filtered: 100.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Extra: Using index; Using join buffer (flat, BNL join)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Looking for Open Source Database Observability Solution?

Open Source Database Focused Observability Solution from Percona
100% Free and Open Source

http://per.co.na/PMM
Thank you, Let’s Connect!

https://www.linkedin.com/in/peterzaitsev/
https://twitter.com/PeterZaitsev