PERFORMANCE_SCHEMA and sys schema

What can we do with it?

FOSDEM 2016, January 30th, Brussels

Oli Sennhauser

Senior MySQL Consultant at FromDual GmbH

oli.sennhauser@fromdual.com

About FromDual GmbH

www.fromdual.com

Support

Consulting

remote-DBA

Training

Contents

PERFORMANCE_SCHEMA and sys schema

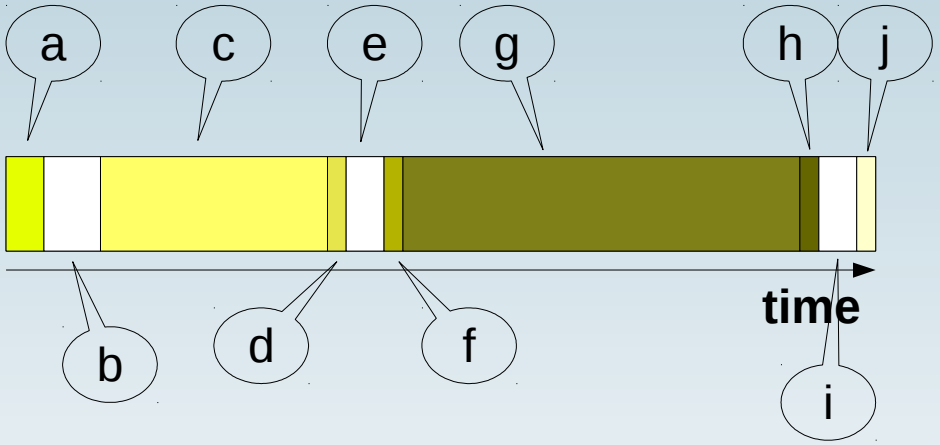
- Database Profiling
- PERFORMANCE_SCHEMA (P_S)
- Installation, configuration, instrumentation, etc.
- sys schema
- Installation, configuration, etc.
- Use cases

Database Profiling

- Where has my time gone?
- Where have my resources gone?

```
function x() {
  start = current_time();
  count[x]++;
  …
  end = current_time();
  duration[x] += (end - start);
}

function   count   time
x            123     156.25     0.8%
y             19     827.30     4.1%
z              2   19280.00    95.1%
Total        144   20263.55   100.0%
```



MySQL Profiler

- Since MySQL 5.0
- `SET profiling = 1;`
  `SELECT …;`
  `SHOW PROFILE;`
- Deprecated in 5.6
- So what now?

```
+----------------------+----------+
| Status               | Duration |
+----------------------+----------+
| starting             | 0.000059 |
| checking permissions | 0.000008 |
| Opening tables       | 0.000017 |
| init                 | 0.000023 |
| System lock          | 0.000009 |
| optimizing           | 0.000006 |
| statistics           | 0.000014 |
| preparing            | 0.000011 |
| Creating tmp table   | 0.000023 |
| Sorting result       | 0.000005 |
| executing            | 0.000003 |
| Sending data         | 1.253803 |
| Creating sort index  | 0.000049 |
| end                  | 0.000005 |
| removing tmp table   | 0.000019 |
| end                  | 0.000006 |
| query end            | 0.000009 |
| closing tables       | 0.000012 |
| freeing items        | 0.000019 |
| logging slow query   | 0.000004 |
| cleaning up          | 0.000011 |
+----------------------+----------+
```

PERFORMANCE_SCHEMA (P_S)

- Discussed at least since 2006
- Introduced in MySQL 5.5 (2010)
  - Disabled by default
  - Not really useful yet
- Improved in MySQL 5.6 (2011/12)
  - Enabled by default
  - Became useful and interesting
- More improvements in MySQL 5.7 (2013-15)
  - More probes, more information:
  - Memory, Transactions, Replication, Session/Status variables
- Overhead 2% - 200%
  - Depends on how you do it → careful!

Goal of P_S

- Monitoring MySQL server execution at a low level
- Inspect internal execution of the server at runtime

- How: Server is instrumented to collect timing information
- Output: SQL queries on normal MySQL ring-buffer tables

- Events is anything the server does:
  - Function call
  - Wait for the O/S
  - SQL statement, stages
  - Memory

P_S installation & configuration

- Installed by default
  - `mysql_upgrade` → do restart afterwards
- Configuration in `my.cnf`
  - 5.5 off
  - `performance_schema = 1`
  - 5.6 and later on
- Tuning (`my.cnf`)
  - `performance_schema_events_waits_long = 10000`
- `SHOW GLOBAL VARIABLES LIKE 'PERF%';`
- `SHOW GLOBAL STATUS LIKE 'PERF%';`

Online enabling / disabling

- `setup_*` tables

```
UPDATE setup_instruments
   SET ENABLED = 'YES', TIMED = 'YES'
 WHERE name = 'wait/synch/mutex/innodb/autoinc_mutex';
```

- Actors
  - Which user and host is profiled (default %)
- Consumers
  - Which `P_S` tables are filled with data (e.g. `events_state_current`, some are off!)
- Instruments
  - Which probes are activated (expensive ones are disabled)
- Objects
  - Database objects (table, trigger, procedure, function, event) to profile (on for user obj.)
- Timers
  - Default is in nanoseconds (10^{-9} s)

Instrumentation depth

- Enable probes:
- events_transactions_* → unknown
  - events_statements_* → moderate expensive
    - events_stages_* → expensive!
      - events_waits_* → very expensive!
      - end
      - events_memory_* → very expensive?
      - end
    - end
  - end
- End
- Can have a performance impact!

Consumer types and hierarchies

- `events_*_current` → one per thread
- `events_*_history` → most recent n events per thread (default 10)
- `events_*_history_long` → most recent n events (10000)

- `events_*_summary_global_by_event_name`
  - `events_*_summary_by_account_by_event_name` → (host + user)
    - `events_*_summary_by_host_by_event_name` → e.g. 127.0.0.1
    - `events_*_summary_by_user_by_event_name` → e.g. root
      - `events_*_summary_by_thread_by_event_name` → thread (connection)

- `events_*_summary_by_digest` → normalized query: `WHERE a = ?`
- `events_*_summary_by_program` → procedure, function, trigger, event
- `events_*_summary_by_instance` → file, mutex, prep stmt, rwlock, socket, condition
- Can be truncate with `TRUCATE TABLE ...`

sys schema

- Originally "ps_helper" in 5.6 (2012)
- By Mark Leith
  - Senior SW Dev Mgr at Oracle
- Make it easier to use the P_S!
- A database schema (sys)
- Consisting of tables, views, functions, procedures and triggers.
- To give human readable insight into a MySQL database.
- Based on PERFORMANCE_SCHEMA and INFORMATION_SCHEMA
- Current release v1.5 (2016-01)
- Download: https://github.com/mysql/mysql-sys

sys schema installation

- Since MySQL 5.7.7 installed by default!
  - Can be skipped
- In MySQL 5.6:

```
wget https://github.com/mysql/mysql-sys/archive/master.tar.gz
tar xf master.tar.gz
cd mysql-sys-master
Mysql -user=root --password < ./sys_56.sql
```

- mysql_upgrade installs/upgrades sys
  - Can be skipped

`sys` view/table types

- 2 types, e.g.:
  - `io_by_thread_by_latency` → human understandable numbers
  - `x$io_by_thread_by_latency` → base tables in picoseconds (10^{-12}s)
- 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

Configure P_S with sys

- Reset defaults:
  - `CALL sys.ps_setup_reset_to_default(TRUE);`
- Change setup tables:
  - `CALL sys.ps_setup_enable_instrument('wait');`
  - `CALL sys.ps_setup_enable_instrument('stage');`
  - `CALL sys.ps_setup_enable_instrument('statement');`
  - `CALL sys.ps_setup_enable_consumer('current');`
  - `CALL sys.ps_setup_enable_consumer('history_long');`

Use cases

- **The following use cases can be found at:**
- **http://fromdual.com/mysql-performance-schema-hints**

SSL encryption used or not


```
SELECT variable_value AS tls_version, processlist_user AS user
     , processlist_host AS host
  FROM performance_schema.status_by_thread AS sbt
  JOIN performance_schema.threads AS t
    ON t.thread_id = sbt.thread_id
 WHERE variable_name = 'Ssl_version'
 ORDER BY tls_version
;

+-------------+------+-----------+
| tls_version | user | host      |
+-------------+------+-----------+
|             | root | localhost |
| TLSv1.1     | root | localhost |
+-------------+------+-----------+
```

Accounts not properly closing connections


```
SELECT ess.user, ess.host
     , (a.total_connections - a.current_connections) - ess.count_star as not_closed
     , ((a.total_connections - a.current_connections) - ess.count_star) * 100 /
       (a.total_connections - a.current_connections) as pct_not_closed
  FROM performance_schema.events_statements_summary_by_account_by_event_name ess
  JOIN performance_schema.accounts a on (ess.user = a.user and ess.host = a.host)
 WHERE ess.event_name = 'statement/com/quit'
   AND (a.total_connections - a.current_connections) > ess.count_star
;

+------+-----------+------------+----------------+
| user | host      | not_closed | pct_not_closed |
+------+-----------+------------+----------------+
| root | localhost |          1 |         5.0000 |
+------+-----------+------------+----------------+
```

Accounts which never connected since last start-up


```
SELECT DISTINCT m_u.user, m_u.host
  FROM mysql.user m_u
  LEFT JOIN performance_schema.accounts ps_a
    ON m_u.user = ps_a.user AND m_u.host = ps_a.host
 WHERE ps_a.user IS NULL
ORDER BY m_u.user
;

+-----------+-----------+
| user      | host      |
+-----------+-----------+
| focmm     | master    |
| mysql.sys | localhost |
| oli       | localhost |
| root      | %         |
+-----------+-----------+
```

Bad SQL queries by user


```
SELECT user, host, event_name
     , sum_created_tmp_disk_tables AS tmp_disk_tables
     , sum_select_full_join AS full_join
  FROM performance_schema.events_statements_summary_by_account_by_event_name
 WHERE sum_created_tmp_disk_tables > 0
    OR sum_select_full_join > 0
 ORDER BY sum_sort_merge_passes DESC
 LIMIT 10
;

+------+-----------+-----------------------------+-----------------+-----------+
| user | host      | event_name                  | tmp_disk_tables | full_join |
+------+-----------+-----------------------------+-----------------+-----------+
| root | localhost | statement/sql/select        |             134 |       204 |
| root | localhost | statement/sql/show_fields   |            3607 |         0 |
| root | localhost | statement/sql/show_triggers |            1792 |         0 |
| root | localhost | statement/com/Field List    |              28 |         0 |
+------+-----------+-----------------------------+-----------------+-----------+
```

Top long running queries


```
UPDATE performance_schema.setup_consumers SET enabled = 1
 WHERE name = 'events_statements_history_long';

SELECT left(digest_text, 64)
     , ROUND(SUM(timer_end-timer_start)/1000000000, 1) AS tot_exec_ms
     , ROUND(SUM(timer_wait)/1000000000, 1) AS tot_wait_ms
     , ROUND(SUM(lock_time)/1000000000, 1) AS tot_lock_ms
     , MIN(LEFT(DATE_SUB(NOW(), INTERVAL (isgs.VARIABLE_VALUE - TIMER_START*10e-13) second), 19)) AS first_seen
     , MAX(LEFT(DATE_SUB(NOW(), INTERVAL (isgs.VARIABLE_VALUE - TIMER_START*10e-13) second), 19)) AS last_seen
     , COUNT(*) as cnt
  FROM performance_schema.events_statements_history_long
  JOIN performance_schema.global_status AS isgs
 WHERE isgs.variable_name = 'UPTIME'
 GROUP BY LEFT(digest_text,64)
 ORDER BY tot_exec_ms DESC
;

+----------------------------------------------------------------+-------------+-------------+-------------+-----+
| left(digest_text, 64)                                          | tot_exec_ms | tot_wait_ms | tot_lock_ms | cnt |
+----------------------------------------------------------------+-------------+-------------+-------------+-----+
| INSERT INTO `history` ( `itemid` , `clock` , `ns` , VALUE ) VALU |        12.3 |        12.3 |         2.4 |   6 |
| SELECT `i` . `itemid` , `i` . `state` , `i` . `delta` , `i` . `m` |        10.4 |        10.4 |         1.9 |   6 |
| SELECT LEFT ( `digest_text` , ? ) , `ROUND` ( SUM ( `timer_end` |         7.5 |         7.5 |         0.0 |   1 |
| SELECT `i` . `itemid` , `i` . `key_` , `h` . `host` , `i` . `typ` |         3.2 |         3.2 |         1.1 |   6 |
| SELECT `h` . `hostid` , `h` . `host` , `h` . `name` , `t` . `htt` |         2.0 |         2.0 |         0.9 |   4 |
+----------------------------------------------------------------+-------------+-------------+-------------+-----+
```

Unused indexes


```
SELECT object_schema, object_name, index_name
  FROM performance_schema.table_io_waits_summary_by_index_usage
 WHERE index_name IS NOT NULL
   AND count_star = 0
   AND index_name != 'PRIMARY'
 ORDER BY object_schema, object_name
;

+---------------+-------------+------------+
| object_schema | object_name | index_name |
+---------------+-------------+------------+
| a             | audit       | data_2     |
| a             | audit       | data       |
| a             | c1          | c2_id      |
+---------------+-------------+------------+

SELECT *
  FROM sys.schema_unused_indexes
;
```

Redundant indexes


```
SELECT table_schema, table_name
     , redundant_index_columns, dominant_index_columns
  FROM sys.schema_redundant_indexes
;
```

```
+--------------+------------------+-------------------------+-----------------------+
| table_schema | table_name       | redundant_index_columns | dominant_index_columns |
+--------------+------------------+-------------------------+-----------------------+
| crm          | accounts         | id,deleted              | id                     |
| crm          | accounts_bugs    | account_id              | account_id,bug_id      |
| crm          | acl_actions      | id,deleted              | id                     |
| crm          | acl_roles        | id,deleted              | id                     |
| crm          | acl_roles_actions | role_id                | role_id,action_id      |
+--------------+------------------+-------------------------+-----------------------+
```

Statements with errors or warnings


```
SELECT db, query, exec_count, errors, warnings
  FROM sys.statements_with_errors_or_warnings
;

+--------------------+---------------------------+------------+--------+----------+
| db                 | query                     | exec_count | errors | warnings |
+--------------------+---------------------------+------------+--------+----------+
| mysql              | UPDATE `setup_consumers... |          1 |      1 |        0 |
| mysql              | SELECT LEFT ( `digest_t... |          2 |      1 |        0 |
| sys                | SELECT * FROM `setup_co... |          1 |      1 |        0 |
| performance_schema | SELECT * FROM `user_sum... |          1 |      1 |        0 |
| test               | SELECT * FROM `test` LI... |          6 |      1 |        0 |
+--------------------+---------------------------+------------+--------+----------+
```

Tables with full table scan


```
SELECT *
  FROM sys.schema_tables_with_full_table_scans
;

+---------------+-------------+--------------------+-----------+
| object_schema | object_name | rows_full_scanned | latency   |
+---------------+-------------+--------------------+-----------+
| zabbix        | triggers    |               1436 | 5.05 ms   |
| zabbix        | hosts       |                 34 | 41.81 us  |
| zabbix        | interface   |                  6 | 4.33 ms   |
| zabbix        | expressions |                  4 | 648.98 us |
| zabbix        | config      |                  1 | 3.62 ms   |
| zabbix        | globalmacro |                  1 | 2.61 ms   |
| zabbix        | media       |                  1 | 548.01 us |
+---------------+-------------+--------------------+-----------+
```

I/O by user


```
SELECT *
  FROM sys.user_summary_by_file_io
;

+------------+------+------------+
| user       | ios  | io_latency |
+------------+------+------------+
| background | 6926 | 8.52 s     | :-(
| zabbix     |  738 | 127.52 ms  |
| root       |  101 | 6.57 ms    |
| fpmmm      |  363 | 644.00 us  |
+------------+------+------------+
```

Latency on file I/O


```
SELECT event_name, total, total_latency
     , read_latency, write_latency, misc_latency
  FROM sys.io_global_by_wait_by_latency
 LIMIT 5
;
```

```
+-------------------------+-------+-------------+-----------+-----------+--------------+
| event_name              | total | tot_latency | r_latency | w_latency | misc_latency |
+-------------------------+-------+-------------+-----------+-----------+--------------+
| innodb/innodb_log_file  |  1323 | 8.84 s      | 29.08 us  | 25.08 ms  | 8.81 s       | :(
| innodb/innodb_data_file |  6299 | 8.81 s      | 429.56 ms | 660.95 ms | 7.72 s       | :(
| sql/binlog_index        |    40 | 35.20 ms    | 4.07 us   | 0 ps      | 35.19 ms     |
| sql/binlog              |   610 | 26.09 ms    | 14.46 us  | 4.12 ms   | 21.96 ms     |
| sql/relaylog            |    13 | 18.77 ms    | 1.97 us   | 5.56 us   | 18.77 ms     |
+-------------------------+-------+-------------+-----------+-----------+--------------+
```

Memory by user


```
UPDATE performance_schema.setup_instruments
   SET enabled = 'YES'
 WHERE name LIKE 'memory%'
;

SELECT user, current_allocated, current_avg_alloc
     , current_max_alloc, total_allocated
  FROM sys.memory_by_user_by_current_bytes
;

+------------+-----------+---------------+---------------+-----------+
| user       | curr_alloc | curr_avg_alloc | curr_max_alloc | tot_alloc |
+------------+-----------+---------------+---------------+-----------+
| root       | 348.35 KiB | 9.95 KiB       | 248.04 KiB     | 8.54 MiB  |
| zabbix     | 44.72 KiB  | 2.48 KiB       | 32.02 KiB      | 11.87 MiB |
| fpmmm      | 43.56 KiB  | 4.36 KiB       | 43.56 KiB      | 1.37 MiB  |
| background | 9.85 KiB   | 1.97 KiB       | 9.64 KiB       | 1.42 MiB  |
+------------+-----------+---------------+---------------+-----------+
```

SELECT, INSERT, UPDATE and DELETE per table


```
SELECT object_type, object_schema, object_name
     , count_star, count_read, count_write
     , count_insert
  FROM performance_schema.table_io_waits_summary_by_table
 WHERE count_star > 0
 ORDER BY count_write DESC
 LIMIT 5
;
```

object_type	object_schema	object_name	count_star	count_read	count_write	count_insert
TABLE	zabbix	history_uint	12791	92	12699	12699
TABLE	zabbix	history	3374	10	3364	3364
TABLE	zabbix	history_str	2003	48	1955	1955
TABLE	zabbix	trends_uint	930	0	930	930
TABLE	zabbix	trends	200	0	200	200

How to proceed?

1. Define the question to answer:
   "Which transaction was locked by which other transaction?"

2. Can sys schema anwser the question?

```
SELECT * FROM sys.innodb_lock_waits;
+-----------------+---------+-------------+-------------+------------+
| wait_started    | wait_age | locked_table | locked_index | locked_type |
+-----------------+---------+-------------+-------------+------------+
| 16-01-29 17:53:09 | 00:00:02 | `test`.`test` | PRIMARY      | RECORD      |
+-----------------+---------+-------------+-------------+------------+
```

3. If not: Can P_S or I_S answer the question?

```
SELECT * FROM information_schema.innodb_locks;
+-------------+-----------+--------------+------------+-----------+
| lock_trx_id | lock_type | lock_table    | lock_index | lock_rec |
+-------------+-----------+--------------+------------+-----------+
| 27514604    | RECORD    | `test`.`test` | PRIMARY    |         2 |
+-------------+-----------+--------------+------------+-----------+
```

Only recent, no history yet! :(

SHOW PROFILE VS P_S


```
SET PROFILING = 1;
SELECT ...;
SHOW PROFILES;
+----------+------------+--------------------+
| Query_ID | Duration   | Query              |
+----------+------------+--------------------+
|        1 | 2.26217725 | select * from test |
+----------+------------+--------------------+

SHOW PROFILE FOR QUERY 1;
+----------------------+----------+
| Status               | Duration |
+----------------------+----------+
| starting             | 0.000060 |
| checking permissions | 0.000009 |
| Opening tables       | 0.000020 |
| init                 | 0.000020 |
| System lock          | 0.000012 |
| optimizing           | 0.000006 |
| statistics           | 0.000014 |
| preparing            | 0.000012 |
| executing            | 0.000003 |
| Sending data         | 2.261963 |
| end                  | 0.000010 |
| query end            | 0.000008 |
| closing tables       | 0.000013 |
| freeing items        | 0.000013 |
| cleaning up          | 0.000015 |
+----------------------+----------+
```

```
UPDATE performance_schema.setup_instruments SET ENABLED = 'YES'
, TIMED = 'YES' WHERE NAME LIKE '%statement/%';
UPDATE performance_schema.setup_instruments SET ENABLED = 'YES'
, TIMED = 'YES' WHERE NAME LIKE '%stage/%';
UPDATE performance_schema.setup_consumers SET ENABLED = 'YES'
  WHERE NAME LIKE '%events_statements_%';
UPDATE performance_schema.setup_consumers SET ENABLED = 'YES'
 WHERE NAME LIKE '%events_stages_%';
SELECT ...;
SELECT eshl.event_id AS Query_ID, TRUNCATE(eshl.timer_wait/1000000000000
, 6) as Duration, LEFT(eshl.sql_text, 120) AS Query
  FROM performance_schema.events_statements_history_long AS eshl
  JOIN performance_schema.threads AS t ON t.thread_id = eshl.thread_id
 WHERE t.processlist_id = CONNECTION_ID();
+----------+-----------+------------------------+
| Query_ID | Duration  | Query                  |
+----------+-----------+------------------------+
|       12 | 13.560737 | select * from test.test |
+----------+-----------+------------------------+

SELECT SUSBTR(event_name, 11) AS Stage
     , TRUNCATE(timer_wait/1000000000000,6) AS Duration
  FROM performance_schema.events_stages_history_long
 WHERE nesting_event_id = 12;
+----------------------+-----------+
| Stage                | Duration  |
+----------------------+-----------+
| starting             |  0.000043 |
| checking permissions |  0.000004 |
| Opening tables       |  0.002700 |
| init                 |  0.000025 |
| System lock          |  0.000009 |
| optimizing           |  0.000002 |
| statistics           |  0.000014 |
| preparing            |  0.000013 |
| executing            |  0.000000 |
| Sending data         | 13.557683 |
| end                  |  0.000002 |
| query end            |  0.000008 |
| closing tables       |  0.000006 |
| freeing items        |  0.000215 |
| cleaning up          |  0.000001 |
+----------------------+-----------+
```

Q & A

Questions ?

Discussion?

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

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

www.fromdual.com/presentations