Performance Schema and Sys Schema

Mark Leith
Senior Software Development Manager
MySQL Enterprise Tools, Oracle

ORACLE

Program Agenda

1. This Introduction
2. Improvements to monitoring in MySQL 5.7
3. Improvements made so far in MySQL 8.0
4. Questions and feedback!

Performance Schema Improvements in MySQL 5.7

- **23 Worklogs completed**
  - **14 Adding new instrumentation**
  - **4 Performance / Scalability improvements**
  - **5 Configuration / Refactoring improvements**
- **35 new tables added** *(5.7 has 87, 5.6 had 52, 5.5 had 17)*
- **419 new instruments** *(5.7 has 972, 5.6 had 553, 5.5 had 222)*

Performance Schema Improvements in MySQL 8.0

- **4 Worklogs completed**
  - **2 Adding new instrumentation**
  - **1 Scalability Improvement**
  - **1 Refactoring**
- **7 new tables added** *(8.0 has 96, 5.7 - 87, 5.6 - 52, 5.5 -17)*
- **143 new instruments** *(8.0 has 1152, 5.7 - 1009, 5.6 - 553, 5.5 - 222)*

- ***So far …***

Program Agenda

1. That Introduction
2. Improvements to monitoring in MySQL 5.7
3. Improvements made so far in MySQL 8.0
4. Questions and feedback!

"Understanding where MySQL can allocate memory can help us to find the cause in most cases. It is not as straightforward as it should be and I'm very hopeful future releases of MySQL, MariaDB or Drizzle bring improvements in this space allowing us to see directly for what purpose memory is allocated and so detect all kinds of memory usage problems easier."

Peter Zaitsev, CEO, Percona

WL#3249 - PERFORMANCE_SCHEMA, Instrument memory usage

http://www.mysqlperformanceblog.com/2012/03/21/troubleshooting-mysql-memory-usage/

WL#3249 - Instrument Memory Usage

- Added **212** different memory instrumentation types (all disabled by default)
- Records **current, high and low water marks** of allocations
- Does ***not*** track latency of memory allocation
- 5 new summary tables
  - **memory_summary_by_account_by_event_name**
  - **memory_summary_by_host_by_event_name**
  - **memory_summary_by_thread_by_event_name**
  - **memory_summary_by_user_by_event_name**
  - **memory_summary_global_by_event_name**

p_s.memory_summary_global_by_event_name

```
mysql> select * from performance_schema.memory_summary_global_by_event_name
    -> order by current_number_of_bytes_used desc limit 1\G
*************************** 1. row ***************************
                  EVENT_NAME: memory/innodb/buf_buf_pool
                 COUNT_ALLOC: 1
                  COUNT_FREE: 0
    SUM_NUMBER_OF_BYTES_ALLOC: 137428992
     SUM_NUMBER_OF_BYTES_FREE: 0
               LOW_COUNT_USED: 0
           CURRENT_COUNT_USED: 1
              HIGH_COUNT_USED: 1
     LOW_NUMBER_OF_BYTES_USED: 0
CURRENT_NUMBER_OF_BYTES_USED: 137428992
    HIGH_NUMBER_OF_BYTES_USED: 137428992
```

InnoDB Buffer Pool

137.429 Megabytes

sys.memory_global_by_current_bytes view

```
 mysql> select * from sys.memory_global_by_current_bytes\G
*************************** 1. row ***************************
       event_name: memory/innodb/buf_buf_pool
    current_count: 1
    current_alloc: 131.06 MiB
current_avg_alloc: 131.06 MiB
       high_count: 1
       high_alloc: 131.06 MiB
   high_avg_alloc: 131.06 MiB
*************************** 2. row ***************************
       event_name: memory/innodb/log0log
    current_count: 9
    current_alloc: 16.01 MiB
current_avg_alloc: 1.78 MiB
       high_count: 9
       high_alloc: 16.01 MiB
   high_avg_alloc: 1.78 MiB
                ...
```

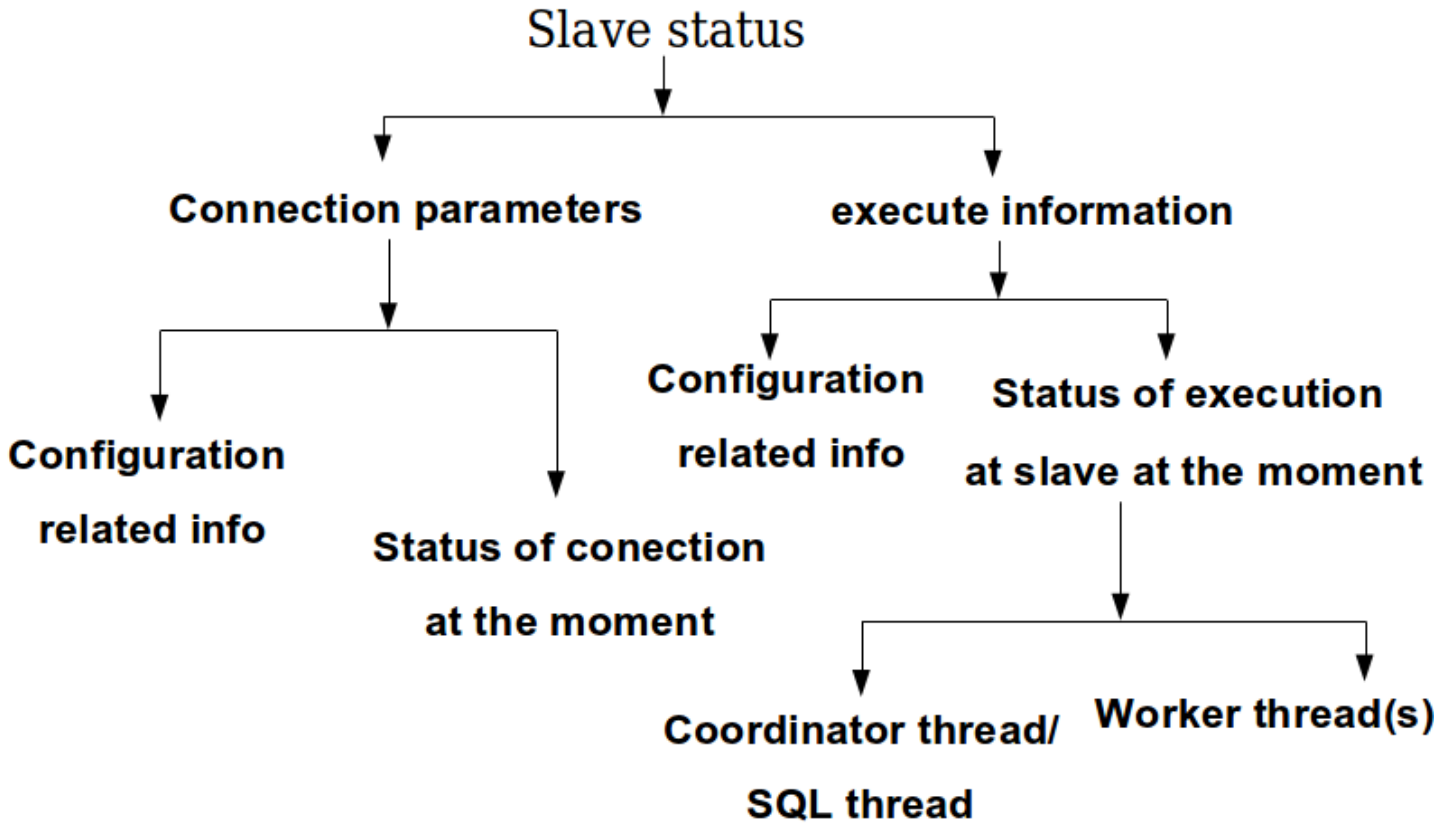
sys.user_summary view

```
mysql> select * from sys.user_summary\G
*************************** 1. row ***************************
                    user: mark
              statements: 3072
      statement_latency: 1.77 s
  statement_avg_latency: 575.29 us
            table_scans: 7
               file_ios: 20043
        file_io_latency: 346.79 ms
    current_connections: 1
      total_connections: 1
           unique_hosts: 1
         current_memory: 515.81 KiB
total_memory_allocated: 30.69 MiB
```

WL#3656 - PERFORMANCE_SCHEMA table for SHOW SLAVE STATUS

- 6 new tables
  - **replication_connection_configuration**
  - **replication_connection_status**
  - **replication_execute_configuration**
  - **replication_execute_status**
  - **replication_execute_status_by_coordinator**
  - **replication_execute_status_by_worker**

SHOW SLAVE STATUS Split



Replication Connection Config

```
mysql> select * from replication_connection_configuration\G
*************************** 1. row ***************************
                 CHANNEL_NAME:
                         HOST: 127.0.0.1
                         PORT: 13253
                         USER: rsandbox
              NETWORK_INTERFACE:
                  AUTO_POSITION: 1
                    SSL_ALLOWED: NO
                    SSL_CA_FILE:
                    SSL_CA_PATH:
                SSL_CERTIFICATE:
                     SSL_CIPHER:
                        SSL_KEY:
SSL_VERIFY_SERVER_CERTIFICATE: NO
                SSL_CRL_FILE:
                SSL_CRL_PATH:
    CONNECTION_RETRY_INTERVAL: 60
       CONNECTION_RETRY_COUNT: 86400
           HEARTBEAT_INTERVAL: 30.000
*************************** 2. row ***************************
                 CHANNEL_NAME: 56-cluster
```

One row per replication channel when using multi-master replication

Replication Connection Status

```
mysql> select * from performance_schema.replication_connection_status\G
*************************** 1. row ***************************
             CHANNEL_NAME:
               GROUP_NAME:
              SOURCE_UUID: 00013253-1111-1111-1111-111111111111
                THREAD_ID: 23
            SERVICE_STATE: ON
COUNT_RECEIVED_HEARTBEATS: 27
 LAST_HEARTBEAT_TIMESTAMP: 2015-09-18 15:01:00
 RECEIVED_TRANSACTION_SET: <GTID SET>
        LAST_ERROR_NUMBER: 0
       LAST_ERROR_MESSAGE:
     LAST_ERROR_TIMESTAMP: 0000-00-00 00:00:00
*************************** 2. row ***************************
             CHANNEL_NAME: 56-cluster
               GROUP_NAME:
              SOURCE_UUID: 16d226ee-f89b-11e4-9d4f-b36d08df49cf
                THREAD_ID: 25
            SERVICE_STATE: ON
COUNT_RECEIVED_HEARTBEATS: 27
                           ...
```

Replication Applier Status Config / State

```
mysql> select * from performance_schema.replication_applier_configuration;
+--------------+---------------+
| CHANNEL_NAME | DESIRED_DELAY |
+--------------+---------------+
|              |             0 |
| 56-cluster   |             0 |
+--------------+---------------+
```

One row per replication channel when using multi-master replication

```
mysql> select * from performance_schema.replication_applier_status;
+--------------+---------------+-----------------+----------------------------+
| CHANNEL_NAME | SERVICE_STATE | REMAINING_DELAY | COUNT_TRANSACTIONS_RETRIES |
+--------------+---------------+-----------------+----------------------------+
|              | ON            |            NULL |                          0 |
| 56-cluster   | ON            |            NULL |                          0 |
+--------------+---------------+-----------------+----------------------------+
```

Replication Applier Status Details

```
mysql> select * from performance_schema.replication_applier_status_by_coordinator;
+--------------+-----------+---------------+--------------------+--------------------+----------------------+
| CHANNEL_NAME | THREAD_ID | SERVICE_STATE | LAST_ERROR_NUMBER | LAST_ERROR_MESSAGE | LAST_ERROR_TIMESTAMP |
+--------------+-----------+---------------+--------------------+--------------------+----------------------+
|              |        24 | ON            |                  0 |                    | 0000-00-00 00:00:00  |
| 56-cluster   |        26 | ON            |                  0 |                    | 0000-00-00 00:00:00  |
+--------------+-----------+---------------+--------------------+--------------------+----------------------+
```

One row per channel use for all applier state when not using parallel workers

One row per parallel worker, across all replication channels

```
mysql> select * from performance_schema.replication_applier_status_by_worker;
+--------------+-----------+-----------+---------------+-----------------------+-------------------+--------------------+----------------------+
| CHANNEL_NAME | WORKER_ID | THREAD_ID | SERVICE_STATE | LAST_SEEN_TRANSACTION | LAST_ERROR_NUMBER | LAST_ERROR_MESSAGE | LAST_ERROR_TIMESTAMP |
+--------------+-----------+-----------+---------------+-----------------------+-------------------+--------------------+----------------------+
|              |         1 |        27 | ON            | <GTID SET>            |                 0 |                    | 0000-00-00 00:00:00  |
|              |         2 |        29 | ON            | <GTID SET>            |                 0 |                    | 0000-00-00 00:00:00  |
|              |         3 |        31 | ON            | <GTID SET>            |                 0 |                    | 0000-00-00 00:00:00  |
|              |         4 |        33 | ON            | <GTID SET>            |                 0 |                    | 0000-00-00 00:00:00  |
| 56-cluster   |         1 |        28 | ON            | <GTID SET>            |                 0 |                    | 0000-00-00 00:00:00  |
| 56-cluster   |         2 |        30 | ON            | <GTID SET>            |                 0 |                    | 0000-00-00 00:00:00  |
| 56-cluster   |         3 |        32 | ON            | <GTID SET>            |                 0 |                    | 0000-00-00 00:00:00  |
| 56-cluster   |         4 |        34 | ON            | <GTID SET>            |                 0 |                    | 0000-00-00 00:00:00  |
+--------------+-----------+-----------+---------------+-----------------------+-------------------+--------------------+----------------------+
```

"Unfortunately, it’s unlikely that I’ll be able to create a reproducible test case, because there’s no way to actually see what is happening. I hope that a future version of MySQL will include a more comprehensive set of tables for inspecting locks, requests, and waits at all layers of the server."

Baron Schwartz, CEO, VividCortex

WL#5879 - PERFORMANCE_SCHEMA, MDL lock instrumentation

WL#5879 - PERFORMANCE_SCHEMA, MDL lock instrumentation

- Added the **wait/lock/metadata/sql/mdl** instrument
- 2 new current instance tables
  - **metadata_locks**
  - **table_handles**

p_s.metadata_locks structure

Field	Type	Null	Key	Default	Extra
OBJECT_TYPE	varchar(64)	NO		NULL	
OBJECT_SCHEMA	varchar(64)	YES		NULL	
OBJECT_NAME	varchar(64)	YES		NULL	
OBJECT_INSTANCE_BEGIN	bigint(20) unsigned	NO		NULL	
LOCK_TYPE	varchar(32)	NO		NULL	
LOCK_DURATION	varchar(32)	NO		NULL	
LOCK_STATUS	varchar(32)	NO		NULL	
SOURCE	varchar(64)	YES		NULL	
OWNER_THREAD_ID	bigint(20) unsigned	YES		NULL	
OWNER_EVENT_ID	bigint(20) unsigned	YES		NULL	

p_s.metadata_locks sample output

```
mysql> select object_type as scope, object_schema, object_name, lock_type, lock_duration, lock_status
    ->   from metadata_locks
    ->  order by object_type = 'global' desc, object_type = 'schema' desc,
    ->          object_type = 'table' desc, object_type = 'commit' desc;
+--------+--------------------+----------------+---------------------+--------------+-----------+
| scope  | object_schema      | object_name    | lock_type           | lock_duration | lock_status |
+--------+--------------------+----------------+---------------------+--------------+-----------+
| GLOBAL | NULL               | NULL           | SHARED              | EXPLICIT     | GRANTED   |
| GLOBAL | NULL               | NULL           | INTENTION_EXCLUSIVE | STATEMENT    | PENDING   |
| GLOBAL | NULL               | NULL           | INTENTION_EXCLUSIVE | STATEMENT    | PENDING   |
| GLOBAL | NULL               | NULL           | INTENTION_EXCLUSIVE | STATEMENT    | PENDING   |
| GLOBAL | NULL               | NULL           | INTENTION_EXCLUSIVE | STATEMENT    | PENDING   |
| GLOBAL | NULL               | NULL           | INTENTION_EXCLUSIVE | STATEMENT    | PENDING   |
| GLOBAL | NULL               | NULL           | INTENTION_EXCLUSIVE | STATEMENT    | PENDING   |
| GLOBAL | NULL               | NULL           | INTENTION_EXCLUSIVE | STATEMENT    | PENDING   |
|
| TABLE  | mem__events        | action_logs    | SHARED_READ         | TRANSACTION  | GRANTED   |
| TABLE  | mem__events        | events         | SHARED_READ         | TRANSACTION  | GRANTED   |
| TABLE  | performance_schema | metadata_locks | SHARED_READ         | TRANSACTION  | GRANTED   |
| COMMIT | NULL               | NULL           | SHARED              | EXPLICIT     | GRANTED   |
+--------+--------------------+----------------+---------------------+--------------+-----------+
```

sys.schema_table_lock_waits

```
mysql> select * from sys.schema_table_lock_waits\G
*************************** 1. row ***************************
               object_schema: test
                 object_name: t
           waiting_thread_id: 43
                 waiting_pid: 21
             waiting_account: msandbox@localhost
           waiting_lock_type: SHARED_UPGRADABLE
       waiting_lock_duration: TRANSACTION
               waiting_query: alter table test.t add foo int
          waiting_query_secs: 988
 waiting_query_rows_affected: 0
 waiting_query_rows_examined: 0
           blocking_thread_id: 42
                 blocking_pid: 20
             blocking_account: msandbox@localhost
           blocking_lock_type: SHARED_NO_READ_WRITE
       blocking_lock_duration: TRANSACTION
     sql_kill_blocking_query: KILL QUERY 20
sql_kill_blocking_connection: KILL 20
```

sys.innodb_lock_waits

```
mysql> SELECT * FROM sys.innodb_lock_waits\G
*************************** 1. row ***************************
                wait_started: 2014-11-11 13:39:20
                    wait_age: 00:00:07
               wait_age_secs: 7
                locked_table: `db1`.`t1`
                locked_index: PRIMARY
                 locked_type: RECORD
              waiting_trx_id: 867158
         waiting_trx_started: 2014-11-11 13:39:15
             waiting_trx_age: 00:00:12
     waiting_trx_rows_locked: 0
   waiting_trx_rows_modified: 0
                 waiting_pid: 3
               waiting_query: UPDATE t1 SET val = val + 1 WHERE id = 2
             waiting_lock_id: 867158:2363:3:3
           waiting_lock_mode: X
             blocking_trx_id: 867157
                blocking_pid: 4
                    ...
                blocking_query: UPDATE t1 SET val …
              blocking_lock_id: 867157:2363:3:3
            blocking_lock_mode: X
          blocking_trx_started: 2014-11-11 13:39:11
              blocking_trx_age: 00:00:16
    blocking_trx_rows_locked: 1
  blocking_trx_rows_modified: 1
     sql_kill_blocking_query: KILL QUERY 4
sql_kill_blocking_connection: KILL 4
```

"Suppose I accessed your database and listed the entire set of stored functions and procedures. How many of them are you not even sure are in use anymore? How many of them you think you can DROP, but are too afraid to, and keep them in *just in case*?"

Shlomi Noach

WL#5766 - PERFORMANCE_SCHEMA, Stored programs instrumentation

http://code.openark.org/blog/mysql/why-delegating-code-to-mysql-stored-routines-is-poor-engineering-practice

WL#5766 - Stored programs instrumentation

- Tracks **Stored Procedures, Stored Functions, Triggers and Events**
- Added 16 new **statement/sp/%** instruments
  - Expose the different work flows that stored programs use, such as cursor operations, workflow controls etc.
  - Integrated in to the normal statement instrumentation
- 1 new summary table
  - **events_statements_summary_by_program**

p_s.memory_summary_global_by_event_name

```
mysql> select * from events_statements_summary_by_program\G
*************************** 1. row ***************************
                 OBJECT_TYPE: PROCEDURE
               OBJECT_SCHEMA: ps_demo
                 OBJECT_NAME: ps_demo_proc
                  COUNT_STAR: 1
              SUM_TIMER_WAIT: 6970931000
              MIN_TIMER_WAIT: 6970931000
              AVG_TIMER_WAIT: 6970931000
              MAX_TIMER_WAIT: 6970931000
             COUNT_STATEMENTS: 5
         SUM_STATEMENTS_WAIT: 6802181000
         MIN_STATEMENTS_WAIT: 16372000
         AVG_STATEMENTS_WAIT: 1360436000
         MAX_STATEMENTS_WAIT: 6484366000
               SUM_LOCK_TIME: 176401000000
                  SUM_ERRORS: 0
                SUM_WARNINGS: 0
             SUM_ROWS_AFFECTED: 1
                 SUM_ROWS_SENT: 0
             SUM_ROWS_EXAMINED: 1
SUM_CREATED_TMP_DISK_TABLES: 0
     SUM_CREATED_TMP_TABLES: 0
       SUM_SELECT_FULL_JOIN: 0
 SUM_SELECT_FULL_RANGE_JOIN: 0
          SUM_SELECT_RANGE: 0
     SUM_SELECT_RANGE_CHECK: 0
            SUM_SELECT_SCAN: 0
      SUM_SORT_MERGE_PASSES: 0
           SUM_SORT_RANGE: 0
            SUM_SORT_ROWS: 0
            SUM_SORT_SCAN: 0
        SUM_NO_INDEX_USED: 0
   SUM_NO_GOOD_INDEX_USED: 0
```

WL#5864 - PERFORMANCE SCHEMA, instrument TRANSACTIONS

- Added the **transaction** instrument
- 8 new raw and summary tables
  - **events_transactions_current**
  - **events_transactions_history**
  - **events_transactions_history_long**
  - **events_transactions_summary_by_account_by_event_name**
  - **events_transactions_summary_by_host_by_event_name**
  - **events_transactions_summary_by_thread_by_event_name**
  - **events_transactions_summary_by_user_by_event_name**
  - **events_transactions_summary_global_by_event_name**

WL#5864 - PERFORMANCE SCHEMA, instrument TRANSACTIONS

- Exposes details of transactions in raw or summary views
- Show details such as
  - Transaction latency
  - Isolation levels
  - Auto commit
  - Savepoint info
  - GTID or transaction IDs
- Link transactions to the statements within the transactions

p_s.events_transactions_current

```
mysql> select * from events_transactions_current\G
*************************** 1. row ***************************
                    THREAD_ID: 1
                     EVENT_ID: 23733
                 END_EVENT_ID: 23742
                   EVENT_NAME: transaction
                        STATE: COMMITTED
                       TRX_ID: 281479898269256
                         GTID: NULL
                          XID: NULL
                     XA_STATE: NULL
                       SOURCE: handler.cc:1246
                  TIMER_START: 31140612726000
                    TIMER_END: 31140647445000
                   TIMER_WAIT: 34719000
                  ACCESS_MODE: READ WRITE
              ISOLATION_LEVEL: REPEATABLE READ
                   AUTOCOMMIT: YES
           NUMBER_OF_SAVEPOINTS: 0
NUMBER_OF_ROLLBACK_TO_SAVEPOINT: 0
    NUMBER_OF_RELEASE_SAVEPOINT: 0
          OBJECT_INSTANCE_BEGIN: NULL
               NESTING_EVENT_ID: NULL
             NESTING_EVENT_TYPE: NULL
```

sys.ps_thread_trx_info(<thread_id>)

```
SELECT sys.ps_thread_trx_info(48) as trx_info\G
*************************** 1. row ***************************
trx_info: [
  {
    "time": "790.70 us",
    "state": "COMMITTED",
    "mode": "READ WRITE",
    "autocommitted": "NO",
    "gtid": "AUTOMATIC",
    "isolation": "REPEATABLE READ",
    "statements_executed": [
      {
        "sql_text": "INSERT INTO info VALUES (1, 'foo')",
        "time": "471.02 us",
        "schema": "trx",
        "rows_examined": 0,
        "rows_affected": 1,
        "rows_sent": 0,
        "tmp_tables": 0,
        "tmp_disk_tables": 0,
        "sort_rows": 0,
        "sort_merge_passes": 0
      },
      ...
```

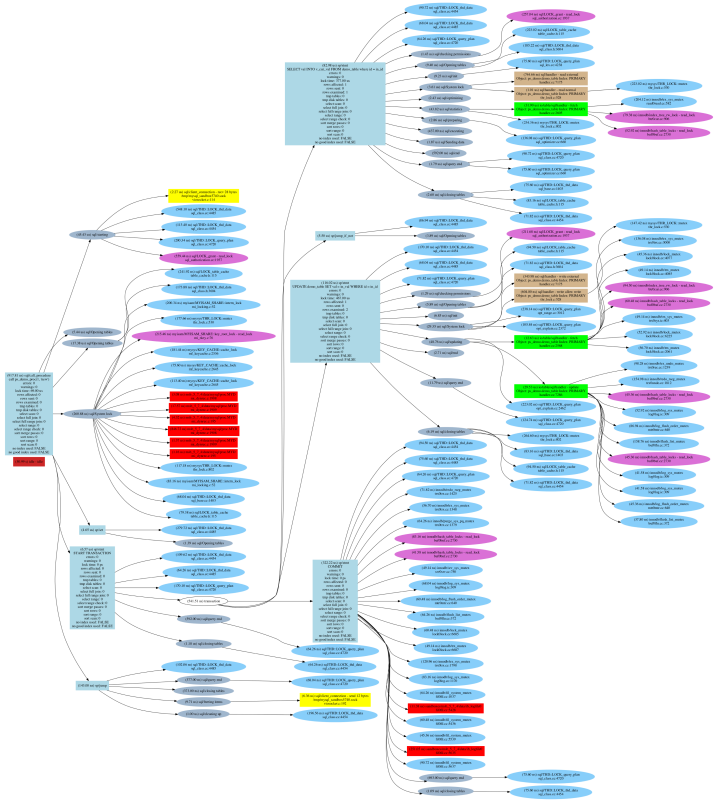
sys.ps_trace_thread procedure

- **ps_trace_thread()** monitors a specific thread for a period
- Captures as much information on the thread activity as possible
- Returns a “dot” formatted file, that can graph the event hierarchy
  - http://en.wikipedia.org/wiki/DOT_(graph_description_language)

sys.ps_trace_thread procedure example

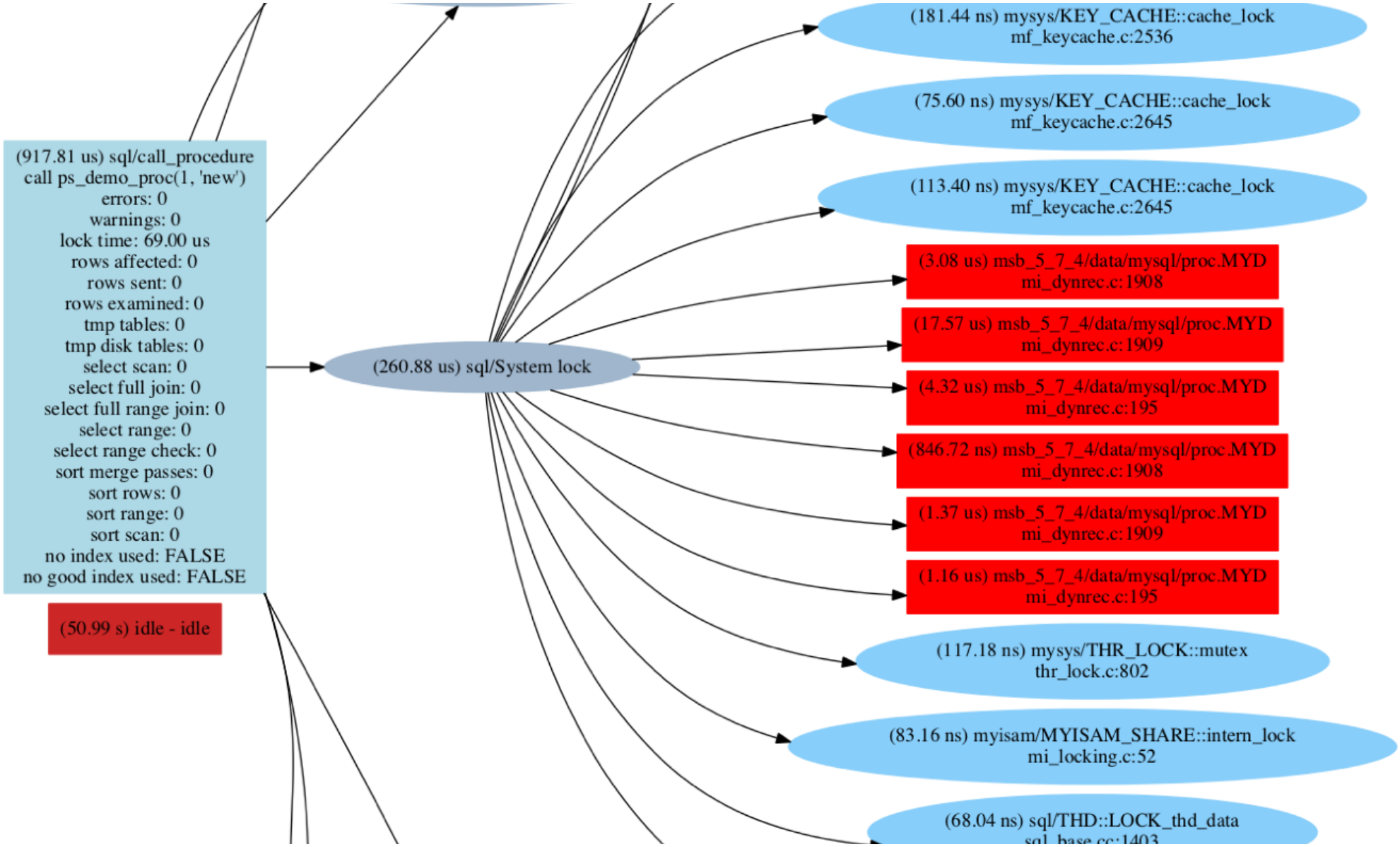
```
mysql> call ps_trace_thread(27768, '/tmp/stack_27768.dot', 60, 0.1, true, true, true);
+-------------------------------------------+
| Info                                      |
+-------------------------------------------+
| Data collection starting for THREAD_ID = 27768 |
+-------------------------------------------+
1 row in set (4.82 sec)

+-----------------------------------------+
| Info                                    |
+-----------------------------------------+
| Stack trace written to /tmp/stack_27768.dot |
+-----------------------------------------+
1 row in set (60.90 sec)

+-----------------------------------------------------+
| Convert to PDF                                      |
+-----------------------------------------------------+
| dot -Tpdf -o /tmp/stack_27768.pdf /tmp/stack_27768.dot |
+-----------------------------------------------------+
1 row in set (60.90 sec)

+-----------------------------------------------------+
| Convert to PNG                                      |
+-----------------------------------------------------+
| dot -Tpng -o /tmp/stack_27768.png /tmp/stack_27768.dot |
+-----------------------------------------------------+
1 row in set (60.90 sec)+
```

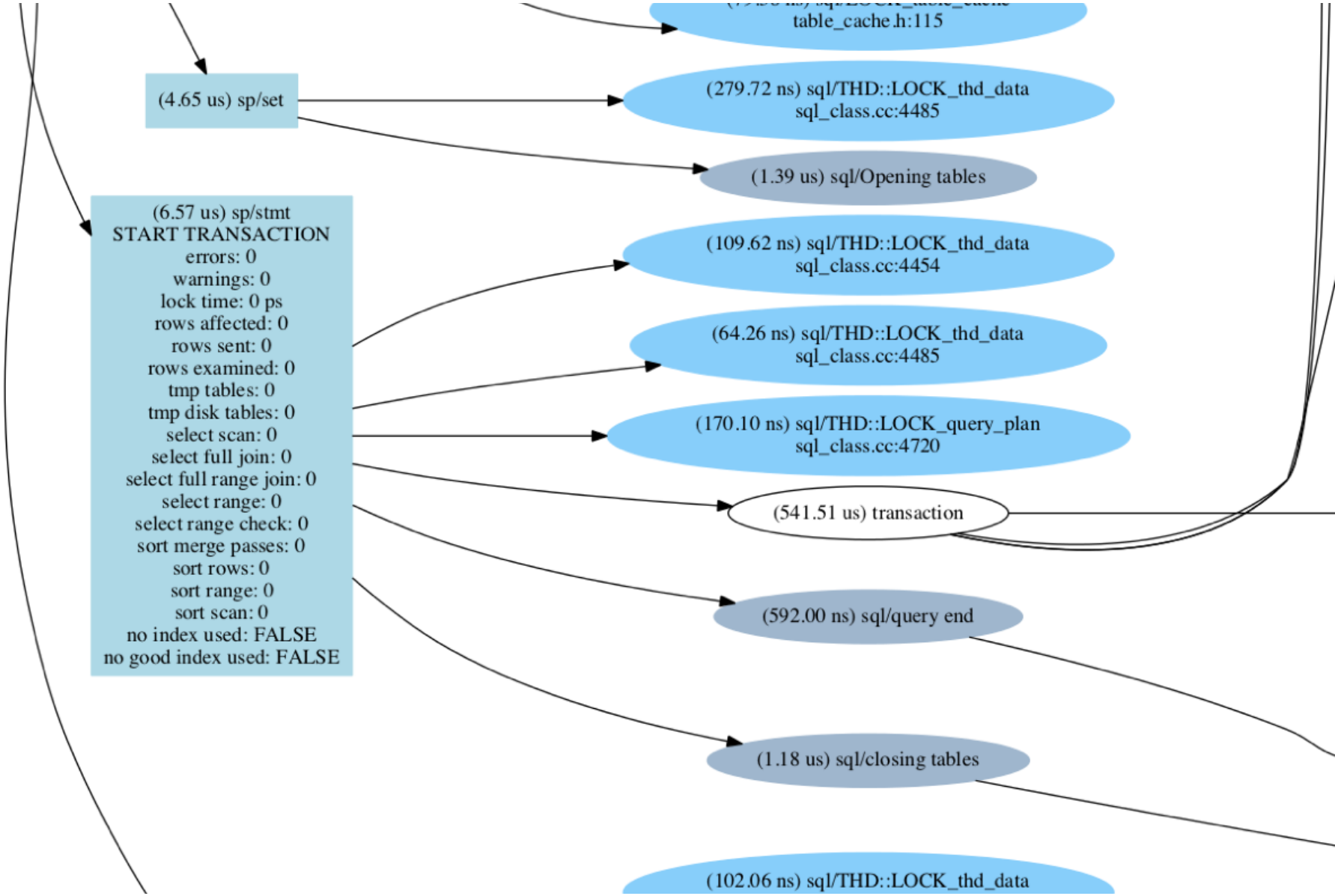
ps_trace_thread procedure example

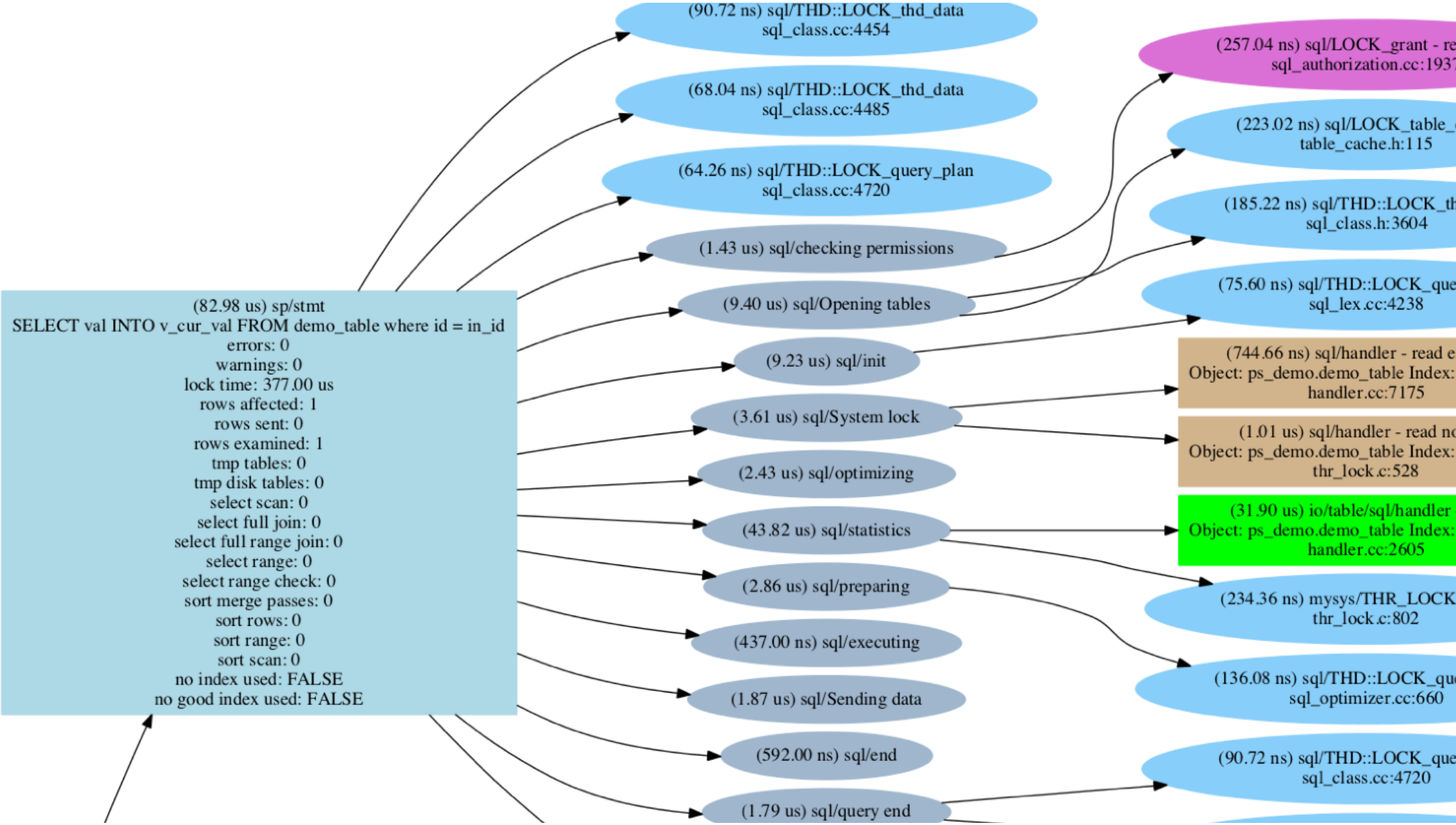


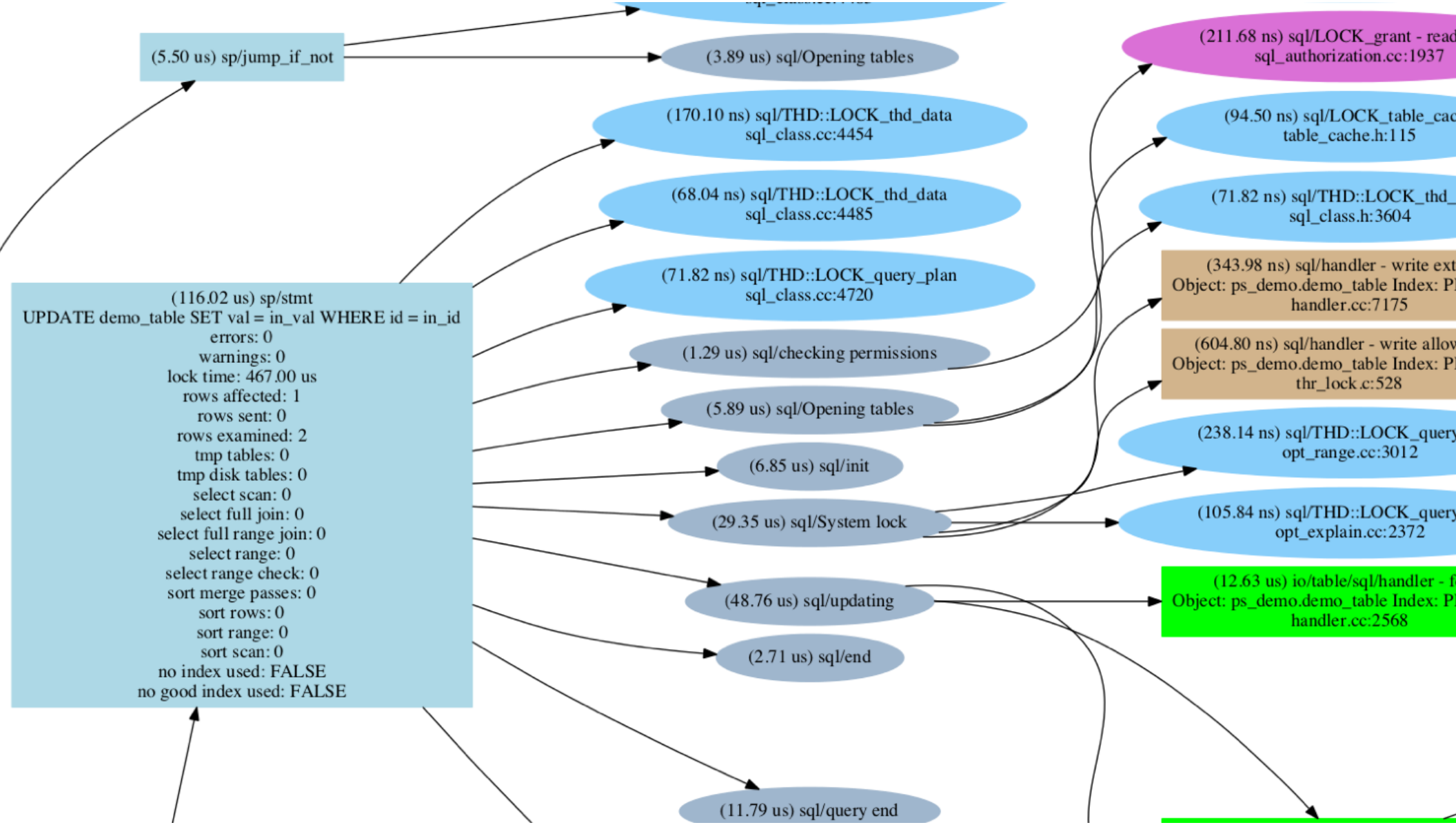
MySQL 5.7

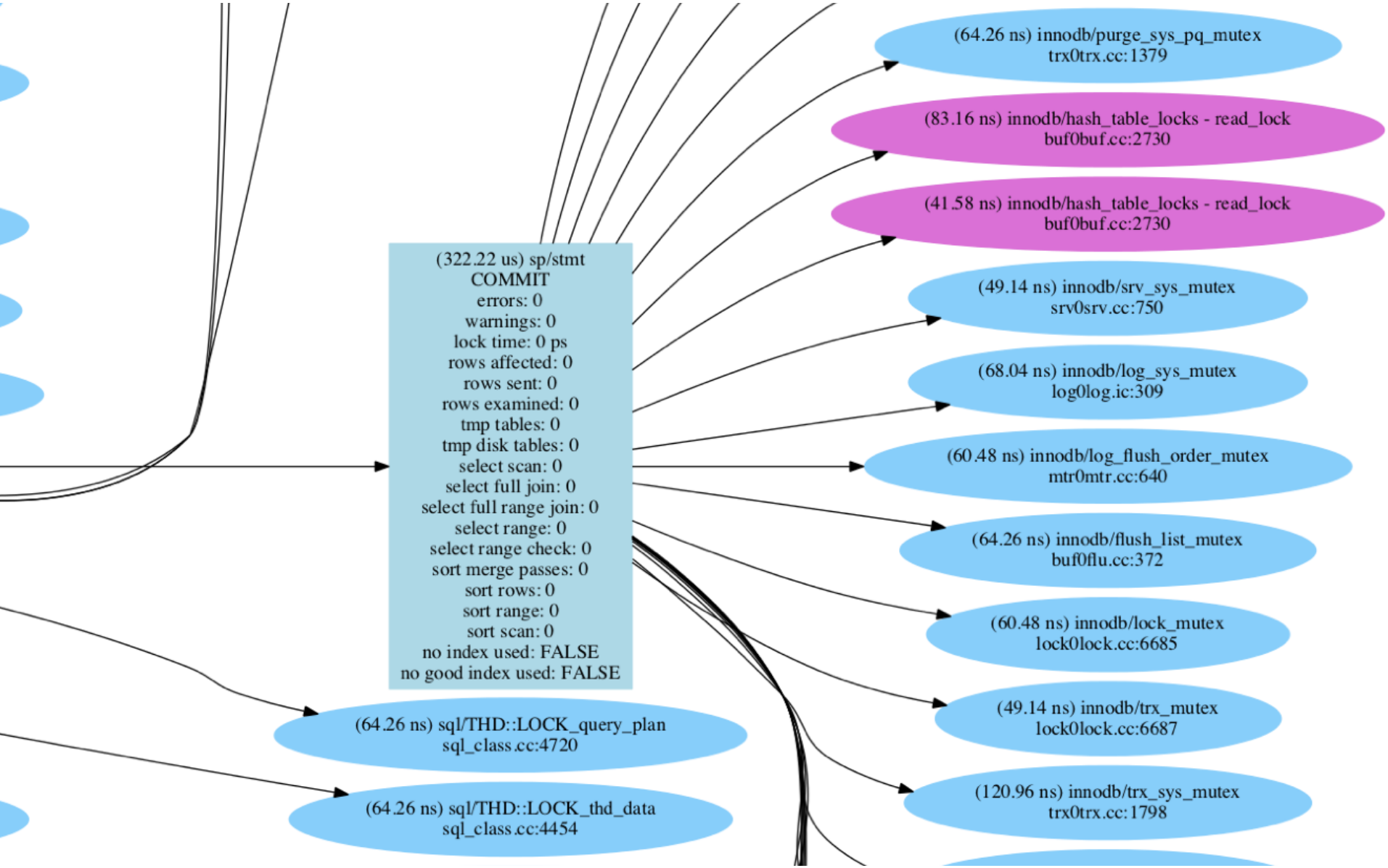
Procedures
Transactions
Statements
Stages
Waits

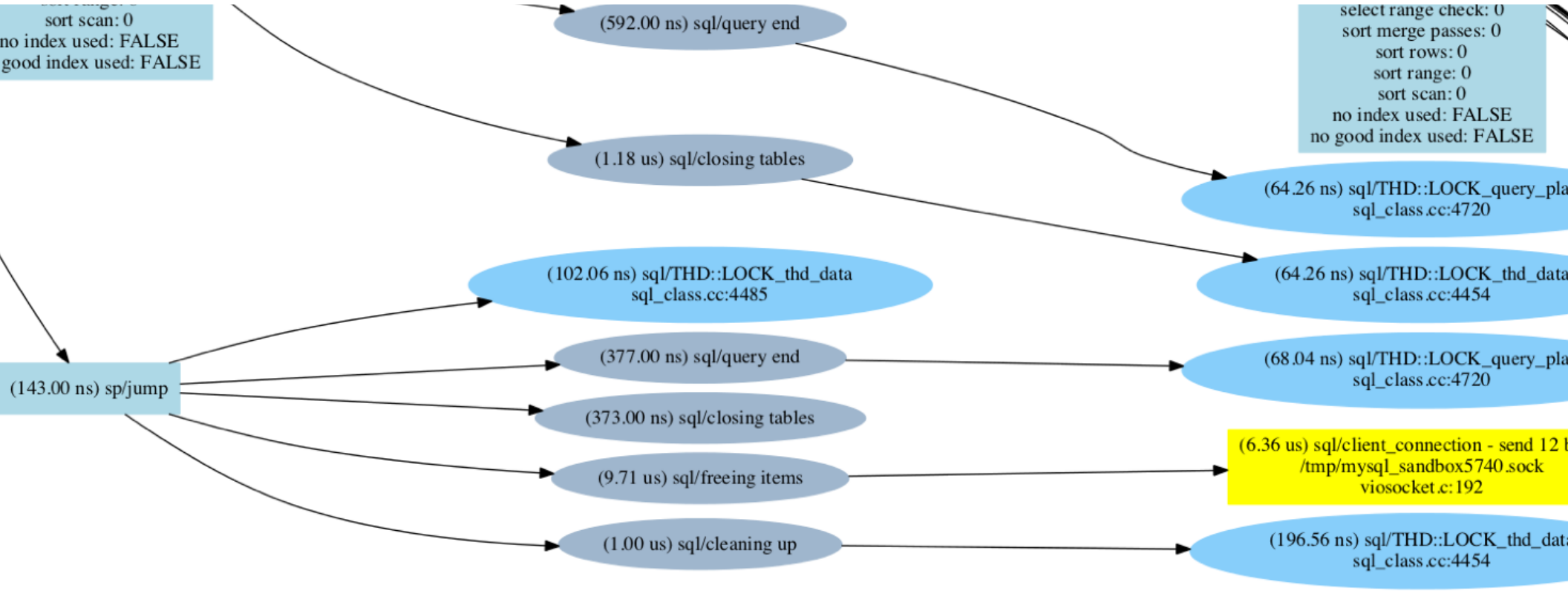

(917.81 us) sql/call_procedure
call ps_demo_proc(1, 'new')
errors: 0
warnings: 0
lock time: 69.00 us
rows affected: 0
rows sent: 0
rows examined: 0
tmp tables: 0
tmp disk tables: 0
select scan: 0
select full join: 0
select full range join: 0
select range: 0
select range check: 0
sort merge passes: 0
sort rows: 0
sort range: 0
sort scan: 0
no index used: FALSE
no good index used: FALSE
(50.99 s) idle - idle
(260.88 us) sql/System lock
(181.44 ns) mysys/KEY_CACHE::cache_lock
mf_keycache.c:2536
(75.60 ns) mysys/KEY_CACHE::cache_lock
mf_keycache.c:2645
(113.40 ns) mysys/KEY_CACHE::cache_lock
mf_keycache.c:2645
(3.08 us) msb_5_7_4/data/mysql/proc.MYD
mi_dynrec.c:1908
(17.57 us) msb_5_7_4/data/mysql/proc.MYD
mi_dynrec.c:1909
(4.32 us) msb_5_7_4/data/mysql/proc.MYD
mi_dynrec.c:195
(846.72 ns) msb_5_7_4/data/mysql/proc.MYD
mi_dynrec.c:1908
(1.37 us) msb_5_7_4/data/mysql/proc.MYD
mi_dynrec.c:1909
(1.16 us) msb_5_7_4/data/mysql/proc.MYD
mi_dynrec.c:195
(117.18 ns) mysys/THR_LOCK::mutex
thr_lock.c:802
(83.16 ns) myisam/MYISAM_SHARE::intern_lock
mi_locking.c:52
(68.04 ns) sql/THD::LOCK_thd_data


table_cache.h:115
(4.65 us) sp/set
(279.72 ns) sql/THD::LOCK_thd_data
sql_class.cc:4485
(1.39 us) sql/Opening tables
(6.57 us) sp/stmt
START TRANSACTION
errors: 0
warnings: 0
lock time: 0 ps
rows affected: 0
rows sent: 0
rows examined: 0
tmp tables: 0
tmp disk tables: 0
select scan: 0
select full join: 0
select full range join: 0
select range: 0
select range check: 0
sort merge passes: 0
sort rows: 0
sort range: 0
sort scan: 0
no index used: FALSE
no good index used: FALSE
(109.62 ns) sql/THD::LOCK_thd_data
sql_class.cc:4454
(64.26 ns) sql/THD::LOCK_thd_data
sql_class.cc:4485
(170.10 ns) sql/THD::LOCK_query_plan
sql_class.cc:4720
(541.51 us) transaction
(592.00 ns) sql/query end
(1.18 us) sql/closing tables
(102.06 ns) sql/THD::LOCK_thd_data


(82.98 us) sp/stmt
SELECT val INTO v_cur_val FROM demo_table where id = in_id
errors: 0
warnings: 0
lock time: 377.00 us
rows affected: 1
rows sent: 0
rows examined: 1
tmp tables: 0
tmp disk tables: 0
select scan: 0
select full join: 0
select full range join: 0
select range: 0
select range check: 0
sort merge passes: 0
sort rows: 0
sort range: 0
sort scan: 0
no index used: FALSE
no good index used: FALSE
(90.72 ns) sql/THD::LOCK_thd_data
sql_class.cc:4454
(68.04 ns) sql/THD::LOCK_thd_data
sql_class.cc:4485
(64.26 ns) sql/THD::LOCK_query_plan
sql_class.cc:4720
(1.43 us) sql/checking permissions
(9.40 us) sql/Opening tables
(9.23 us) sql/init
(3.61 us) sql/System lock
(2.43 us) sql/optimizing
(43.82 us) sql/statistics
(2.86 us) sql/preparing
(437.00 ns) sql/executing
(1.87 us) sql/Sending data
(592.00 ns) sql/end
(1.79 us) sql/query end
(257.04 ns) sql/LOCK_grant - re
sql_authorization.cc:193
(223.02 ns) sql/LOCK_table_
table_cache.h:115
(185.22 ns) sql/THD::LOCK_th
sql_class.h:3604
(75.60 ns) sql/THD::LOCK_que
sql_lex.cc:4238
(744.66 ns) sql/handler - read e
Object: ps_demo.demo_table Index:
handler.cc:7175
(1.01 us) sql/handler - read ne
Object: ps_demo.demo_table Index:
thr_lock.c:528
(31.90 us) io/table/sql/handler
Object: ps_demo.demo_table Index:
handler.cc:2605
(234.36 ns) mysys/THR_LOCK
thr_lock.c:802
(136.08 ns) sql/THD::LOCK_qu
sql_optimizer.cc:660
(90.72 ns) sql/THD::LOCK_que
sql_class.cc:4720


(5.50 us) sp/jump_if_not
(3.89 us) sql/Opening tables
(170.10 ns) sql/THD::LOCK_thd_data
sql_class.cc:4454
(68.04 ns) sql/THD::LOCK_thd_data
sql_class.cc:4485
(71.82 ns) sql/THD::LOCK_query_plan
sql_class.cc:4720
(116.02 us) sp/stmt
UPDATE demo_table SET val = in_val WHERE id = in_id
errors: 0
warnings: 0
lock time: 467.00 us
rows affected: 1
rows sent: 0
rows examined: 2
tmp tables: 0
tmp disk tables: 0
select scan: 0
select full join: 0
select full range join: 0
select range: 0
select range check: 0
sort merge passes: 0
sort rows: 0
sort range: 0
sort scan: 0
no index used: FALSE
no good index used: FALSE
(1.29 us) sql/checking permissions
(5.89 us) sql/Opening tables
(6.85 us) sql/init
(29.35 us) sql/System lock
(48.76 us) sql/updating
(2.71 us) sql/end
(11.79 us) sql/query end
(211.68 ns) sql/LOCK_grant - read
sql_authorization.cc:1937
(94.50 ns) sql/LOCK_table_cac
table_cache.h:115
(71.82 ns) sql/THD::LOCK_thd_
sql_class.h:3604
(343.98 ns) sql/handler - write ext
Object: ps_demo.demo_table Index: P
handler.cc:7175
(604.80 ns) sql/handler - write allow
Object: ps_demo.demo_table Index: P
thr_lock.c:528
(238.14 ns) sql/THD::LOCK_query
opt_range.cc:3012
(105.84 ns) sql/THD::LOCK_query
opt_explain.cc:2372
(12.63 us) io/table/sql/handler - f
Object: ps_demo.demo_table Index: P
handler.cc:2568


(322.22 us) sp/stmt
COMMIT
errors: 0
warnings: 0
lock time: 0 ps
rows affected: 0
rows sent: 0
rows examined: 0
tmp tables: 0
tmp disk tables: 0
select scan: 0
select full join: 0
select full range join: 0
select range: 0
select range check: 0
sort merge passes: 0
sort rows: 0
sort range: 0
sort scan: 0
no index used: FALSE
no good index used: FALSE
(64.26 ns) innodb/purge_sys_pq_mutex
trx0trx.cc:1379
(83.16 ns) innodb/hash_table_locks - read_lock
buf0buf.cc:2730
(41.58 ns) innodb/hash_table_locks - read_lock
buf0buf.cc:2730
(49.14 ns) innodb/srv_sys_mutex
srv0srv.cc:750
(68.04 ns) innodb/log_sys_mutex
log0log.ic:309
(60.48 ns) innodb/log_flush_order_mutex
mtr0mtr.cc:640
(64.26 ns) innodb/flush_list_mutex
buf0flu.cc:372
(60.48 ns) innodb/lock_mutex
lock0lock.cc:6685
(49.14 ns) innodb/trx_mutex
lock0lock.cc:6687
(120.96 ns) innodb/trx_sys_mutex
trx0trx.cc:1798
(64.26 ns) sql/THD::LOCK_query_plan
sql_class.cc:4720
(64.26 ns) sql/THD::LOCK_thd_data
sql_class.cc:4454


sort scan: 0
no index used: FALSE
good index used: FALSE
(592.00 ns) sql/query end
select range check: 0
sort merge passes: 0
sort rows: 0
sort range: 0
sort scan: 0
no index used: FALSE
no good index used: FALSE
(1.18 us) sql/closing tables
(64.26 ns) sql/THD::LOCK_query_pla
sql_class.cc:4720
(102.06 ns) sql/THD::LOCK_thd_data
sql_class.cc:4485
(64.26 ns) sql/THD::LOCK_thd_data
sql_class.cc:4454
(143.00 ns) sp/jump
(377.00 ns) sql/query end
(68.04 ns) sql/THD::LOCK_query_pla
sql_class.cc:4720
(373.00 ns) sql/closing tables
(6.36 us) sql/client_connection - send 12 b
/tmp/mysql_sandbox5740.sock
viosocket.c:192
(9.71 us) sql/freeing items
(1.00 us) sql/cleaning up
(196.56 ns) sql/THD::LOCK_thd_dat
sql_class.cc:4454

p_s.prepared_statement_instances

```
mysql> select * from performance_schema.prepared_statements_instances\G
*************************** 1. row ***************************
      OBJECT_INSTANCE_BEGIN: 140198306602144
               STATEMENT_ID: 1
             STATEMENT_NAME: stmt1
                   SQL_TEXT: select * from test.t1
             OWNER_THREAD_ID: 54003
              OWNER_EVENT_ID: 935
            OWNER_OBJECT_TYPE: NULL
          OWNER_OBJECT_SCHEMA: NULL
            OWNER_OBJECT_NAME: NULL
                TIMER_PREPARE: 15386274000
              COUNT_REPREPARE: 0
                COUNT_EXECUTE: 1
           SUM_TIMER_EXECUTE: 217699000
           MIN_TIMER_EXECUTE: 217699000
           AVG_TIMER_EXECUTE: 217699000
           MAX_TIMER_EXECUTE: 217699000
                SUM_LOCK_TIME: 0
                  SUM_ERRORS: 0
                SUM_WARNINGS: 0
           SUM_ROWS_AFFECTED: 0
               SUM_ROWS_SENT: 0
           SUM_ROWS_EXAMINED: 0
SUM_CREATED_TMP_DISK_TABLES: 0
     SUM_CREATED_TMP_TABLES: 0
       SUM_SELECT_FULL_JOIN: 0
SUM_SELECT_FULL_RANGE_JOIN: 0
         SUM_SELECT_RANGE: 0
      SUM_SELECT_RANGE_CHECK: 0
            SUM_SELECT_SCAN: 0
      SUM_SORT_MERGE_PASSES: 0
           SUM_SORT_RANGE: 0
            SUM_SORT_ROWS: 0
            SUM_SORT_SCAN: 0
        SUM_NO_INDEX_USED: 0
   SUM_NO_GOOD_INDEX_USED: 0
```

WL#6884 - PERFORMANCE SCHEMA, USER VARIABLES

```
mysql> set @debug := 1;
Query OK, 0 rows affected (0.10 sec)

mysql> set @totalRows := 300;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from performance_schema.user_variables_by_thread;
+-----------+---------------+----------------+
| THREAD_ID | VARIABLE_NAME | VARIABLE_VALUE |
+-----------+---------------+----------------+
|        33 | totalRows     | 300            |
|        33 | debug         | 1              |
|        34 | totalRows     | 150            |
+-----------+---------------+----------------+
```

WL#7415 - PERFORMANCE SCHEMA, STAGE PROGRESS

- Added the **WORK_COMPLETED** and **WORK_ESTIMATED** columns to the **events_stages_current** table
- Work is in units, and depends on the stage what it may be, rows copied, bytes processed, etc.
- Started with adding tracking to:
  - **stage/sql/copy to tmp table**
- *(more to come later in this presentation)*

WL#5889 - Add InnoDB events to Performance Schema's Event Stage table

- Follow up to **WL#7415 - PERFORMANCE SCHEMA, STAGE PROGRESS**
- Added stage progress instruments within InnoDB:
  - **stage/innodb/buffer pool load**
  - **stage/innodb/alter table (read PK and internal sort)**
  - **stage/innodb/alter table (merge sort)**
  - **stage/innodb/alter table (insert)**
  - **stage/innodb/alter table (flush)**
  - **stage/innodb/alter table (log apply index)**
  - **stage/innodb/alter table (log apply table)**
  - **stage/innodb/alter table (end)**

WL#5889 - Add InnoDB events to Performance Schema's Event Stage table

- InnoDB ALTER TABLE instruments work in tandem for the entire ALTER TABLE for example:

```
                              work_complete | work_estimated
'.../alter table (read..)'      0 .. 100    | 700
'.../alter table (merge..)'   101 .. 300    | 700
'.../alter table (insert)'    301 .. 500    | 700
'.../alter table (flush)'     500 .. 650    | 700
'.../alter table (end)'       651 .. 700    | 700
```

sys.sessions progress estimation

```
mysql> select * from sys.sessions\G
*************************** 1. row ***************************
                 thd_id: 44524
                conn_id: 44502
                   user: msandbox@localhost
                     db: test
                command: Query
                  state: alter table (flush)
                   time: 18
      current_statement: alter table t1 add column g int
      statement_latency: 18.45 s
               progress: 98.84
           lock_latency: 265.43 ms
          rows_examined: 0
              rows_sent: 0
          rows_affected: 0
             tmp_tables: 0
        tmp_disk_tables: 0
                    ...
```

WL#7800 - PERFORMANCE SCHEMA, SETUP_ACTORS ENABLED COLUMN

- The **performance_schema.setup_actors** table previously only worked for *including* users to instrument, but it was impossible to instrument all but one or two users easily
- Added **`ENABLED` enum('YES','NO')** to **setup_actors**
- To disable user "mark@localhost":
  - **INSERT INTO performance_schema.setup_actors VALUES ('localhost', 'mark', '%', 'NO');**

WL#6629 - PERFORMANCE_SCHEMA, STATUS VARIABLES

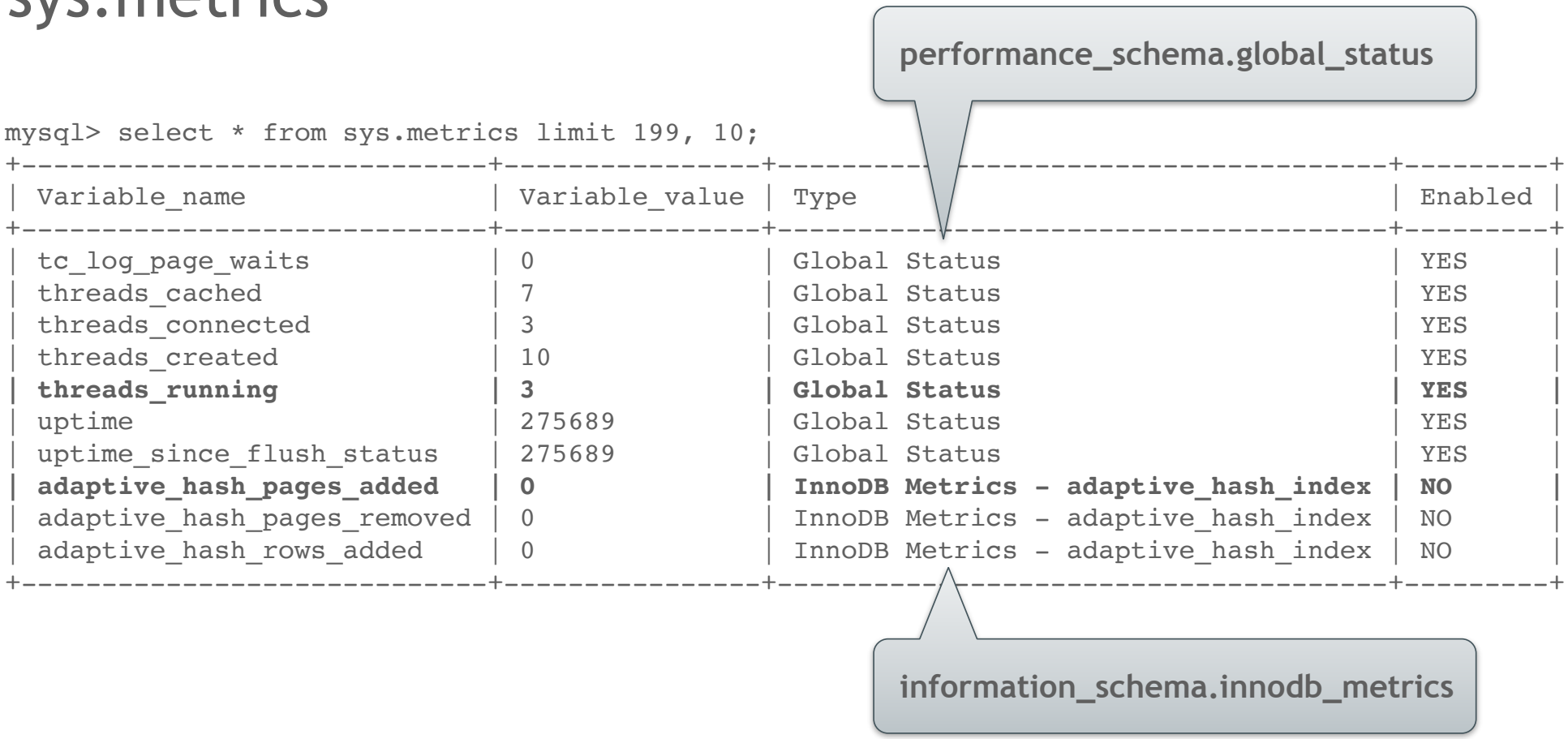
- Added 9 new tables:
  - System Variables
    - **global_variables**
    - **session_variables**
    - **variables_by_thread**
  - Status Variables:
    - **global_status**
    - **session_status**
    - **status_by_thread**
    - **status_by_account**
    - **status_by_user**
    - **status_by_host**

WL#6629 - PERFORMANCE_SCHEMA, STATUS VARIABLES

```
mysql> select * from performance_schema.status_by_thread where variable_name like 'bytes%';
+-----------+----------------+----------------+
| THREAD_ID | VARIABLE_NAME  | VARIABLE_VALUE |
+-----------+----------------+----------------+
|        37 | Bytes_received | 3901           |
|        37 | Bytes_sent     | 135557         |
+-----------+----------------+----------------+
2 rows in set (0.01 sec)

mysql> select * from performance_schema.status_by_user where variable_name like 'bytes%';
+----------+----------------+----------------+
| USER     | VARIABLE_NAME  | VARIABLE_VALUE |
+----------+----------------+----------------+
| NULL     | Bytes_received | 191            |
| NULL     | Bytes_sent     | 160            |
| msandbox | Bytes_received | 4281           |
| msandbox | Bytes_sent     | 136263         |
| root     | Bytes_received | 360            |
| root     | Bytes_sent     | 334            |
| mark     | Bytes_received | 3434           |
| mark     | Bytes_sent     | 101628         |
+----------+----------------+----------------+
```

Or by host, account

sys.metrics



Program Agenda

1. That Introduction
2. Improvements to monitoring in MySQL 5.7
3. Improvements made so far in MySQL 8.0
4. Questions and feedback!

WL#6616 - PERFORMANCE SCHEMA, INDEXES

- Whilst Performance Schema tries to be low overhead from a monitoring perspective, the load of complex queries on the actual data could have significant overhead
- Most of this was due to the server having to process all event data when searching, or joining, or aggregating etc.
- Very noticeable in for instance the sys.session view with many connected sessions

WL#6616 - PERFORMANCE SCHEMA, INDEXES

- **125 “indexes” across the current 95 tables**
- They are faked, we don’t maintain indexes for all instrumentation (would be too much overhead) - we fake them to the optimizer and then do lookups on the in memory data at the storage engine level
- Took the sys.session runtime, with 1,000 connected threads, from **~30 seconds to ~1 second** (it’s one of the more complex joining views in sys)
- Added to most columns *we feel* will be searched upon (need feedback)

WL#8058 - PERFORMANCE SCHEMA, INSTRUMENT SERVER ERRORS

- Understanding the types of errors happening has been hard, you can get raw error data in the statement history tables in performance_schema, but nowhere that summarises what errors have happened over time
- Added a new **error** instrument
- 5 new summary tables (events_errors_summary_[global | by_user | by_host | by_account | by_thread)
- We summarise by errors raised vs handled (i.e. within a routine HANDLER statement)

WL#8058 - PERFORMANCE SCHEMA, INSTRUMENT SERVER ERRORS

```
mysql> select * from events_errors_summary_global_by_error
    ->  where sum_error_raised > 0
    ->  order by sum_error_raised desc\G
*************************** 1. row ***************************
     ERROR_NUMBER: 1287
       ERROR_NAME: ER_WARN_DEPRECATED_SYNTAX
        SQL_STATE: HY000
 SUM_ERROR_RAISED: 190
SUM_ERROR_HANDLED: 0
       FIRST_SEEN: 2016-10-04 08:19:56
        LAST_SEEN: 2016-10-04 09:10:19
*************************** 2. row ***************************
     ERROR_NUMBER: 3554
       ERROR_NAME: ER_NO_SYSTEM_TABLE_ACCESS
        SQL_STATE: HY000
 SUM_ERROR_RAISED: 58
SUM_ERROR_HANDLED: 0
       FIRST_SEEN: 2016-10-03 14:11:22
        LAST_SEEN: 2016-10-04 09:10:21
```

WL#8688 - Support ability to persist SET GLOBAL settings

- Added the **variables_info** table
- Exposes metadata about system variables, such as the min/max values and where the variable was set from
- May be expanded upon (such as who set what when)

WL#8688 - Support ability to persist SET GLOBAL settings

```
mysql> select * from variables_info join global_variables using (variable_name) order by variable_source desc limit 10;
+--------------------------+-----------------+---------------+-----------+----------------------+-----------------------------
| VARIABLE_NAME            | VARIABLE_SOURCE | VARIABLE_PATH | MIN_VALUE | MAX_VALUE            | VARIABLE_VALUE
+--------------------------+-----------------+---------------+-----------+----------------------+-----------------------------
| foreign_key_checks       | DYNAMIC         |               | 0         | 0                    | ON
| host_cache_size          | DYNAMIC         |               | 0         | 65536                | 100
| show_compatibility_56    | DYNAMIC         |               | 0         | 0                    | OFF
| basedir                  | COMMAND_LINE    |               | 0         | 0                    | /Users/mark/git/bld-wl883/
| datadir                  | COMMAND_LINE    |               | 0         | 0                    | /Users/mark/git/bld-wl883/data
| port                     | COMMAND_LINE    |               | 0         | 65535                | 8001
| socket                   | COMMAND_LINE    |               | 0         | 0                    | /tmp/mysql-8001.sock
| auto_increment_increment | COMPILED        |               | 1         | 65535                | 1
| key_cache_division_limit | COMPILED        |               | 1         | 100                  | 100
| tmp_table_size           | COMPILED        |               | 1024      | 18446744073709551615 | 16777216
+--------------------------+-----------------+---------------+-----------+----------------------+-----------------------------
10 rows in set (0.01 sec)
```

WL#8688 - Support ability to persist SET GLOBAL settings

- COMPILED - The default compiled in value
- GLOBAL - Set as part of global options file
- SERVER - Set as part of $MYSQL_HOME/my.cnf options file
- EXPLICIT - Set within —defaults-file options file
- EXTRA - Set within —defaults-extra-file options file
- USER - Set within ~/.my.cnf options file
- LOGIN - Set within login path options file
- COMMAND LINE - Set as a command line parameter
- PERSISTED - Set within the new mysqld-auto.cnf persistent options file
- DYNAMIC - Set from a session dynamically

Question and feedback time!

Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.


ORACLE®