MySQL™ Connect

Improving Performance with the MySQL Performance Schema

Jesper Wisborg Krogh
Principal Technical Support Engineer,
MySQL

Agenda

- Tour of the MySQL Performance Schema
- SHOW PROCESSLIST/The slow query log
- Investigating a slow query and high server load
- Monitor Slave Load
- Stack Trace

Getting Started

- Log into the laptop:

Username	mysqlconnect
Password	mysqlconnect

- Change to the root user

```
shell$ su -l
```

Password	Oracle123

- Download and unpack files:

```
shell$ cd /tmp
shell$ wget http://mysql.wisborg.dk/connect.tgz
shell$ tar -zxvf connect.tgz
```

Getting Started (continued)

- To open manual

```
shell$ evince /tmp/hol/workbook.pdf &
```

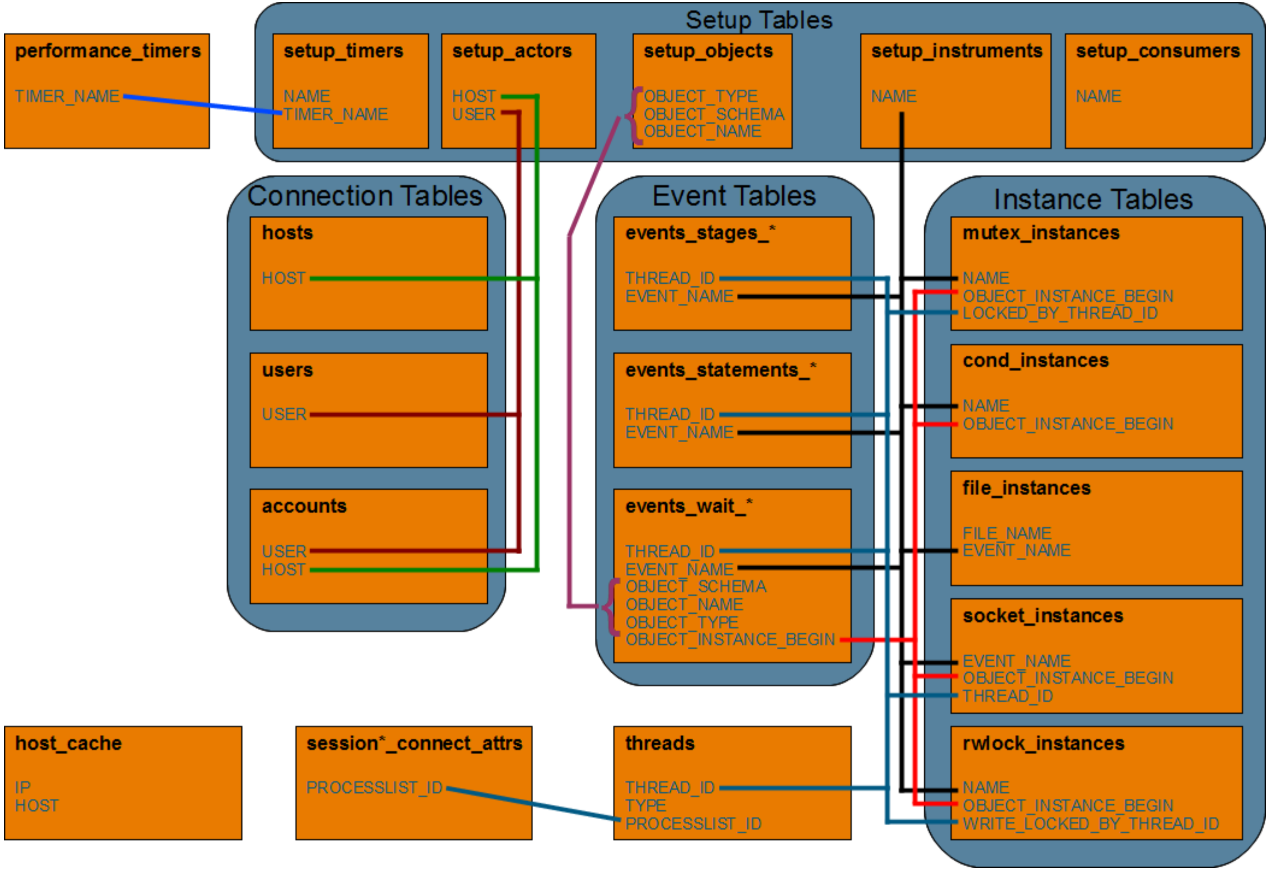
- To open files with queries

```
shell$ gedit /tmp/hol/queries.sql &
```

Tour of the MySQL Performance Schema

- Located in the performance_schema database
- Several groups of tables:
  - Setup tables
  - Instance tables
  - Event tables
  - Summary tables
  - Etc.

Relationship Between Tables



performance_timers
TIMER_NAME

Setup Tables

setup_timers
NAME
TIMER_NAME

setup_actors
HOST
USER

setup_objects
OBJECT_TYPE
OBJECT_SCHEMA
OBJECT_NAME

setup_instruments
NAME

setup_consumers
NAME

Connection Tables

hosts
HOST

users
USER

accounts
USER
HOST

Event Tables

events_stages_*
THREAD_ID
EVENT_NAME

events_statements_*
THREAD_ID
EVENT_NAME

events_wait_*
THREAD_ID
EVENT_NAME
OBJECT_SCHEMA
OBJECT_NAME
OBJECT_TYPE
OBJECT_INSTANCE_BEGIN

Instance Tables

mutex_instances
NAME
OBJECT_INSTANCE_BEGIN
LOCKED_BY_THREAD_ID

cond_instances
NAME
OBJECT_INSTANCE_BEGIN

file_instances
FILE_NAME
EVENT_NAME

socket_instances
EVENT_NAME
OBJECT_INSTANCE_BEGIN
THREAD_ID

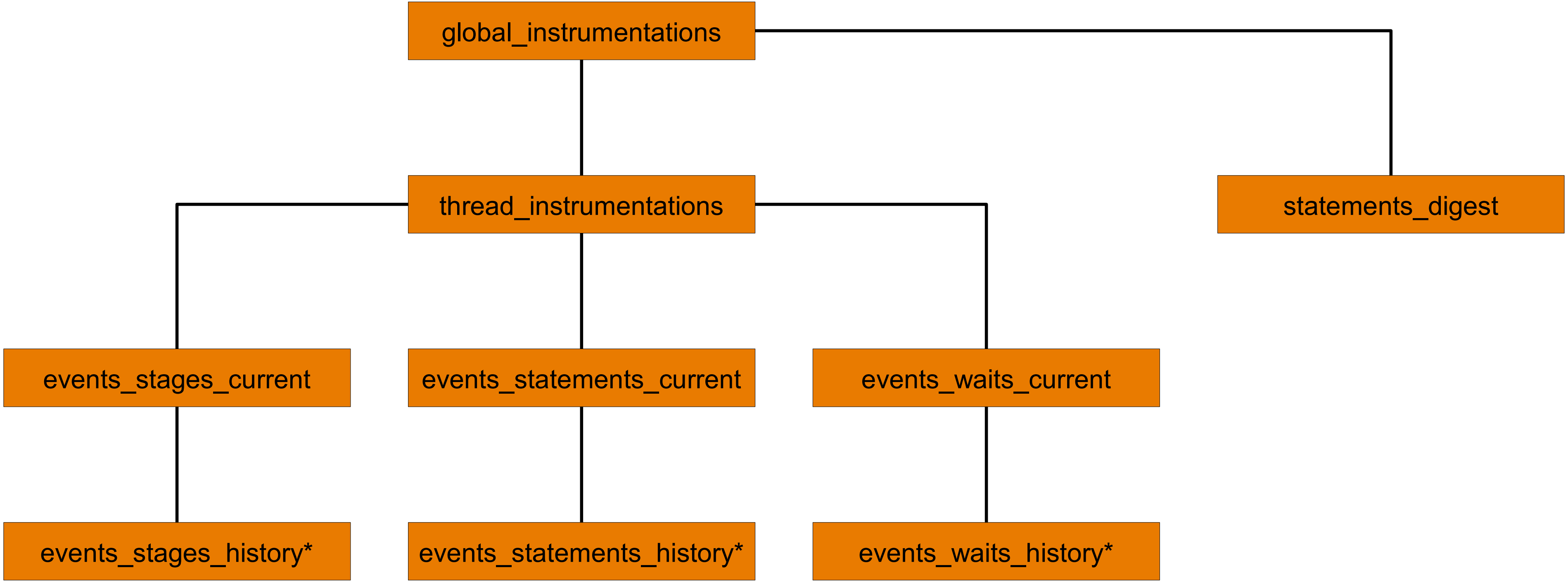
rwlock_instances
NAME
OBJECT_INSTANCE_BEGIN
WRITE_LOCKED_BY_THREAD_ID

host_cache
IP
HOST

session*_connect_attrs
PROCESSLIST_ID

threads
THREAD_ID
TYPE
PROCESSLIST_ID

Relationship Between Tables



SHOW PROCESSLIST

- SHOW PROCESSLIST;
- information_schema.PROCESSLIST table:
  - Essentially the same as SHOW PROCESSLIST;
- performance_schema.threads table:
  - Only mutexes needed are for the query itself; no locking of the listed processes.
  - Can include information about background threads.
  - Can configure which threads to instrument.

Slow Query Log

- Logs queries slower than @@session.long_query_time
- Prints something like the following in the slow query log:

```
# Time: 120923 13:48:40
# User@Host: root[root] @ localhost []  Id:    14
# Query_time: 2.031227  Lock_time: 0.000000 Rows_sent: 1  Rows_examined: 0
use performance_schema;
SET timestamp=1348372120;
SELECT SLEEP(2);
```

- ps_tools.slow_query_long(long_query_time, first_datettime) uses the Performance Schema.

Investigating Slow Query

- For simple investigations use the events_statements_* tables.
- For more complex cases also use the events_waits_* tables.

Investigating General Server Load

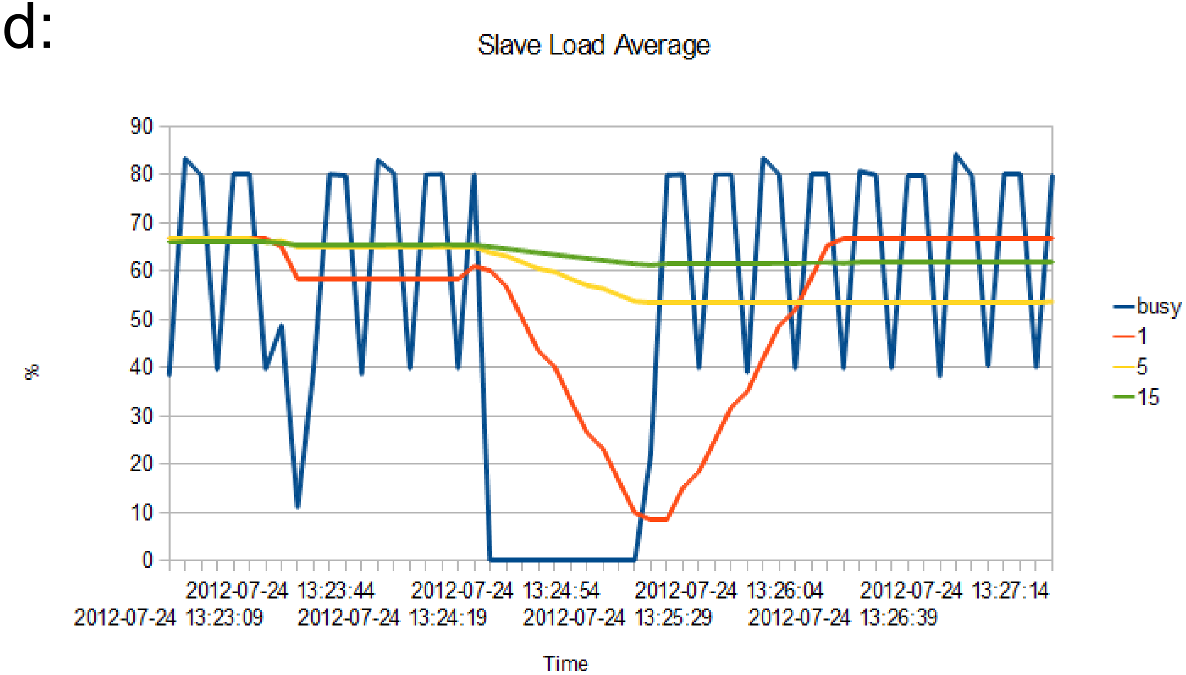
- We will use the events_waits_summary_global_by_event_name table to get an overview of where the server is spending most time.
- Use an iterative approach by taking care of the worst bottleneck first, then retest.
- The idle event will typically dominate, but is not a contention point.
- Table I/O events is an example of molecular events, and will include the time spent in other (atomic) events.

Monitor the Load of a Slave

- Previously it has been difficult to monitor how close a slave has been to the point where it cannot keep up with the master.
- In MySQL 5.5 and later this is possible using the Performance Schema.
- To look at the load for the SQL thread:
  - MySQL 5.5: use wait/synch/cond/sql/MYSQL_RELAY_LOG::update_cond events.
  - MySQL 5.6 the event is called wait/synch/cond/sql/RELAY_LOG::update_cond event

Monitor the Load of a Slave (continued)

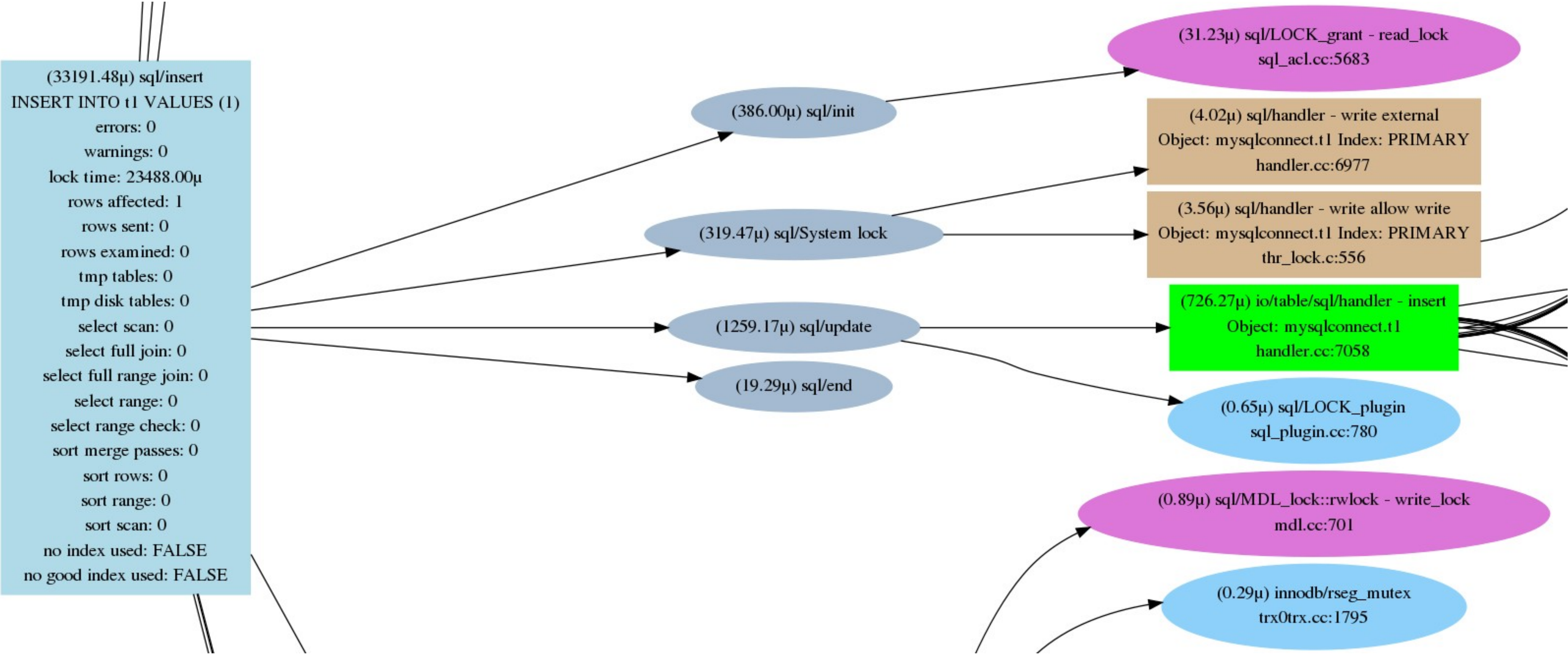
- The compute_slave_load_average procedure (by Mark Leith) can be used:



Stack Trace

- The information in the events_*_history_long tables can be used to create a stack trace for a thread.
- The dump_thread_stack procedure in ps_helper creates the a DOT formatted graph file.
- This file can be used to create a graphical stack trace.

Stack Trace (continued)



MySQL™ Connect