



Introduction to troubleshooting Basic techniques

Sveta Smirnova
Principal Support Engineer
March, 10, 2016

Table of Contents

- Introduction
- How to find problematic query
- Solving issues
 - Syntax errors
 - Logic errors
 - Hidden updates
 - Lost connection
- Next webinars

Introduction

What is basic troubleshooting issue?

- You run a query
- Receive results which you don't expect
 - Wrong dataset
 - No answer from server
 - Anything else
- These results are consistent
 - Repeatable each time when you run the same query

Typical workflow with the basic issue

- Find the query
- Identify class of the problem
 - Syntax error
 - Logical error
 - Data, inserted earlier
 - No answer from the server
- Fix

MySQL architecture

Connectors: C, JDBC, ODBC, Python, ...

Connection Pool: Authentication, Caches

SQL interface

Parser

Optimizer

Caches and Buffers:
Global
Engine-specific

Storage engines: InnoDB, TokuDB, ...

File system: Data, Index, logs, other files

- Base
 - Installation layout
 - Log files
- Connectors
 - Clients, APIs
- Optimizer
- Cache and buffers
- Storage engines
- Management

Typical installation layout

- datadir
 - Schema directory
 - Table and trigger files: *.frm, *.ibd, *.MYD, *.par, etc.
 - Schema
 - ...
 - InnoDB shared tablespace
 - Log files
 - InnoDB redo log files
 - Binary, relay log files
 - Error log
 - Slow query log
 - General query log
- Configurable
- You can setup custom path for each component
- Including custom paths for tables

Plugins

- Storage engines
- Authentication
- Audit
- Query rewrite
- More

Storage engines

- Own data
- Own index format
- Own locking model
- Own diagnostic
- Own log files
- CHECK TABLE

How to find problematic query

Where is the query?

- Log files of your application
- General query log
- performance_schema.events_statements_* tables
- Scriptable Proxies
- Audit Plugin
- User complains

General Query Log

```
mysql> set global log_output='table', global general_log=1;
Query OK, 0 rows affected (0,00 sec)

mysql> select * from mysql.general_log where argument like '%Gender%'
-> and argument not like '%general_log%'\G
***** 1. row *****
event_time: 2016-02-25 03:35:39.903867
user_host: root[root] @ localhost[127.0.0.1]
thread_id: 1
server_id: 1
command_type: Query
argument: select * from employees where first_name = 'Gender'
1 row in set (0.00 sec)
```

performance_schema.events_statements_*

```
mysql> select sql_text from performance_schema.events_statements_history_long
       -> where sql_text like '%gender%' and current_schema='employees';
+-----+
| sql_text                                     |
+-----+
| select * from employees where first_name = 'Gender' |
| select * from performance_schema.events_statements_history_long |
|   where sql_text like '%gender%' and current_schema='employees' |
+-----+
2 rows in set (0.00 sec)
```

MySQL Proxy: all queries, finished with error

```
function read_query( packet )
    proxy.queries:append(1, packet )
    return proxy.PROXY_SEND_QUERY
end

function read_query_result(inj)
    if inj.resultset.query_status == proxy.MYSQLD_PACKET_ERR then
        print(os.date("%d%m%y %H:%M:%S") .. "\t"
            .. "Query \"" .. inj.query:sub(2) .. "\" finished with error")
    end
end
```

Audit Log

```
mysql> INSTALL PLUGIN audit_log SONAME 'audit_log.so';
Query OK, 0 rows affected (0.09 sec)

mysql> \! cat /home/sveta/build/ps-5.6/mysql-test/var/mysqld.1/data/audit.log
<AUDIT_RECORD
  NAME="Query"
  RECORD="3_2016-02-26T11:03:39"
  TIMESTAMP="2016-02-26T11:03:57 UTC"
  COMMAND_CLASS="select"
  CONNECTION_ID="1"
  STATUS="0"                - Check this field for non-0 status!
  SQLTEXT="select * from mysql.user where user like '%sveta%'"
  USER="root[root] @ localhost [127.0.0.1]"
  HOST="localhost" OS_USER="" IP="127.0.0.1" />
```

Solving issues

Query found: what to do next?

- Check kind of the error
 - Syntax
 - Can happen even to experienced users if complicated query created by an application
 - Logic: stored data is correct
 - SELECT returns wrong result
 - DML updates wrong set of rows
 - Hidden update: wrong data was already in the DB
 - First noted after SELECT returned not expected result
 - Source of the problem was an update which happened few steps ago
 - Connection lost
 - Wrong option usage
 - Server crash
- Check what affects the query - **Future webinars**
- Fix!

Query found: what to do next?

- Check kind of the error
- Check what affects the query - **Future webinars**
 - Storage engine
 - Configuration options
 - Privileges
 - Replication
- Fix!

Syntax errors

Syntax error

- `select * fro t1 where f1 in (1,2,1);`
- `select max (f1) from t1;`
- `select * from t1 where "f1"=1;`
- `return ''' . str_pad(md5(rand()),
 rand(1, $column['character_maximum_length']),
 md5(rand()), STR_PAD_BOTH) . ''';`

Syntax error

- `select * fro t1 where f1 in (1,2,1);`
- `select max (f1) from t1;`
 - Correct syntax: `max(f1)` – no space
 - Works with SQL Mode `IGNORE_SPACE` only
- `select * from t1 where "f1"=1;`
- `return ''' . str_pad(md5(rand()),
rand(1, $column['character_maximum_length']),
md5(rand()), STR_PAD_BOTH) . ''';`

Syntax error

- `select * fro t1 where f1 in (1,2,1);`
- `select max (f1) from t1;`
- `select * from t1 where "f1"=1;`
- – Works with SQL Mode ANSI_QUOTES only
- `return ''' . str_pad(md5(rand()),
rand(1, $column['character_maximum_length']),
md5(rand()), STR_PAD_BOTH) . ''';`

Syntax error

- `select * fro t1 where f1 in (1,2,1);`
- `select max (f1) from t1;`
- `select * from t1 where "f1"=1;`
- `return "" . str_pad(md5(rand()),
rand(1, $column['character_maximum_length']),
md5(rand()), STR_PAD_BOTH) . "";`
- In the application

Error message: always check!

```
mysql> select * fro t1 where f1 in (1,2,1);
```

```
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual  
that corresponds to your MySQL server version for the right syntax to use  
near 'fro t1 where f1 in (1,2,1)' at line 1
```

```
mysql> select max (f1) from t1;
```

```
ERROR 1630 (42000): FUNCTION test.max does not exist.  
Check the Function Name Parsing and Resolution' section  
in the Reference Manual
```


Warnings are important

```
mysql> select * from t1 where "f1"=1;  
Empty set, 1 warning (0.05 sec)
```

```
mysql> show warnings;
```

+-----+-----+-----+			+
Level	Code	Message	
+-----+-----+-----+			+
Warning	1292	Truncated incorrect DOUBLE value: 'f1'	
+-----+-----+-----+			+

1 row in set (0.00 sec)

Errors information in the application

- Error information
 - C API, Connector/C, PHP
 - mysql_errno
 - mysql_error
- Warnings
 - C API, Connector/C, PHP
 - mysql_info
 - mysql_sqlstate
 - mysql_warning_count

Errors information in the application

- Error information
 - JDBC, Connector/C++
 - SQLException
- Warnings
 - JDBC, Connector/C++
 - SQLWarning

Errors information in the application

- Error information
 - ODBC
 - SQLGetDiagField
 - SQLGetDiagRec
- Warnings
 - ODBC
 - SQLGetDiagField
 - SQLGetDiagRec

Errors information in the application

- Error information
 - Connector/Python
 - errors.*Error
- Warnings
 - Connector/Python
 - raise_on_warnings option
 - errors.Warning is not used!

Perror: what does the error mean?

```
sveta@thinkie> perror 1292
```

- SQL error

```
MySQL error code 1292 (ER_TRUNCATED_WRONG_VALUE):
```

```
Truncated incorrect %-.32s value: '%-.128s'
```

```
sveta@thinkie> perror 2
```

- OS error

```
OS error code 2: No such file or directory
```

```
sveta@thinkie> perror 150
```

- Engine error

```
MySQL error code 150: Foreign key constraint is incorrectly formed
```

Error handling in stored routines

- GET DIAGNOSTICS
 - GET DIAGNOSTICS rows = ROW_COUNT, conditions = NUMBER;
 - GET DIAGNOSTICS CONDITION 1 code = RETURNED_SQLSTATE, msg = MESSAGE_TEXT;
- SIGNAL/RESIGNAL
 - Can be used for custom errors

Logic errors

Logic errors

- You expect query to work correctly
 - But it does not
 - This can be caused by
 - Error in how you interpret the query should work
 - MySQL bug
- How to start fixing such a query?
 - Simplify it
 - Consider each of parts which complicate the query
 - Find out what it does
 - Fix
- Hardest to teach
- Requires practice

Example 1: movies

- The issue

```
mysql> select title from movies where title=8;
```

```
+-----+  
| title          |  
+-----+  
| 8              |  
| 8 1/2          |  
| 8 Simple Rules |  
| 8 Seconds      |  
+-----+
```

```
4 rows in set, 3 warnings (0.00 sec)
```

Example 1: movies

- The issue
- Investigation

```
mysql> show warnings;
```

```
+-----+-----+-----+
| Level   | Code | Message                                     |
| Warning | 1292 | Truncated incorrect DOUBLE value: '8'     |
| Warning | 1292 | Truncated incorrect DOUBLE value: '8 Simple Rules' |
| Warning | 1292 | Truncated incorrect DOUBLE value: '8 Seconds'      |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

Example 1: movies

- The issue
- Investigation
- Test

```
mysql> select 8 = '8 Simple Rules', '8' = '8 Simple Rules';
```

+-----+	
8 = '8 Simple Rules'	'8' = '8 Simple Rules'
+-----+	
1	0
+-----+	

```
1 row in set, 1 warning (0.00 sec)
```

Example 1: movies

- The issue
- Investigation
- Test
- Fix

```
mysql> select title from movies where title='8';
```

```
+-----+
```

```
| title |
```

```
+-----+
```

```
|      8 |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

Example 2: too many leaders

- The issue

```
mysql> select count(emp_no) from employees where hire_date in  
-> (select hire_date from titles where title='Technique Leader')  
-> union select count(*) from employees;
```

```
+-----+  
| count(emp_no) |  
+-----+  
|          300024 |  
|          300024 |  
+-----+
```

```
1 row in set (3 min 7.20 sec)
```

Example 2: too many leaders

- The issue
- More stranges

```
mysql> select count(emp_no) from employees where hire_date in  
-> (select hire_date from titles where title='Senior engineer');
```

```
+-----+  
| count(emp_no) |  
+-----+  
|          300024 |  
+-----+
```

```
1 row in set (17 min 32.49 sec)
```

Example 2: too many leaders

- The issue
- More stranges
- Investigation

```
mysql> explain extended select count(emp_no) from employees
      -> where hire_date in
      -> (select hire_date from titles where title='Technique Leader')\G
...
mysql> show warnings\G
***** 1. row *****
Level: Note
Code: 1276
Message: Field or reference 'employees.employees.hire_date' of SELECT #2
        was resolved in SELECT #1
```


Example 2: too many leaders

- The issue
- More stranges
- Investigation
- Why so?

```
mysql> desc titles;
```

Field	Type	Null	Key	Default	Extra
emp_no	int(11)	NO	PRI	NULL	
title	varchar(50)	NO	PRI	NULL	
from_date	date	NO	PRI	NULL	
to_date	date	YES		NULL	

Example 3: MySQL bug

- Preparation

```
mysql> alter table titles add index(title, emp_no);  
Query OK, 0 rows affected (8.39 sec)  
Records: 0  Duplicates: 0  Warnings: 0
```

Example 3: MySQL bug

- Preparation
- The issue

```
mysql> select max(emp_no), title from titles
      -> where title = 'Technique Leader'
      -> or emp_no = 10005
      -> group by title;
```

```
+-----+-----+
| max(emp_no) | title           |
+-----+-----+
|      499942 | Technique Leader |
+-----+-----+
1 row in set (0.00 sec)
```

Example 3: MySQL bug

- Preparation
- The issue

But!

```
mysql> select title from titles where emp_no = 10005;
```

```
+-----+
```

```
| title      |
```

```
+-----+
```

```
| Senior Staff |
```

```
| Staff       |
```

```
+-----+
```

```
2 rows in set (0.00 sec)
```

Example 3: MySQL bug

- Preparation
- The issue
- Investigation

```
mysql> explain select max(emp_no), title from titles where  
      -> title = 'Technique Leader' or emp_no = 10005 group by title\G  
***** 1. row *****  
  
...  
possible_keys: PRIMARY,emp_no,title  
      key: title  
  
...  
      Extra: Using where; Using index for group-by  
1 row in set, 1 warning (0.00 sec)
```

Example 3: MySQL bug

- Preparation
- The issue
- Investigation
- Test and Fix

```
mysql> select max(emp_no), title from titles ignore index(title) where  
-> title = 'Technique Leader' or emp_no = 10005 group by title;
```

+-----+-----+		
max(emp_no)	title	
+-----+-----+		
10005	Senior Staff	
10005	Staff	
499942	Technique Leader	
+-----+-----+		

Example 3: MySQL bug

- Preparation
- The issue
- Investigation
- Test and Fix
- Based on
<https://bugs.mysql.com/bug.php?id=70359>

Hidden updates

What do I call "hidden update"?

- Query behaves correctly, but returns wrong data
- Reason can be
 - Wrong update, happened earlier
 - No change when update expected
- How to find what really happened?
 - Logs of your application
 - Binary log - **May not have variables affecting result!**
 - General query log, p_s.events_statements_* tables
 - Scriptable Proxies
- Looks similar to wrong syntax issue?
- We need to watch change time now

Hidden update exmaple

- The issue

```
mysql> select count(*) from titles where title = 'Senior Engineer';
```

```
+-----+
```

```
| count(*) |
```

```
+-----+
```

```
|          0 |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

Hidden update exmaple

- The issue
- Investigation

```
$mysqlbinlog -v master-bin.000001
# at 308
#160305 23:25:16 server id 1  end_log_pos 467 CRC32 0xeef31919  Query
thread_id=3 exec_time=69 error_code=0
use 'employees'/*!*/;
SET TIMESTAMP=1457213116/*!*/;
update titles set title = 'Junior Engineer'
where title = 'Senior Engineer'
/*!*/;
# at 467
...
```

Hidden update exmaple

- The issue
- Investigation
- Possible resolutions
 - In our case we can simply run

```
update titles set title = 'Senior Engineer'
where title = 'Junior Engineer';
```
 - In more general case
 - Restore table from backup
 - Apply all binary logs, created after backup was made

Lost connection

Two main reasons

- Server crash: new connection cannot be established
 - Check error log file
 - Will be discussed in future webinar
- You can reconnect
 - Network issues and timeouts
 - Effect of max_allowed_packet
 - You could see "ERROR 1064 (42000) at line XXX: You have an error in your SQL syntax;" instead of proper "ERROR 1153 (08S01) at line XXX: Got a packet bigger than 'max_allowed_packet' bytes"
 - Fixed in latest versions

Timeout symptoms

- connect_timeout
 - Controls initial communication
 - Check status variable Aborted_connects

Timeout symptoms

- connect_timeout
- interactive_timeout
- wait_timeout
 - How much time to wait activity from the client before aborting connection
 - How much time can a thread be in "Sleep" state
 - Check status variable Aborted_clients

Timeout symptoms

- connect_timeout
- interactive_timeout
- wait_timeout
- net_read_timeout
- net_write_timeout
- How long to wait communication from a client which reads/writes result

Timeout issues: how to fix

- Most obvious and dangerous way: increase timeout
 - Can just hide the issue, but not fix it
- Better way
 - Temporarily increase timeout
 - Check if it solves the issue or make it happen more seldom
 - Find out the reason why the issue happened in the first time
 - Check and fix network

Next webinars

Performance

- Introduction to troubleshooting performance
 - end of March
- Slow queries
 - everything about EXPLAIN, query execution, etc.
- Locking issues
- Storage engines
- Options
- Hardware

Production

- Privileges
- Replication
- Crashes

Effective testing

- Tools
- Workflow
- Advanced testing techniques
- Troubleshooting and monitoring best practices
 - Last webinar, end of the year

Summary

Summary

- Answers from the server are important
- Always check error messages and warnings
- Be ready to use log files

More Informaiton

- MySQL Server Logs
- Percona Audit Plugin
- Writing Audit Plugins
- MySQL Proxy

Place for your questions

???

Thank you!

`http://www.slideshare.net/SvetaSmirnova`

`https://twitter.com/svetismirnova`