# MySQL Break/Fix Lab

Presented by: Alkin Tezuysal, Emanuel Calvo, Okan Buyukyilmaz

April, 2016

# Agenda

- Fix standalone MySQL instance (by Emanuel Calvo - @3manuek - 3manuek.com )
- Replication issues (by Okan Buyukilmaz - @oknbnaba )
- Performance issues (by Alkin Tesuyzal - @ask_dba)

# Getting started

- Each attendee has its own instance (not shared)
  *we break it, you fix it :-)*
- Do not fix other things and follow the sequence of slides.
- One standalone MySQL instance
- Several MySQL instances using MySQL sandbox

# Getting setup

Access:

```
Username/password: user-lab / pythianlab123
ssh user-lab@<IP>
```

Pre-requirement: ssh client (putty, standard ssh, etc).

Host list (take one, be good):
**http://tinyurl.com/PL2016-HOSTS**

Command cheat:
**http://tinyurl.com/BL-PL2016**

# Part 1 - Agenda

- Recover a mysql instance unable to start
  - misconfiguration
  - files permission
  - corrupted files
- Connectivity issues
  - misconfiguration
  - recover password
  - server gone away
- Learn how to troubleshoot crash
  - read error log
  - fix misconfiguration
  - tune variables

# Starting mysqld

```
[root@hostdb ~]# service mysql start
Initializing MySQL database:  Installing MySQL system tables...
2014-10-30 16:29:23 10826 [ERROR] /usr/sbin/mysqld: unknown variable 'tmpd1r=/
   var/tmp'
2014-10-30 16:29:23 10826 [ERROR] Aborting
2014-10-30 16:29:23 10826 [Note] /usr/sbin/mysqld: Shutdown complete


[root@hostdb ~]# !ps
ps aux | grep mysql
root     2185  0.0  0.0 103424   828 pts/2  S+   13:19   0:00 grep mysql
```

# Where is the config file?

```
[root@hostdb ~]# grep tmpd /etc/my.cnf
[root@hostdb ~]# grep tmpd /etc/mysql/my.cnf
grep: /etc/mysql/my.cnf: No such file or directory
```

## Multiple configuration file(s) ?

The easy way:

https://dev.mysql.com/doc/refman/5.6/en/option-files.html

The hard way:

strace

# Strace options

- Option “-e trace=open,stat” will help to filter the long output of the strace.

```
[root@hostdb ~]# strace -e trace=open,stat /usr/sbin/mysqld
...
stat("/etc/my.cnf", {st_mode=S_IFREG|0644, st_size=243, ...}) = 0
open("/etc/my.cnf", O_RDONLY)                = 3
stat("/etc/mysql/my.cnf", 0x7fffea4c0d80) = -1 ENOENT (No such file or
   directory)
stat("/usr/etc/my.cnf", {st_mode=S_IFREG|0644, st_size=25, ...}) = 0
open("/usr/etc/my.cnf", O_RDONLY)            = 3
stat("/root/.my.cnf", {st_mode=S_IFREG|0644, st_size=33, ...}) = 0
open("/root/.my.cnf", O_RDONLY)              = 3
...
```

# Strace: mysqld --print-defaults

```
# strace -e stat64 /usr/sbin/mysqld --print-defaults
/usr/sbin/mysqld would have been started with the following arguments:
--datadir=/var/lib/msql --innodb_data_file_path=ibdata1:18M --
   innodb_buffer_pool_size=100G --innodb_log_file_size=64M --
   sort_buffer_size=60M --tmpd1r=/var/tmp


stat64("/etc/my.cnf", 0xbfb9d750)       = -1 ENOENT (No such file or
   directory)
stat64("/etc/mysql/my.cnf", {st_mode=S_IFREG|0644, st_size=3564, ...}) = 0
stat64("/usr/etc/my.cnf", 0xbfb9d750)   = -1 ENOENT (No such file or
   directory)
stat64("/root/.my.cnf", 0xbfb9d750)     = -1 ENOENT (No such file or
   directory)
```

# Fixing tmpdir variable

```
[root@hostdb ~]# cat /usr/etc/my.cnf
[mysqld]
tmpd1r=/var/tmp
```

```
[root@hostdb ~]# sed -i -e 's/tmpd1r/tmpdir/' /usr/etc/my.cnf
[root@hostdb ~]# cat /usr/etc/my.cnf
[mysqld]
tmpdir=/var/tmp
```

# Starting the mysqld, again.

```
[root@hostdb ~]# service mysql start
MySQL Daemon failed to start.
Starting MySQL... ERROR! The server quit without updating PID file (/var/
   lib/msql/ip-10-87-0-19.pid).
```

# Targeting new errors on the error log

```
[root@hostdb ~]# tail –n 100 /var/log/mysqld.log
1141028 11:40:32 mysqld_safe Starting mysqld daemon with databases from /var/lib/msql
2014-10-28 11:40:34 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated.
    Please use --explicit_defaults_for_timestamp server option (see documentation for
    more details).
2014-10-28 11:40:34 32549 [Note] Plugin 'FEDERATED' is disabled.
/usr/sbin/mysqld: Table 'mysql.plugin' doesn't exist
2014-10-28 11:40:34 32549 [ERROR] Can't open the mysql.plugin table. Please run
    mysql_upgrade to create it.
2014-10-28 11:40:34 32549 [Note] InnoDB: Using atomics to ref count buffer pool pages
2014-10-28 11:40:34 32549 [Note] InnoDB: The InnoDB memory heap is disabled
2014-10-28 11:40:34 32549 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
2014-10-28 11:40:34 32549 [Note] InnoDB: Memory barrier is not used
2014-10-28 11:40:34 32549 [Note] InnoDB: Compressed tables use zlib 1.2.3
2014-10-28 11:40:34 32549 [Note] InnoDB: Using Linux native AIO
2014-10-28 11:40:34 32549 [Note] InnoDB: Using CPU crc32 instructions
/usr/sbin/mysqld: Can't create/write to file '/var/tmp/ib3qf0c3' (Errcode: 13-
    Permission denied )
2014-10-28 11:40:34 7fda23d8b740  InnoDB: Error: unable to create temporary file;
    errno: 13
2014-10-28 11:40:34 32549 [ERROR] Plugin 'InnoDB' init function returned error.
2014-10-28 11:40:34 32549 [ERROR] Plugin 'InnoDB' registration as a STORAGE ENGINE
    failed.
2014-10-28 11:40:34 32549 [ERROR] Unknown/unsupported storage engine: InnoDB
2014-10-28 11:40:34 32549 [ERROR] Aborting
```

# Fixing permissions

```
[root@hostdb ~]# ls -l /var/lib/mysql/mysql/plugin.*
-rw-rw---- 1 root root 8586 Mar 13 12:30 /var/lib/mysql/mysql/plugin.frm
-rw-rw---- 1 root root    0 Mar 13 12:30 /var/lib/mysql/mysql/plugin.MYD
-rw-rw---- 1 root root 1024 Mar 13 12:30 /var/lib/mysql/mysql/plugin.MYI


[root@hostdb ~]# chown -R mysql:mysql /var/lib/mysql/mysql/


[root@hostdb ~]# service mysql start
Starting MySQL... ERROR! The server quit without updating PID file (/var/
   lib/msql/ip-10-87-0-19.pid).
```

# Typo in the datadir variable

```
141030 16:44:55 mysqld_safe Starting mysqld daemon with databases from /var/lib/msql
2014-10-30 16:44:57 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated.
    Please use --explicit_defaults_for_timestamp server option (see documentation for
    more details).
2014-10-30 16:44:57 11576 [Note] Plugin 'FEDERATED' is disabled.
/usr/sbin/mysqld: Table 'mysql.plugin' doesn't exist
2014-10-30 16:44:57 11576 [ERROR] Can't open the mysql.plugin table. Please run
    mysql_upgrade to create it.
2014-10-30 16:44:57 11576 [Note] InnoDB: Using atomics to ref count buffer pool pages
2014-10-30 16:44:57 11576 [Note] InnoDB: The InnoDB memory heap is disabled
2014-10-30 16:44:57 11576 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
2014-10-30 16:44:57 11576 [Note] InnoDB: Memory barrier is not used
2014-10-30 16:44:57 11576 [Note] InnoDB: Compressed tables use zlib 1.2.3
2014-10-30 16:44:57 11576 [Note] InnoDB: Using Linux native AIO
2014-10-30 16:44:57 11576 [Note] InnoDB: Using CPU crc32 instructions
^G/usr/sbin/mysqld: Can't create/write to file '/var/tmp/ibvm13fg' (Errcode: 13 -
    Permission denied)
2014-10-30 16:44:57 7fd985ef8740  InnoDB: Error: unable to create temporary file;
    errno: 13
2014-10-30 16:44:57 11576 [ERROR] Plugin 'InnoDB' init function returned error.
2014-10-30 16:44:57 11576 [ERROR] Plugin 'InnoDB' registration as a STORAGE ENGINE
    failed.
2014-10-30 16:44:57 11576 [ERROR] Unknown/unsupported storage engine: InnoDB
2014-10-30 16:44:57 11576 [ERROR] Aborting
```

# Fix DATADIR path

```
[root@hostdb ~]# grep datadir /etc/my.cnf
datadir=/var/lib/msql

[root@hostdb ~]# sed -i -e 's/datadir=\/var\/lib\/msql/datadir=\/var\/lib\/
   mysql/' /etc/my.cnf

[root@hostdb ~]# grep datadir /etc/my.cnf
datadir=/var/lib/mysql
```

# Permission issues

```
[root@hostdb~]# service mysql start
Starting MySQL... ERROR! The server quit without updating PID file (/var/lib/msql/ip-10-87-0-19.pid).


141028 11:46:25 mysqld_safe Starting mysqld daemon with databases from /var/lib/mysql
2014-10-28 11:46:26 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --
    explicit_defaults_for_timestamp server option (see documentation for more details).
2014-10-28 11:46:27 641 [Note] Plugin 'FEDERATED' is disabled.
2014-10-28 11:46:27 641 [Note] InnoDB: Using atomics to ref count buffer pool pages
2014-10-28 11:46:27 641 [Note] InnoDB: The InnoDB memory heap is disabled
2014-10-28 11:46:27 641 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
2014-10-28 11:46:27 641 [Note] InnoDB: Memory barrier is not used
2014-10-28 11:46:27 641 [Note] InnoDB: Compressed tables use zlib 1.2.3
2014-10-28 11:46:27 641 [Note] InnoDB: Using Linux native AIO
2014-10-28 11:46:27 641 [Note] InnoDB: Using CPU crc32 instructions
/usr/sbin/mysqld: Can't create/write to file '/var/tmp/ibYhjMI4' (Errcode: 13 - Permission denied)
2014-10-28 11:46:27 7f61a2bde740  InnoDB: Error: unable to create temporary file; errno: 13
2014-10-28 11:46:27 641 [ERROR] Plugin 'InnoDB' init function returned error.
2014-10-28 11:46:27 641 [ERROR] Plugin 'InnoDB' registration as a STORAGE ENGINE failed.
2014-10-28 11:46:27 641 [ERROR] Unknown/unsupported storage engine: InnoDB
2014-10-28 11:46:27 641 [ERROR] Aborting
```

# Fix typo in TMPDIR

```
[root@hostdb ~]# ls -ld /var/tmp
drwxrwx--T 2 root root 4096 Mar 13 13:23 /var/tmp
```

```
[root@hostdb ~]# chmod a+rwx /var/tmp
[root@hostdb ~]# ls -ld /var/tmp
drwxrwxrwt 2 root root 4096 Mar 13 13:23 /var/tmp
```

```
[root@hostdb ~]# service mysql start
Starting MySQL ... ERROR! The server quit without updating PID file (/var/
   lib/mysql/ip-10-87-0-19.pid).
```

# Cannot allocate memory.

```
141028 11:47:42 mysqld_safe Starting mysqld daemon with databases from /var/lib/mysql
2014-10-28 11:47:44 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated.
    Please use --explicit_defaults_for_timestamp server option (see documentation for
    more details).
2014-10-28 11:47:44 898 [Note] Plugin 'FEDERATED' is disabled.
2014-10-28 11:47:44 898 [Note] InnoDB: Using atomics to ref count buffer pool pages
2014-10-28 11:47:44 898 [Note] InnoDB: The InnoDB memory heap is disabled
2014-10-28 11:47:44 898 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
2014-10-28 11:47:44 898 [Note] InnoDB: Memory barrier is not used
2014-10-28 11:47:44 898 [Note] InnoDB: Compressed tables use zlib 1.2.3
2014-10-28 11:47:44 898 [Note] InnoDB: Using Linux native AIO
2014-10-28 11:47:44 898 [Note] InnoDB: Using CPU crc32 instructions
2014-10-28 11:47:44 898 [Note] InnoDB: Initializing buffer pool, size = 100.0G
InnoDB: mmap(13736345600 bytes) failed; errno 12
2014-10-28 11:47:44 898 [ERROR] InnoDB: Cannot allocate memory for the buffer pool
2014-10-28 11:47:44 898 [ERROR] Plugin 'InnoDB' init function returned error.
2014-10-28 11:47:44 898 [ERROR] Plugin 'InnoDB' registration as a STORAGE ENGINE
    failed.
2014-10-28 11:47:44 898 [ERROR] Unknown/unsupported storage engine: InnoDB
2014-10-28 11:47:44 898 [ERROR] Aborting
```

# Fixing INNODB_BUFFER_POOL_SIZE

```
[root@hostdb ~]# perror 12
OS error code  12:  Cannot allocate memory

[root@ip-172-31-9-142 ~]# free -m
             total       used       free     shared    buffers     cached
Mem:          1655        654       1000          0         33        546

[root@hostdb ~]# grep innodb_buffer_pool_size /etc/my.cnf
innodb_buffer_pool_size=100G

[root@hostdb ~]# sed -i -e 's/100G/256M/' /etc/my.cnf
[root@hostdb ~]# grep innodb_buffer_pool_size /etc/my.cnf
innodb_buffer_pool_size=256M

[root@hostdb ~]# service mysql start
StartingMySQL... ERROR! The server quit without updating PID file (/var/
    lib/mysql/ip-10-87-0-19.pid).
```

# ERROR 13, again.

```
141028 11:49:50 mysqld_safe Starting mysqld daemon with databases from /var/lib/mysql
2014-10-28 11:49:51 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated.
    Please use --explicit_defaults_for_timestamp server option (see documentation for
    more details).
2014-10-28 11:49:51 1169 [Note] Plugin 'FEDERATED' is disabled.
2014-10-28 11:49:51 1169 [Note] InnoDB: Using atomics to ref count buffer pool pages
2014-10-28 11:49:51 1169 [Note] InnoDB: The InnoDB memory heap is disabled
2014-10-28 11:49:51 1169 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
2014-10-28 11:49:51 1169 [Note] InnoDB: Memory barrier is not used
2014-10-28 11:49:51 1169 [Note] InnoDB: Compressed tables use zlib 1.2.3
2014-10-28 11:49:51 1169 [Note] InnoDB: Using Linux native AIO
2014-10-28 11:49:51 1169 [Note] InnoDB: Using CPU crc32 instructions
2014-10-28 11:49:51 1169 [Note] InnoDB: Initializing buffer pool, size = 256.0M
2014-10-28 11:49:51 1169 [Note] InnoDB: Completed initialization of buffer pool
2014-10-28 11:49:51 1169 [ERROR] InnoDB: ./ibdata1 can't be opened in read-write mode
2014-10-28 11:49:51 1169 [ERROR] InnoDB: The system tablespace must be writable!
2014-10-28 11:49:51 1169 [ERROR] Plugin 'InnoDB' init function returned error.
2014-10-28 11:49:51 1169 [ERROR] Plugin 'InnoDB' registration as a STORAGE ENGINE
    failed.
2014-10-28 11:49:51 1169 [ERROR] Unknown/unsupported storage engine: InnoDB
2014-10-28 11:49:51 1169 [ERROR] Aborting
```

# Fix permissions

```
[root@hostdb ~]# ls -l /var/lib/mysql/ibdata1
-rw-rw---- 1 27 27 18874368 Mar 13 12:34 /var/lib/mysql/ibdata1
[root@hostdb ~]# ls -l /var/lib/mysql
total 83980
-rw-rw---- 1     27       27 18874368 Mar 13 12:34 ibdata1
-rw-rw---- 1     27       27 33554432 Mar 13 12:34 ib_logfile0
-rw-rw---- 1     27       27 33554432 Mar 13 12:34 ib_logfile1
drwx------ 2 mysql mysql  4096 Mar 13 12:30 mysql
drwx------ 2 root  root   4096 Mar 13 12:30 performance_schema
drwx------ 2 root  root   4096 Mar 13 12:30 test
```

```
[root@hostdb ~]# chown -R mysql:mysql /var/lib/mysql
[root@hostdb~]# service mysql start
Starting MySQL......... SUCCESS!

[root@hostdb~]# service mysql status
 SUCCESS! MySQL running (9906)
```

# Wrong structure (Need upgrade)

```
[root@hostdb~]# egrep "ERROR|Errcode" /var/log/mysqld.log | tail
2016-04-14 20:33:46 9906 [ERROR] Native table
    'performance_schema'.'events_statements_summary_global_by_event_name' has the wrong
    structure
2016-04-14 20:33:46 9906 [ERROR] Native table
    'performance_schema'.'events_statements_summary_by_digest' has the wrong structure
2016-04-14 20:33:46 9906 [ERROR] Native table 'performance_schema'.'users' has the
    wrong structure
2016-04-14 20:33:46 9906 [ERROR] Native table 'performance_schema'.'accounts' has the
    wrong structure
2016-04-14 20:33:46 9906 [ERROR] Native table 'performance_schema'.'hosts' has the
    wrong structure
2016-04-14 20:33:46 9906 [ERROR] Native table 'performance_schema'.'socket_instances'
    has the wrong structure
2016-04-14 20:33:46 9906 [ERROR] Native table
    'performance_schema'.'socket_summary_by_instance' has the wrong structure
2016-04-14 20:33:46 9906 [ERROR] Native table
    'performance_schema'.'socket_summary_by_event_name' has the wrong structure
2016-04-14 20:33:46 9906 [ERROR] Native table
    'performance_schema'.'session_connect_attrs' has the wrong structure
2016-04-14 20:33:46 9906 [ERROR] Native table
    'performance_schema'.'session_account_connect_attrs' has the wrong structure
```

# Access to mysqld

```
[root@hostdb ~]# mysql
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/
   tmp/mysql.sock' (2)
```

```
[root@hostdb ~]# perror 2
OS error code   2:  No such file or directory

[root@hostdb ~]# ls -l /tmp/mysql.sock
ls: cannot access /tmp/mysql.sock: No such file or directory
```

# Access to mysqld

```
[root@hostdb ~]# grep socket /var/log/mysqld.log | tail -n 1
Version: '5.5.34'  socket: '/var/lib/mysql/mysql.sock'  port: 3306  MySQL
   Community Server (GPL)
[root@hostdb ~]# lsof -n | grep mysql | grep unix
mysqld   21737    mysql   12u      unix 0xffff880002e0dd40   0t0
   22829 /var/lib/mysql/mysql.sock

[root@hostdb ~]# grep -B 1 socket /etc/my.cnf

[client]
socket=/tmp/mysql.sock


[root@hostdb ~]# sed -i -e 's/\/tmp\/mysql.sock/\/var\/lib\/mysql\/
   mysql.sock/' /etc/my.cnf
```

# ACCESS TO MYSQLD

```
[root@hostdb ~]# mysql
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password:
   YES)
```

```
[root@hostdb ~]# strace -e trace=open mysql
…
open("/etc/my.cnf", O_RDONLY)                = 3
open("/usr/etc/my.cnf", O_RDONLY)            = 3
open("/root/.my.cnf", O_RDONLY)              = 3
```

```
[root@hostdb ~]# cat ~/.my.cnf
[client]
password=adummypassword
```

# Accessing the mysqld

```
[root@hostdb ~]# mysql --no-defaults
```

```
[root@hostdb ~]# mysql -p
```

# CHANGE ROOT PASSWORD

```
[root@hostdb ~]# echo "SET PASSWORD=PASSWORD('$RANDOM$RANDOM')" | mysql
[root@hostdb ~]# mysql
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using
   password: NO)
```

Add skip-grant-tables to the [mysqld] section

```
[root@hostdb ~]# sed -i 's/\[mysqld\]/&\nskip-grant-tables/' /etc/my.cnf

[root@hostdb ~]# service mysql restart
Shutting down MySQL.. SUCCESS!
Starting MySQL... SUCCESS!
```

# CHANGE ROOT PASSWORD

```
[root@hostdb ~]# mysql
```

```
mysql> UPDATE mysql.user SET password=PASSWORD('newpass') WHERE
user='root';
mysql> FLUSH PRIVILEGES;
```

Remove `skip-grant-tables` from /etc/my.cnf

```
[root@hostdb ~]# sed -i 's/skip-grant-tables//' /etc/my.cnf
```

```
[root@hostdb ~]# service mysql restart
```

```
[root@hostdb ~]# sed -i 's/password=adummypassword/password=newpass/'
   ~/.my.cnf
```

# `mysql_upgrade` and datadir version

```
[root@hostdb~]# grep "ERROR" /var/log/mysqld.log |tail -n 3
2014-11-01 21:24:09 9336 [ERROR] Native table
    'performance_schema'.'socket_summary_by_event_name' has the wrong structure
2014-11-01 21:24:09 9336 [ERROR] Native table
    'performance_schema'.'session_connect_attrs' has the wrong structure
2014-11-01 21:24:09 9336 [ERROR] Native table
    'performance_schema'.'session_account_connect_attrs' has the wrong structure

  SELECT COUNT(1) column_count FROM information_schema.columns WHERE
    table_schema='mysql' AND table_name='user';

[root@hostdb ~]# mysql_upgrade
Looking for 'mysql' as: mysql
Looking for 'mysqlcheck' as: mysqlcheck
Running 'mysqlcheck with default connection arguments
Running 'mysqlcheck with default connection arguments
mysql.columns_priv                                 OK
...
mysql.user                                         OK
Running 'mysql_fix_privilege_tables'...
Running 'mysqlcheck with default connection arguments
Running 'mysqlcheck with default connection arguments
OK
[root@hostdb ~]# service mysql restart
```

# MySQL server has gone away

```
[root@hostdb ~]# ( echo -n "SELECT '" ; for i in `seq 1 1` ; do echo -n "1234567890" ; done ;
    echo -n "' a") | mysql
1234567890

[root@hostdb ~]# ( echo -n "SELECT '" ; for i in `seq 1 2` ; do echo -n "1234567890" ; done ;
    echo -n "' a") | mysql
a
12345678901234567890

[root@hostdb ~]# ( echo -n "SELECT '" ; for i in `seq 1 400000` ; do echo -n "1234567890" ;
    done ; echo -n "' a") | mysql | wc
 2        2 4000003

[root@hostdb ~]# ( echo -n "SELECT '" ; for i in `seq 1 450000` ; do echo -n "1234567890" ;
    done ; echo -n "' a") | mysql | wc
ERROR 2006 (HY000) at line 1: MySQL server has gone away
```

# MySQL has gone away (network traffic)

```
[root@hostdb~]# mysql -e "SHOW GLOBAL VARIABLES LIKE 'max_allowed_packet'"
+--------------------+---------+
| Variable_name      | Value   |
+--------------------+---------+
| max_allowed_packet | 4194304 |
+--------------------+---------+

[root@hostdb ~]#  mysql  -e "SET GLOBAL max_allowed_packet=5242880”

[root@hostdb ~]#  ( echo -n "SELECT '" ; for i in `seq 1 450000` ; do echo -n
    "1234567890" ; done ; echo -n "' a") | mysql | wc
 2        2 4500003
```

# MySQL has gone away, again due to aborted service

```
mysql -e "SELECT SLEEP(1000);" &
ERROR 2013 (HY000): Lost connection to MySQL server during query
```

```
# kill -6 `pidof mysqld`
```

*Signal 6 means ABORT. For more information check the "kill -l" command for reference.*

# MySQL has gone away

*12:21:09 UTC - mysqld got signal 6 ;*
*This could be because you hit a bug. It is also possible that this binary*
*or one of the libraries it was linked against is corrupt, improperly built,*
*or misconfigured. This error can also be caused by malfunctioning hardware.*
*We will try our best to scrape up some info that will hopefully help*
*diagnose the problem, but since we have already crashed,*
*something is definitely wrong and this may fail.*

# MySQL has gone away can happen when:

- server crash
- server killed
- session terminated/killed
- session timing out (`wait_timeout`)
- big packet (`max_allowed_packet`)

# MySQL has gone away

- OOM Killer
- crashing bug
- signals
- others

Always check logs:
- mysql error log
- syslog, /var/log/messages or any core dumps.

# LOCAL_INFILE = ON

http://dev.mysql.com/doc/refman/5.6/en/load-data-local.html

"In a Web environment where the clients are connecting from a Web server, a user could use LOAD DATA LOCAL to read any files that the Web server process has read access to (assuming that a user could run any command against the SQL server)."

```
[root@hostdb~]# mysql -e "SHOW GLOBAL VARIABLES LIKE 'local_infile'"
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| local_infile  | ON    |
+---------------+-------+
```

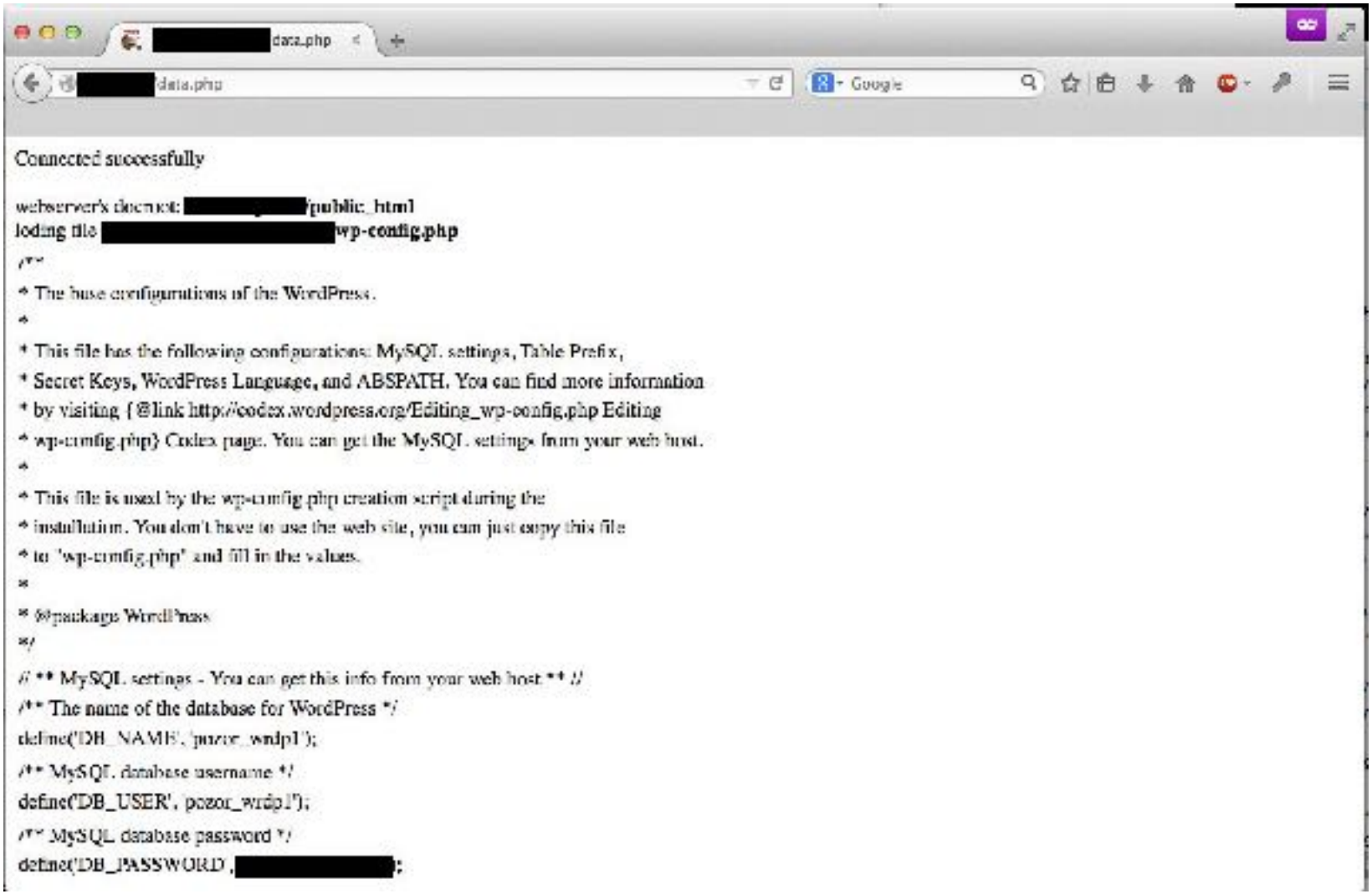
# LOCAL_INFILE = ON

```
[root@hostdb ~]# mysql -e "show grants for evil@localhost”
GRANT USAGE ON *.* TO 'evil'@'localhost' IDENTIFIED BY PASSWORD ‘xxx’
GRANT SELECT, INSERT, CREATE ON `test`.* TO 'evil'@'localhost’

[root@hostdb ~]# mysql -u evil -p4242 test

mysql> select @@datadir;
+-----------------+
| @@datadir       |
+-----------------+
| /var/lib/mysql/ |
+-----------------+

mysql> LOAD DATA LOCAL INFILE  '/var/lib/mysql/mysql/user.MYD' INTO TABLE
   store   LINES TERMINATED BY '*';
```

# LOCAL_INFILE = ON

```
mysql> select * from store\G
...
*************************** 4. row ***************************
col1: D8DECEC305209EEFEC43008E1D420E1AA06B19E0   Uûü::1root
col2: NULL
col3: NULL
*************************** 6. row ***************************
col1: 89C6B530AA78695E257E55D63C00A6EC9AD3E977mysql_native_password   jûü%webapp
col2: NULL
col3: NULL

mysql>  LOAD DATA LOCAL INFILE  '/root/.my.cnf' INTO TABLE store   LINES
    TERMINATED BY '\n';
mysql> select * from store\G
…
*************************** 35. row ***************************
col1: password=newpass
col2: NULL
col3: NULL
```

# LOCAL_INFILE = ON - POC



# Removed ib_logfile (DON'T!)

*Free space was low: I deleted some log files...*

```
[root@ ~]# cd /var/lib/mysql
[root@hostdb mysql]# rm -rf ib_logfile*

[root@hostdb mysql]# lsof -n | grep ib_logfile
mysqld    12076     mysql     8uW       REG   202,1 33554432          403503 /var/lib/
    mysql/ib_logfile0 (deleted)
mysqld    12076     mysql     9uW       REG   202,1 33554432          403505 /var/lib/
    mysql/ib_logfile1 (deleted)

Or
[root@hostdb mysql]# ll /proc/`pidof mysqld`/fd/
total 0
[..]
lrwx------ 1 root root 64 Oct 30 17:19 8 -> /var/lib/mysql/ib_logfile0 (deleted)
lrwx------ 1 root root 64 Oct 30 17:19 9 -> /var/lib/mysql/ib_logfile1 (deleted)
```

# Removed ib_logfile (DON'T!)

mysqld and InnoDB continue working normally

```
mysql> use test;
mysql> create table tbl1 (id int auto_increment primary key, v
   varchar(100)) engine=innodb;
mysql> insert into tbl1 values(null,'aa');
mysql> insert into tbl1 select null, v from tbl1;
mysql> SELECT * FROM tbl1;
```

```
[root@hostdb ~]# service mysql restart
```

# Removed ib_logfile (DON'T!)

InnoDB Redo Log are automatically recreated

```
2014-10-30 17:22:24 14206 [Note] InnoDB: Setting log file ./ib_logfile101 size to 64 MB
2014-10-30 17:22:28 14206 [Note] InnoDB: Setting log file ./ib_logfile1 size to 64 MB
2014-10-30 17:22:32 14206 [Note] InnoDB: Renaming log file ./ib_logfile101 to ./
    ib_logfile0
2014-10-30 17:22:32 14206 [Warning] InnoDB: New log files created, LSN=1605224
```

# Removed ibdata1

*Free space was low: I deleted a large file...*

```
[root@hostdb mysql]# rm -f ibdata1

[root@hostdb mysql]# lsof -n | grep ibdata1
mysqld    7478 mysql 3uW  REG  202,1  27262976
    ibdata1 (deleted)
```

- data dictionary aka metadata of InnoDB tables
- change buffer
- doublewrite buffer
- undo logs

Keep innodb_file_per_table ON.

# Removed ibdata1

mysqld and InnoDB continue working normally … :

```
mysql> use test;
mysql> insert into tbl1 select null, v from tbl1;
mysql> SELECT * FROM tbl1;
```

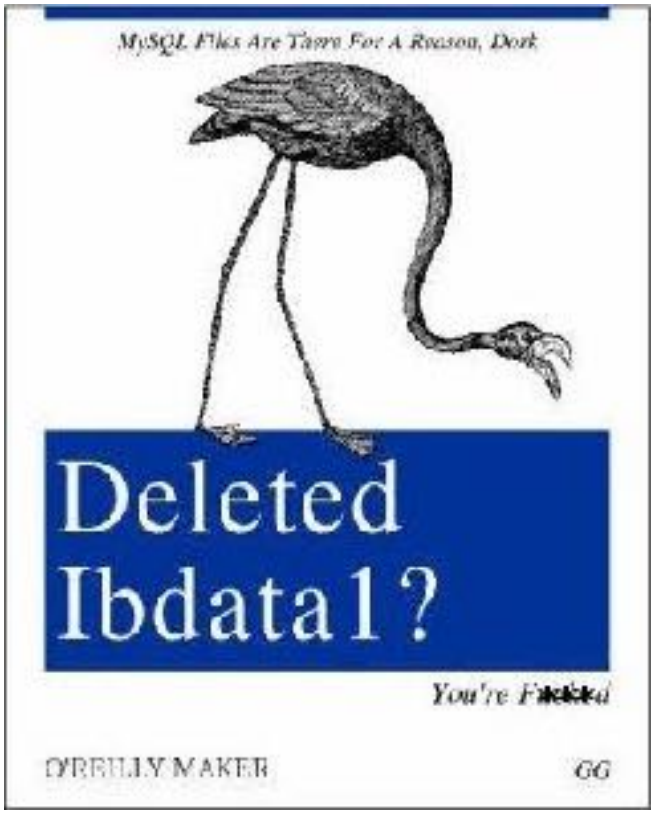
… until restart:

```
[root@hostdb ~]# service mysql restart
service mysql restart
Shutting down MySQL.. SUCCESS!
Starting MySQL..... ERROR! The server quit without updating PID file (/
var/lib/msql/ip-10-87-0-19.pid).
```

# Removed ibdata1

```
2014-10-30 17:34:33 14786 [Note] InnoDB: Initializing buffer pool, size = 256.0M
2014-10-30 17:34:33 14786 [Note] InnoDB: Completed initialization of buffer pool
2014-10-30 17:34:33 14786 [Note] InnoDB: Restoring page 0 of tablespace 0
2014-10-30 17:34:33 14786 [Warning] InnoDB: Doublewrite does not have page_no=0
    of space: 0
2014-10-30 17:34:33 14786 [ERROR] InnoDB: space header page consists of zero
    bytes in data file ./ibdata1
2014-10-30 17:34:33 14786 [ERROR] InnoDB: Could not open or create the system
    tablespace. If you tried to add new data files to the system tablespace, and
    it failed here, you should now edit innodb_data_file_path in my.cnf back to
    what it was, and remove the new ibdata files InnoDB created in this failed
    attempt. InnoDB only wrote those files full of zeros, but did not yet use
    them in any way. But be careful: do not remove old data files which contain
    your precious data!
2014-10-30 17:34:33 14786 [ERROR] Plugin 'InnoDB' init function returned error.
2014-10-30 17:34:33 14786 [ERROR] Plugin 'InnoDB' registration as a STORAGE
    ENGINE failed.
2014-10-30 17:34:33 14786 [ERROR] Unknown/unsupported storage engine: InnoDB
2014-10-30 17:34:33 14786 [ERROR] Aborting
```

# Removed ibdata1 (Really, don't)




**You can now recover from backup!**

# Replication troubleshooting

Presented by: Alkin Tezuysal, Emanuel Calvo, Okan Buyukyilmaz

April, 2016

# AGENDA - PART 2

- Replication overview and tools
- Bad server-id
  - slave id identical to master
  - slave id identical to other slave
- Incorrect slave data
  - Duplicate key error
  - Data drift

# Replication overview

- *asynchronous*: based on copying and executing of binary logs and don't wait to the commit on the slave
  - allows delayed slaves and incremental backups
- binary logs have two formats and a MIXED.
  - statement: SQL
  - row events: since with 5.1
- replication is run by two threads on slave
  - *IO thread*: copies master binary logs to slave relay logs
  - *SQL thread*: executes relay log events on slave
- binary log position: sequence of replicated events
- GTID: global transaction ID unique across all transactions and servers in a replication setup

# Tools and credits

- MySQL Sandbox
  - http://mysqlsandbox.net/
  - created by Giuseppe Maxia (The Data Charmer)
  - Linux/FreeBSD/MacOSX only
  - not for production instances
  - allows you to install and run multiple instances of MySQL on same server
  - can easily set up replication groups
    - default is master and two slaves
    - can be run as regular (non-root) user

# Replication and binlog tools

- Percona Toolkit
  - pt-table-checksum
  - pt-table-sync
  - pt-slave-restart
- mysqlbinlog
- error logs

# Replication commands (CONT.)

- slave commands
  - `SHOW SLAVE STATUS\G`
  - `STOP SLAVE; START SLAVE;`
  - `STOP/START SLAVE IO_THREAD;`
  - `STOP/START SLAVE SQL_THREAD;`
- master commands
  - `SHOW MASTER STATUS;`
  - `SHOW SLAVE HOSTS;`

# Starting the replication test

```
[user-lab@hostdb repl_test]$ pwd
/home/user-lab/sandboxes/repl_test

[user-lab@hostdb repl_test]$ ls
check_slaves             m       restart_all    status_all
clear_all                master  s1             stop_all
connection.json          node1   s2             use_all
default_connection.json  node2   send_kill_all
initialize_slaves        README  start_all

[user-lab@hostdb repl_test]$ ./start_all
# executing "start" on /home/user-lab/sandboxes/repl_test
executing "start" on master
... sandbox server started
executing "start" on slave 1
... sandbox server started
executing "start" on slave 2
... sandbox server started
```

# Sandbox master

```
[user-lab@hostdb repl_test]$ cd master/

[user-lab@hostdb master]$ ls
change_paths              grants.mysql    proxy_start
   status
change_ports              json_in_db      README
   stop
clear                     load_grants     rescue_mysql_dump.sql  tmp
connection.json           msb             restart                use
data                      my              send_kill
   USING
default_connection.json   my.sandbox.cnf  start

[user-lab@hostdb master]$ ls data/
ibdata1      msandbox.err      mysql-bin.index         test
ib_logfile0  mysql             mysql_sandbox23992.pid
ib_logfile1  mysql-bin.000001  performance_schema
```

# Sandbox slave

```
[user-lab@hostdb master]$ cd ../node1

[user-lab@hostdb node1]$ ls
change_paths     default_connection.json  my              send_kill  use
change_ports     grants.mysql             my.sandbox.cnf  start
   USING
clear            json_in_db               proxy_start     status
connection.json  load_grants              README          stop
data             msb                      restart         tmp

[user-lab@hostdb node1]$ ls data
ibdata1            mysql_sandbox23993.pid
ib_logfile0        mysql_sandbox23993-relay-bin.000001
ib_logfile1        mysql_sandbox23993-relay-bin.000002
master.info        mysql_sandbox23993-relay-bin.index
msandbox.err       performance_schema
mysql              relay-log.info
mysql-bin.000001   test
mysql-bin.index
```

# Connecting to mysql nodes

```
[user-lab@hostdb node1]$ cd ..


[user-lab@hostdb repl_test]$ ls
check_slaves             m        restart_all    status_all
clear_all                master   s1             stop_all
connection.json          node1    s2             use_all
default_connection.json  node2    send_kill_all
initialize_slaves        README   start_all
```

# Replication process list threads (master)

```
[user-lab@hostdb repl_test]$ ./m
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
…
```

*NOTE: Id and Time columns removed for visibility.*

```
master [localhost] {msandbox} ((none)) > show processlist;
+----------+-----------------+------+-------------+-----------------------------------------------------------------------+------------------+
| User     | Host            | db   | Command     | State                                                                 | Info             |
+----------+-----------------+------+-------------+-----------------------------------------------------------------------+------------------+
| rsandbox | localhost:44094 | NULL | Binlog Dump | Master has sent all binlog to slave; waiting for binlog to be updated | NULL             |
| rsandbox | localhost:44095 | NULL | Binlog Dump | Master has sent all binlog to slave; waiting for binlog to be updated | NULL             |
| msandbox | localhost       | NULL | Query       | NULL                                                                  | show processlist |
+----------+-----------------+------+-------------+-----------------------------------------------------------------------+------------------+
```

# Replication process list threads (slave)

```
[user-lab@hostdb repl_test]$ ./s1 -uroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
…

slave1 [localhost] {msandbox} ((none)) > show processlist;
+----+-------------+-----------+------+---------+------+-----------------------------------------------------------------------+------------------+
| Id | User        | Host      | db   | Command | Time | State                                                                 | Info             |
+----+-------------+-----------+------+---------+------+-----------------------------------------------------------------------+------------------+
|  1 | system user |           | NULL | Connect |   31 | Slave has read all relay log; waiting for the slave I/O thread to update it | NULL             |
|  2 | system user |           | NULL | Connect |   31 | Waiting for master to send event                                      | NULL             |
|  3 | msandbox    | localhost | NULL | Query   |    0 | NULL                                                                  | show processlist |
+----+-------------+-----------+------+---------+------+-----------------------------------------------------------------------+------------------+
```

# PROBLEM #1: Duplicate server-id value

- The server-id uniquely identifies a server in a replication topology
  - recorded in the binary log
- Dynamic variable (doesn't require restart)
- Is set in the my.cnf file (my.sandbox.cnf)
- If a server is cloned (e.g., ec2 snapshot), the server-id needs to be edited in new instance

# PROBLEM #1: DUPLICATE SERVER-ID VALUE

Example of slave with identical server_id to master:

```
$ ./s1 -e "show slave status\G"
*************************** 1. row ***************************
               Slave_IO_State:
…
             Slave_IO_Running: No
            Slave_SQL_Running: Yes
…
        Seconds_Behind_Master: NULL
Master_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 1593
                Last_IO_Error: Fatal error: The slave I/O thread
   stops because master and slave have equal MySQL server ids; these
   ids must be different for replication to work (or the --replicate-
   same-server-id option must be used on slave but this does not
   always make sense; please check the manual before using it).
           Master_Server_Id: 1
```

# PROBLEM #1: Duplicate server-id values

- With two slaves with identical server-ids, the symptoms and error messages are misleading
- master will show only no slaves in the processlist

# PROBLEM #1: Duplicate server-id values

- Master looks OK, but errors will appear on slave:

```
$ ./s1 -e "start slave; show slave status\G"
$ cd /home/user-lab/sandboxes/dupl_server_id1
$ find ./ -name msandbox.err | xargs tail -n2
==> ./master/data/msandbox.err <==
2014-10-22 11:46:00 1650 [Note] /home/user-lab/sandboxes/5.6.21/bin/
   mysqld: ready for connections.
Version: '5.6.21-log'  socket: '/tmp/mysql_sandbox21000.sock'  port:
   21000  MySQL Community Server (GPL)
==> ./node1/data/msandbox.err <==
2014-10-22 11:49:56 1905 [ERROR] Slave I/O: Fatal error: The slave I/
   O thread stops because master and slave have equal MySQL server
   ids; these ids must be different for replication to work (or the
   --replicate-same-server-id option must be used on slave but this
   does not always make sense; please check the manual before using
   it). Error_code: 1593
2014-10-22 11:49:56 1905 [Note] Slave I/O thread exiting, read up to
   log 'mysql-bin.000001', position 2081
```

# SOLUTION #1: Modifying slave's server-id

## Set *server_id* variable:

```
[user-lab@hostdb dupl_server_id1]$ ./s1

slave1 [localhost] {msandbox} ((none)) > select @@server_id;
+-------------+
| @@server_id |
+-------------+
|           1 |
+-------------+

1 row in set (0.00 sec)

slave1 [localhost] {msandbox} ((none)) > set global server_id = 102;
Query OK, 0 rows affected (0.00 sec)

slave1 [localhost] {msandbox} ((none)) > stop slave; start slave;
Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.00 sec)

slave1 [localhost] {msandbox} ((none)) > show slave status\G
```

# SOLUTION #1: Modifying slave's server-id

2. edit node1/my.sandbox.cnf to use different server-id

```
[user-lab@hostdb dupl_server_id1]$ cd node1

[user-lab@hostdb node1]$ sed -i 's/server-
  id=1/server-id=102/g' my.sandbox.cnf

[user-lab@hostdb node1]$ grep server-id
  my.sandbox.cnf
server-id=102
```

# PROBLEM #2: Duplicate key errors

- Very common replication error
- Number of possible causes
  - Slave created from bad backup
  - Slave replication started in wrong file/pos
  - Direct write to the slave
  - *“data drift”*

# PROBLEM #2: Duplicate key errors

## Example:

```
[user-lab@hostdb sandboxes]$ cd dupl_key/

[user-lab@hostdb dupl_key]$ ./start_all
# executing "start" on /home/user-lab/sandboxes/dupl_key
executing "start" on master
... sandbox server started
executing "start" on slave 1
... sandbox server started

[user-lab@hostdb dupl_key]$ ./s1
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
```

# PROBLEM #2: Duplicate key errors

Example:

```
slave1 [localhost] {msandbox} (test) > show slave status\G
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 127.0.0.1
                  Master_User: rsandbox
...
             Slave_IO_Running: Yes
            Slave_SQL_Running: No
...
                   Last_Errno: 1062
                   Last_Error: Error 'Duplicate entry '3' for key
   'PRIMARY'' on query. Default database: 'test'. Query: 'insert
   into dupe_test (val) values ('master3'), ('master4')'
...
        Seconds_Behind_Master: NULL
```

# PROBLEM #2: Duplicate key errors

**Example (cont.)**

```
slave1 [localhost] {msandbox} ((none)) > use test;

slave1 [localhost] {msandbox} (test) > show create table dupe_test\G
*************************** 1. row ***************************
       Table: dupe_test
Create Table: CREATE TABLE `dupe_test` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `val` varchar(20) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1
1 row in set (0.00 sec)

slave1 [localhost] {msandbox} (test) > select * from dupe_test where id = 3;
+----+--------+
| id | val    |
+----+--------+
|  3 | slave1 |
+----+--------+
1 row in set (0.00 sec)
```

# PROBLEM #2: Duplicate key errors

**Example (cont.)**

```
[user-lab@hostdb dupl_key]$ cd master

[user-lab@hostdb master]$ mysqlbinlog data/mysql-bin.000002
...
# at 521
#140401  7:05:14 server id 1  end_log_pos 589   Query     thread_id=3
    exec_time=0     error_code=0
SET TIMESTAMP=1396335914/*!*/;
BEGIN
/*!*/;
# at 589
#140401  7:05:14 server id 1  end_log_pos 617   Intvar
SET INSERT_ID=3/*!*/;
# at 617
#140401  7:05:14 server id 1  end_log_pos 739   Query     thread_id=3
    exec_time=0     error_code=0
SET TIMESTAMP=1396335914/*!*/;
insert into dupe_test (val) values ("master3"), ('master4')
/*!*/;
...
```

# SOLUTION #2: Duplicate key errors

## General solutions:

1. recreate slave from fresh backup of master (preferred)
2. get replication running again, fix data differences later

We will be doing solution #2.

# SOLUTION #2: Duplicate key errors

To check if the error is due to a direct write to slave:

```
[user-lab@hostdb dupl_key]$ cd node1

[user-lab@hostdb node1]$ pwd
/home/user-lab/sandboxes/dupl_key/node1

[user-lab@hostdb node1]$ ls data/mysql-bin.*
data/mysql-bin.000001  data/mysql-bin.000002  data/mysql-
bin.index

[user-lab@hostdb node1]$ mysqlbinlog --server-id=101 data/mysql-
bin.000002 | grep dupe_test
insert into dupe_test (val) values ('slave1')
```

# SOLUTION #2: Duplicate key errors

Do this (if it's a read slave):

```
[mysqld]
read_only
```

Do ***not*** do this:

```
[mysqld]
slave-skip-errors = all
```

# SOLUTION #2: Duplicate key errors

## Hack #1: Get slave caught up

Manually:

```
[user-lab@hostdb dupl_key]$ ./s1

slave1 > stop slave; set global sql_slave_skip_counter = 1;
   start slave;

slave1 > show slave status\G
```

If you have a lot of events to skip, you can use **pt-slave-restart**.

# SOLUTION #2: Duplicate key errors

## Hack #1: Get slave caught up

- note this skips binary log “events”
  - for InnoDB tables: entire transaction
- can’t easily guarantee that master and slave data are equivalent
- later: run checksum to compare master and slave

# SOLUTION #2: Duplicate key errors

## Hack #1: Get slave caught up w/GTID

- The previous syntax applies to regular replication (available in 5.6 and previous versions)
- In MySQL 5.6 with GTID mode enabled the concept changes somewhat - skipping statements is not allowed when "`enforce-gtid-consistency = 1`"
- Instead of skipping transactions we need to "inject" empty transactions

# SOLUTION #2: Duplicate key errors

## Hack #1: Get slave caught up w/GTID

First identify the transaction causing the issue by connecting to the slave node and checking replication:

```
[user-lab@hostdb dupl_key_gtid]$ ./s1

slave1 > show slave status\G
...
            Last_SQL_Error: Error 'Duplicate entry '6' for
  key 'PRIMARY'' on query. Default database: 'test'.
  Query: 'insert into dupe_test (id,val) values
  (null,'stringa'), (null,'stringb'),(null,'stringc'),
  (null,'stringd'), (null,'stringe'),(null,'stringf')'
```

# SOLUTION #2: Duplicate key errors

## Hack #1: Get slave caught up w/GTID

```
...
            Retrieved_Gtid_Set: 987abf63-5454-11e4-
   ae18-22000af80ef3:1-4
            Executed_Gtid_Set: 987abf63-5454-11e4-
   ae18-22000af80ef3:1-3,
a0acd3c7-5454-11e4-ae18-22000af80ef3:1
                Auto_Position: 0
```

In the output we can see the set of GTIDs 1 - 4 has been retrieved however only GTIDs 1 - 3 have been *executed.

So we need to “skip” GTID 4 (the failing transaction) by replacing it with an empty transaction.

# SOLUTION #2: Duplicate key errors

## Hack #1: Get slave caught up w/GTID

To do this we set the value of GTID_NEXT to the value of the GTID we want to inject and commit an empty transaction:

```
slave1 > SET GTID_NEXT='987abf63-5454-11e4-
   ae18-22000af80ef3:4';
   slave1 > BEGIN;COMMIT;  -- empty transaction
   slave1 > SET GTID_NEXT='AUTOMATIC';
   slave1 > START SLAVE; SHOW SLAVE STATUS\G
```

# PROBLEM #3: Data drift

- Statement-based binary logging can result in incorrect data on the slave
  - “nondeterministic”
  - updates with LIMIT but no ORDER BY
  - certain system and math functions
  - etc.
- ROW and MIXED binary logging avoids this
- CREATE TABLE … SELECT statements are not safe for statement based replication or row-based w/GTID enabled (as statement is split in two parts with the same GTID)

# PROBLEM #3: Data drift

```
[user-lab@ip-10-74-3-30 dupl_key]$ ./m

master [localhost] {msandbox} ((none)) > use test;

master [localhost] {msandbox} (test) > insert into dupe_test (val) values (uuid());
Query OK, 1 row affected, 2 warnings (0.04 sec)

master [localhost] {msandbox} (test) > show warnings\G
*************************** 1. row ***************************
  Level: Warning
   Code: 1265
Message: Data truncated for column 'val' at row 1
*************************** 2. row ***************************
  Level: Note
   Code: 1592
Message: Unsafe statement written to the binary log using statement format since
BINLOG_FORMAT = STATEMENT. Statement is unsafe because it uses a system function
that may return a different value on the slave.
2 rows in set (0.00 sec)
```

# PROBLEM #3: Data Drift

```
master [localhost] {msandbox} (test) > select * from dupe_test order by id
desc limit 1;
+----+----------------------+
| id | val                  |
+----+----------------------+
|  9 | f6fe0011-b932-11e3-a |
+----+----------------------+
1 row in set (0.02 sec)

slave1 [localhost] {msandbox} ((none)) > select * from test.dupe_test order
by id desc limit 1;
+----+----------------------+
| id | val                  |
+----+----------------------+
|  9 | f6ff6a8c-b932-11e3-a |
+----+----------------------+
1 row in set (0.00 sec)
```

# SOLUTION #3: Fixing data drift/Incorrect slave

## Hack #2: Checksum and sync slave data

- pt-table-checksum
  - checksums tables in “chunks” of rows
  - can use replication to compare master and slave
  - can write to checksum table, results on slave

- pt-table-sync
  - can be run without pt-table-checksum
  - can sync data via REPLACE/DELETE statements run on master or printed out for review

# SOLUTION #3: Fixing data drift/Incorrect slave

Checksum *

```
[user-lab@hostdb dupl_key]$ pt-table-checksum -uroot --ask-pass --
replicate test.checksum --host 127.0.0.1 --port 20000
Enter MySQL password: msandbox
...
            TS ERRORS  DIFFS     ROWS  CHUNKS SKIPPED    TIME TABLE
04-01T00:48:17      0      0        0       1       0   0.051
mysql.columns_priv
04-01T00:48:17      0      0        0       1       0   0.052 mysql.db
04-01T00:48:17      0      0        0       1       0   0.049
mysql.event
04-01T00:48:17      0      0        0       1       0   0.050
mysql.func
…
10-22T12:16:57      0      1        5       1       0   0.037
test.dupe_test
```

# SOLUTION #3: Fixing data drift/Incorrect slave

## Checksum results check:

```
[user-lab@hostdb dupl_key]$ ./s1

slave1 [localhost] {msandbox} (test) > SELECT db, tbl, SUM(this_cnt) AS
total_rows, COUNT(*) AS chunks
FROM test.checksum
WHERE (
  master_cnt <> this_cnt
  OR master_crc <> this_crc
  OR ISNULL(master_crc) <> ISNULL(this_crc))
  GROUP BY db, tbl;
+------+-----------+------------+--------+
| db   | tbl       | total_rows | chunks |
+------+-----------+------------+--------+
| test | dupe_test |          4 |      1 |
+------+-----------+------------+--------+
1 row in set (0.00 sec)
```

# SOLUTION #3: Fixing data drift/Incorrect slave

Checksum Results - Simplified

```
slave1 [localhost] {msandbox} (test) > select
distinct db, tbl from test.checksum where master_crc
<> this_crc;
+------+-----------+
| db   | tbl       |
+------+-----------+
| test | dupe_test |
+------+-----------+
1 row in set (0.00 sec)
```

# SOLUTION #3: Fixing data drift/Incorrect slave

Sync Demonstration:

```
[user-lab@hostdb dupl_key]$ pt-table-sync -uroot --ask-pass --print --
replicate test.checksum --sync-to-master --socket=/tmp/
mysql_sandbox20001.sock D=test,t=dupe_test
Enter password for DSN D=test,S=/tmp/
mysql_sandbox20001.sock,t=dupe_test,u=root:
msandbox

REPLACE INTO `test`.`dupe_test`(`id`, `val`) VALUES ('3', 'master3') /
*percona-toolkit src_db:test src_tbl:dupe_test src_dsn:D=test,P=23992,S=/
tmp/mysql_sandbox20001.sock,h=127.0.0.1,p=...,t=dupe_test,u=root
dst_db:test dst_tbl:dupe_test dst_dsn:D=test,S=/tmp/
mysql_sandbox20001.sock,p=...,t=dupe_test,u=root lock:1 transaction:1
changing_src:test.checksum replicate:test.checksum bidirectional:0 pid:
26192 user:user-lab host:hostdb*/;
...
```

# SOLUTION #3: Fixing data drift/Incorrect slave

Sync to master data:

```
[user-lab@hostdb dupl_key]$ pt-table-sync -uroot --ask-
pass --execute --replicate test.checksum --sync-to-master
--socket=/tmp/mysql_sandbox20001.sock D=test,t=dupe_test
Enter password for DSN D=test,S=/tmp/
mysql_sandbox20001.sock,t=dupe_test,u=root:
msandbox
```

**Note**: pt-table-sync is connecting to the slave socket to start, as that's the location of the test.checksum table results.

# SOLUTION #3: Fixing data drift/Incorrect slave

Sync Check:

```
[user-lab@hostdb dupl_key]$ ./m -e "checksum table test.dupe_test"
+----------------+------------+
| Table          | Checksum   |
+----------------+------------+
| test.dupe_test | 305013415 |
+----------------+------------+

[user-lab@hostdb dupl_key]$ ./s1 -e "checksum table
test.dupe_test"
+----------------+------------+
| Table          | Checksum   |
+----------------+------------+
| test.dupe_test | 305013415 |
+----------------+------------+
```

# Significant performance issues

Presented by: Alkin Tezuysal, Emanuel Calvo, Okan Buyukyilmaz

April 18, 2016

# Significant Performance Issues

- Diagnosing performance issues via OS tools, performance counters/graphs and MySQL utilities/ commands.
- Mitigating/Triaging issues with such techniques as relaxing durability, dynamic variable changes, command line fixes and managing connections/ commands.

# Agenda - Part 3

- System Bottlenecks
  - Verify Operating System metrics
  - Run diagnostics
- MySQL Bottlenecks
  - MySQL CLI
  - MySQL tools
  - External MySQL tools
- Configuration Setting Changes
  - Dynamic
  - Static

# Bottlenecks Explained

- Trends
  - Memory Utilisation
  - CPU Utilisation
  - Disk Utilisation
  - Network Utilisation
- Current Status
  - High Load
  - Swapping
  - I/O Wait

# What to look first?

- Sudden performance issues - what changed?
  - Software release
  - Drive failure, temperature warnings
  - Database schema and configuration changes
  - OS patches, packages, updates
- Issues over time - study the graphs
  - Application servers added or user traffic increase
  - Swap space usage
  - Drive rebuilding
  - Memory leaks
  - Table growth
  - Buffer pool size + overhead

# None of the above? Dig more…

- Operating System Diagnostics
  - vmstat
  - iostat
  - ps
  - top
  - sar
  - Strace
  - lsof
  - ifstat
  - dmesg

# OS stats - vmstat

```
root@sandbox:~# vmstat 1 10
procs -----------memory---------- ---swap-- -----io---- -system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa
 9  1     64   3812 103808  86128    0    0     0     6   14   26  0  0 100  0
16  1     64   3992 104124  85680    0    0     0 12192 1009 38386 50 48  0  2
11  0     64   3992 104472  85212    0    0     0 15508 1111 36948 46 53  0  1
 5  0     64   4172 104644  84908    0    0     0 12440 1056 37181 49 49  0  1
10  0     64   3872 104972  84904    0    0     0 11988  973 32739 54 45  0  1
 8  1     64   6808 105244  81720    0    0     0 15572 1065 37857 55 44  0  1
16  1     64   6448 105600  81708    0    0     0 15552 1042 36428 51 48  0  1
 8  1     64   6028 105972  81708    0    0     0 17340 1073 35714 54 42  0  4
11  1     64   5728 106360  81700    0    0     0 14040 1071 36160 48 51  0  1
14  1     64   5248 106736  81716    0    0     0 15596 1057 37619 51 47  0  2
```

# OS Stats - iostat

- iostat -y -x 3
  - -y - throw away stats from the last system boot
  - -x - extended statistics
  - 3 - interval seconds

```
root@sandbox:~# iostat -k -d -x 1 3 /dev/sd?
Linux 2.6.32-38-generic (sandbox)   03/27/2014  _i686_      (1 CPU)

Device:         rrqm/s   wrqm/s     r/s     w/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await  svctm  %util
sda               0.00     0.92    0.01    0.25     0.29     4.45    36.43     0.00    0.53   0.30   0.01

Device:         rrqm/s   wrqm/s     r/s     w/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await  svctm  %util
sda               0.00  2618.00    1.00  881.00    24.00 13176.00    29.93     0.32    0.38   0.28  24.40

Device:         rrqm/s   wrqm/s     r/s     w/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await  svctm  %util
sda               0.00  2653.54    0.00  848.48     0.00 13163.64    31.03     0.38    0.45   0.32  27.07
```

# OS stats – top

- nice top - defer priority to the issue
  - 1 - what are the cores doing
  - -u mysql - what is mysql doing
  - H - show threads
  - < > - change sort order

```
root@sandbox:~# top
top - 20:32:25 up 5 days, 13:22,  2 users,  load average: 2.40, 0.56, 0.24
Tasks:  80 total,   2 running,  78 sleeping,   0 stopped,   0 zombie
Cpu(s): 50.2%us, 45.1%sy,  0.0%ni,  0.0%id,  4.0%wa,  0.0%hi,  0.7%si,  0.0%st
Mem:    250628k total,   244184k used,     6444k free,    35012k buffers
Swap:  1114104k total,       64k used,  1114040k free,   152528k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 1981 mysql     20   0  162m  30m 6676 S 79.5 12.6   1:36.01 mysqld
 4494 root      20   0 15484 3632 1420 S 13.6  1.4   0:02.80 sysbench
  192 root      20   0     0    0    0 R  1.7  0.0   0:00.71 jbd2/dm-0-8
  174 root      20   0     0    0    0 S  1.3  0.0   0:00.48 kdmflush
    1 root      20   0  2672 1560 1212 S  0.0  0.6   0:00.73 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.02 kthreadd
    3 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/0
```

# #top – Shift +H , u - mysql

```
top - 20:38:21 up 5 days, 13:28,  2 users,  load average: 8.39, 6.40, 3.09
Tasks: 129 total,  15 running, 114 sleeping,   0 stopped,   0 zombie
Cpu(s): 53.0%us, 45.6%sy,  0.0%ni,  0.0%id,  0.7%wa,  0.0%hi,  0.7%si,
0.0%st
Mem:    250628k total,   246868k used,     3760k free,   126316k buffers
Swap:  1114104k total,      116k used,  1113988k free,    61984k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 4506 mysql     20   0  174m  32m 6588 R  5.3 13.5   0:15.92 mysqld
 4541 mysql     20   0  174m  32m 6588 S  5.3 13.5   0:04.87 mysqld
 1998 mysql     20   0  174m  32m 6588 R  5.0 13.5   0:16.10 mysqld
 3166 mysql     20   0  174m  32m 6588 R  5.0 13.5   0:16.69 mysqld
 4502 mysql     20   0  174m  32m 6588 R  5.0 13.5   0:16.02 mysqld
 4505 mysql     20   0  174m  32m 6588 R  5.0 13.5   0:15.96 mysqld
 4507 mysql     20   0  174m  32m 6588 R  5.0 13.5   0:16.01 mysqld
 4537 mysql     20   0  174m  32m 6588 R  5.0 13.5   0:04.79 mysqld
 4539 mysql     20   0  174m  32m 6588 S  5.0 13.5   0:04.84 mysqld
 4540 mysql     20   0  174m  32m 6588 R  5.0 13.5   0:04.82 mysqld
```

# OS stats - ps

- ps –efww
  - -e - all processes
  - -f - full format listing with command arguments
  - -ww - unlimited width

```
[root@localhost ~]# ps -efww | grep -i mysql | grep -v grep
root 2147 1 0 07:14 ? 00:00:00 /bin/sh /usr/bin/mysqld_safe --datadir=/var/lib/mysql --pidfile=/
var/lib/mysql/localhost.localdomain.pid
mysql 2349 2147 0 07:14 ? 00:00:00 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --
plugin-dir=/usr/lib64/mysql/plugin --user=mysql --log-error=/var/log/mysqld.log --pidfile=/
var/lib/mysql/localhost.localdomain.pid --socket=/var/lib/mysql/mysql.sock
```

# MySQL Tools (built-in)

- mysql client (CLI)
- mysqladmin
- mysqlbinlog
- mysql error log
- mysql slow query log

# External Mysql tools

- Percona tools
  - pt-query-digest
  - pt-config-diff
  - pt-stalk
- tcpdump
- Innotop
- Bash commands

# MySQL – Information/Performance Schema

- **SHOW FULL PROCESSLIST**

```
SELECT id, user, host, db, command, time, state, LEFT(info, 80) AS info
  FROM information_schema.processlist
 WHERE command NOT IN ('Sleep', 'Binlog Dump')
 ORDER BY time ASC;
```

```
SELECT PROCESSLIST_ID AS id, PROCESSLIST_USER AS user, PROCESSLIST_HOST AS host,
PROCESSLIST_DB AS db
     , PROCESSLIST_COMMAND AS command, PROCESSLIST_TIME AS time,
PROCESSLIST_STATE AS state, LEFT(PROCESSLIST_INFO, 80) AS info
  FROM performance_schema.threads
 WHERE PROCESSLIST_ID IS NOT NULL
   AND PROCESSLIST_COMMAND NOT IN ('Sleep', 'Binlog Dump')
 ORDER BY PROCESSLIST_TIME ASC;
```

# MySQL CLI – Information/Performance Schema II

**List of long running queries**

UPDATE 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_end-timer_start)/1000000000/COUNT(*), 1) AS avg_exec_ms
    , ROUND(MIN(timer_end-timer_start)/1000000000, 1) AS min_exec_ms
    , ROUND(MAX(timer_end-timer_start)/1000000000, 1) AS max_exec_ms
    , ROUND(SUM(timer_wait)/1000000000, 1) AS tot_wait_ms
    , ROUND(SUM(timer_wait)/1000000000/COUNT(*), 1) AS avg_wait_ms
    , ROUND(MIN(timer_wait)/1000000000, 1) AS min_wait_ms
    , ROUND(MAX(timer_wait)/1000000000, 1) AS max_wait_ms
    , ROUND(SUM(lock_time)/1000000000, 1) AS tot_lock_ms
    , ROUND(SUM(lock_time)/1000000000/COUNT(*), 1) AS avglock_ms
    , ROUND(MIN(lock_time)/1000000000, 1) AS min_lock_ms
    , ROUND(MAX(lock_time)/1000000000, 1) AS max_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 events_statements_history_long
 JOIN information_schema.global_status AS isgs
WHERE isgs.variable_name = 'UPTIME'
GROUP BY LEFT(digest_text,64)
ORDER BY tot_exec_ms DESC
;
```

# MySQL CLI – Information/Performance Schema III

- **Who created temporary (disk) tables**

```
SELECT user, host, event_name, count_star AS cnt, sum_created_tmp_disk_tables AS
tmp_disk_tables, sum_created_tmp_tables AS tmp_tables
  FROM performance_schema.events_statements_summary_by_account_by_event_name
 WHERE sum_created_tmp_disk_tables > 0
    OR sum_created_tmp_tables > 0;
```

```
SELECT schema_name, substr(digest_text, 1, 40) AS statement, count_star AS cnt,
sum_created_tmp_disk_tables AS tmp_disk_tables, sum_created_tmp_tables AS tmp_tables
  FROM performance_schema.events_statements_summary_by_digest
 WHERE sum_created_tmp_disk_tables > 0
    OR sum_created_tmp_tables > 0;
```

# MySQL CLI – Information/Performance Schema IV

- **Storage Engines types per schema**

```
SELECT table_schema AS `schema`, engine, COUNT(*) AS `tables`
     , ROUND(SUM(data_length)/1024/1024, 0) AS data_mb,
ROUND(SUM(index_length)/1024/1024, 0) index_mb
  FROM information_schema.tables
 WHERE table_schema NOT IN ('mysql', 'information_schema',
'performance_schema')
   AND engine IS NOT NULL
 GROUP BY table_schema, engine;
```

# MySQL CLI – Information/Performance Schema V

- **Harmful SQL queries of users**

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

# MySQL Tools - #mysqlbinlog - I

$mysqlbinlog proddb2-433574-bin-log.000420 | egrep '^#.*exec_time' | egrep -v 'exec_time=(4294967295|0)' | sed -e 's/exec_time=//' | sort -r -n -k 10 | head -n 20

```
$mysqlbinlog /path/to/mysql-bin.000999 | grep -i -e "^update" -e "^insert" -e
"^delete" -e "^replace" -e "^alter"  |    cut -c1-100 | tr '[A-Z]' '[a-z]' |
sed -e "s/\t/ /g;s/\`//g;s/(.*$//;s/ set .*$//;s/ as .*$//" | sed -e "s/
where .*$//" | sort | uniq -c | sort -nr
  33389 update e_acc
  17680 insert into r_b
  17680 insert into e_rec
  14332 insert into rcv_c
  13543 update e_rec
  10805 update loc
   3339 insert into r_att
   2781 insert into o_att
```

# MySQL Tools - #mysqlbinlog - II

```
$ mysqlbinlog al-db2.001079|  pt-query-digest --type=binlog --group-
by=distill > /tmp/writes.txt

$ head –n 10000 /tmp/writes.txt > /tmp/writes_10000.txt

$ egrep '^#' /tmp/writes_1000.txt | awk '{print $10}' | grep exec_time | sort
| uniq -c
      2 exec_time=0
    156 exec_time=4
   1376 exec_time=5

$ mysqlbinlog pathtobinlog | pt-query-digest --type binlog --limit 30 --
order-by 'Query_time:cnt' > output.txt
```

# MySQL slow query log

- Set slow query log 0 seconds (warning)

```
mysql> SET GLOBAL log_slow_verbosity='standard';
Query OK, 0 rows affected (0.00 sec)

mysql> SET GLOBAL slow_query_log_use_global_control='long_query_time';
Query OK, 0 rows affected (0.00 sec)

mysql> SET GLOBAL long_query_time=0;
Query OK, 0 rows affected (0.00 sec)

mysql> \! mv /var/log/mysql/mysql-slow.log /var/log/mysql/mysql-slow.log__
mysql> FLUSH LOGS;
Query OK, 0 rows affected (0.08 sec)
```

- Run detailed pt-query-digest

# External tools – #pt-query-digest

- All queries ordered by time:

```
#pt-query-digest --limit 100% /var/log/mysql/mysql-slow.log > /root/bb/mysql-
 slow-db1.time.digest
```

- All queries ordered by count:

```
#pt-query-digest --limit 100% /var/log/mysql/mysql-slow.log --order-by 'Query_time:cnt' > /root/bb/
 mysql-slow-db1.cnt.digest
```

- All queries ordered by row examined:

```
#pt-query-digest --limit 100% /var/log/mysql/mysql-slow.log --order-by
 'Rows_examined:sum' > /root/palominodb/mysql-slow-db1.rows.digest
```

- All queries longer than 1 seconds:

```
#pt-query-digest mysql-slow.log.1 --filter '$event->{Query_time} > 1' > /tmp/
 mysql-slow.log.1_1sec.txt
```

# External tools – #pt-query-digest II

- All queries by time range:

```
#pt-query-digest --limit 100% --since "2013-09-10 13:00:00" --until
 "2013-09-10 15:00:00"  > spike10sept.diges
```

- All queries by time range compressed:

```
#zcat slow-log.2.gz | pt-query-digest --limit 100% --since "2013-09-10
 13:00:00" --until "2013-09-10 15:00:00"  > spike10sept.digest
```

- All queries by time range ordered by rows examined:

```
#pt-query-digest --limit 100% mysql_slow.log  --since "2014-01-29 19:00:00"
 --until "2014-01-30 03:00:00" --order-by 'Rows_examined:sum' >
 013014.spike.txt
```

# External tools – #tcpdump

- Use tcpdump to capture all traffic:

```
#tcpdump -s 65535 -x -nn -q -tttt -i any -c 1000000 port 3306  >
  mysql.tcp.txt
```

- Generate slowlog from output of tcpdump:

```
#pt-query-digest --output tcpdump.slow.log --no-report --type tcpdump
  mysql.tcp.txt
```

# External tools – #innotop

## Top queries

```
When    Load  QPS     Slow  QCacheHit  KCacheHit  BpsIn    BpsOut
Now     0.00  8.30k      0      0.00%    100.00%  206.07k   7.10M
Total   0.00  0.00       0      0.00%      0.00%    0.00    0.11

Cmd       ID      State                 User   Host           DB        Time
 Query
Execute       46  statistics            alkin  localhost      sysbench  00:00
 SELECT DISTINCT c from sbtest where id between 49875 and 49975
Execute       47  statistics            alkin  localhost      sysbench  00:00
 SELECT c from sbtest where id=50259
Execute       48  Sorting result        alkin  localhost      sysbench  00:00
 SELECT c from sbtest where id between 49925 and 50024 order by
Execute       49  Updating              alkin  localhost      sysbench  00:00
 UPDATE sbtest set k=k+1 where id=?
Execute       50  statistics            alkin  localhost      sysbench  00:00
 SELECT c from sbtest where id=43618
```

# External tools – Bash Commands

- Space left

```
[root@localhost ~]# echo $(($(stat -f --format="%a*%s" /var/lib/mysql)/1024/1024))M
14446M
```

- Release swap

```
[root@localhost ~]# swapoff -a && swapon –a
```

- Unused InnoDB tables

```
[root@localhost ~]# find /var/lib/mysql/ -name *.ibd -mtime +60 | sort | xargs du -hc
```

# Configuration Setting Changes

- Dynamic variables
  - innodb_io_capacity
  - innodb_lock_wait_timeout
  - lock_wait_timeout
  - query_cache_size
  - table_open_cache
- Static variables
  - innodb_buffer_pool_size
  - innodb_buffer_pool_instances
  - open_files_limit
  - skip_name_resolve
  - tmpdir

# Thank you

## Follow me at

- Twitter: @ask_dba
- Linked in: http://tr.linkedin.com/in/askdba/