Basic External Troubleshooting Tools

March, 15, 2018

Sveta Smirnova

Table of Contents

PERCONA

What This Webinar is About?

- Instruments for troubleshooting
- Which I use
- Helps to discover and solve problems

PERCONA

What This Webinar is Not?

- Complete guide
- Cover of
  - All existent tools
  - MySQL Test Framework
  - Build-related tools, such as mysql_config
  - Third-party benchmarking tools

PERCONA

MySQL Sandbox, DBdeployer and Docker

Ideal For Quick Installation

- Software deploys
- Development servers
- Upgrade/downgrade tests

PERCONA

MySQL Sandbox and DBdeployer

A tool that deploys MySQL database servers easily. ©

PERCONA

MySQL Sandbox and DBdeployer

A tool that deploys MySQL database servers easily. ©

- Installs
  - Single server
  - Replicated Servers
    - Supports Group Replication
  - Multiple instances

PERCONA

MySQL Sandbox and DBdeployer

A tool that deploys MySQL database servers easily. ©

- Installs
  - Single server
  - Replicated Servers
    - Supports Group Replication
  - Multiple instances
- Supports custom configuration

PERCONA

MySQL Sandbox Usage

- Installation

```
perl Makefile.PL
make
[make test] - optionally
sudo make install
```

PERCONA

MySQL Sandbox Usage

- Single Server

```
$ make_sandbox Percona-Server-5.7.21-20-Linux.x86_64.ssl100.tar.gz
unpacking /home/sveta.smirnova/Percona-Server-5.7.21-20-Linux.x86_64.ssl100.tar.gz
Executing low_level_make_sandbox
    ...
do you agree? ([Y],n)
# Starting server
... sandbox server started
# Loading grants
Your sandbox server was installed in $SANDBOX_HOME/msb_5_7_21
$ cd $SANDBOX_HOME/msb_5_7_21
$ ./use -e "select @@innodb_buffer_pool_size/1024/1024"
+-------------------------------------+
| @@innodb_buffer_pool_size/1024/1024 |
+-------------------------------------+
|                          128.00000000 |
+-------------------------------------+
```

PERCONA

MySQL Sandbox Usage

- Replicated Setup

```
$ make_replication_sandbox Percona-Server-5.7.21-20-Linux.x86_64.ssl100.tar.gz
installing and starting master
installing slave 1
installing slave 2
starting slave 1
. sandbox server started
starting slave 2
.. sandbox server started
initializing slave 1
initializing slave 2
replication directory installed in $SANDBOX_HOME/rsandbox_Percona-Server-5_7_21
```

PERCONA

MySQL Sandbox Usage

- Multiple Instances

```
$ make_multiple_sandbox --how_many_nodes=5 --circular --sandbox_base_port=8000 \
> Percona-Server-5.7.21-20-Linux.x86_64.ssl100.tar.gz
installing node 1
installing node 2
installing node 3
installing node 4
installing node 5
...
Circular replication activated
group directory installed in $SANDBOX_HOME/multi_msb_Percona-Server-5_7_21
```

PERCONA

MySQL Sandbox Usage

```
$ cd  $SANDBOX_HOME/multi_msb_Percona-Server-5_7_21
$ ls -l
total 100
-rwxr-xr-x check_slaves
-rwxr-xr-x clear_all
-rw-r--r-- connection.json
-rw-r--r-- default_connection.json
-rwxr-xr-x n1          -rw-r--r-- README
-rwxr-xr-x n2          -rwxr-xr-x restart_all
-rwxr-xr-x n3          -rwxr-xr-x send_kill_all
-rwxr-xr-x n4          -rwxr-xr-x set_circular_replication.sh
-rwxr-xr-x n5          -rwxr-xr-x start_all
drwxr-xr-x node1       -rwxr-xr-x status_all
drwxr-xr-x node2       -rwxr-xr-x stop_all
drwxr-xr-x node3       -rwxr-xr-x use_all
drwxr-xr-x node4
drwxr-xr-x node5
```

PERCONA

MySQL Sandbox Usage

- Custom Options

```
$ cat > webinar.cnf
[mysqld]
innodb_buffer_pool_size=1G

^C
$ make_sandbox Percona-Server-5.7.21-20-Linux.x86_64.ssl100.tar.gz -- --sandbox_port=35721 \
> -m ./webinar.cnf
...
# Loading grants
Your sandbox server was installed in $SANDBOX_HOME/msb_5_7_21
$ cd $SANDBOX_HOME/msb_5_7_21
$ ./use -e "select @@innodb_buffer_pool_size/1024/1024"
+-------------------------------------+
| @@innodb_buffer_pool_size/1024/1024 |
+-------------------------------------+
|                       1024.00000000 |
+-------------------------------------+
```

PERCONA

DBdeployer Usage

- Installation ©

```
$ VERSION=0.3.1
$ origin=https://github.com/datacharmer/dbdeployer/releases/download/$VERSION
$ wget $origin/dbdeployer-$VERSION.linux.tar.gz
$ tar -xzf dbdeployer-$VERSION.linux.tar.gz
$ chmod +x dbdeployer-$VERSION.linux
$ mv dbdeployer-$VERSION.linux dbdeployer
```

PERCONA

DBdeployer Usage

- Prepare Server

```
$ ./dbdeployer --unpack-version=5.7.21 unpack Percona-Server-5.7.21-20-Linux.x86_64.ssl100.tar.gz
Unpacking tarball Percona-Server-5.7.21-20-Linux.x86_64.ssl100.tar.gz
to /home/sveta.smirnova/opt/mysql/5.7.21
.........100.........200.........300.........400.........500.........600.........700.........800..
.......900.........1000.........1100.........1200.........1300.........1400.........1500.........
1600.........1700.........1800.........1900.........2000.........2100.........2200.........2300...
...
17400.........17500.........17600.........17700.........17800.........17900.........18000.........
18100.........18200.........18300.........18400..18421
```

DBdeployer Usage

- Single Server

```
$ ./dbdeployer deploy single 5.7.21 --port 35721
run 'dbdeployer usage single' for basic instructions'
. sandbox server started
$ cd /bigdisk/sveta/sandbox_webinar/msb_5_7_21
$ ./use -e "select @@innodb_buffer_pool_size/1024/1024"
+--------------------------------------+
| @@innodb_buffer_pool_size/1024/1024 |
+--------------------------------------+
|                         128.00000000 |
+--------------------------------------+
```

PERCONA

DBdeployer Usage

- Replicated Setup

```
$ ./dbdeployer deploy replication 5.7.21
Installing and starting master
. sandbox server started
Installing and starting slave 1
. sandbox server started
Installing and starting slave 2
. sandbox server started
/bigdisk/sveta/sandbox_webinar/rsandbox_5_7_21/initialize_slaves
initializing slave 1
initializing slave 2
Replication directory installed in /bigdisk/sveta/sandbox_webinar/rsandbox_5_7_21
run 'dbdeployer usage multiple' for basic instructions'
```

PERCONA

DBdeployer Usage

- Multiple Instances

```
$ ./dbdeployer deploy multiple 5.7.21
Installing and starting node 1
. sandbox server started
Installing and starting node 2
. sandbox server started
Installing and starting node 3
. sandbox server started
multiple directory installed in /bigdisk/sveta/sandbox_webinar/multi_msb_5_7_21
run 'dbdeployer usage multiple' for basic instructions'
```

PERCONA

DBdeployer Usage

- Custom Options

```
$ ./dbdeployer deploy single 5.7.21 --port 35721 --my-cnf-options=innodb_buffer_pool_size=1G
run 'dbdeployer usage single' for basic instructions'
. sandbox server started
$ cd /bigdisk/sveta/sandbox_webinar/msb_5_7_21
$ ./use -e "select @@innodb_buffer_pool_size/1024/1024"
+-------------------------------------+
| @@innodb_buffer_pool_size/1024/1024 |
+-------------------------------------+
|                       1024.00000000 |
+-------------------------------------+
```

PERCONA

DBdeployer Usage

- Run command globally

```
$ ./dbdeployer global use "select @@server_id"
# Running "use" on msb_5_7_21
@@server_id
0

# Running "use_all" on multi_msb_5_7_21
# server: 1
@@server_id
100
# server: 2
@@server_id
200
# server: 3
@@server_id
300

# Running "use_all" on rsandbox_5_7_21
# master
@@server_id
100
# server: 1
@@server_id
200
# server: 2
@@server_id
300
```

Docker

The world's leading software container platform to modernize applications without disruption. ©

PERCONA

Docker

The world's leading software container platform to modernize applications without disruption. ©

- You can create any pre-defined setup

PERCONA

Docker

The world's leading software container platform to modernize applications without disruption. ©

- You can create any pre-defined setup
- Ready-to-use containers by
  - Oracle MySQL Team
  - Percona
  - Docker MySQL
  - Docker MariaDB

PERCONA

Docker Usage

```
$ docker run --name percona-server -e MYSQL_ROOT_PASSWORD=secret -d percona/percona-server
676c502c72aa2e5137f3bd81a13ad99329e52f4f38ee24ea354956acfba7cd74
$ docker exec -it percona-server bash
mysql@676c502c72aa:/$ mysql -uroot -psecret
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.21-20 Percona Server (GPL), Release '20', Revision 'ed217b06ca3'

Copyright (c) 2009-2018 Percona LLC and/or its affiliates
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> \q
Bye
```


PERCONA

When to Use?

Feature	Sandbox	Docker
Single Server	✓	✓
Replication	✓	Scriptable
Complex Setup	✓	Scriptable
PXC/Galera	No support	✓
Custom Options	✓	Scriptable

PERCONA

Tools from Standard MySQL Server Distribution

Installation and Setup

- Come with standard distribution
- Located in `bin` directory
- No additional maintenance required

PERCONA

mysql CLI

- Runs every command you like
- Tested by almost every MySQL user
- Use it to check if wrong behavior is affected by options of your application

PERCONA

mysqladmin

- Runs administrative commands

mysqladmin

- Some can be called from mysql CLI

mysqladmin	mysql CLI
`flush-hosts`	`FLUSH HOSTS`
`kill id,id,...`	`KILL id;`
`processlist`	`SHOW PROCESSLIST`
`reload`	`FLUSH PRIVILEGES`
`start-slave`	`START SLAVE`
...	...

mysqladmin

- Runs administrative commands
- Some can be called from mysql CLI
- Others can be called only from the tool
  - `debug`
  - `ping`
  - `shutdown`

PERCONA

mysqlshow

- Shows structure
- Databases

```
$ mysqlshow  --host=127.0.0.1 --port=13001 --user=root
+--------------------+
|     Databases      |
+--------------------+
| information_schema |
| mtr                |
| mysql              |
| performance_schema |
| sys                |
| test               |
+--------------------+
```

PERCONA

mysqlshow

- Shows structure
- Tables

```
$ mysqlshow  --host=127.0.0.1 --port=13001 --user=root mtr
Database: mtr
+----------------------+
|       Tables         |
+----------------------+
| global_suppressions |
| test_suppressions   |
+----------------------+
```

PERCONA

mysqlshow

- Shows structure
- Columns

```
$ mysqlshow  --host=127.0.0.1 --port=13001 --user=root -t -v sys host_summary
Database: sys  Wildcard: host_summary
+--------------+------------+----------+
|    Tables    | table_type | Columns  |
+--------------+------------+----------+
| host_summary | VIEW       |       12 |
+--------------+------------+----------+
1 row in set.
```

PERCONA

mysqlshow

- Shows structure
- Columns

```
$ mysqlshow  --host=127.0.0.1 --port=13001 --user=root --count -t -v mtr test_suppressions
Database: mtr  Wildcard: test_suppressions
+-------------------+------------+----------+------------+
|      Tables       | table_type | Columns  | Total Rows |
+-------------------+------------+----------+------------+
| test_suppressions | BASE TABLE |        1 |          0 |
+-------------------+------------+----------+------------+
1 row in set.
```

PERCONA

mysqlshow

- Shows structure
- Columns

```
$ mysqlshow  --host=127.0.0.1 --port=13001 --user=root sys version
Database: sys  Table: version
+--------------+------------+-----------------+------+-----+---------+------+------------+---------+
| Field        | Type       | Collation       | Null| Key| Default| Extra| Privileges | Comment |
+--------------+------------+-----------------+------+-----+---------+------+------------+---------+
| sys_version  | varchar(5) | utf8_general_ci | NO  |    |        |      | select,... |         |
| mysql_version| varchar(19)| utf8_general_ci | NO  |    |        |      | select,... |         |
+--------------+------------+-----------------+------+-----+---------+------+------------+---------+
```

mysqlbinlog

- Parses binary log
- Shows what slave applied
- Contains updates which server received
- MySQL Replication Troubleshooting webinar

PERCONA

mysqldumpslow

- Parses slow query log

mysqldumpslow

- Parses slow query log
- `SELECT * FROM t2 WHERE f1=1` and `SELECT * FROM t2 WHERE f1=2` treated as the same query

```
$mysqldumpslow mysql-slow.log
Reading mysql slow query log from
/home/sveta/build/mysql-5.7/data/mysqld57-Thinkie-slow.log
Count: 3 Time=0.03s (0s) Lock=0.03s (0s) Rows=0.7 (2),
root[root]@localhost
SELECT * FROM t2 WHERE f1=N
```

PERCONA

mysqldumpslow

- Parses slow query log
- `SELECT * FROM t2 WHERE f1=1` and `SELECT * FROM t2 WHERE f1=2` treated as the same query
- Does not work with extended slow query log in Percona Server

```
$ ~/build/ps-5.7/bin/mysqldumpslow
"mysqldumpslow.sh" is not currently compatible with Percona extended slow query
log format. Please use "pt-query-digest" from Percona Toolkit instead
(https://www.percona.com/doc/percona-toolkit/2.2/pt-query-digest.html).
```

mysqlslap

- Benchmarking tool which is always available

mysqlslap

- Benchmarking tool which is always available
- Very simple

```
$ mysqlslap --socket=/tmp/mysql.sock --user=root delimiter=";" --create-schema=mstest \
> create="CREATE TABLE mstest(id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,\
> f1 VARCHAR(255)) ENGINE=InnoDB" \
> --query="INSERT INTO mstest(f1) VALUES(MD5(RAND())); SELECT f1 FROM mstest;" \
> --concurrency=10 --iterations=1000
Benchmark
Average number of seconds to run all queries:
0.039 seconds
Minimum number of seconds to run all queries:
0.025 seconds
Maximum number of seconds to run all queries:
0.173 seconds
Number of clients running queries: 10
Average number of queries per client: 2
```

PERCONA

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
```

PERCONA

resolveip

- Links hostname and IP-address

```
$ resolveip percona.com
IP address of percona.com is 74.121.199.234
$ resolveip 74.121.199.234
Host name of 74.121.199.234 is www.percona.com
```


PERCONA

resolve_stack_dump

- Resolves stack dump from symbols

```
$ nm -D -n `which mysqld` > mysqld.symbols
$ resolve_stack_dump -s mysqld.symbols -n temp | c++filt
0xc0b9ad do_sj_dups_weedout(THD*, SJ_TMP_TABLE*) + 1869
0xc0b65c do_sj_dups_weedout(THD*, SJ_TMP_TABLE*) + 1020
0xc10e02 sub_select(JOIN*, QEP_TAB*, bool) + 386
0xc1137a join_materialize_semijoin(QEP_TAB*) + 122
0xc09d35 QEP_TAB::prepare_scan() + 69
0xc10cc0 sub_select(JOIN*, QEP_TAB*, bool) + 64
0xc098c8 JOIN::exec() + 968
0xc7da03 handle_query(THD*, LEX*, Query_result*, unsigned long ...
0x739404 _init + 11628
```

- Troubleshooting MySQL Crashes webinar

PERCONA

Engine-Specific Tools

- MyISAM
  - `myisamchk`
    - Checks and repairs MyISAM tables
    - Can solve issues which `CHECK TABLE` cannot

Engine-Specific Tools

- MyISAM
  - `myisamchk`
    - Checks and repairs MyISAM tables
    - Can solve issues which `CHECK TABLE` cannot
  - `myisam_ftdump`
    - Dumps `FULLTEXT` index

```
$ ~/build/ps-5.7/bin/myisam_ftdump  m1 0
Total rows: 3
Total words: 11
Unique words: 9
Longest word: 11 chars (performance)
Median length: 7
Average global weight: 0.385082
Most common word: 2 times, weight: -0.693147 (percona)
```

PERCONA

Engine-Specific Tools

- InnoDB
  - `innochecksum`
    - Offline checksum utility
    - Checks InnoDB table for corruptions

```
$ innochecksum var/mysqld.1/data/test/i1.ibd; echo $?
0
$ innochecksum corrupted.ibd; echo $?
Fail: page 0 invalid
Exceeded the maximum allowed checksum mismatch count::0
1
```

PERCONA

Engine-Specific Tools

- TokuDB
  - `tokuftdump`
    - Investigates fractal tree
    - Very detail analysis
    - Various options

PERCONA

Engine-Specific Tools

- TokuDB
  - tokuftdump

```
$ tokuftdump --summary /data/employees/employees_main_12_2_1d.tokudb
leaf nodes: 5
non-leaf nodes: 1
Leaf size: 6,647,296
Total size: 6,647,808
Total uncompressed size: 16,022,656
Messages count: 0
Messages size: 0
Records count: 300024
Tree height: 1
height: 0, nodes count: 5; avg children/node: 49.800000
 basement nodes: 249; msg size: 0; disksize: 6,647,296;
 uncompressed size: 16,022,375; ratio: 2.410360
height: 1, nodes count: 1; avg children/node: 5.000000
 msg cnt: 0; msg size: 0; disksize: 512;
 uncompressed size: 281; ratio: 0.548828
```


PERCONA

Engine-Specific Tools

- TokuDB
  - tokuft_logprint
    - Parses TokuDB log files

```
$ cat data/log000000000000.tokulog29 | tokuft_logprint | head
tokulog v.29
xbegin                  'b': lsn=1 xid=1,0 parentxid=0,0 crc=f2617838 len=53
fcreate                 'F': lsn=2 xid=1,0 filenum=1 iname={len=18 data="tokudb.envi...
enq_insert              'I': lsn=3 filenum=1 xid=1,0 key={len=16 data="original_vers...
enq_insert              'I': lsn=4 filenum=1 xid=1,0 key={len=15 data="current_versi...
enq_insert              'I': lsn=5 filenum=1 xid=1,0 key={len=13 data="creation_time...
fcreate                 'F': lsn=6 xid=1,0 filenum=3 iname={len=16 data="tokudb.dire...
xcommit                 'C': lsn=7 xid=1,0 crc=fff04439 len=37
begin_checkpoint        'x': lsn=8 timestamp=1521053564699278 last_xid=1 crc=16f712...
fassociate              'f': lsn=9 filenum=0 treeflags=0 iname={len=15 data="tokudb....
```

Engine-Specific Tools

- MyRocks
  - `sst_dump`
    - Checks, scans and dumps SST files

PERCONA

Engine-Specific Tools

- MyRocks
  - `sst_dump`
    - Checks, scans and dumps SST files
  - `mysql_ldb`, `ldb`
    - Manages, backups, modify LevelDB files
    - Checks for consistency
    - Dumps WAL files

Engine-Specific Tools

- MyRocks
  - `sst_dump`
    - Checks, scans and dumps SST files
  - `mysql_ldb`, `ldb`
    - Manages, backups, modify LevelDB files
    - Checks for consistency
    - Dumps WAL files
  - MyRocks Troubleshooting

PERCONA

Engine-Specific Tools

- All engines
  - `mysqlcheck`
    - `CHECK`s, `REPAIR`s, `ANALYZE`s, `OPTIMIZE`s tables
    - Engine must support corresponding operation

```
$ mysqlcheck --host=127.0.0.1 --port=13001 --user=root employees employees
employees.employees                                OK
$ mysqlcheck --host=127.0.0.1 --port=13001 --user=root --analyze employees employees
employees.employees                                OK
$ mysqlcheck --host=127.0.0.1 --port=13001 --user=root --repair employees employees
employees.employees
note     : The storage engine for the table doesn't support repair
```

PERCONA

MySQL Utilities

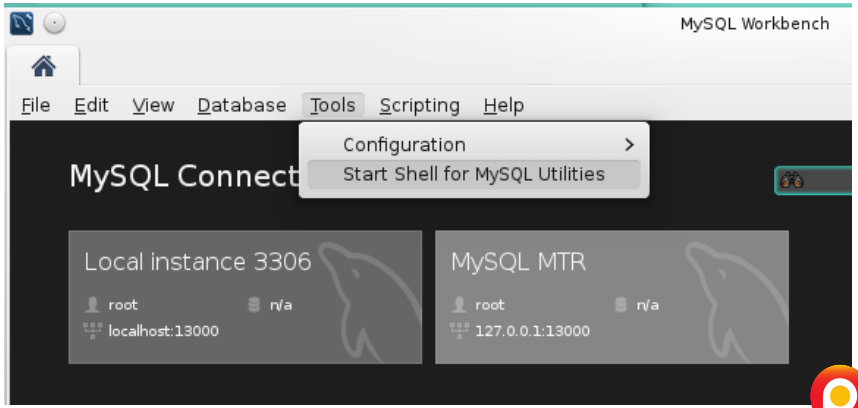
Prerequisites

- Python 2.6
- MySQL Connector Python version 2.0.4/2.1.2 or later
- Available in distros
- MySQL command line utilities (`my_print_defaults` etc.) must be in `$PATH`

PERCONA

How to access MySQL Utilities

- In MySQL Workbench



PERCONA

How to access MySQL Utilities

- In MySQL Workbench
- From command line
  - Login path, set by `mysql_config_editor`
  - Configuration file
  - Command-line - Deprecated!

PERCONA

How to access MySQL Utilities

- In MySQL Workbench
- From command line
  - Login path, set by `mysql_config_editor`

```
sveta@thinkie:~> mysqlserverinfo --server=PerconaMTR --format=vertical
# Source on localhost: ... connected.
************************       1. row ************************
        server: localhost:13001
   config_file: /etc/my.cnf
    binary_log: master-bin.000001
binary_log_pos: 151
     relay_log:
 relay_log_pos:
       version: 5.6.26-73.2-debug-log
       datadir: /home/sveta/build/ps-5.6/mysql-test/var/mysqld.1/data/
       basedir: /home/sveta/build/ps-5.6
    plugin_dir: /home/sveta/build/ps-5.6/lib/mysql/plugin/
...
```

How to access MySQL Utilities

- In MySQL Workbench
- From command line
  - Configuration file

```
sveta@thinkie:~> mysqlgrants --server=$HOME/.my.cnf[PerconaMTR] mysql
# Source on 127.0.0.1: ... connected.

# DATABASE `mysql`:
# - 'root'@'127.0.0.1' : ALL PRIVILEGES, GRANT OPTION
# - 'root'@'::1' : ALL PRIVILEGES, GRANT OPTION
# - 'root'@'localhost' : ALL PRIVILEGES, GRANT OPTION
# - 'root'@'thinkie' : ALL PRIVILEGES, GRANT OPTION
#...done.
```

PERCONA

How to access MySQL Utilities

- In MySQL Workbench
- From command line
  - Command-line - Deprecated!

```
sveta@thinkie:~> mysqlmetagrep --server=root:@127.0.0.1:13001 \
> --pattern='innodb%' --format=vertical
WARNING: Using a password on the command line interface can be insecure.
*************************       1. row *************************
Connection: root:*@127.0.0.1:13001
Object Type: TABLE
Object Name: INNODB_BUFFER_PAGE
Database: information_schema
Field Type: TABLE
Matches: INNODB_BUFFER_PAGE
*************************       2. row *************************
Connection: root:*@127.0.0.1:13001
Object Type: TABLE
Object Name: INNODB_BUFFER_PAGE_LRU
```

Utilities can be used for

- Oracle groups them
  - Binary log maintenance
  - Database operations
  - General operations
  - High Availability operations
  - Server Operations
  - Specialized Operations

Utilities can be used for

- To be short
  - Replication maintenance
  - Data integrity
  - Information about MySQL Server

PERCONA

mysqlslavetrx

- Skips 1-N transactions
- Works with `GTID`
- Easier to work with than using manual method, described at Percona blog

PERCONA

mysqlslavetrx example

```
sveta@thinkie> mysqlslavetrx --gtid-set=fb776095-8474-11e5-ad41-30b5c2208a0f:3 \
               --slaves=root:@127.0.0.1:13001
WARNING: Using a password on the command line interface can be insecure.
#
# GTID set to be skipped for each server:
# - 127.0.0.1@13001: fb776095-8474-11e5-ad41-30b5c2208a0f:3
#
# Injecting empty transactions for '127.0.0.1:13001'...
#
#...done.
#
```

PERCONA

mysqlserverclone

- Spawns a new process, using same binaries which source server uses

mysqlserverclone

- Spawns a new process, using same binaries which source server uses
- Does not copy configuration

PERCONA

mysqlserverclone

- Spawns a new process, using same binaries which source server uses
- Does not copy configuration
- Does not copy data

PERCONA

mysqlserverclone

- Spawns a new process, using same binaries which source server uses
- Does not copy configuration
- Does not copy data
- Practically starts same server binary with different datadir with default options
  - Defaults can be overwritten on command line

mysqlserverclone

- Spawns a new process, using same binaries which source server uses
- Does not copy configuration
- Does not copy data
- Practically starts same server binary with different datadir with default options
  - Defaults can be overwritten on command line
- Can be used in conjunction with `mysqldbcopy` to create test instance

mysqlserverclone example

```
sveta@thinkie> mysqlserverclone --server=root:@127.0.0.1:13000 \
> --new-data=~/tmp/mysqlserverclone
WARNING: Using a password on the command line interface can be insecure.
# WARNING: Root password for new instance has not been set.
# Cloning the MySQL server running on 127.0.0.1.
# Configuring new instance...
# Locating mysql tools...
# Setting up empty database and mysql tables...
# Starting new instance of the server...
# Testing connection to new instance...
# Success!
# Connection Information:
#  -uroot --socket=/home/sveta/tmp/mysqlserverclone/mysql.sock
#...done.
```

PERCONA

How useful mysqlserverclone?

```
sveta@thinkie> ls -lh var/mysqld.1/data/ib_log*
-rw-r--r-- 1 sveta users 5.0M Oct 23 15:51 var/mysqld.1/data/ib_logfile0
-rw-r--r-- 1 sveta users 5.0M Oct 22 22:58 var/mysqld.1/data/ib_logfile1

sveta@thinkie> ls -lh /home/sveta/tmp/mysqlserverclone/ib_log*
-rw-r----- 1 sveta users 48M Oct 23 16:12 ~/tmp/mysqlserverclone/ib_logfile0
-rw-r----- 1 sveta users 48M Oct 23 16:11 ~/tmp/mysqlserverclone/ib_logfile1
```

PERCONA

mysqluserclone

- Clones existing user accounts

mysqluserclone

- Clones existing user accounts
- Can help to provide access to groups of users without adding authentication plugins

```
sveta@thinkie>  mysqluserclone --source=root:@127.0.0.1:13000 \
--destination=root:@127.0.0.1:13000 bar@% baz@%
WARNING: Using a password on the command line interface can be insecure.
# Source on 127.0.0.1: ... connected.
# Destination on 127.0.0.1: ... connected.
# Cloning 1 users...
# Cloning bar@% to user baz@%
# ...done.
```

PERCONA

mysqluserclone

- Clones existing user accounts
- Can help to provide access to groups of users without adding authentication plugins
- Works with global and per-object grants

PERCONA

mysqlfrm

- Reads `frm` files without data

PERCONA

mysqlfrm

- Reads `frm` files without data
- Two modes: with and without spawning server

PERCONA

mysqlfrm

- Reads `frm` files without data
- Two modes: with and without spawning server
- Offline mode to read corrupted `.frm` files

PERCONA

mysqlfrm

- Reads `frm` files without data
- Two modes: with and without spawning server
- Offline mode to read corrupted `.frm` files
- Cannot read all possible corrupted files

PERCONA

mysqlfrm

- Reads `frm` files without data
- Two modes: with and without spawning server
- Offline mode to read corrupted `.frm` files
- Cannot read all possible corrupted files
- Can be only tool to re-create table definition

PERCONA

mysqlfrm

- Reads `frm` files without data
- Two modes: with and without spawning server
- Offline mode to read corrupted `.frm` files
- Cannot read all possible corrupted files
- Can be only tool to re-create table definition
- My favorite utility! ☺

PERCONA

mysqlfrm example

```
sveta@thinkie> mysqlfrm --basedir=.. --port=3333 c1.frm
# Starting the spawned server on port 3333 ... done.
# Reading .frm files
#
# Reading the c1.frm file.
#
# CREATE statement for c1.frm:
#

CREATE TABLE ‘c1‘ (
‘f1‘ varchar(255) CHARACTER SET cp1251 DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1

#...done.
```

PERCONA

mysqlfrm –diagnostic

```
sveta@thinkie> truncate --size=-1 c1.frm
sveta@thinkie> mysqlfrm --basedir=.. --port=3333 c1.frm
# Starting the spawned server on port 3333 ... done.
# Reading .frm files
#
# Reading the c1.frm file.
ERROR: Failed to correctly read the .frm file. Please try reading the file with the --diagnostic mode.
...
sveta@thinkie> mysqlfrm --diagnostic c1.frm
# WARNING: Cannot generate character set or collation names without the --server option.
# CAUTION: The diagnostic mode is a best-effort parse of the .frm file. As such, it may not identify al
# Reading .frm file for c1.frm:
# The .frm file is a TABLE.
# CREATE TABLE Statement:

CREATE TABLE `c1` (
`f1` varchar(255) CHARACTER SET <UNKNOWN> DEFAULT NULL
) ENGINE=InnoDB;

#...done.
```


PERCONA

mysqlgrants

- Shows grants for objects
- Answers on questions like
  - Who has `UPDATE` privilege on X?
  - Who has `SELECT` privilege on Y?
  - Who has "`GRANT OPTION`"?

PERCONA

mysqlgrants example

```
sveta@thinkie> mysqlgrants --server=root:@127.0.0.1:13000 test
# Source on 127.0.0.1: ... connected.
# DATABASE ‘test‘:
# - ’’@’%’ : ALTER, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE VIEW, DELETE, DROP, EVENT,
# - ’root’@’127.0.0.1’ : ALL PRIVILEGES, GRANT OPTION
# - ’root’@’::1’ : ALL PRIVILEGES, GRANT OPTION
# - ’root’@’localhost’ : ALL PRIVILEGES, GRANT OPTION
# - ’root’@’thinkie’ : ALL PRIVILEGES, GRANT OPTION
#...done.

sveta@thinkie> mysqlgrants --server=root:@127.0.0.1:13000 test.t1
# Source on 127.0.0.1: ... connected.
# TABLE ‘test‘.‘t1‘:
# - ’’@’%’ : ALL PRIVILEGES
# - ’geek’@’%’ : UPDATE
# - ’root’@’127.0.0.1’ : ALL PRIVILEGES, GRANT OPTION
# - ’root’@’::1’ : ALL PRIVILEGES, GRANT OPTION
# - ’root’@’localhost’ : ALL PRIVILEGES, GRANT OPTION
# - ’root’@’thinkie’ : ALL PRIVILEGES, GRANT OPTION
#...done.
```


PERCONA

Compare: pt-show-grants

```
sveta@thinkie> pt-show-grants h=127.0.0.1,P=13000,u=root,D=test
-- Grants dumped by pt-show-grants
-- Dumped from server 127.0.0.1 via TCP/IP, MySQL 5.7.9-debug-log at 2015-11-05 13:49:14
-- Grants for 'geek'@'%'
GRANT UPDATE ON `test`.`t1` TO 'geek'@'%';
GRANT USAGE ON *.* TO 'geek'@'%';
-- Grants for 'mysql.sys'@'localhost'
GRANT SELECT ON `sys`.`sys_config` TO 'mysql.sys'@'localhost';
GRANT TRIGGER ON `sys`.* TO 'mysql.sys'@'localhost';
GRANT USAGE ON *.* TO 'mysql.sys'@'localhost';
-- Grants for 'root'@'127.0.0.1'
GRANT ALL PRIVILEGES ON *.* TO 'root'@'127.0.0.1' WITH GRANT OPTION;
-- Grants for 'root'@'::1'
GRANT ALL PRIVILEGES ON *.* TO 'root'@'::1' WITH GRANT OPTION;
-- Grants for 'root'@'localhost'
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION;
GRANT PROXY ON ''@'' TO 'root'@'localhost' WITH GRANT OPTION;
-- Grants for 'root'@'thinkie'
GRANT ALL PRIVILEGES ON *.* TO 'root'@'thinkie' WITH GRANT OPTION;
GRANT PROXY ON ''@'' TO 'root'@'thinkie' WITH GRANT OPTION;
```


PERCONA

Compare: pt-show-grants

```
sveta@thinkie> pt-show-grants h=127.0.0.1,P=13000,u=root,D=test --only=geek
-- Grants dumped by pt-show-grants
-- Dumped from server 127.0.0.1 via TCP/IP, MySQL 5.7.9-debug-log at 2015-11-05 13:49:46
-- Grants for 'geek'@'%'
GRANT UPDATE ON `test`.`t1` TO 'geek'@'%';
GRANT USAGE ON *.* TO 'geek'@'%';
```

- `pt-show-grants` separates by user
- `mysqlgrants` separates by object
- Complements!

PERCONA

mysqlmetagrep

- Searches names which satisfy pattern
- Only particular kind: table, routine, etc.
- Examines routines body
- Supports `LIKE` syntax and regular expressions

PERCONA

mysqlmetagrep example

```
sveta@thinkie> mysqlmetagrep --server=root:@127.0.0.1:13000 -e '%memory%thread%' \
               --database=sys --search-objects=view -f vertical
WARNING: Using a password on the command line interface can be insecure.
*************************       1. row *************************
 Connection: root:*@127.0.0.1:13000
Object Type: VIEW
Object Name: memory_by_thread_by_current_bytes
   Database: sys
 Field Type: VIEW
    Matches: memory_by_thread_by_current_bytes
*************************       2. row *************************
 Connection: root:*@127.0.0.1:13000
Object Type: VIEW
Object Name: x$memory_by_thread_by_current_bytes
   Database: sys
 Field Type: VIEW
    Matches: x$memory_by_thread_by_current_bytes
2 rows.
```

PERCONA

mysqlindexcheck

- Shows duplicate indexes
- Reports if table has neither primary key, nor unique index
- Displays index statistics (if exist)

PERCONA

mysqlindexcheck example

```
sveta@thinkie> mysqlindexcheck --server=root:@127.0.0.1:13000 -r test.t1
WARNING: Using a password on the command line interface can be insecure.
# Source on 127.0.0.1: ... connected.
# The following indexes are duplicates or redundant for table test.t1:
#
CREATE UNIQUE INDEX `a` ON `test`.`t1` (`a`) USING BTREE
#     may be redundant or duplicate of:
ALTER TABLE `test`.`t1` ADD PRIMARY KEY (`a`)
#
CREATE INDEX `a_2` ON `test`.`t1` (`a`, `b`) USING BTREE
#     may be redundant or duplicate of:
ALTER TABLE `test`.`t1` ADD PRIMARY KEY (`a`)
# The following indexes for table test.t1 contain the clustered index and
  might be redundant:
#
CREATE UNIQUE INDEX `a` ON `test`.`t1` (`a`) USING BTREE
#
CREATE INDEX `a_2` ON `test`.`t1` (`a`, `b`) USING BTREE
```

PERCONA

Compare: pt-duplicate-key-checker

```
sveta@thinkie> pt-duplicate-key-checker h=127.0.0.1,P=13000,u=root,D=test -t t1
# ########################################################################
# test.t1
# ########################################################################
# Uniqueness of a ignored because PRIMARY is a duplicate constraint
# a is a duplicate of PRIMARY
# Key definitions:
#   UNIQUE KEY `a` (`a`),
#   PRIMARY KEY (`a`),
# Column types:
#         `a` int(11) not null
# To remove this duplicate index, execute:
ALTER TABLE `test`.`t1` DROP INDEX `a`;
# ########################################################################
# Summary of indexes
# ########################################################################
# Size Duplicate Indexes   4
# Total Duplicate Indexes  1
# Total Indexes            4
```

PERCONA

mysqldiskusage

- Displays disk usage
- First tries to access `datadir`, if fails prints info from system tables
- No magic like how big is InnoDB table in shared tablespace
- Users can specify which information to show

mysqldiskusage example

```
sveta@thinkie:> mysqldiskusage --server=root:@127.0.0.1:13000 --all
WARNING: Using a password on the command line interface can be insecure.
# Source on 127.0.0.1: ... connected.
# Database totals:
+--------------------+------------+
| db_name            |      total |
+--------------------+------------+
| mtr                | 25,991     |
| mysql              | 2,898,515  |
| performance_schema | 822,113    |
| sys                | 496,572    |
| test               | 557,924    |
+--------------------+------------+

Total database disk usage = 4,801,115 bytes or 4.58 MB
```

PERCONA

mysqldiskusage example cont.

```
# Log information.
# log_error information is not accessible. Check your permissions.
+------------------+----------+
| log_name         |    size  |
+------------------+----------+
| mysqld.log       | 185,581  |
| mysqld-slow.log  |     237  |
+------------------+----------+
Total size of logs = 185,818 bytes or 181.46 KB

# Binary log information:
Current binary log file = master-bin.000001
+--------------------+-------+
| log_file           | size  |
+--------------------+-------+
| master-bin.000001  | 2031  |
| master-bin.index   | 20    |
+--------------------+-------+
```



```
Total size of binary logs = 2,051 bytes or 2.00 KB
```

mysqldiskusage example cont.

```
# Server is not an active slave - no relay log information.
# InnoDB tablespace information:
+--------------+-------------+
| innodb_file  |       size  |
+--------------+-------------+
| ib_logfile0  |  5,242,880  |
| ib_logfile1  |  5,242,880  |
| ibdata1      | 12,582,912  |
+--------------+-------------+

Total size of InnoDB files = 23,068,672 bytes or 22.00 MB

#...done.
```

mysqlprocgrep

- Searches `processlist` for a pattern

PERCONA

mysqlprocgrep

- Searches `processlist` for a pattern
- Supports `LIKE` and regular expressions

PERCONA

mysqlprocgrep

- Searches `processlist` for a pattern
- Supports `LIKE` and regular expressions
- Can filter by
  - `ID`, `user`, `host`, `db`
  - `command`
  - `info`
  - `state`

PERCONA

mysqlprocgrep

- Searches `processlist` for a pattern
- Supports `LIKE` and regular expressions
- Can filter by
- Uses stored routine `kill_processes` to kill

PERCONA

mysqlprocgrep

- Searches `processlist` for a pattern
- Supports `LIKE` and regular expressions
- Can filter by
- Uses stored routine `kill_processes` to kill
- Easier to use, but less powerful than `pt-kill`

PERCONA

mysqlprocgrep example

```
sveta@thinkie> mysqlprocgrep --server=root:@127.0.0.1:13000 --match-db=sbtest \
               --match-state=updating -f vertical
WARNING: Using a password on the command line interface can be insecure.
*************************       1. row *************************
Connection: root:*@127.0.0.1:13000
        Id: 284
      User: root
      Host: localhost:54260
        Db: sbtest
   Command: Query
      Time: 0
     State: updating
      Info: UPDATE sbtest5 SET k=k+1 WHERE id=496909
*************************       2. row *************************
Connection: root:*@127.0.0.1:13000
...
     State: updating
      Info: DELETE FROM sbtest5 WHERE id=503665
...
```

mysqluc

- Console to run utilities
- You can set local variables
- Does not remember options

PERCONA

mysqluc example

```
sveta@thinkie> mysqluc
Launching console ...
Welcome to the MySQL Utilities Client (mysqluc) version 1.6.2
Copyright (c) 2010, 2015 Oracle and/or its affiliates. All rights reserved
...
Type 'help' for a list of commands or press TAB twice for list of utilities.
mysqluc> set server=root:@127.0.0.1:13000
mysqluc> mysqlprocgrep --server=$server -f vertical
WARNING: Using a password on the command line interface can be insecure.
*************************       1. row *************************
Connection: root:*@127.0.0.1:13000
        Id: 299
      User: root
      Host: localhost:54460
        Db: None
   Command: Query
      Time: 0
     State: executing
      Info: SELECT Id, User, Host, Db, Command, Time, State, Info
...
```


PERCONA

Extending Utilities

- They all have basic functionality
  - `mysqlserverclone` is useless by itself
  - But can be used by another routine which needs to spawn MySQL Server

Extending Utilities

- They all have basic functionality
- Can be extended

Extending Utilities

- They all have basic functionality
- Can be extended
- Python library

PERCONA

Extending Utilities

- They all have basic functionality
- Can be extended
- Python library
- Have parsers, specialized operations

PERCONA

Extending Utilities

- They all have basic functionality
- Can be extended
- Python library
- Have parsers, specialized operations
- Oracle accepts pull requests at GitHub

PERCONA

Percona Toolkit

PERCONA

First Data Source in Many Support Tickets

- `pt-summary`: hardware and OS
  - System overall report
  - Details
    - Processor
    - Memory
    - Disks
    - Network
  - Processlist
    - Notable processes

PERCONA

First Data Source in Many Support Tickets

- `pt-summary`: hardware and OS
- `pt-mysql-summary`: MySQL server
  - Overall information
  - Processlist summary
  - Groupped configuration and statistics
    - Misses new additions: innodb_io_capacity, Performance Schema, ...
  - Incompatible with MariaDB 10.2

PERCONA

First Data Source in Many Support Tickets

- `pt-summary`: hardware and OS
- `pt-mysql-summary`: MySQL server
- `pt-stalk`: everything runtime
  - Operating system
  - Hardware
  - Processlist
  - InnoDB
  - Statistics
  - More

PERCONA

Find Duplicate Keys

- pt-duplicate-key-checker

```
$ pt-duplicate-key-checker h=127.0.0.1,P=16000,u=root -d test -t employees
# ########################################################################
# test.employees
# ########################################################################

# Key birth_date_2 ends with a prefix of the clustered index
# Key definitions:
#   KEY `birth_date_2` (`birth_date`,`emp_no`)
#   PRIMARY KEY (`emp_no`),
# Column types:
#   `birth_date` date not null
#   `emp_no` int(11) not null
# To shorten this duplicate clustered index, execute:
ALTER TABLE `test`.`employees` DROP INDEX `birth_date_2`,
ADD INDEX `birth_date_2` (`birth_date`);
```

Find Duplicate Keys

- pt-duplicate-key-checker

```
# ############################################################################
# Summary of indexes
# ############################################################################

# Size Duplicate Indexes   2099230
# Total Duplicate Indexes  1
# Total Indexes            3
```

PERCONA

Find Duplicate Keys

- `pt-duplicate-key-checker`
- Known issues
  - Does not work with `UNIQUE` indexes well
    - PT-677, PT-1414

```
$ pt-duplicate-key-checker h=127.0.0.1,P=16000,u=root -d test -t employees
...
# Key birth_date ends with a prefix of the clustered index
# Key definitions:
#   UNIQUE KEY `birth_date` (`birth_date`,`emp_no`),
#   PRIMARY KEY (`emp_no`),
# Column types:
#   `birth_date` date not null
#   `emp_no` int(11) not null
# To shorten this duplicate clustered index, execute:
ALTER TABLE `test`.`employees` DROP INDEX `birth_date`,
ADD INDEX `birth_date` (`birth_date`);
...
```

PERCONA

Analyze Slow Log

- `pt-query-digest`
- Supports:
  - Progress report

```
$ pt-query-digest /mariadb-10.2.5-linux-x86_64/mysqld.1/mysqld-slow.log
/mariadb-10.2.5-linux-x86_64/mysqld.1/mysqld-slow.log:  10% 04:06 remain
/mariadb-10.2.5-linux-x86_64/mysqld.1/mysqld-slow.log:  23% 03:19 remain
/mariadb-10.2.5-linux-x86_64/mysqld.1/mysqld-slow.log:  33% 02:56 remain
/mariadb-10.2.5-linux-x86_64/mysqld.1/mysqld-slow.log:  44% 02:28 remain
/mariadb-10.2.5-linux-x86_64/mysqld.1/mysqld-slow.log:  54% 02:06 remain
/mariadb-10.2.5-linux-x86_64/mysqld.1/mysqld-slow.log:  64% 01:37 remain
/mariadb-10.2.5-linux-x86_64/mysqld.1/mysqld-slow.log:  74% 01:12 remain
/mariadb-10.2.5-linux-x86_64/mysqld.1/mysqld-slow.log:  84% 00:43 remain
/mariadb-10.2.5-linux-x86_64/mysqld.1/mysqld-slow.log:  94% 00:14 remain
```

Analyze Slow Log

- pt-query-digest
- Supports:
  - Summary

```
# 281.8s user time, 210ms system time, 37.46M rss, 105.50M vsz
# Current date: Fri Apr  7 17:52:42 2017
# Hostname: Thinkie
# Files: /mariadb-10.2.5-linux-x86_64/mysqld.1/mysqld-slow.log
# Overall: 2.30M total, 82 unique, 1.48k QPS, 1.27x concurrency __________
# Time range: 2017-04-07 17:17:24 to 17:43:15
# Attribute          total     min     max     avg     95%  stddev  median
# ============     ======= ======= ======= ======= ======= ======= =======
# Exec time          1971s     13us      2s    855us     98us     13ms     23us
# Lock time            31s       0   721ms     13us     36us      2ms
# Rows sent          3.40M       0     100    1.55    0.99    11.69
# Rows examine       7.64M       0     300    3.48    0.99    26.43
# Rows affecte      47.36k       0    1000    0.02       0     1.28
# Query size        44.77M       5 187.54k   20.38   51.63  249.22   11.95
```

Analyze Slow Log

- `pt-query-digest`
- Supports:
  - Top queries profile

```
# Profile
# Rank Query ID           Response time   Calls   R/Call V/M   Item
# ==== ================== =============== ======= ====== ===== ===========
#    1 0x813031B8BBC3B329 1754.2446 89.0%   11077 0.1584  0.05 COMMIT
#    2 0xD30AD7E3079ABCE7   58.8457  3.0%   11238 0.0052  0.33 UPDATE sbtest?
#    3 0x6C099B0B73EA7633   52.9969  2.7% 2010599 0.0000  0.00
#    4 0xE96B374065B13356   23.8898  1.2%   11191 0.0021  0.38 UPDATE sbtest?
# MISC 0xMISC               81.3937  4.1%  259100 0.0003   0.0 <78 ITEMS>
```

PERCONA

Analyze Slow Log

- pt-query-digest
- Supports:
  - Individual query reports

```
# Query 2: 44.77 QPS, 0.23x concurrency, ID 0xD30AD7E3079ABCE7 at byte 40361702
# This item is included in the report because it matches --limit.
# Scores: V/M = 0.33
# Time range: 2017-04-07 17:18:17 to 17:22:28
# Attribute    pct   total     min     max     avg     95%  stddev  median
# ============ === ======= ======= ======= ======= ======= ======= =======
# Count          0   11238
# Exec time      2     59s    68us      2s     5ms   287us    42ms   108us
# Lock time     21      7s    25us   721ms   585us    73us    13ms    38us
...
# Converted for EXPLAIN
# EXPLAIN /*!50100 PARTITIONS*/
select  k=k+1 from sbtest2 where  id=500\G
```


PERCONA

Analyze Slow Log

- `pt-query-digest`
- Supports:
  - More
    - Filters
    - History tables
    - Samples
    - Different sources: slow, general, binary log files; `SHOW PROCESSLIST`, `tcpdump`

PERCONA

Kill Connections Stealing Resources

- pt-kill

```
pt-kill h=127.0.0.1,P=16000,u=root --match-command Sleep --kill
```


PERCONA

Kill Connections Stealing Resources

- `pt-kill`

```
pt-kill h=127.0.0.1,P=16000,u=root --match-command Sleep --kill
```

- What to kill
  - `match-[all|command|db|host|info|state|user]`
  - `ignore-*`
  - `busy-time`
    - Query, which runs longer than specified

PERCONA

Kill Connections Stealing Resources

- pt-kill

```
pt-kill h=127.0.0.1,P=16000,u=root --match-command Sleep --kill
```

- What to kill
- When to kill
  - interval
    - 30 seconds
    - busy-time/2

PERCONA

Kill Connections Stealing Resources

- `pt-kill`

```
pt-kill h=127.0.0.1,P=16000,u=root --match-command Sleep --kill
```

- What to kill
- When to kill
- How long to run
  - `run-time`
  - Default: forever
    - sentinel file supported

PERCONA

Manage and Configure Replication

- pt-heartbeat

```
$ pt-heartbeat -D test --update -h 127.0.0.1 -P3371 -u root --daemonize
$ pt-heartbeat -D test --monitor -h 127.0.0.1 -P3372 -u root
0.00s [  0.02s,  0.00s,  0.00s ]
0.00s [  0.02s,  0.00s,  0.00s ]
0.00s [  0.02s,  0.00s,  0.00s ]
 1.00s [  0.03s,  0.01s,  0.00s ]
2.00s [  0.07s,  0.01s,  0.00s ]
3.00s [  0.12s,  0.02s,  0.01s ]
...
19.00s [  3.18s,  0.64s,  0.21s ]
11.00s [  3.37s,  0.67s,  0.22s ]
10.00s [  3.53s,  0.71s,  0.24s ]
6.00s [  3.63s,  0.73s,  0.24s ]
0.00s [  3.63s,  0.73s,  0.24s ]
0.00s [  3.63s,  0.73s,  0.24s ]
...
```


PERCONA

Manage and Configure Replication

- pt-heartbeat
- pt-slave-find
  - Short format

```
$ pt-slave-find --report-format hostname h=127.0.0.1,P=3371,u=root
127.0.0.1:3371
+- 127.0.0.1:3374
   +- 127.0.0.1:3372
+- 127.0.0.1:3373
```

PERCONA

Manage and Configure Replication

- pt-heartbeat
- pt-slave-find
  - Default format

```
$ pt-slave-find h=127.0.0.1,P=3371,u=root
127.0.0.1:3371
Version         5.7.17-11-27.20-log
Server ID       1
Uptime          17:32:12 (started 2017-04-07T00:22:47)
Replication     Is a slave, has 2 slaves connected, is not read_only
Filters
Binary logging  ROW
Slave status    0 seconds behind, running, no errors
Slave mode      STRICT
...
```

PERCONA

Manage and Configure Replication

- pt-heartbeat
- pt-slave-find
- pt-table-checksum

```
$ pt-table-checksum h=127.0.0.1,P=16000,u=root
            TS ERRORS  DIFFS     ROWS  CHUNKS SKIPPED    TIME TABLE
04-08T02:07:21      0      0       99       1       0   0.776 mtr.global_suppressions
04-08T02:07:22      0      0        0       1       0   0.707 mtr.test_suppressions
04-08T02:07:23      0      0        0       1       0   0.803 mysql.column_stats
04-08T02:07:25      0      0        0       1       0   1.583 mysql.columns_priv
...
04-08T02:07:47      0      0        4       1       0   1.334 mysql.user
04-08T02:07:49      0      3     1000       5       0   2.555 sbtest.sbtest1
```

Manage and Configure Replication

- pt-heartbeat
- pt-slave-find
- pt-table-checksum
- pt-table-sync

```
$ pt-table-sync h=127.0.0.1,P=16001,u=root --sync-to-master --execute --verbose
# Syncing P=16001,h=127.0.0.1,u=root
# DELETE REPLACE INSERT UPDATE ALGORITHM START    END      EXIT DATABASE.TABLE
#      0       0      0      0 Nibble    02:18:26 02:18:26 0    mysql.column_stats
...
#      0     101      0      0 Chunk     02:18:26 02:18:27 2    sbtest.sbtest1
```

PERCONA

Summary

- With Percona Toolkit you easily can
  - Troubleshoot
  - Improve performance
  - Manage data
  - Manage replication
  - Perform custom tasks automatically
  - More

PERCONA

PMM

PERCONA

Percona Monitoring and Management (PMM)

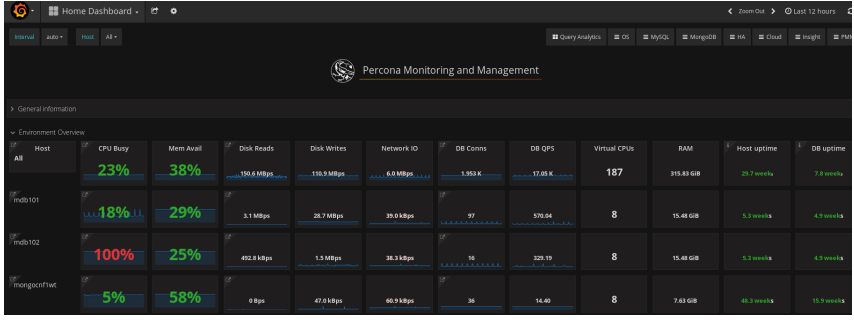
Percona Monitoring and Management (PMM) is an open-source platform for managing and monitoring MySQL and MongoDB performance. It is developed by Percona in collaboration with experts in the field of managed database services, support and consulting.

PMM is a free and open-source solution that you can run in your own environment for maximum security and reliability. It provides thorough time-based analysis for MySQL and MongoDB servers to ensure that your data works as efficiently as possible. ©

PERCONA

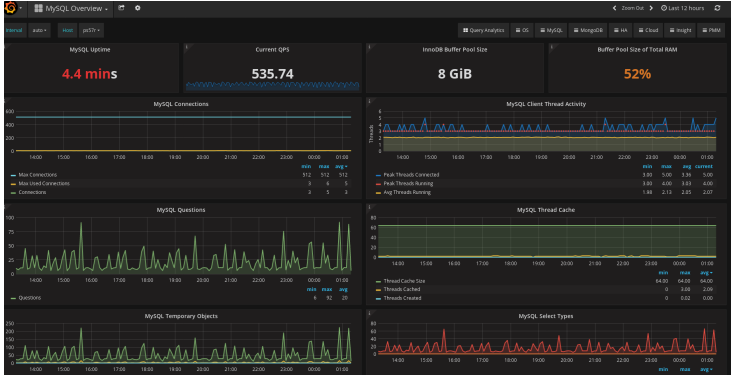
Inside PMM

- Performance overview
  - Overall



Inside PMM

- Performance overview
  - Database



Inside PMM

- Performance overview
  - System

PERCONA

Inside PMM

- Performance overview
  - Cloud

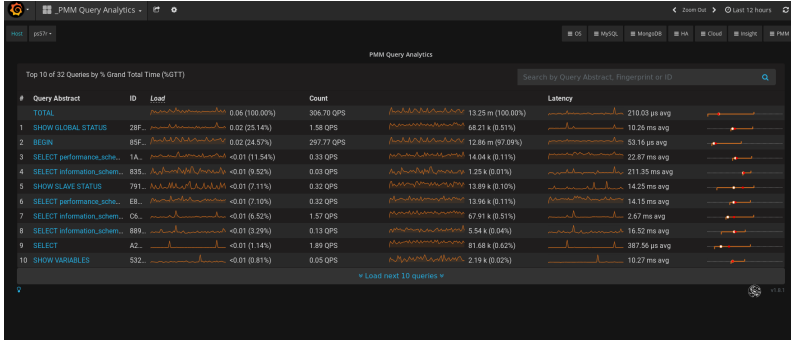
PERCONA

Inside PMM

- Performance overview
  - Insights and Trends

Inside PMM

- Performance overview
- Query Analytics



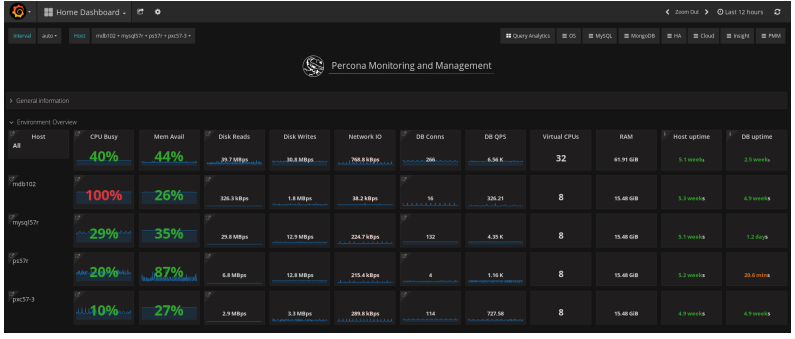
Tips for Working with PMM

- Study what metrics mean

PERCONA

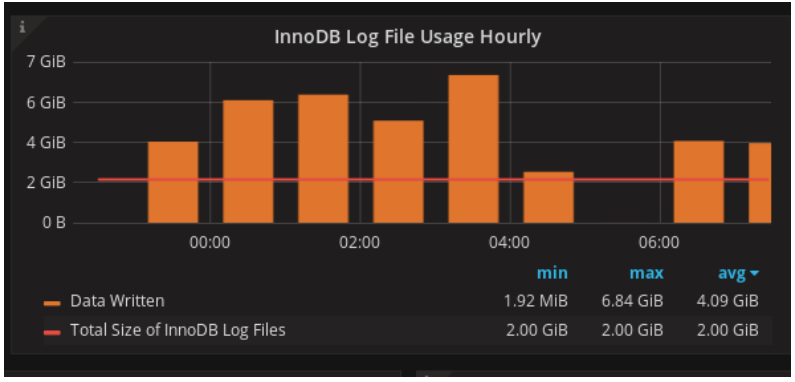
Tips for Working with PMM

- Look into alerts



Tips for Working with PMM

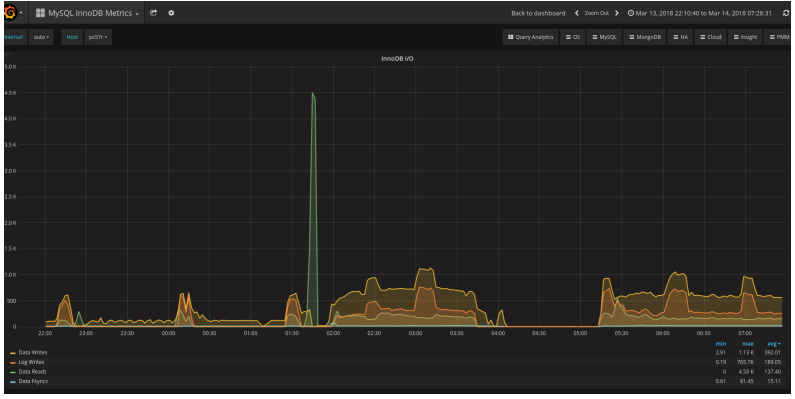
- Pay attention for everything above limits



PERCONA

Tips for Working with PMM

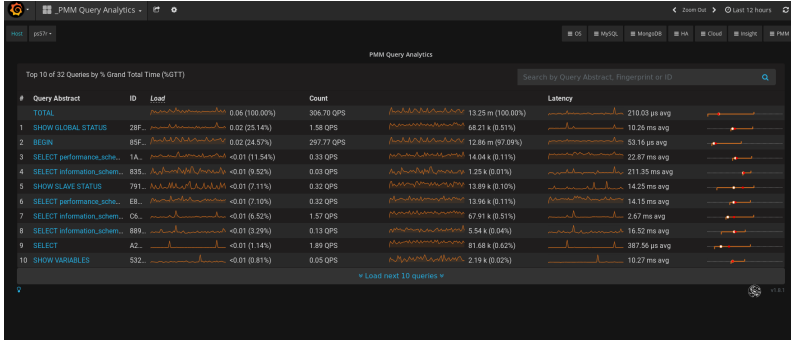
- Watch peaks



PERCONA

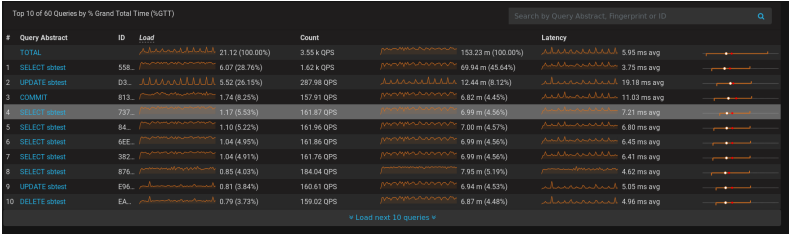
Tips for Working with Query Analytics

- Start from top queries



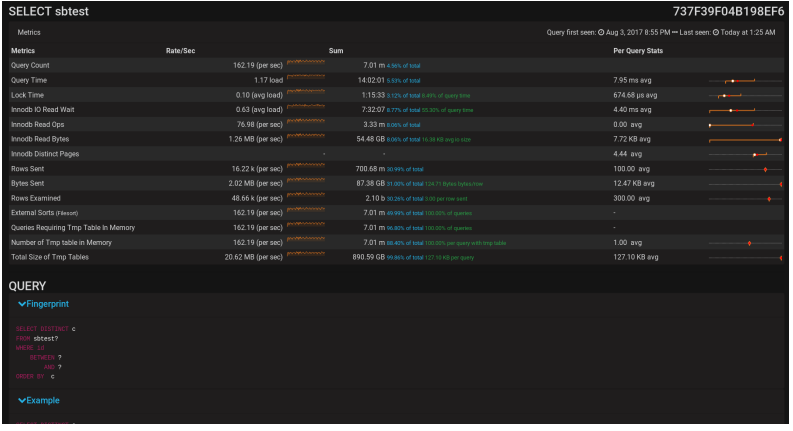
Tips for Working with Query Analytics

- Choose user's query



Tips for Working with Query Analytics

- Investigate



Tips for Working with Query Analytics

- Tune
  - Troubleshooting Slow Queries webinar

PERCONA

More Information

MySQL User reference Manual

MySQL Sandbox

dbdeployer

Percona Server in Docker

MySQL Utilities 1.6 User Manual

Official Documentation for Percona Toolkit

Percona Toolkit at Percona Blog

pmmdemo.percona.com

PERCONA

Thank you!

http://www.slideshare.net/SvetaSmirnova

https://twitter.com/svetsmirnova

PERCONA