Operational DBA in a nutshell

Tom De Cooman and Dimitri Vanoverbeke
SDM Team Manager
Solution engineer
Welcome!

- Welcome!
- Who are we?
  - Fans of funny gifs...
- What will we talk about?
- When are the breaks?
Basic housekeeping

• Full Day Tutorial
• Hands On! Laptop Required!
• Breaks
• Wireless SSID/PWD = Percona 2019 / PL19live
• How to ask for help
• Using the documentation
• Text in red you can execute on the commandline or MySQL prompt
  – # for commandline
  – > for mysql
Beware!

This tutorial is not for you if ... ...

• You know what `replicate_same_server_id` is!
• You know all `GRANTs` by heart!
• `pt-query-digest` is your favourite tool!
• You’ve been dealing with replication inconsistencies for a while already!
• Your backup environment is the most awesome there is!
Beware! … Part 2

- This tutorial may go very fast at some times ...
- There is a lot of material to cover
- Don’t worry!
- The main idea is to introduce you to approaches, concepts, tech ...
- Don’t hesitate to get our help!
Operational DBA In A Nutshell

- Setting up today’s environment
- MySQL Installation
- Logging In
- MySQL Privileges
- Diagnostics
- Troubleshooting
- Backups
- Replication
- Schema Changes
- Configuration Optimization
Operational DBA In A Nutshell

• Setting Up Today’s Environment
• MySQL Installation
• Logging In
• MySQL Privileges
• Diagnostics
• Troubleshooting
• Backups

• Replication
• Schema Changes
• Configuration Optimization
Intermezzo: What versions of MySQL do you run?

- Do you know?
- MySQL 5.7/8
- Percona Server 5.7/8?
- MariaDB 10.2/10.3/10.4?
- Another blend?
What is MySQL Community edition?

- Standard Oracle Community Version (previously MySQL AB, Sun)
- Contains a standard MySQL version with InnoDB and MyISAM as a standard
- The original source
- Packages for a large array of operating systems including Windows and MacOS
- Open Source GPL v2
What is Percona Server?

- Enhanced version of MySQL Community edition
- Added plugins for
  - Authentication (PAM)
  - Auditing
  - Diagnostics
  - Improved and Extra storage engines
- Improved Performance (Patches)
- Improved backup and management capabilities
- Hotbackup using Percona XtraBackup
- Monitoring using PMM or other tools
- Packages for Linux distributions
What is MariaDB server

• Fork of MySQL community Edition
  – New features of MySQL not included
• Improved Functionalities
  – Connect storage engine
  – Additional storage engines
  – Additional plugins
• Packages for Linux and Windows
Other existing or older options

- WebScaleSQL
  - database developed by large web organisations based on MySQL community 5.6
  - No ready available packages
  - No longer maintained
- Drizzle
- MySQL Cluster
  - NDB storage engine (specific use case)
- MySQL 5.7 Enterprise edition
  - MySQL community edition with additional plugins
    - Authentication
    - Firewall
    - MySQL enterprise backup and monitor
Hold my beer: Percona Toolkit

- Commandline Tools
- System tasks
- Dataset management and tooling
- Percona Server for MySQL, MySQL, MariaDB, Percona Server for MongoDB and MongoDB
- Open Source
- Packages for RHEL and Debian based systems
Setting up Today’s environment

• Copy Files From USB Stick (if you have not downloaded them yet)
• Install VirtualBox, open (doubleclick) the vbox files, start all VMs
• Test Connectivity
  – ssh -p 2221 root@localhost (password: vagrant)
  – ssh -p 2222 root@localhost (password: vagrant)
  – http://localhost:8080/ (you should see an apache test page)
Setting up Today’s environment (Amazon)

• Amazon based infrastructure (IAAS)
  – Automation using Hashicorp’s Terraform
  – Combination of bash scripts, terraform scripts, etc
    • Based on CentOS 7.6
    • Installs multiple repos
    • Has a separate VPC and security group
  – You are RC testers!
Setting up Today’s environment

• Connection details
  – IP
  – User
  – PWD

• Connecting to via a terminal/putty
  – # ssh centos@<IP>
  – Become root: # sudo -i
  – [root@perconalive-1-master ~]#
Setting up Today’s environment (Amazon)

- Download the private key: https://tinyurl.com/yyglnrmm
- ssh into the new machines:
  - take the perconalive.pem key
- Test Connectivity
  - ssh -i perconalive.pem centos@<perconalive master host>
  - ssh -i perconalive.pem centos@<perconalive slave host>
  - http://<perconalive master host IP>/ (you should see an apache test page)
- Become root: # sudo -i  → Not a best practice!
Operational DBA In A Nutshell

- Setting Up Today’s Environment
- **MySQL Installation**
- Logging In
- MySQL Privileges
- Diagnostics
- Troubleshooting
- Backups

- Replication
- Schema Changes
- Configuration Optimization
Installation of MySQL

• Install Percona Server
  # yum install Percona-Server-server-57
  # service mysql start <ONLY ON MASTER>
• Verify if you can connect
  # mysql
• Verify the application:
  – http://<MASTER_IP>/my-movies
Today's application

MyMovies

There are 50005 users, 1543720 movies and 2812743 actors in the system.

Random Movie
Entre Calais et Douvres (1897)

Latest comments

Featured User
Larry.B.Fargo@dodgit.com

Online right now

Being viewed right now
Paul Calderon
Pulp Fiction (1994)
Tight Security (2001)
Assrar fi rammel (1954)
John Travolta
Operational DBA In A Nutshell

- Setting Up Today’s Environment
- Installation
- Logging In
- MySQL Privileges
- Diagnostics
- Troubleshooting
- Backups

- Replication
- Schema Changes
- Configuration Optimization
Intermezzo! What are DDL queries?

- Data Definition Language
- Example:
  CREATE TABLE Accounts
  (Account_number Bigint(16),
  Account_name varchar,
  Amount Bigint(16),
  PRIMARY KEY (Account_number)
);

- Usage
DDL queries define the structure on which you develop your application. Your structure will also define how the database server searches for information in a table.
What are DML statements?

• Data Manipulation Language
• Example:
  INSERT into resto_visitor values(5,'Julian','highway 5',12);
  INSERT INTO tbl_name (col1,col2) VALUES(15,col1*2);
  UPDATE resto_visitor set name='Evelyn',age=17 where id=103;

• Usage
  It’s how you can interact with a Database. Fetch information, update information, remove or add information
  as in: SELECT, UPDATE, DELETE, ADD
Other types of queries

- **DCL (Data control language)**
  - GRANT, REVOKE, ...
- **TCL (Transaction control language)**

```plaintext
-- start a new transaction
start transaction;

-- Debit
UPDATE sb_accounts
SET balance = balance - 1000
WHERE account_no = 932656;

-- Credit
UPDATE ca_accounts
SET balance = balance + 1000
WHERE account_no = 933456;

-- commit changes
commit;
```
# Logging in - Hands on!

## Database Diagram

![Database Diagram](image)

## Table

<table>
<thead>
<tr>
<th>name</th>
<th>email</th>
<th>created_at</th>
<th>updated_at</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shawn</td>
<td><a href="mailto:shawn@spbd.gov">shawn@spbd.gov</a></td>
<td>2012-12-10 T 10:00 UTC</td>
<td>2013-02-16 T 14:00 UTC</td>
</tr>
<tr>
<td>Gus</td>
<td><a href="mailto:gus@sbpd.gov">gus@sbpd.gov</a></td>
<td>2012-12-01 T 07:00 UTC</td>
<td>2013-02-17 T 09:00 UTC</td>
</tr>
</tbody>
</table>
Logging in - Handson!

[root@perconalive-1-master ~]# mysql
mysql> help
mysql> SHOW DATABASES;
mysql> show schemas;
mysql> USE imdb;
mysql> SHOW TABLES;
mysql> describe users;
mysql> show create table users;
mysql> SELECT * FROM users LIMIT 1;
mysql> SHOW PROCESSLIST;
mysql> exit
MySQL CLI - Basic features

• Prompt

```sql
mysql> prompt Master >
PROMPT set to 'Master > '
Master > prompt mysql>
PROMPT set to 'mysql>,'
```

Undo:
```sql
mysql> prompt
```

• Edit

```sql
mysql> edit
```
Opens up a beautiful little editor:
```sql
use sakila;
select distinct(customer_id) from payment limit 5;
```
To Tee or not to Tee ...

• Tee

mysql> tee /tmp/tee.log
Logging to file '/tmp/tee.log'
mysql> select * from sakila.city limit 10;
mysql> notee
mysql> exit
# cat /tmp/tee.log

• Formatting help
  • help!
  • \g
  • \G
MySQL CLI - basic functions

• Pager

```sql
mysql> pager grep queries
mysql> show engine innodb status \G
0 queries inside InnoDB, 0 queries in queue
mysql> pager md5sum
PAGER set to 'md5sum'
```

```sql
mysql> select * from city limit 10;
449d5bca6e0e5b19e7101478934a7e6 -
10 rows in set (0.00 sec)
```

```sql
mysql> select city_id, city, country_id, last_update FROM city
LIMIT 10 ;
449d5bca6e0e5b19e7101478934a7e6 -
10 rows in set (0.00 sec)
```
MySQL Workbench

- Visual Tool
- Design
- SQL Development
- Database Administration
- Database Migration
- Editions ([Webpage](#))
  - Community (GPL)
  - Standard
  - Enterprise
**Client Connections**

<table>
<thead>
<tr>
<th>Id</th>
<th>User</th>
<th>Host</th>
<th>DB</th>
<th>Command</th>
<th>Time</th>
<th>State</th>
<th>Thread...</th>
<th>Type</th>
<th>Name</th>
<th>Parent...</th>
<th>Instrumented</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>root</td>
<td>localhost</td>
<td>None</td>
<td>Sleep</td>
<td>72</td>
<td>None</td>
<td>41</td>
<td>FOREGO...</td>
<td>thread/sql...</td>
<td>0</td>
<td>YES</td>
<td>NULL</td>
</tr>
<tr>
<td>24</td>
<td>plmce</td>
<td>gateway</td>
<td>None</td>
<td>Sleep</td>
<td>53</td>
<td>None</td>
<td>50</td>
<td>FOREGO...</td>
<td>thread/sql...</td>
<td>0</td>
<td>YES</td>
<td>NULL</td>
</tr>
<tr>
<td>25</td>
<td>plmce</td>
<td>gateway</td>
<td>None</td>
<td>Sleep</td>
<td>53</td>
<td>None</td>
<td>51</td>
<td>FOREGO...</td>
<td>thread/sql...</td>
<td>1</td>
<td>YES</td>
<td>NULL</td>
</tr>
<tr>
<td>26</td>
<td>plmce</td>
<td>gateway</td>
<td>None</td>
<td>Query</td>
<td>0</td>
<td>Sending d...</td>
<td>52</td>
<td>FOREGO...</td>
<td>thread/sql...</td>
<td>1</td>
<td>YES</td>
<td>SELECT t.PROCE</td>
</tr>
<tr>
<td>27</td>
<td>plmce</td>
<td>gateway</td>
<td>None</td>
<td>Sleep</td>
<td>2</td>
<td>None</td>
<td>53</td>
<td>FOREGO...</td>
<td>thread/sql...</td>
<td>1</td>
<td>YES</td>
<td>NULL</td>
</tr>
<tr>
<td>1</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>Daemon</td>
<td>322</td>
<td>Suspending</td>
<td>28</td>
<td>FOREGO...</td>
<td>thread/sql...</td>
<td>1</td>
<td>YES</td>
<td>NULL</td>
</tr>
<tr>
<td>Name</td>
<td>Engine</td>
<td>Version</td>
<td>Row Format</td>
<td>Rows</td>
<td>Avg Row Length</td>
<td>Data Length</td>
<td>Max Data Length</td>
<td>Index Length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>---------</td>
<td>---------</td>
<td>------------</td>
<td>--------</td>
<td>----------------</td>
<td>-------------</td>
<td>-----------------</td>
<td>--------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>aka_title</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>290478</td>
<td>88</td>
<td>24.5 MiB</td>
<td>0.0 bytes</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cast_info</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>2118776</td>
<td>51</td>
<td>1.0 GiB</td>
<td>0.0 bytes</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>char_name</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>2396994</td>
<td>53</td>
<td>121.6 MiB</td>
<td>0.0 bytes</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>comments</td>
<td>InnoDB</td>
<td>10</td>
<td>Dynamic</td>
<td>0</td>
<td>0</td>
<td>16.0 KiB</td>
<td>0.0 bytes</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>comp_cast_type</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>4</td>
<td>4096</td>
<td>16.0 KiB</td>
<td>0.0 bytes</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>company_name</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>234398</td>
<td>69</td>
<td>15.5 MiB</td>
<td>0.0 bytes</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>company_type</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>4</td>
<td>4096</td>
<td>16.0 KiB</td>
<td>0.0 bytes</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>complete_cast</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>97356</td>
<td>48</td>
<td>4.5 MiB</td>
<td>0.0 bytes</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>favorites</td>
<td>InnoDB</td>
<td>10</td>
<td>Dynamic</td>
<td>0</td>
<td>0</td>
<td>16.0 KiB</td>
<td>0.0 bytes</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>info_type</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>113</td>
<td>144</td>
<td>16.0 KiB</td>
<td>0.0 bytes</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>keyword</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>87477</td>
<td>54</td>
<td>4.5 MiB</td>
<td>0.0 bytes</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kind_type</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>7</td>
<td>2340</td>
<td>16.0 KiB</td>
<td>0.0 bytes</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>link_type</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>18</td>
<td>910</td>
<td>16.0 KiB</td>
<td>0.0 bytes</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>movie_companies</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>1891345</td>
<td>55</td>
<td>100.6 MiB</td>
<td>0.0 bytes</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>movie_info</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>9550897</td>
<td>88</td>
<td>810.0 MiB</td>
<td>0.0 bytes</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>movie_info_idx</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>932372</td>
<td>41</td>
<td>36.6 MiB</td>
<td>0.0 bytes</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>movie_keyword</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>2771370</td>
<td>33</td>
<td>87.6 MiB</td>
<td>0.0 bytes</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>movie_link</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>920920</td>
<td>38</td>
<td>33.6 MiB</td>
<td>0.0 bytes</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>movie_ratings</td>
<td>InnoDB</td>
<td>10</td>
<td>Dynamic</td>
<td>0</td>
<td>0</td>
<td>16.0 KiB</td>
<td>0.0 bytes</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>2812405</td>
<td>58</td>
<td>156.7 MiB</td>
<td>0.0 bytes</td>
<td>125</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>page_views</td>
<td>InnoDB</td>
<td>10</td>
<td>Dynamic</td>
<td>4</td>
<td>4096</td>
<td>16.0 KiB</td>
<td>0.0 bytes</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>person_info</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>2268056</td>
<td>161</td>
<td>349.8 MiB</td>
<td>0.0 bytes</td>
<td>41</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>role_type</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>12</td>
<td>1365</td>
<td>16.0 KiB</td>
<td>0.0 bytes</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>InnoDB</td>
<td>10</td>
<td>Compact</td>
<td>1533886</td>
<td>68</td>
<td>100.6 MiB</td>
<td>0.0 bytes</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>user_friends</td>
<td>InnoDB</td>
<td>10</td>
<td>Dynamic</td>
<td>0</td>
<td>0</td>
<td>16.0 KiB</td>
<td>0.0 bytes</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>users</td>
<td>InnoDB</td>
<td>10</td>
<td>Dynamic</td>
<td>49803</td>
<td>95</td>
<td>4.5 MiB</td>
<td>0.0 bytes</td>
<td>3.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Top File I/O Activity Report

Show the Files doing the most I/Os in bytes.

<table>
<thead>
<tr>
<th>File</th>
<th>Total IOs (#)</th>
<th>Read Requests</th>
<th>Total Read I/O</th>
<th>Avg Read I/O</th>
<th>Write Requests</th>
<th>Total Write</th>
<th>Avg Write I/O</th>
<th>Write Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>/var/lib/mysql/ibtmp1</td>
<td>14516224</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>130</td>
<td>1116632</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>/var/lib/mysql/ibdata1</td>
<td>8372224</td>
<td>380</td>
<td>8339456</td>
<td>21946</td>
<td>2</td>
<td>32786</td>
<td>16384</td>
<td>0.39</td>
</tr>
<tr>
<td>/var/lib/mysql/xb_doublewrite</td>
<td>3964928</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>3964928</td>
<td>792986</td>
<td>100</td>
</tr>
<tr>
<td>/var/lib/mysql/myisamproc.MYD</td>
<td>2469840</td>
<td>880</td>
<td>2469840</td>
<td>2807</td>
<td>5</td>
<td>792986</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>/var/lib/mysql/myisamusers.ibd</td>
<td>753664</td>
<td>46</td>
<td>753664</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>/var/lib/mysql/imdb.title.ibd</td>
<td>98304</td>
<td>6</td>
<td>98304</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/innodb_index_stats.ibd</td>
<td>98304</td>
<td>6</td>
<td>98304</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/imdb/aka_name.ibd</td>
<td>81920</td>
<td>5</td>
<td>81920</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/imdb/aka_title.ibd</td>
<td>81920</td>
<td>5</td>
<td>81920</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/imdb/cast_info.ibd</td>
<td>81920</td>
<td>5</td>
<td>81920</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/imdb/char_name.ibd</td>
<td>81920</td>
<td>5</td>
<td>81920</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/imdb/movie_companies.ibd</td>
<td>81920</td>
<td>5</td>
<td>81920</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/imdb/movie_info.ibd</td>
<td>81920</td>
<td>5</td>
<td>81920</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/imdb/movie_info_idx.ibd</td>
<td>81920</td>
<td>5</td>
<td>81920</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/imdb/movie_keyword.ibd</td>
<td>81920</td>
<td>5</td>
<td>81920</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/imdb/movie_link.ibd</td>
<td>81920</td>
<td>5</td>
<td>81920</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/imdb/movie_link.ibd</td>
<td>81920</td>
<td>5</td>
<td>81920</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/imdb/movie_link.ibd</td>
<td>81920</td>
<td>5</td>
<td>81920</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/imdb/person_info.ibd</td>
<td>81920</td>
<td>5</td>
<td>81920</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/imdb/producer.ibd</td>
<td>81920</td>
<td>5</td>
<td>81920</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/imdb/production.ibd</td>
<td>81920</td>
<td>5</td>
<td>81920</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/imdb/script.ibd</td>
<td>81920</td>
<td>5</td>
<td>81920</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/imdb/script.ibd</td>
<td>81920</td>
<td>5</td>
<td>81920</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/imdb/script.ibd</td>
<td>81920</td>
<td>5</td>
<td>81920</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>.../mysql/imdb/script.ibd</td>
<td>81920</td>
<td>5</td>
<td>81920</td>
<td>16384</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Operational DBA In A Nutshell

- Setting Up Today’s Environment
- Installation
- Logging In
- MySQL Privileges
- Diagnostics
- Troubleshooting
- Backups

- Replication
- Schema Changes
- Configuration Optimization
MySQL privileges

- Users
- `mysql` database
- Percona Toolkit
- Default Permissions
- Creating an Application user
User privileges overview

1. Users
2. Grants
3. Passwords
1) Users

- Identify users based on: **user@host**
  - user: username
  - host: hostname/ip/network of the client that connects
    - different host, different user, different ‘grants’
    - use of wildcards

- Examples:
  - ‘dim0’@‘localhost’
  - ‘tom’@‘app0001.company.com’
  - ‘awesomeuser’@‘192.168.1.212’
  - ‘kenju’@‘192.168.%’

- Creating A User:
  - > CREATE USER 'dim0'@'app0001';
  - Drop user: change CREATE into DROP
2) Grants

- Grant the user some kind of privilege
- Example: INSERT, SELECT, UPDATE, DELETE
- Grant ... to:

  - SQL Command:
    - > GRANT SELECT ON db.* TO ‘dim0’@‘app0001’;
    - > GRANT INSERT ON *.* TO ‘dim0’@‘app0001’;
  - Revoking privileges: change GRANT into REVOKE
Grants Table/Column level constraints

- Possible:
  - `> GRANT SELECT ON db.table TO 'dim0'@'app';`
  - `> GRANT SELECT (col) ON db.table to 'monsieur'@'app';`
- Using too many columns might make authentication slower
- Not commonly used
Complete list of grants

CREATE
DROP
GRANT OPTION
LOCK TABLES
EVENT
ALTER
DELETE
INDEX
INSERT
SELECT
UPDATE
CREATE TEMPORARY TABLES
TRIGGER
CREATE VIEW
SHOW VIEW
ALTER ROUTINE
CREATE ROUTINE
EXECUTE
FILE
CREATE USER
PROCESS
PROXY
RELOAD
REPLICATION CLIENT
REPLICATION SLAVE
SHOW DATABASES
SHUTDOWN
SUPER
ALL [PRIVILEGES]
USAGE
3) Password

> SELECT PASSWORD('pass')

Password('pass'): *196BDEDE2AE4F84CA44C47D54D78478C7E2BD7B7

Password generating statements:

> SET PASSWORD FOR 'dim0'@'app0001' = PASSWORD('pass');
> SET PASSWORD FOR 'dim0'@'app0001' = '*196BDEDE2...';

> CREATE USER 'dim0'@'app' IDENTIFIED BY PASSWORD 'pass';

MYSQL > 5.7.16

> SET PASSWORD FOR 'dim0'@'app0001' = 'plaintextpass';
KISS: All in one

> GRANT SELECT ON imdb.* TO 'dim0'@'192.168.24.6' IDENTIFIED BY 'plaintextpass';
Showing the grants

- Show grants: manual
  - Check the mysql.user table
  - Check privileges for a certain combo

```sql
> select user, host from mysql.user;
+-------+-----------+
| user  | host      |
+-------+-----------+
| root  | %         |
| root  | 127.0.0.1 |
| root  | ::1       |
| plmce | localhost |

> show grants for 'plmce'@'localhost';
```
Showing the grants: The Percona way

- Show grants: The Percona Way!
  - pt-show-grants

[root@ps1 ~]# pt-show-grants
-- Grants dumped by pt-show-grants
-- Dumped from server Localhost via UNIX socket, MySQL 5.7.14-79.1-log at
2019-04-10 16:02:56
-- Grants for 'plmce'@'localhost'
GRANT USAGE ON *.* TO 'plmce'@'localhost'
IDENTIFIED BY PASSWORD
'**221D28791318A46BA8556B182ADF26D8D360220F';
GRANT ALL PRIVILEGES ON `imdb`.* TO 'plmce'@'localhost';
Grants in the MySQL database

```sql
node1 mysql> show tables;
+---------------------------+
| Tables_in_mysql           |
| columns_priv              |
| db                        |
| event                     |
| func                      |
| general_log               |
| help_category             |
| help_keyword              |
| help_relation             |
| help_topic                |
| host                      |
| ndb_binlog_index          |
+---------------------------+
```
Grant in the MySQL database

```
mysql> select * from user where user = 'imdb' \G
*************************** 1. row ***************************
    Host: localhost
     User: imdb
  Select_priv: N
  Insert_priv: N
  Update_priv: N
  Delete_priv: N
  Create_priv: N
   Drop_priv: N
 Reload_priv: N
 Shutdown_priv: N
   Process_priv: N
   File_priv: N
   Grant_priv: N
References_priv: N
  Index_priv: N
  Alter_priv: N
 Show_db_priv: N
  Super_priv: N
Create_tmp_table_priv: N
 Lock_tables_priv: N
 Execute_priv: N
Repl_slave_priv: N

Repl_client_priv: N
  Create_view_priv: N
 Show_view_priv: N
Create_routine_priv: N
 Alter_routine_priv: N
Create_user_priv: N
  Event_priv: N
  Trigger_priv: N
Create_tablespace_priv: N
  ssl_type: 
ssl_cipher: 
x509_issuer: 
x509_subject: 
max_questions: 0
max_updates: 0
max_connections: 0
max_user_connections: 0
plugin: *97312E87D257F45E02FC031D5DF926A8398533C5
authentication_string: NULL
password_expired: N
password_last_changed: NULL
password_lifetime: NULL
account_locked: N
1 row in set (0.00 sec)
```
Default user/permissions

• If using CentOS/RHEL a password is created for root automagically:
  • /var/log/mysql/mysql.log
  [Note] A temporary password is generated for root@localhost: 8)13ftQG50Y
• Debian/Ubuntu will prompt during installation, if no password included, it will only listen to the local socket.
• https://www.percona.com/blog/2016/05/18/where-is-the-mysql-5-7-root-password/
Grants: mysql_secure_installation

- Reworked since 5.7
- Password validation plugin included
- You can set a password for root accounts.
- You can remove root accounts that are accessible from outside the local host.
- You can remove anonymous-user accounts.
- You can remove the test database (which by default can be accessed by all users, even anonymous users), and privileges that permit anyone to access databases with names that start with test_.

Privileges: localhost vs 127.0.0.1

- On *nix, MySQL programs treat localhost in a way that is likely different from what you expect compared to other network-based programs.
- For connections to localhost, MySQL programs attempt to connect to the local server by using a socket file.
  - This occurs even if a --port or -P option is given to specify a port number.
  - To ensure that the client makes a TCP/IP connection to the local server, use --host or -h to specify a host name value of 127.0.0.1, or the IP address or name of the local server.
  - You can also specify the connection protocol explicitly, by using the --protocol=TCP option.
Privileges: Guidelines

- Do not give rights for apps or general users to the mysql database.
- DML statements are possible on mysql database
  - Use FLUSH PRIVILEGES to apply direct changes.
- Don't GRANT ALL to every user.
- Use hostname, netmask, etc when possible to restrict accidental access.
Grants - Handson!

• Install Percona Toolkit
  # yum install percona-toolkit
  # pt-show-grants
  – show all grants on a system (in order)
  • good for version control

[root@node1 lib]# pt-show-grants
-- Grants dumped by pt-show-grants
-- Dumped from server Localhost via UNIX socket, MySQL 5.7.12-30.2-log at 2017-03-21 00:18:28
-- Grants for '@localhost'
GRANT USAGE ON *.* TO '@localhost';
-- Grants for '@node1'
GRANT USAGE ON *.* TO '@node1';
Handson - Default permissions are wrong

```
root@ip-172-31-0-120:~# mysql_secure_installation
Securing the MySQL server deployment.
Enter password for user root:

VALIDATE PASSWORD PLUGIN
Press y|Y for Yes, any other key for No: No
Using existing password for root.

Change the password for root ? ((Press y|Y for Yes, any other key for No) : Y

Remove anonymous users? (Press y|Y for Yes, any other key for No) : Y
Success.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : Y
- Dropping test database...
Success.
- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : Y
Success.

All done!
```
Handson - Default permissions are wrong

- Now we have ‘secure’ privileges
  
  # mysql -u root -p

- Create a user for ‘my-movies’ application
  
  mysql> GRANT ALL PRIVILEGES ON imdb.* TO 'mymovies'@'localhost' IDENTIFIED BY 'password';

- Change application user&password:
  
  # vim /var/www/html/my-movies/lib/config.inc.php

- Verify Application:
  
  http://localhost:8080/my-movies

- Doublecheck with pt-show-grants
  
  # pt-show-grants -u root --ask-pass

- Create a .my.cnf file
  
  # vim .my.cnf

  [mysql]
  user=root
  password=root
MySQL 8 Security features

- **Password rotation policy enforcement**
  - Password expiration, to require passwords to be changed periodically.
  - Dual passwords, to enable clients to connect using either a primary or secondary password.

  ```sql
  ALTER USER 'jeffrey'@'localhost' PASSWORD EXPIRE;
  
  Default expiration -> unlimited
  ```

- **Automatic assignment of default roles when new users are created**
- **Creation of Roles (like groups)**
  - `GRANT ALL ON application-database.* TO 'favorit_developers';`
  - `GRANT SELECT ON app_db.* TO 'application_readers';`
  - `GRANT INSERT, UPDATE, DELETE ON app_db.* TO 'application_writers';`
Operational DBA In A Nutshell

- Setting Up Today’s Environment
- Installation
- Logging In
- MySQL Privileges
- Diagnostics
- Troubleshooting
- Backups

- Replication
- Schema Changes
- Configuration Optimization
Troubleshooting

- Diagnostics
  - Operating System
  - MySQL
- Percona Toolkit
- Queries
- PMM
- Alerting
Getting Ready - Handson

• Open up New Terminal window
  # yum install pmm-client innotop
  # pmm-admin config --server 10.0.0.202 --server-user admin
      --server-password percona2019
  # pmm-admin add linux:metrics
  # pmm-admin add mysql --password root
  # pmm-admin list

• Services:
  – mysql:queries
  – linux:metrics
  – mysql:metrics

• Add load
  # /home/centos/percona/add_load.py
Diagnostics OS - Handson

• OS metrics (memory, cpu...)
  – vmstat
  – free
  – mpstat
  – iostat
  – top
  – ps
  – sar
  – # cat /proc/meminfo
Diagnostics OS - Handson

• vmstat: Virtual Memory Statistics

```bash
[root@master percona]# vmstat -S m 5
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
4  0      0      5      0    193    0    0 626   130  139 1090  0  1 98  0  0
2  0      0      6      0    195    0    0 90716     4 1506 500 70 29  0  0  0
5  0      0      6      0    190    0    0 97937    33 1678 578 69 31  0  0  0
7  0      0      6      0    188    0    0 116584   66 2029 592 65 34  1  0  0
```

• free

```bash
[root@master percona]# free -m
total        used        free      shared  buff/cache   available
Mem:            488         278           8           3         201         169
Swap:          2015           1        2014
```

• mpstat: Processor Statistics

```bash
[root@master percona]# mpstat 5
Linux 3.10.0-514.10.2.el7.x86_64 (master) 04/17/2017 _x86_64_ (1 CPU)
 02:09:05 PM  CPU  %usr  %nice  %sys %iowait  %irq  %soft  %steal  %guest  %gnice  %idle
 02:09:10 PM  all  4.72   0.00   0.82   0.00   0.00   0.00   0.00   0.00   0.00   0.00
```
Diagnostics OS - Handson

- **iostat** commonly used: `iostat -x -m 1`

```
  %vfcpu:  %user  %nice  %system  %iowait  %steal  %idle
     0.24    0.00    0.63    0.07    0.00   99.06

Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await  r_await  w_await  svctm  %util
sda               0.02     1.54    9.26    2.07     0.31     0.05    64.10     0.06    5.48    2.31   19.69   0.50   0.56
dm-0              0.00     0.00    6.49    1.96     0.22     0.02    59.37     0.20    23.38    2.47   92.82   0.61   0.51
dm-1              0.00     0.00    0.15    0.23     0.00     0.00    11.26     0.00    9.63    0.83   15.47   0.56   0.02
dm-2              0.00     0.00    1.02    0.20     0.06     0.02    135.74     0.01    5.82    4.12   14.62   0.56   0.07
```

- **%util**: how many % of time at least one request was busy
- **await+svctm**:
  - response time
  - writes and reads combined
Disk subsystem diagnostics - Handson

• **# pt-diskstats**
  • similar to iostat
  • more detailed
  • interactive
  • hide inactive disks
  • space for header
  • `/` for regex
  • `?` for help
  • reads `/proc/diskstats`
  • show r/w response time
**pt-diskstats - Handson!**

- reads `/proc/diskstats`, shows wr/rd response time

```bash
# pt-diskstats

<table>
<thead>
<tr>
<th>ts</th>
<th>device</th>
<th>rd_s</th>
<th>rd_avkb</th>
<th>rd_mb_s</th>
<th>rd_mrg</th>
<th>rd_cnc</th>
<th>rd_rt</th>
<th>wr_s</th>
<th>wr_avkb</th>
<th>wr_mb_s</th>
<th>wr_mrg</th>
<th>wr_cnc</th>
<th>wr_rt</th>
<th>busy</th>
<th>in_prg</th>
<th>io_s</th>
<th>qtime</th>
<th>stime</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>dm-2</td>
<td>932.2</td>
<td>127.5</td>
<td>116.1</td>
<td>0%</td>
<td>4.7</td>
<td>5.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0%</td>
<td>0.0</td>
<td>0.0</td>
<td>0%</td>
<td>0.0</td>
<td>25%</td>
<td>0</td>
<td>932.2</td>
</tr>
<tr>
<td>1.0</td>
<td>dm-2</td>
<td>1021.3</td>
<td>127.4</td>
<td>127.1</td>
<td>0%</td>
<td>5.5</td>
<td>5.4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0%</td>
<td>0.0</td>
<td>0.0</td>
<td>0%</td>
<td>0.0</td>
<td>30%</td>
<td>25</td>
<td>1021.3</td>
</tr>
</tbody>
</table>
```
Percona toolkit

- # pt-summary
  - System info like
    - Memory
    - CPU
    - Network
    - …
MySQL diagnostics - Handson

- MySQL Logfiles
  - MySQL Log: /var/log/mysqld.log
  - Errorlog: /var/lib/mysql/error.log
  - General log: connects, queries, ...
MySQL diagnostics - Handson

• MySQL
  - `mysql> SHOW GLOBAL STATUS \G;
  - `mysql> SHOW ENGINE INNODB STATUS \G
  - `mysql> SHOW PROCESSLIST;

• Shell
  - `# mysqladmin extended-status
  - `# mysqladmin proc stat
## More metrics! pt-mext - Handson!

```shell
# pt-mext -r -- mysqladmin -p ext -i 10 -c 3

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value1</th>
<th>Value2</th>
<th>Value3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binlog_cache_disk_use</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Binlog_cache_use</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bytes_received</td>
<td>2875788973602</td>
<td>1738235</td>
<td>346057</td>
</tr>
<tr>
<td>Bytes_sent</td>
<td>863929033790</td>
<td>588078</td>
<td>536398</td>
</tr>
<tr>
<td>Com_begin</td>
<td>6298644573</td>
<td>3516</td>
<td>5102</td>
</tr>
<tr>
<td>Com_delete</td>
<td>23721852</td>
<td>26</td>
<td>51</td>
</tr>
<tr>
<td>Com_insert</td>
<td>4454794705</td>
<td>1518</td>
<td>3287</td>
</tr>
<tr>
<td>Com_replace</td>
<td>527848577</td>
<td>197</td>
<td>121</td>
</tr>
<tr>
<td>Com_select</td>
<td>6993291133</td>
<td>8114</td>
<td>7594</td>
</tr>
<tr>
<td>Com_set_option</td>
<td>5112076</td>
<td>250</td>
<td>262</td>
</tr>
<tr>
<td>Connections</td>
<td>7331059</td>
<td>250</td>
<td>262</td>
</tr>
<tr>
<td>Created_tmp_disk_tables</td>
<td>113568</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Created_tmp_files</td>
<td>7803</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Created_tmp_tables</td>
<td>729281259</td>
<td>1816</td>
<td>479</td>
</tr>
</tbody>
</table>
```
More metrics! pt-mext

- Look at current global behaviour of the database
- Query Optimization necessary? (sorting_%, handler_%, range_%, tmp_table_%)
- InnoDB misbehaving?
- ...

P E R C O N A
L I V E
Percona toolkit

• # pt-mysql-summary
  – Replication status
  – Processlist
  – Counters (pt-mext/show global status)
  – InnoDB section: BP Info
  – ...
Slow log statistics

Check:
mysql> select @@slow_query_log;
mysql> select @@slow_query_log_file;
mysql> select @@long_query_time;

Set:
mysql> SET GLOBAL slow_query_log=on;
mysql> SET GLOBAL slow_query_log_file='/var/lib/mysql/master-slow.log';
mysql> SET GLOBAL long_query_time=0.4;
mysql> SET GLOBAL log_slowVerbosity=full;

Load!
# cat /home/centos/percona/queries.sql | mysql
# tail -n 200 /var/lib/mysql/master-slow.log
Slowed log statistics

```
root@master ~]# cat /var/lib/mysql/master-slow.log
# Time: 2017-09-22T10:00:51.815819Z
# User@Host: root[root] @ localhost []  Id: 4149
# Schema: Last_errno: 0  Killed: 0
# Query_time: 12.349040  Lock_time: 0.000286  Rows_sent: 9  Rows_examined: 9748370  Rows_affected: 0
# Bytes_sent: 612  Tmp_tables: 0  Tmp_disk_tables: 0  Tmp_table_sizes: 0
# QC_Hit: No  Full_scan: Yes  Full_join: No  Tmp_table: No  Tmp_table_on_disk: No
# Filesort: No  Filesort_on_disk: No  Merge_passes: 0
#   InnoDB_IO_r_ops: 51622  InnoDB_IO_r_bytes: 845774848  InnoDB_IO_r_wait: 2.570710
#   InnoDB_rec_lock_wait: 0.000000  InnoDB_queue_wait: 0.000000
#   InnoDB_pages_distinct: 51133
SET timestamp=1506074451;
SELECT * FROM imdb.movie_info WHERE movie_id = 93221;
# Time: 2017-09-22T10:01:03.145903Z
# User@Host: root[root] @ localhost []  Id: 4149
# Schema: Last_errno: 0  Killed: 0
# Query_time: 11.326319  Lock_time: 0.000101  Rows_sent: 6  Rows_examined: 9748370  Rows_affected: 0
# Bytes_sent: 515  Tmp_tables: 0  Tmp_disk_tables: 0  Tmp_table_sizes: 0
# QC_Hit: No  Full_scan: Yes  Full_join: No  Tmp_table: No  Tmp_table_on_disk: No
# Filesort: No  Filesort_on_disk: No  Merge_passes: 0
#   InnoDB_IO_r_ops: 51622  InnoDB_IO_r_bytes: 845774848  InnoDB_IO_r_wait: 2.570710
#   InnoDB_rec_lock_wait: 0.000000  InnoDB_queue_wait: 0.000000
#   InnoDB_pages_distinct: 51133
```
pt-query-digest

• generate reports from
  – slow query log
    `pt-query-digest /var/lib/mysql/master-slow.log`
  – general log (not so useful)
  – tcpdump
### pt-query-digest

150ms user time, 40ms system time, 26.23M rss, 220.63M vsz

# Current date: Fri Sep 22 10:04:13 2017
# Hostname: master
# Files: /var/lib/mysql/master-slow.log
# Overall: 11 total, 8 unique, 0.24 QPS, 1.00x concurrency _______________
# Time range: 2017-09-22T10:03:20 to 2017-09-22T10:04:06

<table>
<thead>
<tr>
<th>Attribute</th>
<th>total</th>
<th>min</th>
<th>max</th>
<th>avg</th>
<th>95%</th>
<th>stddev</th>
<th>median</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exec time</td>
<td>46s</td>
<td>7us</td>
<td>21s</td>
<td>4s</td>
<td>13s</td>
<td>7s</td>
<td>2ms</td>
</tr>
<tr>
<td>Exec time</td>
<td>46s</td>
<td>7us</td>
<td>21s</td>
<td>4s</td>
<td>13s</td>
<td>7s</td>
<td>2ms</td>
</tr>
<tr>
<td>Lock time</td>
<td>1ms</td>
<td>0</td>
<td>269us</td>
<td>111us</td>
<td>176us</td>
<td>72us</td>
<td>103us</td>
</tr>
<tr>
<td>Rows sent</td>
<td>21</td>
<td>0</td>
<td>9</td>
<td>1.91</td>
<td>5.75</td>
<td>2.69</td>
<td>0.99</td>
</tr>
<tr>
<td>Rows examine</td>
<td>39.75M</td>
<td>0</td>
<td>21.16M</td>
<td>3.61M</td>
<td>8.86M</td>
<td>6.31M</td>
<td>0.99</td>
</tr>
<tr>
<td>Rows affecte</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bytes sent</td>
<td>4.65k</td>
<td>0</td>
<td>805</td>
<td>433.09</td>
<td>755.64</td>
<td>277.09</td>
<td>487.67</td>
</tr>
<tr>
<td>Merge passes</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Tmp tables</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.09</td>
<td>0</td>
<td>0.29</td>
<td>0</td>
</tr>
<tr>
<td>Tmp disk tbl</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.09</td>
<td>0</td>
<td>0.29</td>
<td>0</td>
</tr>
<tr>
<td>Tmp tbl size</td>
<td>16.00k</td>
<td>0</td>
<td>16.00k</td>
<td>1.45k</td>
<td>0</td>
<td>4.59k</td>
<td>0</td>
</tr>
<tr>
<td>Query size</td>
<td>645</td>
<td>27</td>
<td>199</td>
<td>58.64</td>
<td>65.89</td>
<td>44.08</td>
<td>42.48</td>
</tr>
<tr>
<td>InnoDB:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IO r bytes</td>
<td>2.59G</td>
<td>0</td>
<td>1.01G</td>
<td>294.23M</td>
<td>1006.04M</td>
<td>410.09M</td>
<td>62.55k</td>
</tr>
<tr>
<td>IO r ops</td>
<td>165.50k</td>
<td>0</td>
<td>64.66k</td>
<td>18.39k</td>
<td>62.55k</td>
<td>25.50k</td>
<td>3.89</td>
</tr>
<tr>
<td>IO r wait</td>
<td>9s</td>
<td>0</td>
<td>3s</td>
<td>960ms</td>
<td>3s</td>
<td>1s</td>
<td>2ms</td>
</tr>
<tr>
<td>pages distin</td>
<td>163.61k</td>
<td>0</td>
<td>63.72k</td>
<td>18.18k</td>
<td>62.55k</td>
<td>25.50k</td>
<td>3.89</td>
</tr>
<tr>
<td>queue wait</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>rec lock wai</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Boolean:

- Filesort: 9% yes, 90% no
- Full scan: 45% yes, 54% no
- Tmp table: 9% yes, 90% no
- Tmp table on: 9% yes, 90% no

### Profile

<table>
<thead>
<tr>
<th>Rank</th>
<th>Query ID</th>
<th>Response time</th>
<th>Calls</th>
<th>R/Call</th>
<th>V/M</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x8C9D4708FCE85CF3</td>
<td>25.6656</td>
<td>55.6%</td>
<td>2</td>
<td>12.8328</td>
<td>SELECT imdb.movie_info</td>
</tr>
<tr>
<td>2</td>
<td>0x212DEB0A59444468</td>
<td>20.5102</td>
<td>44.4%</td>
<td>1</td>
<td>20.5102</td>
<td>SELECT imdb.cast_info</td>
</tr>
</tbody>
</table>

**InnoDB IO:**

- Read bytes: 2.59G
- Read operations: 165.50k
- Read wait: 9s
- Index read: 64.66k
- Index read wait: 3s
- Page reads: 163.61k
- Page reads wait: 0s
- Queue reads: 0
- Recruitment lock wait: 0

**Boolean:**

- Filesort: 9% yes, 90% no
- Full scan: 45% yes, 54% no
- Tmp table: 9% yes, 90% no
- Tmp table on: 9% yes, 90% no
pt-query-digest

Overall: 11 total, 8 unique, 0.30 QPS, 0.99x concurrency

# Time range: 2017-04-20T18:37:07 to 2017-04-20T18:37:44

<table>
<thead>
<tr>
<th>Attribute</th>
<th>total</th>
<th>min</th>
<th>max</th>
<th>avg</th>
<th>95%</th>
<th>stddev</th>
<th>median</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exec time</td>
<td>37s</td>
<td>7us</td>
<td>19s</td>
<td>3s</td>
<td>9s</td>
<td>6s</td>
<td>6ms</td>
</tr>
<tr>
<td>Lock time</td>
<td>39ms</td>
<td>0</td>
<td>30ms</td>
<td>4ms</td>
<td>4ms</td>
<td>8ms</td>
<td>495us</td>
</tr>
<tr>
<td>Rows sent</td>
<td>21</td>
<td>0</td>
<td>9</td>
<td>1.91</td>
<td>5.75</td>
<td>2.69</td>
<td>0.99</td>
</tr>
<tr>
<td>Rows examine</td>
<td>39.75M</td>
<td>0</td>
<td>21.16M</td>
<td>3.61M</td>
<td>8.86M</td>
<td>6.31M</td>
<td>0.99</td>
</tr>
<tr>
<td>Rows affected</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bytes sent</td>
<td>4.65k</td>
<td>0</td>
<td>805</td>
<td>432.73</td>
<td>755.64</td>
<td>277.09</td>
<td>487.67</td>
</tr>
<tr>
<td>Merge passes</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Tmp tables</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.09</td>
<td>0</td>
<td>0.29</td>
<td>0</td>
</tr>
<tr>
<td>Tmp disk tbl</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.09</td>
<td>0</td>
<td>0.29</td>
<td>0</td>
</tr>
<tr>
<td>Tmp tbl size</td>
<td>16.00k</td>
<td>0</td>
<td>16.00k</td>
<td>1.45k</td>
<td>65.89</td>
<td>44.08</td>
<td>42.48</td>
</tr>
<tr>
<td>Query size</td>
<td>645</td>
<td>27</td>
<td>199</td>
<td>58.64</td>
<td>65.89</td>
<td>44.08</td>
<td>42.48</td>
</tr>
<tr>
<td>Rank</td>
<td>Query ID</td>
<td>Response time</td>
<td>Calls</td>
<td>R/Call</td>
<td>V/M</td>
<td>Item</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>---------------</td>
<td>---------------</td>
<td>-------</td>
<td>--------</td>
<td>-----</td>
<td>-------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0x212DEB0A59444468</td>
<td>18.7334</td>
<td>50.9</td>
<td>1</td>
<td>18.7334</td>
<td>SELECT imdb.cast_info imdb.title</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0x8C9D4708FCE85CF3</td>
<td>18.0220</td>
<td>49.0</td>
<td>2</td>
<td>9.0110</td>
<td>SELECT imdb.movie_info</td>
<td></td>
</tr>
<tr>
<td>MISC</td>
<td>0xMISC</td>
<td>0.0444</td>
<td>0.1</td>
<td>8</td>
<td>0.0056</td>
<td>&lt;6 ITEMS&gt;</td>
<td></td>
</tr>
</tbody>
</table>
### Query 1

0.01 QPS, 0.36x concurrency, ID 0x212DEB0A59444468 at byte 645

This item is included in the report because it matches `--limit`.

Scores: V/M = 3.70


<table>
<thead>
<tr>
<th>Attribute</th>
<th>pct</th>
<th>total</th>
<th>min</th>
<th>max</th>
<th>avg</th>
<th>95%</th>
<th>stddev</th>
<th>median</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>33</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exec time</td>
<td>50</td>
<td>54s</td>
<td>20s</td>
<td>34s</td>
<td>27s</td>
<td>34s</td>
<td>10s</td>
<td>27s</td>
</tr>
<tr>
<td>Lock time</td>
<td>90</td>
<td>10ms</td>
<td>388us</td>
<td>10ms</td>
<td>5ms</td>
<td>10ms</td>
<td>7ms</td>
<td>5ms</td>
</tr>
<tr>
<td>Rows sent</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Rows examine</td>
<td>53</td>
<td>42.32M</td>
<td>21.16M</td>
<td>21.16M</td>
<td>21.16M</td>
<td>21.16M</td>
<td>0</td>
<td>21.16M</td>
</tr>
<tr>
<td>Rows affecte</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bytes sent</td>
<td>30</td>
<td>970</td>
<td>485</td>
<td>485</td>
<td>485</td>
<td>485</td>
<td>0</td>
<td>485</td>
</tr>
<tr>
<td>Merge passes</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Tmp tables</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Tmp disk tbl</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Tmp tbl size</td>
<td>100</td>
<td>16.00k</td>
<td>16.00k</td>
<td>16.00k</td>
<td>16.00k</td>
<td>16.00k</td>
<td>0</td>
<td>16.00k</td>
</tr>
<tr>
<td>Query size</td>
<td>65</td>
<td>398</td>
<td>199</td>
<td>199</td>
<td>199</td>
<td>199</td>
<td>0</td>
<td>199</td>
</tr>
<tr>
<td>InnoDB:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IO r bytes</td>
<td>39</td>
<td>1.01G</td>
<td>1.01G</td>
<td>1.01G</td>
<td>1.01G</td>
<td>1.01G</td>
<td>0</td>
<td>1.01G</td>
</tr>
<tr>
<td>IO r ops</td>
<td>39</td>
<td>64.66k</td>
<td>64.66k</td>
<td>64.66k</td>
<td>64.66k</td>
<td>64.66k</td>
<td>0</td>
<td>64.66k</td>
</tr>
<tr>
<td>IO r wait</td>
<td>38</td>
<td>4s</td>
<td>4s</td>
<td>4s</td>
<td>4s</td>
<td>4s</td>
<td>0</td>
<td>4s</td>
</tr>
<tr>
<td>pages distin</td>
<td>38</td>
<td>63.72k</td>
<td>63.72k</td>
<td>63.72k</td>
<td>63.72k</td>
<td>63.72k</td>
<td>0</td>
<td>63.72k</td>
</tr>
<tr>
<td>queue wait</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
pt-query-digest

Filesort     100% yes,   0% no
# Full scan     100% yes,   0% no
# Tmp table     100% yes,   0% no
# Tmp table on 100% yes,   0% no
# String:
# Hosts        localhost
# Users        root
# Query_time distribution
#   1us
#  10us
# 100us
#   1ms
#  10ms
# 100ms
#   1s
#  10s+  ################################################################
# Tables
#    SHOW TABLE STATUS FROM `imdb` LIKE 'cast_info`
#    SHOW CREATE TABLE `imdb`.`cast_info`
#    SHOW TABLE STATUS FROM `imdb` LIKE 'title`
#    SHOW CREATE TABLE `imdb`.`title`
# EXPLAIN /*!50100 PARTITIONS*/
SELECT cast_info.* FROM imdb.cast_info INNER JOIN imdb.title on (cast_info.movie_id=title.id) WHERE cast_info.person_id = 2409 AND title.kind_id = 1 ORDER BY title.production_year DESC, title.id DESC
Query 2: 0.22 QPS, 2.00x concurrency, ID 0x8C9D4708FCE85CF3 at byte 2567
# This item is included in the report because it matches --limit.
# Scores: V/M = 0.00
# Attribute    pct   total     min     max     avg     95%  stddev  median
# ============ === ======= ======= ======= ======= ======= ======= =======
# Count         18       2
# Exec time     48     18s      9s      9s      9s      9s   147ms      9s
# Lock time      6     3ms   257us     2ms     1ms     2ms     2ms     1ms
# Rows sent    71      15       6       9    7.50       9    2.12    7.50
# Rows affecte   0       0       0       0       0       0       0       0
# Bytes sent    23   1.10k     515     612  563.50     612   68.59  563.50
# Merge passes   0       0       0       0       0       0       0       0
# Tmp tables     0       0       0       0       0       0       0       0
# Tmp disk tbl   0       0       0       0       0       0       0       0
# Tmp tbl size   0       0       0       0       0       0       0       0
# Query size    16    104      52      52      52      52      52      52
pt-query-digest

# Query_time distribution
#  1us
# 10us
# 100us
#  1ms
# 10ms
# 100ms
#  1s
# 10s+

# Tables
# SHOW TABLE STATUS FROM `imdb` LIKE 'movie_info'\G
# SHOW CREATE TABLE `imdb`.`movie_info`\G
# EXPLAIN /**!50100 PARTITIONS*/
SELECT * FROM imdb.movie_info WHERE movie_id = 93221\G
Optimizing the application - Handson

- The problem is that the database is not optimized, indexing is not done properly
Optimizing this application

• Indexes
  – to find the rows matching a WHERE clause quickly
  – reduce amount of data to be examined
  – helps avoid sorting and temp tables
  – random IO gets turned into sequential IO

• EXPLAIN !

• Examples
  
```sql
explain SELECT * FROM imdb.movie_info WHERE movie_id = 635;
```
Optimizing - Hands-on

```
MASTER> explain SELECT * FROM imdb.title WHERE `id` = 455;

<table>
<thead>
<tr>
<th>id</th>
<th>select_type</th>
<th>table</th>
<th>partitions</th>
<th>type</th>
<th>possible_keys</th>
<th>key</th>
<th>key_len</th>
<th>ref</th>
<th>rows</th>
<th>filtered</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SIMPLE</td>
<td>title</td>
<td>NULL</td>
<td>const</td>
<td>PRIMARY</td>
<td>PRIMARY</td>
<td>4</td>
<td>const</td>
<td>1</td>
<td>100.00</td>
<td>NULL</td>
</tr>
</tbody>
</table>
```

```
MASTER> explain SELECT * FROM imdb.movie_info WHERE movie_id = 635;

<table>
<thead>
<tr>
<th>id</th>
<th>select_type</th>
<th>table</th>
<th>partitions</th>
<th>type</th>
<th>possible_keys</th>
<th>key</th>
<th>key_len</th>
<th>ref</th>
<th>rows</th>
<th>filtered</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SIMPLE</td>
<td>movie_info</td>
<td>NULL</td>
<td>ALL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>9550897</td>
<td>10.00</td>
<td>Using where</td>
</tr>
</tbody>
</table>
```
Optimizing this application

- Apply the changes:
  
  ```
  # cat /home/centos/percona/billkarwin-opt.sql | mysql -D imdb
  ```

- Have another look with pt-query-digest
  
  ```
  # > /var/lib/mysql/master-slow.log
  # cat /home/centos/percona/queries.sql | mysql
  # pt-query-digest /var/lib/mysql/master-slow.log | less
  ```
# Explain Columns

<table>
<thead>
<tr>
<th>Column</th>
<th>JSON Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>select_id</td>
<td>The SELECT identifier</td>
</tr>
<tr>
<td>select_type</td>
<td>None</td>
<td>The SELECT type</td>
</tr>
<tr>
<td>table</td>
<td>table_name</td>
<td>The table for the output row</td>
</tr>
<tr>
<td>partitions</td>
<td>partitions</td>
<td>The matching partitions</td>
</tr>
<tr>
<td>type</td>
<td>access_type</td>
<td>The join type</td>
</tr>
<tr>
<td>possible_keys</td>
<td>possible_keys</td>
<td>The possible indexes to choose</td>
</tr>
<tr>
<td>key</td>
<td>key</td>
<td>The index actually chosen</td>
</tr>
<tr>
<td>key_len</td>
<td>key_length</td>
<td>The length of the chosen key</td>
</tr>
<tr>
<td>ref</td>
<td>ref</td>
<td>The columns compared to the index</td>
</tr>
<tr>
<td>rows</td>
<td>rows</td>
<td>Estimate of rows to be examined</td>
</tr>
<tr>
<td>filtered</td>
<td>filtered</td>
<td>Percentage of rows filtered by table condition</td>
</tr>
<tr>
<td>Extra</td>
<td>None</td>
<td>Additional information</td>
</tr>
</tbody>
</table>
Explain join types

- system
- const
- eq_ref
- ref
- fulltext
- ...
- range
- index (→ index tree is scanned)
- all (→ rows are scanned)
Info & References

• Percona Online Resource: Indexes
  • https://dev.mysql.com/doc/refman/5.7/en/explain-output.html
• Percona: Matthew Boehm on Indexes
  • https://noelherrick.com/explained/#

MySQL Explain Explained

Query Plan

Hover over the terms to see their definition.

<table>
<thead>
<tr>
<th>id</th>
<th>select_type</th>
<th>table</th>
<th>type</th>
<th>key</th>
<th>key_len</th>
<th>ref</th>
<th>rows</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PRIMARY</td>
<td>ALL</td>
<td>ALL</td>
<td></td>
<td></td>
<td></td>
<td>497352</td>
<td>Using temporary; Using filesort</td>
</tr>
<tr>
<td>2</td>
<td>DERIVED</td>
<td>dm</td>
<td>ALL</td>
<td></td>
<td></td>
<td></td>
<td>24</td>
<td>Using where; Using temporary; Using filesort</td>
</tr>
<tr>
<td>2</td>
<td>DERIVED</td>
<td>m</td>
<td>eq_ref</td>
<td>PRIMARY</td>
<td>4</td>
<td>employees.dm.emp_no</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>DERIVED</td>
<td>de</td>
<td>ref</td>
<td>dept_no</td>
<td>4</td>
<td>employees.dm.dept_no</td>
<td>20723</td>
<td>Using where</td>
</tr>
<tr>
<td>2</td>
<td>DERIVED</td>
<td>e</td>
<td>eq_ref</td>
<td>PRIMARY</td>
<td>4</td>
<td>employees.de.emp_no</td>
<td>1</td>
<td>Using Index</td>
</tr>
</tbody>
</table>

5 row(s) returned in 0.0763 second(s).
Monitoring and Management: PMM
Monitoring and management: PMM

• Percona Monitoring and Management
  • Open Source
  • MySQL®, MariaDB®, MongoDB® and PostgreSQL performance

• Tooling
  • Metrics
    • Prometheus
    • Grafana
  • QAN: Query Analytics (MySQL and Mongo)
Monitoring and management: PMM
PMM - deploying

- PMM Server
  - Docker images
  - Amazon amis
  - Virtualbox images
- PMM Client
  - packages available for
    - RHEL/CentOS
    - Ubuntu/Debian
PMM - usage

- PMM Landing page
  - Add load: `/home/centos/percona/add_load.py`
  - http://52.87.153.255
    - user: admin
    - pwd: percona2019
- All info in Grafana dashboards
  - Metrics
  - QAN
Overview
Percona Monitoring and Management (PMM) is a free and open-source platform for managing and monitoring MySQL and MongoDB performance, and provides time-based analysis to ensure that your data works as efficiently as possible.

Documentation
Please consult the official PMM documentation to learn more about PMM. Also of interest are the Release notes and FAQ for common questions about PMM.

Community and Blogs
On the PMM Community Forums you will find help from Perconians and the Community at large. Further, we publish PMM announcements and use cases regularly on the Percona Database Performance Blog.

Get Help from Percona
Are you looking for assistance that comes with a response-time guarantee? Let our Support team help you! Already a customer? Log in to your Percona Support Portal.

Environment Overview

- Host: All
- CPU Busy: 17%
- Mem Avail: 47%
- Disk Reads: 109.0 MB/s
- Disk Writes: 77.6 MB/s
- Network IO: 6.6 MB/s
- DB Conns: 1.813 K
- DB QPS: 21.98 K
- Virtual CPUs: 157
- RAM: 261.67 GB
- Host uptime: 1.3 years
- DB uptime: 41.9 weeks
PMM: Handson!

- http://52.87.153.255
- Check out The OS and MySQL Dashboards
- Check out the Query Analytics Dashboard
Operational DBA In A Nutshell

• Setting Up Today’s Environment
• Installation
• Logging In
• MySQL Privileges
• Diagnostics
• Troubleshooting
• Backups

• Replication
• Schema Changes
• Configuration Optimization
Backups

• Exciting and essential
• MySQL is multi-engine
  – Different storage engines call for different strategies
• MySQL can be quite busy
  – Do we need to shut it down?
  – How much load will we generate?
  – How quickly can we restore?
Backing up your data - Concept

- Impact of Backup
  - **Hot backup** - does not prevent readers or writers from performing operations.
  - **Warm backup** - readers may continue, but write activity must be queued until the backup is completed.
  - **Cold backup** - system is unavailable during backup window.
Backing up your data - concepts

• Type of Backup
  – **Physical** - stores data files in the “native” format.
  – **Logical** - stores data in a generic representation, such as a SQL dump.
A few existing backup technologies

<table>
<thead>
<tr>
<th>Backup Technology</th>
<th>Impact (Perf)</th>
<th>Warmth</th>
<th>Backup Time</th>
<th>Restore Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cold Backup</td>
<td>very high</td>
<td>cold</td>
<td>very fast</td>
<td>fast</td>
</tr>
<tr>
<td>mysqlDump</td>
<td>high</td>
<td>warm</td>
<td>medium</td>
<td>slow</td>
</tr>
<tr>
<td>mydumper</td>
<td>high</td>
<td>warm</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>Snapshotting</td>
<td>high/medium</td>
<td>hot/warm</td>
<td>fast</td>
<td>fast</td>
</tr>
<tr>
<td>MySQL Enterprise Backup</td>
<td>low/medium</td>
<td>hot/warm</td>
<td>fast</td>
<td>fast</td>
</tr>
<tr>
<td>XtraBackup</td>
<td>low/medium</td>
<td>hot/warm</td>
<td>fast</td>
<td>fast</td>
</tr>
</tbody>
</table>
MySQLdump

• Included by default with MySQL
• Logical backup solution - creates SQL file to reproduce data
• Pro’s
  – Easy to use, reliable
• Cons
  – High impact, long restore time
MySQLdump - Handson

• Open 2 terminals on the master
• Take backup:
  
  # mkdir /var/backup/
  # mysqldump -p --all-databases --single-transaction >
  /var/backup/dump.sql

  — --all-databases for every database (including mysql).
  — --single-transaction is only safe when all InnoDB.
  — This ensures a hot backup.
  — --master-data=1 (optional) records the binary log coordinates.

• Restore as follows (don’t do this):
  
  # mysql < dump.sql
| r | b  | swpd | free | buff | cache | si | so | bi | bo | in | cs | us | sy | id | wa | st |
|---|----|------|------|------|-------|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 1  | 41024| 3108 | 828  | 69764 | 3  | 10 | 12292| 1723| 306 | 1613| 7  | 5  | 82 | 6  | 0  |
| 0 | 2  | 42280| 3704 | 644  | 69968 | 2  | 255| 5422 | 31890| 1132 | 1418| 63 | 14 | 0  | 22 | 0  |
| 1 | 2  | 48212| 10372| 1532 | 42024 | 87 | 1193| 12894| 23186| 1609 | 3254 | 55 | 27 | 0  | 18 | 0  |
| 3 | 1  | 49620| 6992 | 744  | 47788 | 0  | 282| 2782 | 12288| 1859 | 5983 | 57 | 43 | 0  | 0  | 0  |
| 2 | 1  | 53540| 20300| 512  | 55436 | 28 | 790 | 2918 | 29270| 1236 | 1517 | 74 | 16 | 0  | 10 | 0  |
| 2 | 4  | 53992| 28872| 344  | 46988 | 70 | 106 | 2297 | 28966| 1095 | 1386 | 71 | 11 | 0  | 18 | 0  |
| 1 | 6  | 53892| 4016 | 664  | 71856 | 46 | 0  | 2948 | 33727| 1119 | 1916 | 64 | 11 | 0  | 25 | 0  |
| 1 | 6  | 53860| 4020 | 588  | 71916 | 12 | 0  | 2262 | 30219| 1083 | 1830 | 63 | 11 | 0  | 26 | 0  |
| 2 | 4  | 54192| 3340 | 576  | 72936 | 10 | 70 | 3054 | 29504| 1127 | 1822 | 64 | 10 | 0  | 26 | 0  |
| 2 | 9  | 54664| 3324 | 648  | 73668 | 2  | 102| 3980 | 28035| 1187 | 2116 | 61 | 11 | 0  | 28 | 0  |
| 2 | 6  | 55016| 3980 | 440  | 73724 | 5  | 78 | 4618 | 26222| 1220 | 2149 | 64 | 10 | 0  | 26 | 0  |
| 1 | 8  | 55580| 5608 | 436  | 72176 | 3  | 119| 5265 | 27130| 1232 | 2318 | 59 | 11 | 0  | 30 | 0  |
| 2 | 7  | 56012| 3608 | 412  | 75092 | 3  | 88 | 5745 | 25135| 1237 | 2319 | 60 | 10 | 0  | 30 | 0  |
| 2 | 7  | 55968| 4084 | 440  | 74572 | 1  | 0  | 5637 | 27538| 1294 | 2468 | 58 | 12 | 0  | 30 | 0  |
| 2 | 9  | 55968| 3380 | 408  | 75352 | 3  | 6869| 23874| 1238 | 2418 | 53 | 11 | 0  | 36 | 0  |
| 2 | 6  | 55880| 3496 | 412  | 74808 | 64 | 0  | 6806 | 27942| 1312 | 2732 | 53 | 12 | 0  | 34 | 0  |
| 2 | 6  | 55852| 3440 | 408  | 74888 | 0  | 0  | 7103 | 27166| 1298 | 2619 | 54 | 11 | 0  | 35 | 0  |
| 3 | 6  | 55852| 3672 | 398  | 74648 | 0  | 3  | 6821 | 27295| 1304 | 2676 | 54 | 11 | 0  | 35 | 0  |
| 2 | 7  | 55824| 4052 | 372  | 74352 | 3  | 3  | 7539 | 23926| 1347 | 2592 | 56 | 11 | 0  | 33 | 0  |
| 1 | 9  | 55836| 3708 | 388  | 74652 | 4  | 7  | 8014 | 26310| 1331 | 2698 | 54 | 12 | 0  | 34 | 0  |
| 2 | 8  | 55820| 3524 | 348  | 74900 | 1  | 0  | 8590 | 24613| 1322 | 2712 | 54 | 11 | 0  | 35 | 0  |
| 4 | 6  | 38096| 3976 | 1640 | 51804 | 4033| 0  | 14980| 21471| 1730 | 3325 | 48 | 22 | 0  | 30 | 0  |
Percona Xtrabackup

- Completely free and open source
- Physical backup solution - Stores data in similar format as the original.

Pro’s
- Powerful, scriptable, feature-packed

Cons
- Multiple steps required to restore, complexity
Percona Xtrabackup - Features

- Non-blocking
- Support for MyISAM
- Compression
- Partial Backups
- Throttling
- Incremental Backups
- Exporting/Importing individual tables
- Streaming
- Parallel copying
- Compact Backups (since 2.1!)
- Encryption (since 2.1!)
- ...
Percona Xtrabackup - operations (1)

- Two separate "manual" steps are required for taking and storing the backup:
  - (1) - Backing up
  - (2) - Preparing InnoDB files for recovery
- Restoring it happens in a single step:
  - (3) - Restoring the files
Performing and restoring a mixed full backup:

- Backing up (1):
  - Copy datafiles while recording transaction log changes
  - FLUSH TABLES WITH READ LOCK;
  - Get binlog position
  - Copy all .MYD, .MYI, .TRG, .TRN, ... files
  - Stop recording transaction log changes
  - UNLOCK TABLES;

Preparing/Restoring:

- Prepare backup by applying recorded tlog changes (2)

Restoring files to original location (3)
Simple full backup and restore:

Install Xtrabackup

# yum install percona-xtrabackup-24
# mkdir /var/backup
(1)
# xtrabackup -proot --backup --target-dir=/var/backup/latest
(2)
# xtrabackup --prepare --target-dir=/var/backup/latest
(3)
# service mysql stop
# rm -rf /var/lib/mysql/*
# xtrabackup --copy-back --target-dir=/var/backup/latest
# chown -R mysql:mysql /var/lib/mysql
Understanding backup requirements

- Recovery Time Objective - How quickly must I restore?
- Recovery Point Objective - How much data can I lose?
- Risks - What failures should this data be protected against?
Advanced backup topics

• Incremental Backups
  – HandsOn!

• Delayed Slaves
  – Using `pt-slave-delay` to keep a slave behind on its master by a certain amount of time.
  – Good help against potential disastrous user mistakes
  – Example:
    
    # pt-slave-delay --delay 1m --interval 15s
Advanced backup topics - Handson

• Empty the previous backups:
  # rm -rf /var/backup/*

• Incremental Backup
  • Create a full backup
    # innobackupex --password=root /var/backup/
  • Create incremental backup
    # innobackupex --password=root --incremental
    /var/backup/inc/
    --incremental-basedir=/var/backup/ [backupfolder]
Advanced backups - handson

• Incremental Restore
  • Apply log
    # innobackupex --apply-log --redo-only /var/backup/[backupfolder]
  • Apply incremental backup to the full backup
    # innobackupex --apply-log --redo-only /var/backup/[backupfolder]
      --incremental-dir=/var/backup/inc/[incbackupfolder]
  • Finish preparing the backup
    # innobackupex --apply-log /var/backup/[backupfolder]
  • Restore as usual
Point in time recovery

- Point-In-Time recovery
  - Making use of the binary log
  - mysqlbinlog can be used to pipe commands into mysql
  - Use in combination with incremental backups to restore right up to the point where the disaster occurred.
Advanced backup topics

- Checking backup binlog position

[root@master latest]# cat xtrabackup_binlog_info
master-bin.0000013374603

- Checking binlog contents

[root@master mysql]# mysqlbinlog --start-position=243118 --stop-position=243300 node1-bin.000006

/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;
/*!40019 SET @@session.max_insert_delayed_threads=0*/;
/*!50003 SET @OLD_COMPLETION_TYPE=@@COMPLETION_TYPE,COMPLETION_TYPE=0*/;
[...]

- Applying contents to mysql

[root@master mysql]# mysqlbinlog --start-position=243118 --stop-position=243300 node1-bin.000006 | mysql
Advanced backup topics

• Managing & Retaining your backups
  – Storing offsite in case of disaster
  – Recovery testing
  – Retention policies
  – Monitoring Backup & Restore success rate
Operational DBA In A Nutshell

- Setting Up Today’s Environment
- Installation
- Logging In
- MySQL Privileges
- Diagnostics
- Troubleshooting
- Backups

- Replication
- Schema Changes
- Configuration Optimization
MySQL replication

• What is Replication?
• What are Binary Logs
• How should we be setting Up Replication?
• Commands
• Breaking & Fixing Replication
• Finding Inconsistencies
What is replication?

Replication enables data from one MySQL database server (the master) to be replicated to one or more MySQL database servers (the slaves)

What is Replication?

- Happens at MySQL level, not Storage Engine Level
- Asynchronous! (Semi-sync available in 5.5)
- A server can have multiple masters (Since MySQL 5.7)
- IO Thread: Fetches from master
- SQL Thread: Executes on slave
- Single Threaded Execution (Until 5.6)
Example Asynchronous replication
Asynchronous replication, how?
Semisynchronous replication
Semisync, how?
Other types of "replication"

- Semi-synchronous replication
- Galera replication (MySQL with Galera, PXC, MariaDB Cluster)
- MySQL Cluster
- MySQL group replication
- MySQL InnoDB cluster
What is HA

Reliability engineering
  Elimination of single points of failure
  Reliable failover/crossover
  Detection of failures whenever failure occurs

Scheduled and unscheduled downtime

Closely related concepts
  ETR
  RTO
  MTTR
  Data availability
In the past HA was ancient tech based :D

High availability was typically more limited

DRBD – Block device syncing
Contains: DRBD, Pacemaker, Corosync
DRBD based replication
Current tech: Group replication
Current Tech: Galera replication
Binary Logging
What are binary logs

- Set of files
- Contains all writes and schema changes
- Incrementing numbers (000001, 000002, 000003,...)
- 2 Formats:
  - Statement Based (SBR)
  - Row Based (RBR, since MySQL 5.1)
  - Prior to MySQL 5.7.7, statement-based logging format was the default. In MySQL 5.7.7 and later, row-based logging format is the default.
- Binary files (extract with MySQLbinlog)
What are binary logs

- Set of files
- Contains all writes and schema changes
- Incrementing numbers (000001, 000002, 000003, ...)
- 2 Formats:
  - Statement Based (SBR)
  - Row Based (RBR, since MySQL 5.1)
  - Prior to MySQL 5.7.7, statement-based logging format was the default. In MySQL 5.7.7 and later, row-based logging format is the default.
- Binary files (extract with MySQLbinlog)
Binary log - formats (binlog_format)

• Statement Based Replication (SBR):
  – Writes statements to binary logs, slave executes the statement
    • eg: DELETE FROM wp_options WHERE option_name = '_transient_doing_cron'
Binary log - formats (binlog_format)

• Row Based Replication (RBR, since 5.1):
  – Write row changes (larger binlogs)
  – Does not need to parse/execute queries, just make the changes necessary
  – Mixed: Combination of both: defaults to SBR, use RBR when necessary
Looking to the binary log

• mysqlbinlog
• SHOW BINLOG EVENTS
Example SBR (1)

```sql
> SHOW GLOBAL VARIABLES LIKE 'binlog_format';
+---------------+-----------+
| Variable_name | Value     |
+---------------+-----------+
| binlog_format | ROW       |
+---------------+-----------+
1 row in set (0.00 sec)

> CREATE DATABASE replication;
Query OK, 1 row affected (0.14 sec)

> use replication
Database changed

> CREATE TABLE repl (a int) ENGINE=innodb;
Query OK, 0 rows affected (0.25 sec)

> INSERT INTO repl VALUES (1);
Query OK, 1 row affected (0.14 sec)
```
Example SBR (2)

```sql
# mysqlbinlog mysql-bin.000193
...
# at 106
#120106 15:19:13 server id 9999  end_log_pos 203  Query  thread_id=11
CREATE DATABASE replication
/*!*/;
# at 203
#120106 15:19:32 server id 9999  end_log_pos 312  Query  thread_id=11  exec_time=1  error_code=0
use replication/*!*/;
SET TIMESTAMP=1325859572/*!*/;
CREATE TABLE repl (a INT) ENGINE=innodb
/*!*/;
# at 312
#120106 15:19:55 server id 9999  end_log_pos 387  Query  thread_id=11  exec_time=0  error_code=0
SET TIMESTAMP=1325859595/*!*/;
BEGIN
/*!*/;
# at 387
#120106 15:19:55 server id 9999  end_log_pos 484  Query  thread_id=11  exec_time=0  error_code=0
SET TIMESTAMP=1325859595/*!*/;
INSERT INTO repl VALUES (1)
/*!*/;
# at 484
#120106 15:19:55 server id 9999  end_log_pos 511  Xid = 14
COMMIT/*!*/;
```
Example SBR (2)

```sql
# mysqlbinlog mysql-bin.000193
...  
# at 106
#120106 15:19:13 server id 9999  end_log_pos 203  Query  thread_id=11
CREATE DATABASE replication
/*!*/;
# at 203
#120106 15:19:32 server id 9999  end_log_pos 312  Query  thread_id=11 exec_time=1 error_code=0
use replication//*!*/;
SET TIMESTAMP=1325859572//*!*/;
CREATE TABLE repl (a INT) ENGINE=innodb
/*!*/;
# at 312
#120106 15:19:55 server id 9999  end_log_pos 387  Query  thread_id=11 exec_time=0 error_code=0
SET TIMESTAMP=1325859595//*!*/;
BEGIN
/*!*/;
# at 387
#120106 15:19:55 server id 9999  end_log_pos 484  Query  thread_id=11 exec_time=0 error_code=0
SET TIMESTAMP=1325859595//*!*/;
INSERT INTO repl VALUES (1)
/*!*/;
# at 484
#120106 15:19:55 server id 9999  end_log_pos 511  Xid = 14
COMMIT//*!*/;
```
Example SBR inside mysql

> SHOW BINLOG EVENTS FROM 106\G

*************************** 1. row ***************************
  Log_name: mysql-bin.000193
  Pos: 106
  Event_type: Query
  Server_id: 1
End_log_pos: 203
  Info: CREATE DATABASE replication

*************************** 2. row ***************************
  Log_name: mysql-bin.000193
  Pos: 203
  Event_type: Query
  Server_id: 1
End_log_pos: 312
  Info: use `replication`; CREATE TABLE repl (a INT) ENGINE=InnoDB

*************************** 3. row ***************************
  Log_name: mysql-bin.000193
  Pos: 312
  Event_type: Query
  Server_id: 1
End_log_pos: 387
  Info: BEGIN

*************************** 4. row ***************************
  Log_name: mysql-bin.000193
  Pos: 387
  Event_type: Query
  Server_id: 1
End_log_pos: 484
  Info: use `replication`; INSERT INTO repl VALUES (1)

*************************** 5. row ***************************
  Log_name: mysql-bin.000193
  Pos: 484
  Event_type: Xid
  Server_id: 1
End_log_pos: 511
  Info: COMMIT

5 rows in set (0.00 sec)
Example SBR inside mysql

SHOW BINLOG EVENTS FROM 106\G

*************************** 1. row ***************************

  Log_name: mysql-bin.000193
    Pos: 106
  Event_type: Query
  Server_id: 1
End_log_pos: 203

  Info: CREATE DATABASE replication

*************************** 2. row ***************************

  Log_name: mysql-bin.000193
    Pos: 203
  Event_type: Query
  Server_id: 1
End_log_pos: 312

  Info: use `replication`; CREATE TABLE repl (a INT) ENGINE=innodb

*************************** 3. row ***************************

  Log_name: mysql-bin.000193
    Pos: 312

  Info: use `replication`; INSERT INTO repl VALUES (1)

5 rows in set (0.00 sec)
Example RBR (1)

```sql
> SHOW GLOBAL VARIABLES LIKE 'binlog_format';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| binlog_format | ROW   |
+---------------+-------+
1 row in set (0.00 sec)

> CREATE DATABASE replication;
Query OK, 1 row affected (0.14 sec)

> use replication
Database changed

> CREATE TABLE repl (a int) ENGINE=innodb;
Query OK, 0 rows affected (0.25 sec)

> INSERT INTO repl VALUES (1);
Query OK, 1 row affected (0.14 sec)
```
Example RBR (2)

```sql
# mysqlbinlog mysql-bin.000193 --start-position=606
... 
# at 606
#120106 15:54:54 server id 1  end_log_pos 703  Query  thread_id=11 exec_time=0  error_code=0
CREATE DATABASE replication  /*!*!/;
# at 703
#120106 15:55:02 server id 1  end_log_pos 812  Query  thread_id=11 exec_time=0  error_code=0
use replication/*!*!/;
SET TIMESTAMP=1325861702/*!*!/;
CREATE TABLE repl (a int) ENGINE=innodb  /*!*!/;
# at 812
...
# at 937
#120106 15:55:06 server id 1  end_log_pos 937  Table_map: `replication`.`repl` mapped to number 17
#120106 15:55:06 server id 1  end_log_pos 971  Write_rows: table id 17 flags: STMT_END_F
```

BINLOG ' SgsHTxMBAAAAAMgAAAKkDAAAAABEAAAAAAEAC3JlcGxpY2F0aW9uAARyZXVBsAAEDAAE= SgsHTxcBAAAAIgAAAMsDAAAAABEAAAAAAEAAf/+AQAAAA==' /*!*!/;
# at 971
Example RBR (3)

```sql
# mysqlbinlog mysql-bin.000193 --verbose --verbose
...
# at 937
#120106 15:55:06 server id 1  end_log_pos 937  Table_map: `replication`.`repl` mapped to number 17
#120106 15:55:06 server id 1  end_log_pos 971  Write_rows: table id 17 flags: STMT_END_F

BINLOG '
SgsHTxMBAAAAAMgAAAkKDAAAAAABEEAAAAAEAC3JlcGxpY2F0aW9uAARyZXBsAAEDAAE=
SgsHTxcBAAAAIgAAAMsDAAAAAABEEAAAAAEEAf/+AQAAAA==
'/*!**/;
### INSERT INTO replication.repl
### SET
### @1=1 /* INT meta=0 nullable=1 is_null=0 */
# at 971
```
Example RBR (3)

```sql
# mysqlbinlog mysql-bin.000193 --verbose --verbose
...
# at 937
#120106 15:55:06 server id 1  end_log_pos 937  Table_map: `replication`.`repl` mapped to number 17
#120106 15:55:06 server id 1  end_log_pos 971  Write_rows: table id 17 flags: STMT_END_F

BINLOG ' SgsHTxMBAAAMgAAAKkDAAAAABEAAAAAASEAC3JlcGxpY2F0aW9uY29vZ2_mode=91gAAAGkDAAAE
SgsHTxcBAAAAIAgAAAMsDAAAAABEAAAAAEEAf/+AQAAAA=
'/*!*'/
### INSERT INTO replication.repl
### SET
###   @1=1 /* INT meta=0 nullable=1 is_null=0 */
# at 971
Example RBR (4)

```sql
> SHOW BINLOG EVENTS FROM 606\G
*************************** 1. row ***************************
  Log_name: mysql-bin.000193
  Pos: 606
  Event_type: Query
  Server_id: 1
  End_log_pos: 703
  Info: CREATE DATABASE replication
*************************** 2. row ***************************
  Log_name: mysql-bin.000193
  Pos: 703
  Event_type: Query
  Server_id: 1
  End_log_pos: 812
  Info: use `replication`; CREATE TABLE repl (a int) ENGINE=innodb
...
*************************** 4. row ***************************
  Log_name: mysql-bin.000193
  Pos: 887
  Event_type: Table_map
  Server_id: 1
  End_log_pos: 937
  Info: table_id: 17 (replication.repl)
```
Setting up replication

• Install the slave
• Change master/slave MySQL configuration
• Prerequisites
• Configure Replication
• Start Replication/Check Status
### Example RBR (4)

```
> SHOW BINLOG EVENTS FROM 606\G

*************************** 1. row ***************************
  Log_name: mysql-bin.000193
  Pos: 606
  Event_type: Query
  Server_id: 1
  End_log_pos: 703
  Info: CREATE DATABASE replication

*************************** 2. row ***************************
  Log_name: mysql-bin.000193
  Pos: 703
  Event_type: Query
  Server_id: 1
  End_log_pos: 812
  Info: use `replication`; CREATE TABLE repl (a int) ENGINE=innodb

...  

*************************** 4. row ***************************
  Log_name: mysql-bin.000193
  Pos: 887
  Event_type: Table_map
  Server_id: 1
  End_log_pos: 937
  Info: table_id: 17 (replication.repl)

*************************** 5. row ***************************
```
Change master/slave configuration

- On the master, make the following changes to /etc/percona-server.conf.d/mysqld.cnf:
  - `server-id=<unique number>`
  - `log-slave-updates`
  - `log-bin=log-bin`
- After making the changes, restart the master
  ```
  [root@master ~]# service mysql restart
  ```
- For future steps, drop the IMDB database
  ```
  master mysql> DROP DATABASE IMDB;
  ```
Why server-id?

- Avoid events to be written more than once
- Identifier for your system to keep track of replication
Prerequisites (creating a backup) - Hands-on

- Please perform the following steps on the master

```
[root@master ~]# rm -rf /var/backup/latest/*

[root@master ~]# xtrabackup -p root --backup
   --target-dir=/var/backup/latest/

[root@master ~]# xtrabackup -p root --prepare
   --target-dir=/var/backup/latest/

[root@master ~]# ln -s /var/backup/latest /var/backup/last_full
```
Creation of the slave (2) - Handson

- No way to easily create slave with 1 command
- It’s required to Create/Restore **consistent** backup

```
root@slave# yum install Percona-Server-server-57
root@slave# rm -rf /var/lib/mysql/*
root@master# chown -R centos /var/backup/latest
root@slave# scp -i /home/centos/.ssh/id_rsa -r centos@<privateip>:/var/backup/last_full/* /var/lib/mysql/
root@slave# chown -R mysql:mysql /var/lib/mysql/
root@slave# service mysql start
root@slave# vi ~/.my.cnf
```

```
[mysql]
user=root
password=root
```
Creation of the slave (3)

- Add to file on slave: /etc/percona-server.conf.d/mysqld.cnf
  server-id=102

# service mysql restart
Creation of the slave (4)

- On Master, add permissions:

  ```bash
  master mysql> GRANT REPLICATION SLAVE ON *.* TO "repl"@"%" IDENTIFIED BY "slaveme";
  ```

- Get Binlog file/Position from backup on the slave:

  ```bash
  root@slave# cat /var/lib/mysql/xtrabackup_binlog_info
  ```

  Example output
  
<table>
<thead>
<tr>
<th>Filename</th>
<th>log_position</th>
</tr>
</thead>
<tbody>
<tr>
<td>log-bin.000001</td>
<td>311</td>
</tr>
</tbody>
</table>

- On Slave, configure replication:

  ```bash
  slave mysql> CHANGE MASTER TO MASTER_HOST="<privateIP MASTER", MASTER_USER="repl",
          MASTER_PASSWORD="slaveme", MASTER_LOG_FILE="BINLOGFILENAME", MASTER_LOG_POS=[position];
  ```

  ```bash
  slave mysql> start slave;
  ```
Creation of the slave (5)

```
mysql> show master status \G
*************************** 1. row ***************************
    File: log-bin.000001
    Position: 246938
  Binlog_Do_DB:
  Binlog_Ignore_DB:
Executed_Gtid_Set:
1 row in set (0.00 sec)
```
Creation of the slave (6)

```
slave mysql> show slave status
```

```
+-------------------+-------------------+
|       Field        | Value             |
+-------------------+-------------------+
| Slave_IO_State    | Waiting for master to send event |
| Master_Host       | 192.168.70.1      |
| Master_User       | repl              |
| Master_Port       | 3306              |
| Connect_Retry     | 60                |
| Master_Log_File   | log-bin.000001    |
| Read_Master_Log_Pos| 246938            |
| Relay_Log_File    | slave-relay-bin.000003 |
| Relay_Log_Pos     | 246445            |
| Relay_Master_Log_File| log-bin.000001 |
| Slave_IO_Running | Yes               |
| Slave_SQL_Running| Yes               |
| Replicate_Do_DB   |                  |
| Replicate_Ignore_DB|                |
| Replicate_Do_Table|                  |
| Replicate_Ignore_Table|            |
| Replicate_Wild_Do_Table|            |
| Replicate_Wild_Ignore_Table|      |
| Last_Errno        | 0                 |
| Last_Error        |                  |
| Skip_Counter      | 0                 |
| Exec_Master_Log_Pos| 246938            |
| Relay_Log_Space   | 246316            |
| Until_Condition   | None              |
| Until_Log_File    |                  |
| Until_Log_Pos     | 0                 |
| Master_SSL_Allowed| No                |
| Master_SSL_CA_File|                  |
| Master_SSL_CA_Path|                  |
| Master_SSL_Cert   |                  |
| Master_SSL_Cipher |                  |
| Master_SSL_Crl    |                  |
| Master_SSL_Crlpath|                  |
| Master_UUID       | 4ad3bd95-210d-11e7-8ee8-08002789cd2e |
| Master_Info_File  | /var/lib/mysql/master.info |
| SQL_Delay         | 0                 |
| SQL_Remaining_Delay| NULL             |
| Slave_SQL_Running_State| Slave has read all relay log; waiting for more updates |
| Master_Retry_Count| 86400             |
| Master_Bind       |                  |
| Last_IO_Error_Timestamp|     |
| Last_SQL_Error_Timestamp|     |
| Master_SSL_Verify_Server_Cert| No     |
| Last_IO_Errno     | 0                 |
| Last_SQL_Errno    | 0                 |
| Skip_Counter      | 0                 |
| Retrieved_Gtid_Set| 9854d69b-49fb-11e3-a549-080027771fca:1-2 |
| Executed_Gtid_Set | 9854d69b-49fb-11e3-a549-080027771fca:1-2, fc945001-4a34-11e3-a6c0-08002736e589:1 |
| Auto_Positon      | 1                 |
```
Replication - Administrative commands

- Rotate binary log: `FLUSH BINARY LOGS`
- Rotate relay log: `FLUSH RELAY LOGS`
- Remove binary logs:
  - `PURGE MASTER LOGS TO 'mysql-bin.000005'`;
  - `PURGE MASTER LOGS`
  - `BEFORE '2012-01-01 00:00:00'`;
- Remove all binary logs: `RESET MASTER`
- Remove slave configuration and files: `RESET SLAVE`
Replication - Diagnostic commands

• On Master
  – SHOW MASTER STATUS
  – SHOW PROCESSLIST
  – SHOW SLAVE HOSTS
  – SHOW BINLOG EVENTS

• On Slave
  – SHOW SLAVE STATUS
  – SHOW PROCESSLIST
Replication - Master - Replication status

- Current binary log file and position:

```sql
mysql> show master status \G
File: log-bin.000001
Position: 246938
Binlog_Do_DB:
Binlog_Ignore_DB:
Executed_Gtid_Set:
1 row in set (0.00 sec)
```
Find Connected Slaves using SHOW PROCESSLIST:

```
master> SHOW PROCESSLIST\G
*************************** 1. row ***************************
    Id: 1171
  User: repl
    Host: 192.168.70.2:48590
    db: NULL
Command: Binlog Dump
          Time: 3720
     State: Master has sent all binlog to slave; waiting for more updates
          Info: NULL
      Rows_sent: 0
      Rows_examined: 0
```
Replication - Slave - Show processlist

• Slave Thread Status:

```sql
mysql> show processlist \G
*************************** 1. row ***************************
    Id: 6
    User: system user
    Host: 
    db: NULL
    Command: Connect
    Time: 3782
    State: Waiting for master to send event
    Info: NULL
    Rows_sent: 0
    Rows_examined: 0

*************************** 2. row ***************************
    Id: 7
    User: system user
    Host: 
    db: NULL
    Command: Connect
    Time: 3317
    State: Slave has read all relay log; waiting for more updates
    Info: NULL
    Rows_sent: 0
    Rows_examined: 0
```
Other common configuration options

- Don’t start replication at start: `skip_slave_start`
- Put relay log events in it’s own binary log: `log_slave_updates`
- Disallow writing on slaves: `read_only`
- Automatically remove binlogs: `expire_logs_days`
- Change binlog format: `binlog_format`
  - STANDARD ROW in 5.7
Breaking and fixing replication - handson

- Create table on Slave First, then Master:

```sql
CREATE TABLE sakila.ple
(
  id int auto_increment primary key,
  name varchar(20)
);
```

- Check Slave Status:

```
slave mysql> SHOW SLAVE STATUS\G
```

- Replication broke :(
Breaking and fixing replication

• Replication is Broken.
• Here we understand what went wrong, so let’s just skip this record! (ONLY DO THIS WHEN YOU ARE SURE)
Detecting inconsistencies

• Let’s delete some data from the slave:
  ```
  slave mysql> DELETE FROM sakila.film_actor LIMIT 5;
  ```

• On the Master:
  ```
  master mysql> grant all privileges on *.* to 'pttc'@'%' identified by 'password';
  ```

• After Checksumming, let’s verify:
  ```
  # pt-table-checksum h=<privateip>,u=pttc,p=password --no-check-binlog-format
  ```
Operational DBA In A Nutshell

- Setting Up Today’s Environment
- Installation
- Logging In
- MySQL Privileges
- Diagnostics
- Troubleshooting
- Backups

- Replication
- Schema Changes
- Configuration Optimization
Schema changes - Techniques

• Why?
  – Achieving schema perfection can be painful
  – Performance optimizations
  – Migrations
  – …

• MySQL 5.5 vs 5.6 vs 5.7 vs 8
  – MySQL 5.6/5.7/8 supports online schema changes
  – Prior to MySQL 5.6 use pt-online-schema-change
Schema changes - Techniques

• Connect to your MySQL instance
• Open second terminal

node1> insert into sakila.actor
(first_name, last_name) values
('LeT','watDim0ooootjeen');

• alter table on your system if you want the changes locally

# node1> ALTER TABLE sakila.actor ADD INDEX new_idx
(first_name, last_name);
Schema changes in 5.5

- In 5.5, use `pt-online-schema-change` if you want to replicate changes

```bash
# time pt-online-schema-change \
--execute --alter \
"DROP INDEX new_idx" \
D=sakila,t=actor
```
Schema changes in 5.7

New in MySQL 5.6 - sweet hot alter tables

mysql> ALTER TABLE sakila.actor DROP INDEX new_idx, LOCK=NONE;
Query OK, 0 rows affected (0.02 sec)

• However:
Schema changes in 5.7

New in MySQL 5.6 - sweet hot alter tables

```sql
mysql> ALTER TABLE sakila.actor DROP INDEX new_idx, LOCK=NONE;
Query OK, 0 rows affected (0.02 sec)
```

• However:

Operational DBA In A Nutshell

- Setting Up Today’s Environment
- Installation
- Logging In
- MySQL Privileges
- Diagnostics
- Troubleshooting
- Backups

- Replication
- Schema Changes
- Configuration Optimization
Configuration optimisation

- Get variables
  ```
  mysql> show global variables like 'innodb_buffer_pool_size';
  mysql> select @@innodb_buffer_pool_size;
  mysql> show global variables like 'innodb_%';
  ```

- Set
  ```
  mysql> select @@table_open_cache;
  mysql> set global table_open_cache = 3000;
  ```

- Make the changes persistent
  - my.cnf
Configuration optimisations - to optimise

- `innodb_buffer_pool_size = 1G` # (adjust value here, 50%-70% of total RAM)
- `innodb_log_file_size = 128M – 2G` (does not need to be larger than buffer pool) (default is 48 MB)
- `innodb_flush_log_at_trx_commit`
- `sync_binlog` (default is 1)
- `innodb_flush_method = O_DIRECT` (avoid double buffering) (default: `fsync`)
- OS: Swappiness
- OS: Disk scheduler: cfq vs deadline/noop
Configuration optimisations - not optimise

- `innodb_io_capacity, innodb_io_capacity_max`: don’t oversize
- `sort_buffer_size, join_buffer_size, read_buffer_size, read_rnd_buffer_size`: leave default globally
- `innodb_thread_concurrency`: Default is fine for most environments
- `query_cache_type, query_cache_size`: disable or do not oversize (+64MB)
- `innodb_flush_method=O_DIRECT` and ext4
Optimisation process

Low Hanging Fruit
- MySQL Configuration
- OS settings
- Indexes
- Caching

Medium Level
- InfraStructure
- Minor MySQL Version

Hard Changes
* Major schema changes
* Major MySQL Versions
* Application architecture changes
Want to learn more?

• Some good resources to help you along:
  – Blogs
    • Planet MySQL: planet.mysql.com
    • MySQL Performance Blog: www.mysqlperformanceblog.com
  – Recorded Webinars
    • www.percona.tv
That is all folks!
Thank You to Our Sponsors
Rate My Session

**Details**

Introducing gh-ost: triggerless, painless, trusted online schema migrations

- Time: 11:20 → 12:10
- Location: Matterhorn 2

**Rate & Review**

Tap a star to rate

Feedback (optional)

Submit
Servers

PerconaLive-1-Master  PerconaLive-13-Master
PerconaLive-2-Master  PerconaLive-14-Master
PerconaLive-3-Master  PerconaLive-15-Master
PerconaLive-4-Master  PerconaLive-16-Master
PerconaLive-5-Master  PerconaLive-17-Master
PerconaLive-6-Master  PerconaLive-18-Master
PerconaLive-7-Master  PerconaLive-19-Master
PerconaLive-8-Master  PerconaLive-20-Master
PerconaLive-9-Master
PerconaLive-9-Slave
PerconaLive-10-Master
PerconaLive-11-Master
PerconaLive-12-Master