Infrastructure review

AboutMe

- Miklos ‘Mukka’ Szel
- From Hungary
- Started using Linux in 1997 and MySQL in 2000, used to be a backend developer
- Worked for Own Startup, ISP, Walt Disney(ESPN,Cricinfo), PalominoDB/Pythian
- Freelance Consultant since 2015 - Edmodo

Overview

- Architecture
  - Topology
  - Config
- Monitoring
  - Queries
- Backup/restore
- Schema
- +1

Architecture

Inventory – finding the servers

- Configuration files
- Monitoring
- Check the process list on every server
- Scan for port

```
nmap -p 3306 –sV 10.77.3.0/24
Nmap scan report for 10.77.3.238
Host is up (0.00087s latency).
PORT     STATE SERVICE VERSION
3306/tcp open  mysql   MySQL 5.6.27-75.0-log
```

Inventory – finding the servers(cont'd)

```
./pt-slave-find --host 10.77.3.181 -u root--ask-pass

Enter password:
10.77.24.181
Version         5.6.32-78.1-log
Server ID       134
Uptime          318+06:01:55 (started 2015-11-19T02:42:03)
Replication     Is not a slave, has 1 slaves connected, is not read_only
Filters
Binary logging  MIXED
Slave status
Slave mode      STRICT
InnoDB version  5.6.32-78.1

+- 10.77.25.17
   Version         5.6.27-75.0-log
   Server ID       143
   Uptime          167+18:21:03 (started 2016-04-17T14:22:55)
   Replication     Is a slave, has 0 slaves connected, is read_only
   Filters
   Binary logging  MIXED
   Slave status    0 seconds behind, running, no errors
   Slave mode      STRICT
   Auto-increment  increment 2, offset 1
   InnoDB version  5.6.27-75.0
```

Inventory – Server version mismatch

```
 - name: Install MySQL-Oracle Server
     yum: pkg={{ item }} state=present disablerepo=*
enablerepo=edmodo
     with_items:
         - MySQL-server
         - MySQL-client
```

```
yum install MySQL-server
apt-get install percona-server-server-5.6
```

Inventory – finding the servers(cont'd)

pt-summary

- OS related info
  - Disks, IO usage, processes, open ports, CPU, OS version, kernel, uptime

pt-mysql-summary

  - storage engine distribution, config values, replication, users, innodb

Config diff

- Between config file and runtime variables
- Between servers in the same replication chain

```
./pt-config-diff /etc/my.cnf h=localhost
3 config differences
Variable /etc/my.cnf c1-adhoc4-i.us-west2
=========================================
innodb_thread_concurrency   0        4
wait_timeout                600      3600
read_only                   ON       OFF
```

https://www.percona.com/doc/percona-toolkit/2.2/pt-config-diff.html

Inventory – finding the servers

Name	IP	AZ	Instance Type	DATA EBS	monthly cost USD	RAM	MySQL Version	OS
db1_m	10.77.2.2	us-west-2b	r3.2xlarge	1000	485+100	61	Percona Server 5.6.32	Ubuntu 14.04 LTS
db1_s1	10.77.2.66	us-west-2c	r3.xlarge	1000	243+100	30.5	Percona Server 5.6.27	Ubuntu 14.04 LTS
db1_s2	10.77.2.8	us-west-2a	r3.xlarge	1000	243+100	30.5	Percona Server 5.6.28	Ubuntu 14.04 LTS
db1_bck	10.77.2.212	us-west-2a	r3.xlarge	2x1000	243+200	30.5	Percona Server 5.6.28	Ubuntu 14.04 LTS
db2_m	10.77.3.23	us-west-2b	r3.4xlarge	3000	973+300	122	Percona Server 5.6.28	Ubuntu 16.04 LTS
db2_s1	10.77.3.26	us-west-2b	r3.4xlarge	3000	973+300	122	Percona Server 5.6.27	Ubuntu 16.04 LTS
db2_bck	10.77.3.92	us-west-2b	r3.4xlarge	2x3000	973+600	122	Percona Server 5.6.28	Ubuntu 16.04 LTS
db_dwh	10.77.3.123	us-west-2c	i2.2xlarge	ephemeral	1248	61	Oracle MySQL 5.5	Ubuntu 16.04 LTS

Inventory

type	# of instances	monthly price	sum/month	USD/year	Total Cost/Year	Saving %	Total old cost/Year[USD]	254664	
cluster1							Total new cost/Year[USD]	146076	**57.36%**
i2.2xlarge	13	1248	16224	**194688**	**194688**				
r3.4xlarge	7	973	6811	**81732**	**92652**	47.59%			
1200 GB Generic SSD	7	130	910	**10920**					
cluster2									
i2.xlarge	3	624	1872	**22464**	**22464**				
r3.2xlarge	3	486	1458	**17496**	**22176**	98.72%			
1200 GB Generic SSD	3	130	390	**4680**					
cluster3									
i2.xlarge	3	624	1872	**22464**	**22464**				
r3.2xlarge	3	486	1458	**17496**	**22176**	98.72%			
1200 GB Generic SSD	3	130	390	**4680**					
cluster4									
r3.xlarge	2	512	1024	**12288**	**12288**				
r3.large	2	256	512	**6144**	**6552**	53.32%			
200 GB Generic SSD	2	17	34	**408**					
cluster5									
m3.large	2	98	196	**2352**	**2760**				
m4.large	2	88	176	**2112**	**2520**	91.30%			
200 GB Generic SSD	2	17	34	**408**					

my.cnf – server_id

- Don’t use methods like removing dots from ip addresses
  - *10.77.12.3 (1077123)*
  - *10.77.1.23 (1077123)*

  - *MySQL truncates the server_id if it’s bigger than the max value for an integer(4294967295)*
  - *192.168.111.222 (192168111222 -> 4294967295)*
  - *192.168.222.222 (192168222222 -> 4294967295)*

  - *4508354421957495439 -> 4294967295*
  - *12354356476576 -> 4294967295*

OS

- Disable atime on the data partition
- Disable swap
- Allocate most of the RAM to MySQL
  - This will make it the perfect candidate to kill for the OOM-killer
- If your data is on EBS use the ephemaral that comes with the instance as tmp
  - DDLs on big tables can consume it!

```
[mysqld]
open_files_limit = 10000

/etc/security/limits.conf:
*           hard    nofile  10000
*           soft    nofile  10000
```

User audit

- Remove users with empty password
- GRANT ALL PERMISSION comes with SUPER which can write on instances with read_only = ON
  - You can use super-read-only in Percona Server >5.6
    - https://www.percona.com/doc/percona-server/5.6/management/super_read_only.html
- USE max_user_connections, this will prevent a single app from using all threads(max_connections)
- Wait_timeout 28800 by default

```
gdb -p $pid  -ex "set  max_connections=3000 --batch
```

HA

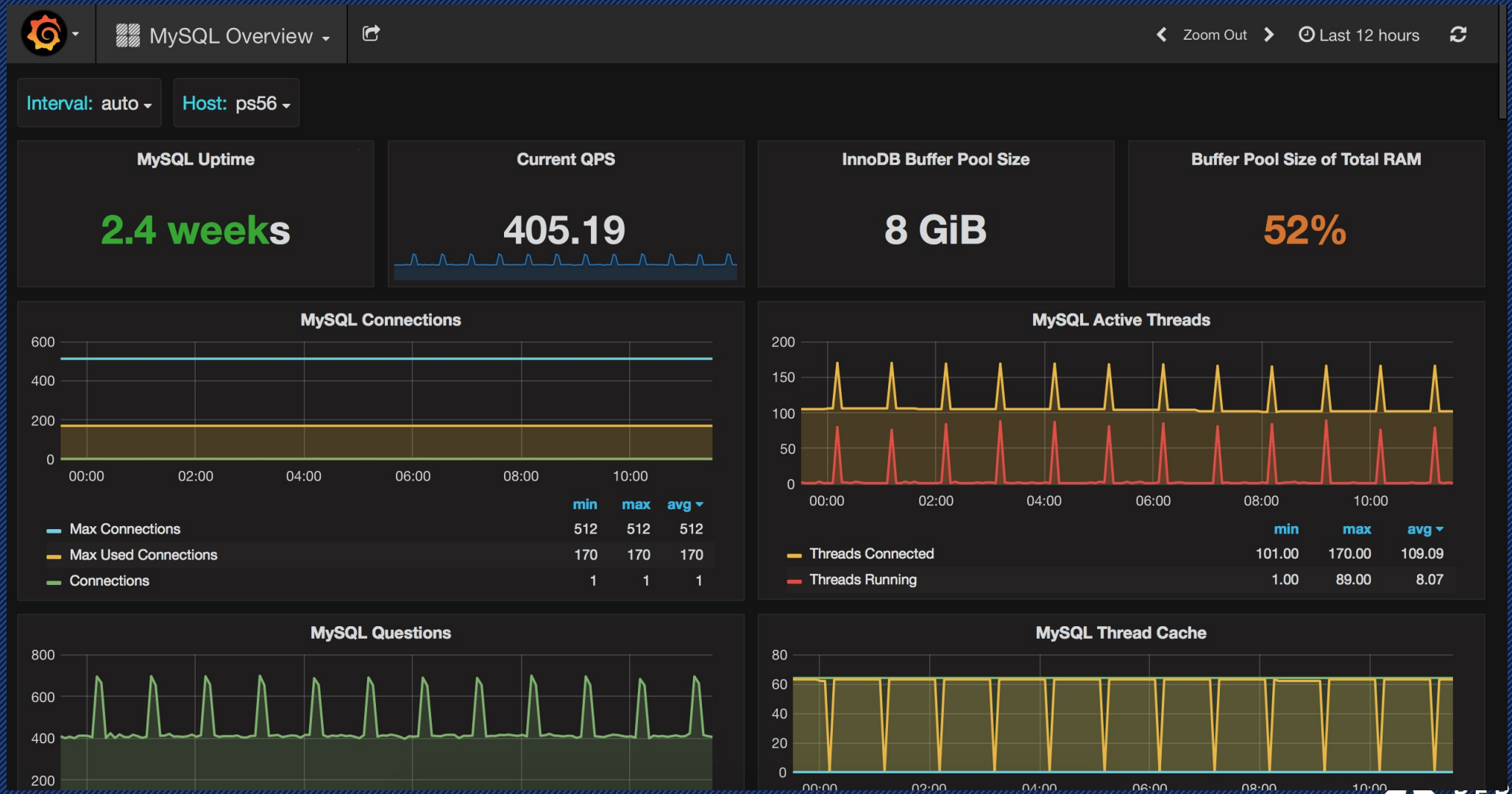
- Galera
- HaProxy
- DRBD
- Orchestrator
- Mysqlfailover
- MHA (with ProxySQL)
  - Seamless falilover without losing a single read

Monitoring

Monitoring

- Trend monitoring
  - Cacti
  - Graphite/Grafana
  - PMM - Percona Monitoring and Management
- State monitoring
  - Nagios/Icinga
  - Sensu

Monitoring


MySQL Overview
Zoom Out
Last 12 hours
Interval: auto
Host: ps56
MySQL Uptime
2.4 weeks
Current QPS
405.19
InnoDB Buffer Pool Size
8 GiB
Buffer Pool Size of Total RAM
52%
MySQL Connections
600
400
200
0
00:00
02:00
04:00
06:00
08:00
10:00
min max avg
Max Connections 512 512 512
Max Used Connections 170 170 170
Connections 1 1 1
MySQL Active Threads
200
150
100
50
0
00:00
02:00
04:00
06:00
08:00
10:00
min max avg
Threads Connected 101.00 170.00 109.09
Threads Running 1.00 89.00 8.07
MySQL Questions
800
600
400
200
MySQL Thread Cache
80
60
40
20
0

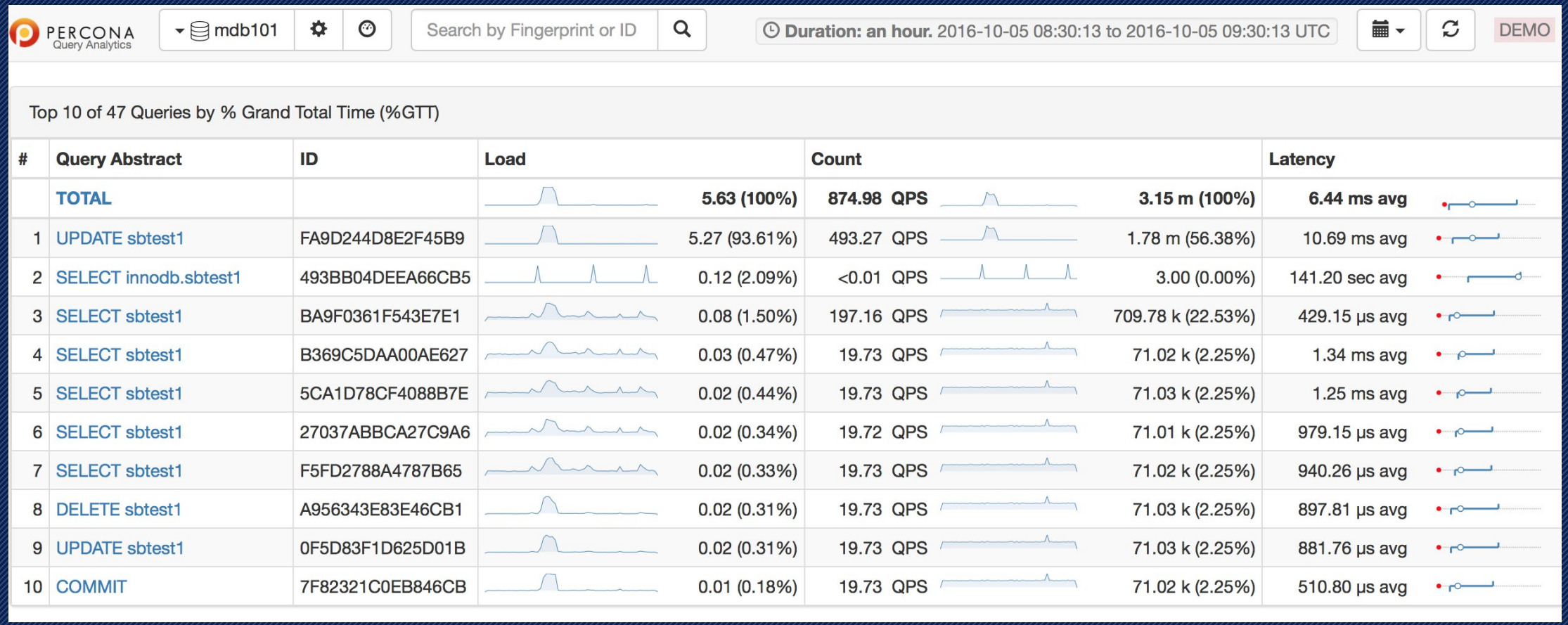
Queries

- pt-query-digest
- Anemometer
- Percona Monitoring and Management Query Analyzer
- Vividcortex

“1 pageload = 1 bad query” can be deadly

- pt-kill
- ProxySQL
- Memcached

Queries



Top 10 of 47 Queries by % Grand Total Time (%GTT)

#	Query Abstract	ID	Load	Count	Latency
	TOTAL		**5.63 (100%)**	**874.98 QPS**	**3.15 m (100%)**	**6.44 ms avg**
1	UPDATE sbtest1	FA9D244D8E2F45B9	5.27 (93.61%)	493.27 QPS	1.78 m (56.38%)	10.69 ms avg
2	SELECT innodb.sbtest1	493BB04DEEA66CB5	0.12 (2.09%)	<0.01 QPS	3.00 (0.00%)	141.20 sec avg
3	SELECT sbtest1	BA9F0361F543E7E1	0.08 (1.50%)	197.16 QPS	709.78 k (22.53%)	429.15 µs avg
4	SELECT sbtest1	B369C5DAA00AE627	0.03 (0.47%)	19.73 QPS	71.02 k (2.25%)	1.34 ms avg
5	SELECT sbtest1	5CA1D78CF4088B7E	0.02 (0.44%)	19.73 QPS	71.03 k (2.25%)	1.25 ms avg
6	SELECT sbtest1	27037ABBCA27C9A6	0.02 (0.34%)	19.72 QPS	71.01 k (2.25%)	979.15 µs avg
7	SELECT sbtest1	F5FD2788A4787B65	0.02 (0.33%)	19.73 QPS	71.02 k (2.25%)	940.26 µs avg
8	DELETE sbtest1	A956343E83E46CB1	0.02 (0.31%)	19.73 QPS	71.03 k (2.25%)	897.81 µs avg
9	UPDATE sbtest1	0F5D83F1D625D01B	0.02 (0.31%)	19.73 QPS	71.03 k (2.25%)	881.76 µs avg
10	COMMIT	7F82321C0EB846CB	0.01 (0.18%)	19.73 QPS	71.02 k (2.25%)	510.80 µs avg

BACKUP/RESTORE

BACKUP

Logical
- mysqldump
- mydumper

Cold
- stop mysql, archive data files

Snapshots
- LVM(performance overhead)
- EBS snapshots 

Hot or Online
- MySQL Enterprise backup(expensive)
- Percona XtraBackup 

Using EBS snapshots

Why EBS snapshots are great for backups/restore:

- easy to implement
- EC2 uses incremental snapshots
- Easy to provision multiple machines

Cons

- EBSs are network drives, check the throughput limit for your instance
- (n disks * throughput)+other nw activity< bandwidth limit
- volumes are COLD in the beginning, penalty!

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-ec2-config.html

Using EBS snapshots – Contd.

Warming up new 100GB EBS disks - m3.2xlarge EBS optimized instance

Type	Warmup Time	Warmup Speed
SSD ‘io1’ PIOPS 3000	869s	**124 MB/s**
generic purpose ‘gp2’ SSD 300/3000	864s	**124 MB/s**
SSD ‘io1’ PIOPS 1500	1604s	**66.8 MB/s**
Magnetic ‘standard’ no iops	4884s	**22.0 MB/s**

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-prewarm.html

Using EBS snapshots– restore from EBS

“If you access a piece of data that hasn't been loaded yet, the volume immediately downloads the requested data from Amazon S3, and then continues loading the rest of the volume's data in the background.

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSSnapshots.html

EBS types:

- gp2 1TB 3.000 IOPS
- io1 1TB 20.000 IOPS
- magnetic 1TB

Warming up the data on 2 threads:
355467264 bytes (355 MB) copied, 159.721 s, 2.2 MB/s
475004928 bytes (475 MB) copied, 159.601 s, 3.0 MB/s

Warming up the data on 3 threads:

360710144 bytes (361 MB) copied, 195.47 s, **1.8 MB/s**
363855872 bytes (364 MB) copied, 195.47 s, **1.9 MB/s**
360710144 bytes (361 MB) copied, 195.471 s, **1.8 MB/s**

Using EBS snapshots– restore from EBS

Warming up 20GB (or restore from a backup kept on a snapshot):

4302.73 s, 5.0 MB/s = 71 minutes

If we want to warm up a 500GB volume = ~ 30 hours!!!

Percona Xrabackup

Online open source backup tool from Percona

- Parallel
- Compression
- Enryption
- Streaming
- Throttle
  - Doesn't work with streaming!

- Having multiple MyISAM needs longer period when Flush Tables With Read Lock

S3 download speed is pretty good:

```
innobackupex  --user=root --password=password --stream=xbstream  --parallel=10 /tmp | pigz –p 2 | pv –l 20M | aws s3 cp - s3://bucket/filename.gz
```

Do you have (encrypted) offsite backup ?

mysqlbinlog

- Backups taken with Xtrabackup are consistent across databases
  - Backups are not consistent across clusters (end time matters)
- Consistent restore requires point in time recovery
- mysqlbinlog supports --read-from-remote-server starting from MySQL 5.5
- expire_logs_days = 10 by default

Warming up cache

- Replay slow logs
  - Only replay SELECTs to a slave!
  - log_slow_rate_limit
- Percona-playback
- Saving/restoring buffer pool
- ProxySQL supports traffic mirroring

Schema

Schema – Primary Keys

- Always use Primary Keys
  - In InnoDB
    - *Every PK is (leftmost) part of ALL the secondary keys*
    - *USE a small one (UUID is a no-go!)*
    - *SIGNED is default use UNSIGNED*

If the table has no PRIMARY KEY or suitable UNIQUE index, InnoDB internally generates a hidden clustered index on a synthetic column containing row ID values. The rows are ordered by the ID that InnoDB assigns to the rows in such a table. The row ID is a 6-byte field that increases monotonically as new rows are inserted. Thus, the rows ordered by the row ID are physically in insertion order.

- https://dev.mysql.com/doc/refman/5.7/en/innodb-index-types.html

Simple tool to find all tables without primary keys on every cluster:

- https://raw.githubusercontent.com/mszel42/mukka/master/no_pk.sh
- Since pt-osc requires PK or UNIQUE key, adding a PK is not possible with it.

Schema – Duplicate Keys

Every PK is (leftmost) part of ALL secondary keys

PRIMARY KEY (`id`),
UNIQUE KEY `last_visit_username` (`last_visit`,`username`),
KEY `last_visit` (`last_visit`),
KEY `id_last_visit` (`id`, `last_visit`)

DB	TABLE	REDUNDANT INDEX NAM	TABLE DATA SIZE[MB]	TABLE INDEX SIZE[MB]	NUM ROWS	RECOMMENDE	FOREIGN KEYS
marketplace	play_useram	id	996	1867.91	18808722	PT-OSC	
marketplace	account_use	user_id	905.83	1005.78	3946361	PT-OSC	
marketplace	account_buy	user_id	596	734.17	6281639	PT-OSC	database.collection;database2.user
marketplace	tracking_edn	user_id	458.89	856.84	6367249	PT-OSC	
marketplace	account_buy	id	24.55	52.05	471564	ALTER TABLE	-
marketplace	tracking_use	user_id2	20.55	47.14	340699	ALTER TABLE	-

https://github.com/mszel42/mukka/blob/master/mt-duplicate-index-review.pl

Schema – Max Integer check

- INT: Most typical column types for Primary Keys
- UNSIGNED NOT NULL
- A rolled back TX still increases the AUTO_INCREMENT

SCHEMA NAME	TABLE NAME	COLUMN NAME	COLUMN TYPE	UNSIGNED?	AUTOINC?	MAX VALUE	COLUMN LIMIT	%FULL
analytics	answers	id	bigint			9222639640569	9223372036854	99.99
snapshot_analytics	questions	question_id	bigint			9222639640569	9223372036854	99.99
reporting	users	id	int	**unsigned**	autoinc	4246461765	4294967295	98.87
reporting	trends	id	int	unsigned	autoinc	713444021	4294967295	16.61

https://github.com/RickPizzi/pztools/blob/master/findmax.sh

Schema – Reclaimable space

- Innodb tables can only grow
- Reclaim space is only possible when innodb_file_per_table=on
- <5.7.4 optimize table is not online

pt-online-schema-change --execute --alter 'ENGINE=InnoDB' D=users,t=userdata

DB	TABLE	ROWS	TOTAL SIZE[MB]	RECAIMABLE SPACE[MB]	RECOMMENDED METHOD
users	userdata	1320689114	119717	36776	PT-OSC
users	comments	796809533	76865	15570	PT-OSC
users	notifications	2657222062	316047	242	PT-OSC
groups	emails	356000	120	40	ALTER

https://github.com/mszel42/mukka/blob/master/mt-check-fregmentation.pl

Schema – Foreign Key

Foreign keys are created by some ORMs(Django, Ruby activerecord)

- Pros:
  - Integrity provided by the Database Server
  - relatively easy to implement
- Cons:
  - The referenced columns(all of them) must be indexed to speed up these lookups
  - Every DML(INSERT/UPDATE/DELETE) will result in other lookups in the referenced tables (in order to check the FK constraint)
  - Schema changes are riskier
  - If a table that is being referenced becomes corrupted or inconsistent, MySQL will drop an error if the foreign key constraint fails and stops executing the given query

LOAD DATA LOCAL INFILE

“In a Web environment where the clients are connecting from a Web server, a user could use LOAD DATA LOCAL to read any files that the Web server process has read access to (assuming that a user could run any command against the SQL server).”
http://dev.mysql.com/doc/refman/5.6/en/load-data-local.html

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

```
[root@hostdb ~]# mysql -e "show grants for wordpress@%”

GRANT USAGE ON *.* TO ’wordpress@'localhost' IDENTIFIED BY PASSWORD ‘xxx’
GRANT SELECT, INSERT ON `wordpress`.* TO 'wordpress'@’%’
```

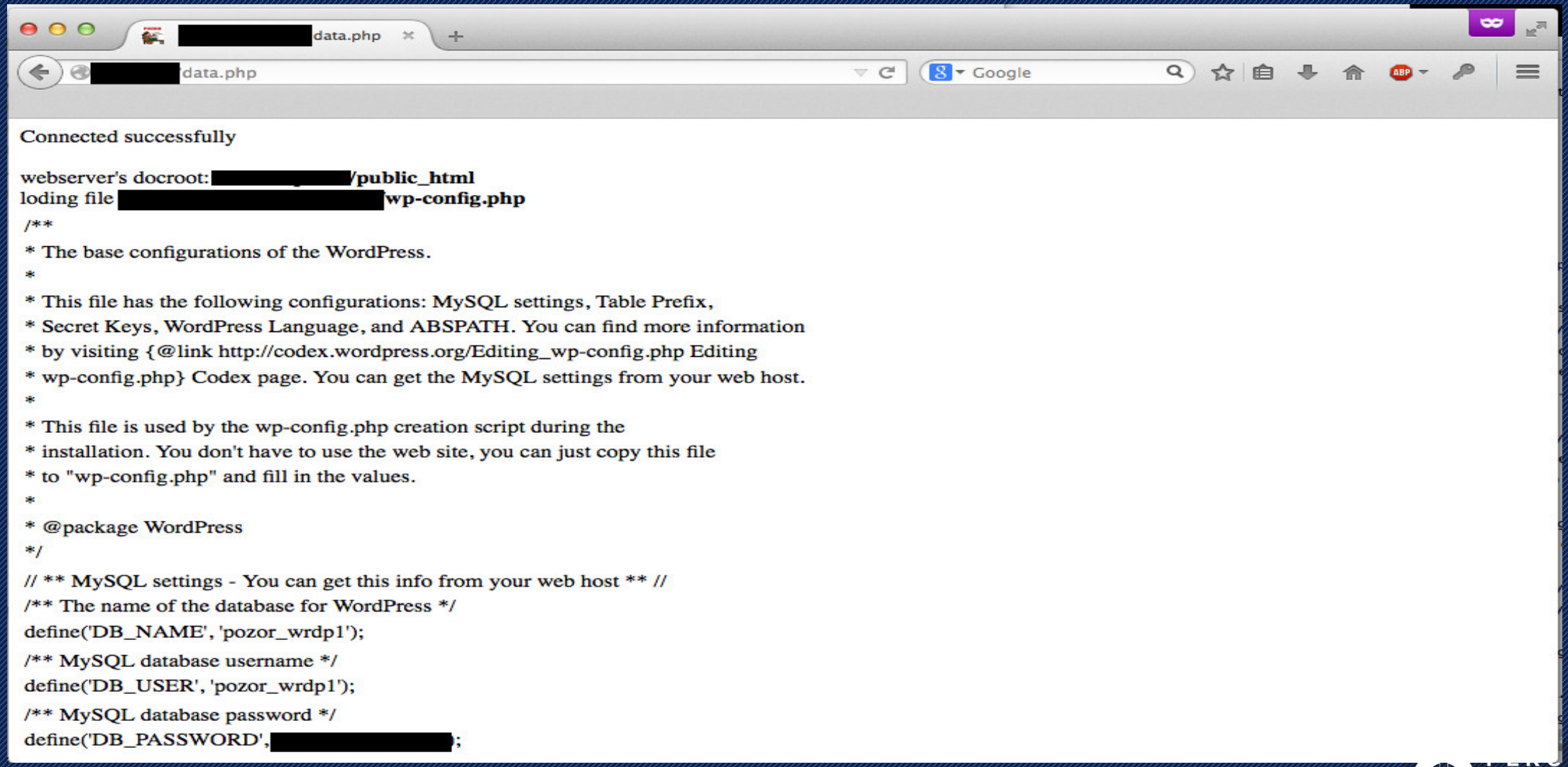
LOAD DATA LOCAL INFILE

```
mysql> LOAD DATA LOCAL INFILE  '/root/.aws/config' INTO TABLE `wp_comments` (comment_content) ;
Query OK, 5 rows affected, 2 warnings (0.00 sec)

mysql> LOAD DATA LOCAL INFILE  '/root/.my.cnf' INTO TABLE `wp_comments` (comment_content) ;
Query OK, 3 rows affected, 2 warnings (0.00 sec)

select comment_content  from wp_comments;
[..]
| [default]                                                                |
| output = json                                                            |
| region = us-west-2                                                       |
| aws_access_key_id = AKIAxxxxxxxxx                                        |
| aws_secret_access_key = S4xxxxxxxxxxxxxxxxxx                             |
| [client]                                                                 |
| user=root                                                                |
| password=mysqlpassword                                                   |
+--------------------------------------------------------------------------+
```

LOAD DATA LOCAL INFILE



Questions?