20 Typical MySQL Mistakes and how to avoid them

Short real world examples and tips to avoid them

Janos Ruszo

Senior Consultant
Percona University Budapest
11/05/2017

OS side mistakes

open files limit, storage, timezones

Open files limit / security.conf / systems

- Default limit: 1024 open files
- Default configuration file: /etc/security/limits.conf
- RHEL 6+ / Fedora 9+: /etc/security/limits.d
- Increase the limits for MySQL!

Data directory on NFS

- Avoid if possible
- Latency of SSD: 0.031ms / Latency of NFS ~= network latency (0.5-1ms)
- Higher latency -> lower IO/s
- NFS server can go away (short network issue / server side issue)
- Usually NFS server is shared, no guaranteed performance
- Usually DB team cannot access the NFS server -> harder to debug performance issues

Timezones

- MySQL uses the system's timezone
- Master/slave can have different timezone -> different results
- Functions which relies on the local time, will produce different results on replication [NOW() / FROM_UNIXTIME()]
  - *Binlog format "ROW" replicates the actual value, which makes it resistant to this issue*

https://dev.mysql.com/doc/refman/5.7/en/replication-features-timezone.html

Configuration mistakes

various configuration values and what to set them to

Buffer Pool Sizing

- General recommendation: 75-80% of total memory
- InnoDB caches data + index
- Data server from memory is fast, served from disk is slower

- SHOW ENGINE INNODB STATUS
- Buffer pool hit rate 999 / 1000

Character sets

- Character set levels:
- Server -> Database -> Table -> Column
- Client and storing column/table can have different character sets -> that's bad

Character sets - example

```
Create Table: CREATE TABLE `text` (
  `content` text
) ENGINE=InnoDB DEFAULT CHARSET=utf8

mysql> SHOW VARIABLES LIKE 'character_set_client%';
+----------------------+--------+
| Variable_name        | Value  |
+----------------------+--------+
| character_set_client | latin1 |
+----------------------+--------+
1 row in set (0.00 sec)
mysql> insert into test.text set content='árvíztűrő tükörfúrógép’;
mysql> select * from text;
+------------------------------+
| content                      |
+------------------------------+
| árvízt?r? tükörfúrógép       |
+------------------------------+
1 rows in set (0.00 sec)
```

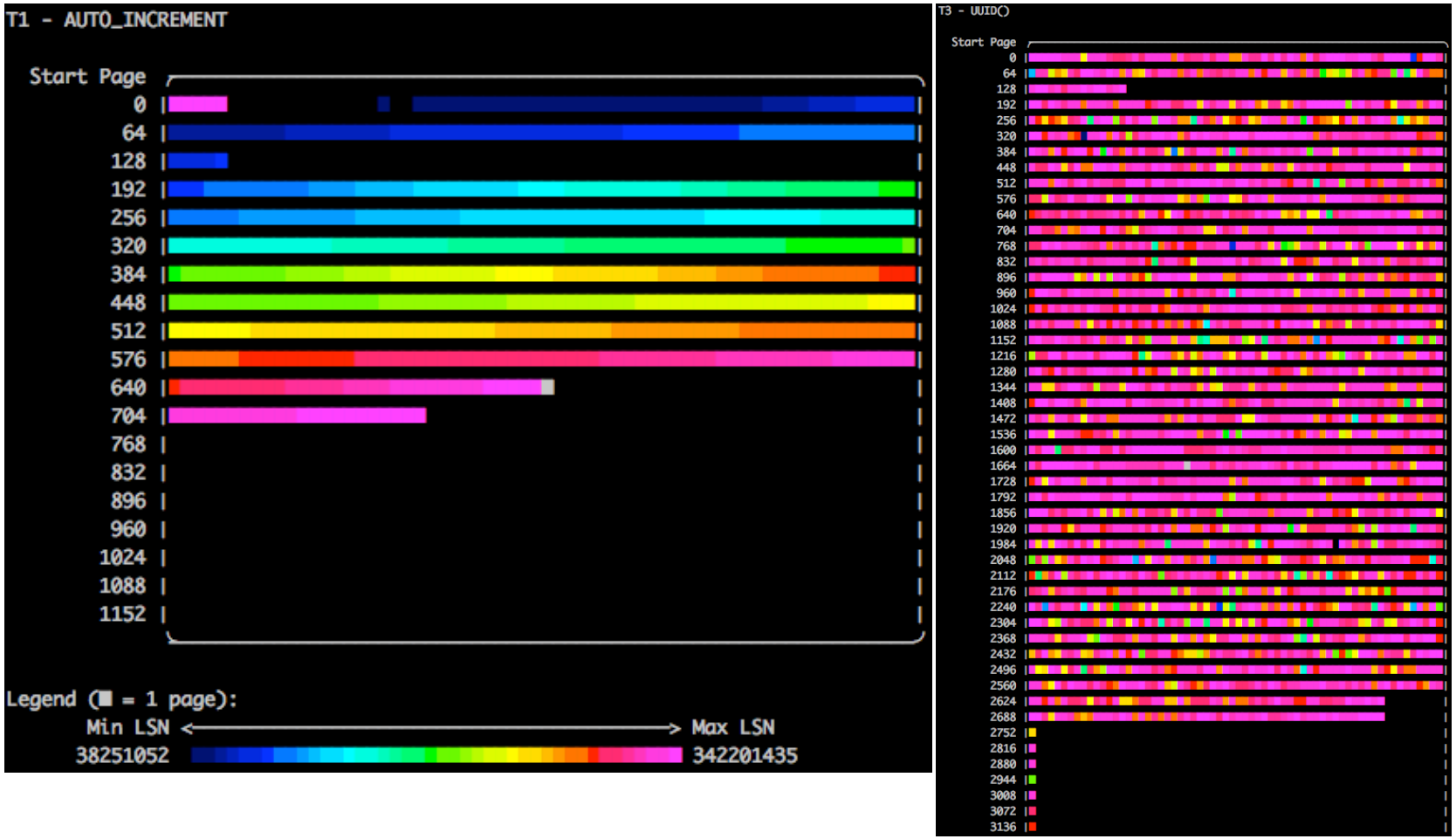
MyISAM tables

- MyISAM is not crash safe -> you can lose data on server crash
- Not caching data, only the indexes
- MyISAM support only table level locking
- Not MVCC compilant, no transactions
- Slowly deprecated

Multi Column / UUID PK on InnoDB

- InnoDB organises data ordered by the Primary Key
- Secondary index always contains the Primary Key, which results in large indexes
- UUID is not sequential which results in fragmentation and random writes

Multi Column / UUID PK on InnoDB



PERCONA

Too many tables on the server

- On filesystem, each database is 1 directory
- Each table is at least 2 files (.frm + .idb) at InnoDB
- Standard drupal install: 74 tables
- Shared host with 1000 drupal installs -> 74.000 tables
- Various information_schema queries have to open all tables
- even filesystem find/ls can be slow

Straight alter (direct alter)

- Straight alter locks write access to the table
- For large tables this can take a long time
- Since MySQL 5.6 online alter possible in some cases (See MySQL documentation)
- For non-blocking schema modifications use pt-osc or gh-ost

Version mismatch

- Ansible:

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

- Centos/Redhat:

```
yum install MySQL-server
```

- Debian/Ubuntu:

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

Version differences

- Due to optimisations version difference can result in performance differences

```
[root@server-1 ~]# mysql --version
mysql  Ver 14.14 Distrib 5.6.34-79.1, for Linux (x86_64) using  6.2

[root@server-2 ~]# mysql --version
mysql  Ver 14.14 Distrib 5.6.35-81.0, for Linux (x86_64) using  6.2
```

- You can use the packages directly
- Use own repository

Config differences

- Config differences between my.cnf and runtime configuration
- After restart things can break / unexpected behaviour
- Different config between slaves

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

Using server's IP as 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 INT(4294967295)
  - 192.168.111.222 (192168111222 -> 4294967295)
  - 192.168.222.222 (192168222222 -> 4294967295)

  - 4508354421957495439 -> 4294967295
  - 12354356476576 -> 4294967295

wait_timeout, max_connections

- Default value of wait_timeout: 28800 seconds
- Use max_user_connections, this will prevent a single user from using all connections

- MySQL reserves an extra connection slot (max_connections + 1) for a user with SUPER privileges
- gdb -p $pid -ex "set max_connections=3000 --batch

Replication lag

- Replication is single threaded!
- A long running delete/update/alter can cause huge replication lags, which causes applications to read stale data

- 5.6 introduced multi threaded replication for different databases
- 5.7 introduced new methods for multi threaded replication

Too many slaves on 1 master

- Master logs all the writes in the binary log
- All slaves needs to pull the binary logs

- With high number of slaves (10-20), they cause huge IO and Network overhead on the master
- eg.: 15 slaves, 100MB write/minute
  - 15*100 -> 1.5GB/minute -> 25MB/sec (200Mbps)

- Use intermediate slaves or shard the database into smaller “blocks”

Bad queries

- Queries which are not using indexes or aggregating lot of data can be deadly

- LIKE '%something' or WHERE without index

```
mysql> explain select * from text where data like '%something';
+----+-------------+-------+------+---------------+------+---------+------+---------+-------------+
| id | select_type | table | type | possible_keys | key  | key_len | ref  | rows    | Extra       |
+----+-------------+-------+------+---------------+------+---------+------+---------+-------------+
|  1 | SIMPLE      | text  | ALL  | NULL          | NULL | NULL    | NULL | 2092230 | Using where |
+----+-------------+-------+------+---------------+------+---------+------+---------+-------------+
```

- JOIN on columns without index

```
mysql> explain select text.* from text join text2 on text.data = text2.data  where text2.data = 'something';
+----+-------------+-------+------+---------------+------+---------+------+---------+-------------+
| id | select_type | table | type | possible_keys | key  | key_len | ref  | rows    | Extra       |
+----+-------------+-------+------+---------------+------+---------+------+---------+-------------+
|  1 | SIMPLE      | text2 | ALL  | NULL          | NULL | NULL    | NULL | 2091815 | Using where |
|  1 | SIMPLE      | text  | ALL  | NULL          | NULL | NULL    | NULL | 2092230 | Using where |
+----+-------------+-------+------+---------------+------+---------+------+---------+-------------+
```

Bad queries

- Queries which are not using indexes or aggregating lot of data can be deadly
- Monitor slow queries:
  - Percona Monitoring and Management Query Analyser
  - pt-query-digest
  - Anemometer
  - Vividcortex
- Handle queries:
  - pt-kill
  - Optimise
- Cache!

Maximum Integer

- INT: Most typical column types for Primary Keys (2147483647)
- UNSIGNED NOT NULL (4294967295)
- A rolled back TX still increases the AUTO_INCREMENT

- Monitor the usage of INT columns with AUTO_INCREMENT
- Start to plan in time.
- Prepare application to use BIGINT (18446744073709551615)

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

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 statement against the SQL server). “
  - https://dev.mysql.com/doc/refman/5.6/en/load-data-local.html

- To disable:
  - SET GLOBAL local_infile = OFF
  - my.cnf: local_infile = OFF

LOAD DATA LOCAL INFILE - example

```
mysql> LOAD DATA LOCAL INFILE  '/etc/shadow' INTO TABLE `text` (data);
Query OK, 27 rows affected (0.00 sec)
Records: 27  Deleted: 0  Skipped: 0  Warnings: 0
mysql> select * from text limit 0,1;
+-----------------------------------------------------------------------------------------+
| data                                                                                    |
+-----------------------------------------------------------------------------------------+
| root:$6$DytrzB7fNRZakzdw$.UM./RzgQ2s1p.hJ<redacted>AJnx3z6u1amvHWUHYBo.n/::0:99999:7::: |
+-----------------------------------------------------------------------------------------+
1 row in set (0.00 sec)
```

Backup caveats

- Most common backup methods
- Logical
  - mysqldump
  - mydumper
- Cold
  - stop mysql, archive data files
- Snapshots
  - LVM(performance overhead)
  - EBS snapshots
- Hot or Online
  - MySQL Enterprise backup(expensive)
  - Percona XtraBackup

Backup caveats - Logical

- mysqldump
  - Runs for a long time for larger databases
  - Single threaded
  - Locks the databases for consistent backup (except InnoDB tables with —single-transaction option

- mydumper
  - Multi threaded
  - Still need to lock MyISAM tables, but for a shorter time
  - Faster restores due to multiple threads

Backup caveats - Snapshot

- LVM snapshot
  - Very large overhead for writes!
  - Read speed degrades as snapshot space is consumed
  - https://www.percona.com/blog/2013/07/09/lvm-read-performance-during-snapshots/

- EBS snapshot
  - quick and light
  - Snapshot is stored in S3, quick to restore
  - !!EBS created from snapshot is cold!!
  - “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”
  - In our tests, read speed for not yet downloaded data is 4-6MB/s
  - http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSSnapshots.html

Backup caveats - Hot or Online

- Percona XtraBackup
  - Online open source backup tool from Percona
  - Parallel
  - Compression
  - Encryption
  - Streaming
  - Throttle (Doesn’t work with streaming!)

  - MyISAM tables still needs to be locked during backup!

Backup caveats - mysqlbinlog

- For Point-in-time restore, we need the binary logs to be backed up as well!

- 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

Database Performance Matters