

MySQL Group replication

Francisco Bordenave - Architect, Percona

Overview

- Introduction
- Single-Primary vs Multi-Primary
- New variables
- New tables
- Status counters
- Demo!

Introduction

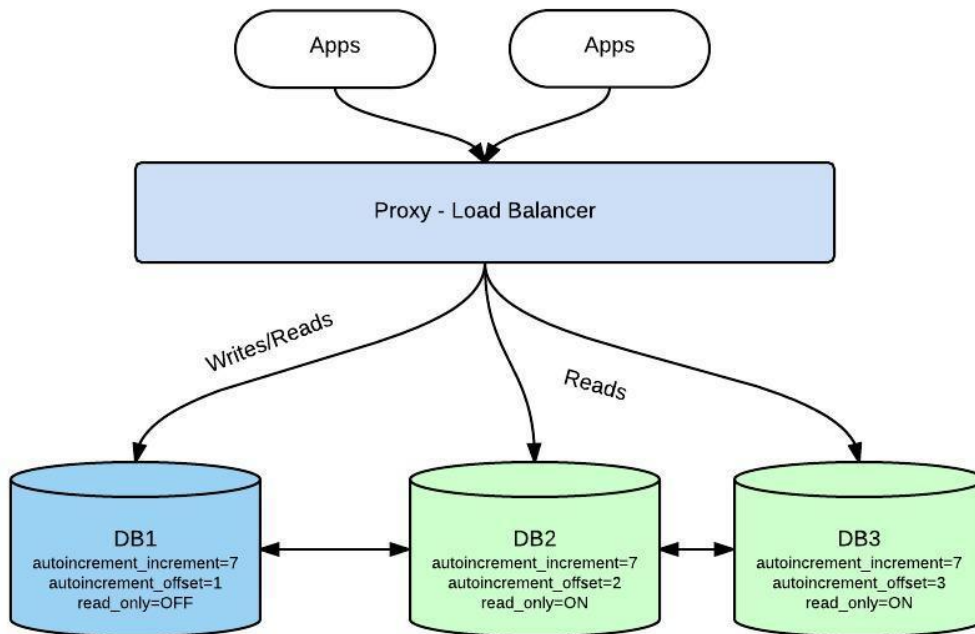
- New HA solution - a.k.a. InnoDB Cluster
 - Actually InnoDB cluster includes some other tools.
- All nodes are writable
 - Single-Primary and Multi-Primary modes available
 - Multi-Primary is not recommended - defaults to Single-Primary
- Relies in writesets - same than Galera
- Only InnoDB and RBR - same than Galera!

Introduction

- No SST concept - all nodes need to start with same data
- IST is based in binlog positions - GTID
- Auto detection of data inconsistencies based in RBR
- Replication error just stop replication to that node
- Inconsistencies can/should be fixed manually (no need to SST)

Single-Primary vs Multi-Primary

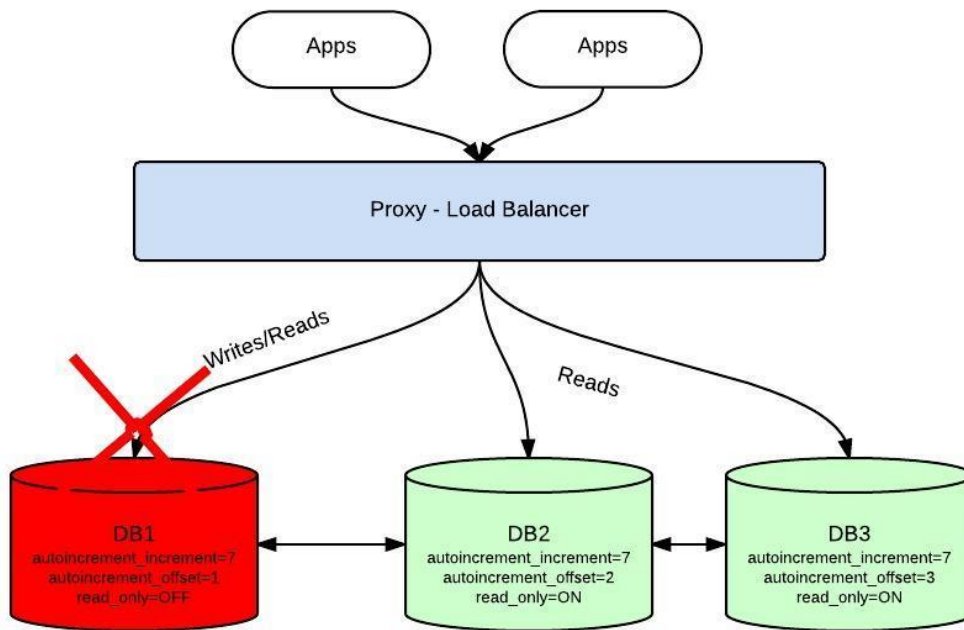
Single-Primary mode



PERCONA

Single-Primary vs Multi-Primary

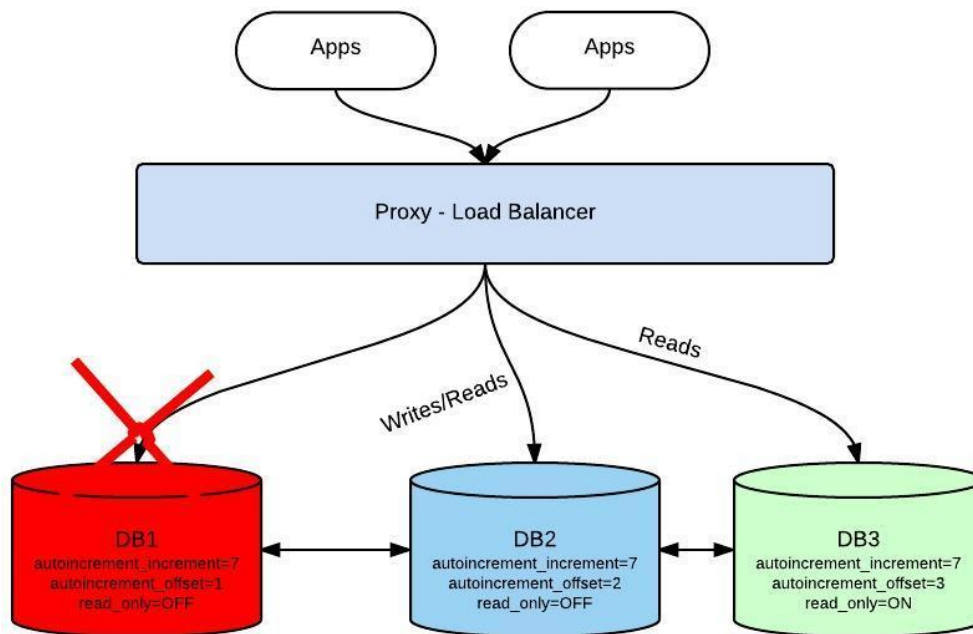
Single-Primary mode - Failure and autopromotion



PERCONA

Single-Primary vs Multi-Primary

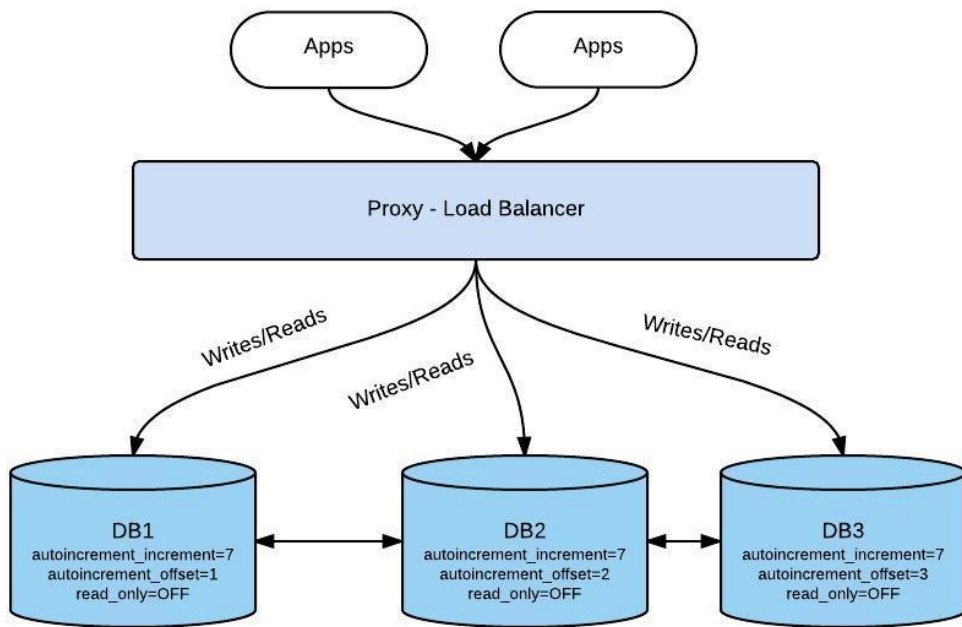
Single-Primary mode - Failure and autopromotion



PERCONA

Single-Primary vs Multi-Primary

Multi-Primary mode - All nodes are equal



PERCONA

New variables

Well not new but mandatory

- `log_bin`
- `binlog_format=ROW`
- `gtid_mode=ON`
- `enforce_gtid_consistency=ON`
- `master_info_repository=TABLE`
- `relay_log_info_repository=TABLE`
- `binlog_checksum=NONE`
- `log_slave_updates=ON`

New variables

Now new vars :) - minimal configuration

- `transaction_write_set_extraction=XXHASH64`
- `[loose-]group_replication_group_name="aaaaaaaa-aaaa-aaaa-aaaa-aaaa-aaaa"`
- `[loose-]group_replication_start_on_boot=off`
- `[loose-]group_replication_local_address= "192.168.70.2:33061"`
- `[loose-]group_replication_group_seeds="192.168.70.2:33061,192.168.70.3:33061,192.168.70.4:33061"`
- `[loose-]group_replication_bootstrap_group= off`

New variables

Some more interesting vars (there are a bunch)

- `group_replication_recovery_retry_count=10`
- `group_replication_recovery_reconnect_interval=60`
- `*ssl* options (recovery, communication and so)`
- `group_replication_recovery_complete_at`
- `group_replication_allow_local_disjoint_gtids_join=0`
- `group_replication_flow_control_mode=QUOTA`
- `group_replication_flow_control_certifier_threshold=25000`
- `group_replication_flow_control_applier_threshold=25000`

New tables

- replication_group_members

```
mysql> select * from replication_group_members;
```

CHANNEL_NAME	MEMBER_ID	MEMBER_HOST	MEMBER_PORT	MEMBER_STATE
group_replication_applier	c95e292c-cdea-11e6-9dc7-08002726bc0a	data2.localdomain	3306	ONLINE
group_replication_applier	d557eb77-cde9-11e6-9a13-08002726bc0a	data1.localdomain	3306	ONLINE
group_replication_applier	fd6aafcl-cde6-11e6-be5b-08002726bc0a	data3.localdomain	3306	ONLINE

New tables

- replication_group_member_stats

```
mysql> select * from replication_group_member_stats\G
***** 1. row *****
      CHANNEL_NAME: group_replication_applier
      VIEW_ID: 14839906723816325:3
      MEMBER_ID: fd6aafc1-cde6-11e6-be5b-08002726bc0a
COUNT_TRANSACTIONS_IN_QUEUE: 0
COUNT_TRANSACTIONS_CHECKED: 1
COUNT_CONFLICTS_DETECTED: 0
COUNT_TRANSACTIONS_ROWS_VALIDATING: 0
TRANSACTIONS_COMMITTED_ALL_MEMBERS: aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa:1-14:1000008
LAST_CONFLICT_FREE_TRANSACTION: aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa:14
```

Status counters

```
mysql> show global status like '%grou%';
```

+-----+-----+	
Variable_name	Value
+-----+-----+	
Com_group_replication_start	1
Com_group_replication_stop	0
group_replication_primary_member	d557eb77-cde9-11e6-9a13-08002726bc0a
+-----+-----+	

DEMO!!!