PERCONA
XtraDB Cluster

GALERA CLUSTER

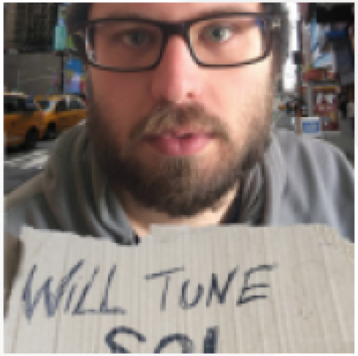
MySQL Group Replication

Us

Ramesh Sivaraman

QA Engineer

Kenny Gryp



MySQL Practice Manager

PERCONA

Table of Contents

1. Overview
2. Similarities
3. Differences GR & Galera
4. Differences PXC & Galera
5. Limitations
6. Known Issues
7. QA
8. Performance Tests

Different Technologies

Overview

Galera Cluster

- Developed by Codership
- http://galeracluster.com
- Included in MariaDB

Galera Cluster is a **synchronous multi-master** database cluster, based on **synchronous replication** and Oracle's MySQL/InnoDB. When Galera Cluster is in use, you can **direct reads and writes to any node**, and you can **lose any individual node without interruption** in operations and **without the need to handle complex failover procedures**.

- Replication is synchronous, Applying is asynchronous

Percona XtraDB Cluster

- Patched Galera Cluster, developed by Percona
- https://www.percona.com/software/mysql-database/percona-xtradb-cluster
- Generally Available Since April 2012
- With additional features
  - Extended PFS support
  - SST/XtraBackup Changes
  - Bug-Fixes
  - PXC Strict mode *
  - ProxySQL integration *
  - Performance Enhancements *

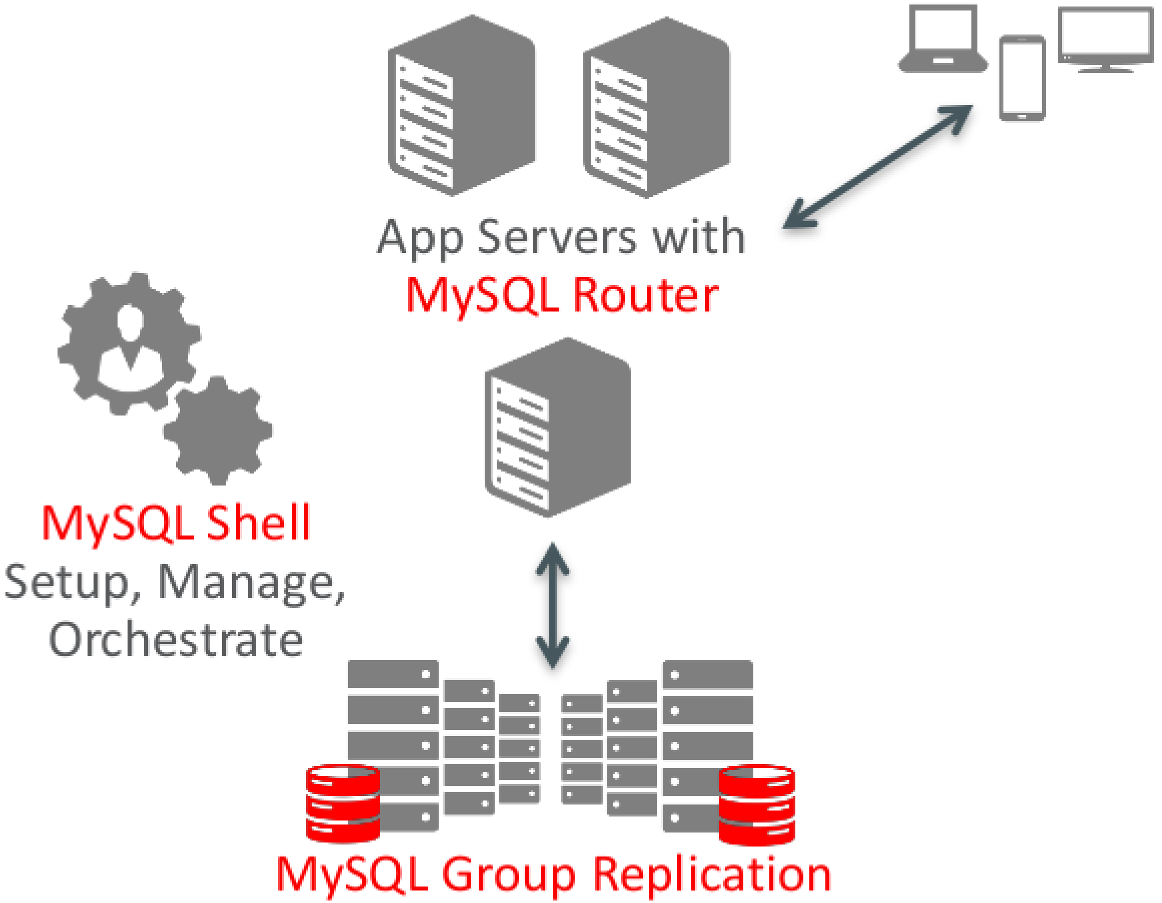
PERCONA XtraDB Cluster

MySQL Group Replication

- Developed by Oracle
- Generally Available in MySQL 5.7.17 on December 2016
- MySQL InnoDB Cluster as Solution

MySQL Group Replication is a MySQL Server plugin that **provides distributed state machine replication** with strong coordination between servers. Servers coordinate themselves automatically, when they are part of the same replication group. **Any server in the group can process updates**. **Conflicts are detected and handled automatically**. There is a **built-in membership service** that keeps the view of the group consistent and available for all servers at any given point in time. **Servers can leave and join the group** and the view will be updated accordingly.

MySQL InnoDB Cluster



They have a lot in common

Similarities

Similarities

- MySQL/MariaDB
- Replication Method
- Data centric - All nodes have all data
  - Reads happen on the local node only
- All require InnoDB/XtraDB as Storage Engine
- **Active-active multi-master Topology**
  - Write to multiple nodes
  - No complex/external failover necessary
- Node membership: join/leave automatically
- Execute writes in Global Total Order
- **Data Consistency!**
- **Optimistic Locking / First Committer Wins**
- Quorum - split brain prevention

Similar - Use Cases

- Environments with strict **durability** requirements
- Write to multiple nodes simultaneously while keeping data **consistent**
- Reduce failover time

Similar Limitations

- Large & Long running transactions
  - Higher chance on failures
  - non-efficient replication of large transations
- Workload hotspots (updating same set of rows in parallel)
- Often writing to 1 node is the best solution

Group Replication vs. Galera Based

Differences

Differences

- Group Communication System
- Binlogs & Gcache
- Node Provisioning
- GTID vs. Seqno
- Partition Handling
- Full Solution or Plugin
- Flow Control
- WAN Support
- OS Support
- Schema Changes

Group Communication

- Galera:
  - Totem Single-ring Ordering
  - All nodes have to ACK message
- Group Replication:
  - Xcom, similar to Paxos Mencius
  - Paxos only requires majority of nodes to ACK the message in order

Binlogs & GCache

Galera Cluster/PXC:

- uses binlog row events
- but does not require binary logging
- writes events to Gcache (configurable size)

Group Replication:

- requires binary logging

Node Provisioning

- Galera Cluster/PXC:
  - has State Snapshot Transfer (SST):
    - Percona XtraBackup (Recommended)
    - `rsync`
    - `mysqldump`
  - incremental State Transfer (IST) using GCache
- MySQL Group Replication:
  - currently no automatic provisioning
    restoring a backup is required
  - asynchronous replication channel for syncing

GTID vs. Seqno

- MySQL Group Replication:
  - built around MySQL GTID.
  - writes to a cluster create GTID events on the GR Cluster UUID
- Galera Cluster/PXC:
  - has a `seqno` which is a incrementing number

Partition Handling

Galera Cluster/PXC:

- A partitioned node will refuse reads/writes (configurable)
- A partitioned node will automatically recover and rejoin

Group Replication:

- A partitioned node will accept reads
- A partitioned node will accept write requests, but will hang forever
- A partitioned node needs to be manually rejoined to the cluster

Full Solution or Plugin

- Plugin:
  - Group Replication is a 'Replication Plugin'
  - several split brain bugs in current code (fixes pending!)
- Solution:
  - Galera Cluster, handling application connections is not included
  - strong split brain prevention compared to current GR
  - MySQL InnoDB Cluster (w. MySQLRouter)
- Full Solution:
  - Percona XtraDB Cluster (w. ProxySQL)
  - integrated ProxySQL
  - strict mode prevents limitations from being used

Flow Control

Prevent a *slower node* from getting too far behind

- Galera Cluster/PXC:
  - block all writes in cluster when a node reaches a limit
  - flow control message is sent
  - low defaults; Galera: `16`(*), PXC: `100`
  - **Tell others to stop writes**
- MySQL Group Replication:
  - every node has statistics about every member
  - each individual node decides to throttle writes
  - high default: `25000`
  - **Slow down your own writes if other nodes are struggling**

WAN Support

MySQL Group Replication:

- not recommended for WAN

Galera Based Systems have WAN features:

- Weighted Quorum
- Tunable network communication settings
- Reduce network traffic with segments
- Arbitrator

Operating System Support

Galera:

- FreeBSD & Linux

Percona XtraDB Cluster:

- Linux

Group Replication:

- Linux, Windows, Solaris, OSX, FreeBSD

Schema Changes - DDL

- Galera Cluster/PXC:
    - Total Order Isolation:
    - All writes will be blocked during
    - Writes on other nodes will be terminated
    - Workarounds:
        - `pt-online-schema-change`
        - `wsrep_osu_method=RSU`
            - More operational work
            - Not for all DDL's
- Group Replication:
    - DDL does not block all writes, like regular InnoDB
    - Only recommended in single-primary mode

Percona XtraDB Cluster vs. Galera

Differences

Percona XtraDB Cluster vs Galera Cluster

PXC has additional features:

- Extended PFS support
- SST/XtraBackup Changes
- Bug-Fixes
- PXC Strict mode *
- ProxySQL integration *
- Performance Enhancements *

PXC Strict Mode

Prevent experimental/unsupported features:

- Only Allow InnoDB Operations
- Prevent Changing `binlog_format!=ROW`
- Require Primary Key on tables
- Disable Unsupported Features:
  - `GET_LOCK`, `LOCK TABLES`, `CTAS`
  - `FLUSH TABLES <tables> WITH READ LOCK`
  - `tx_isolation=SERIALIZABLE`

ProxySQL Integration

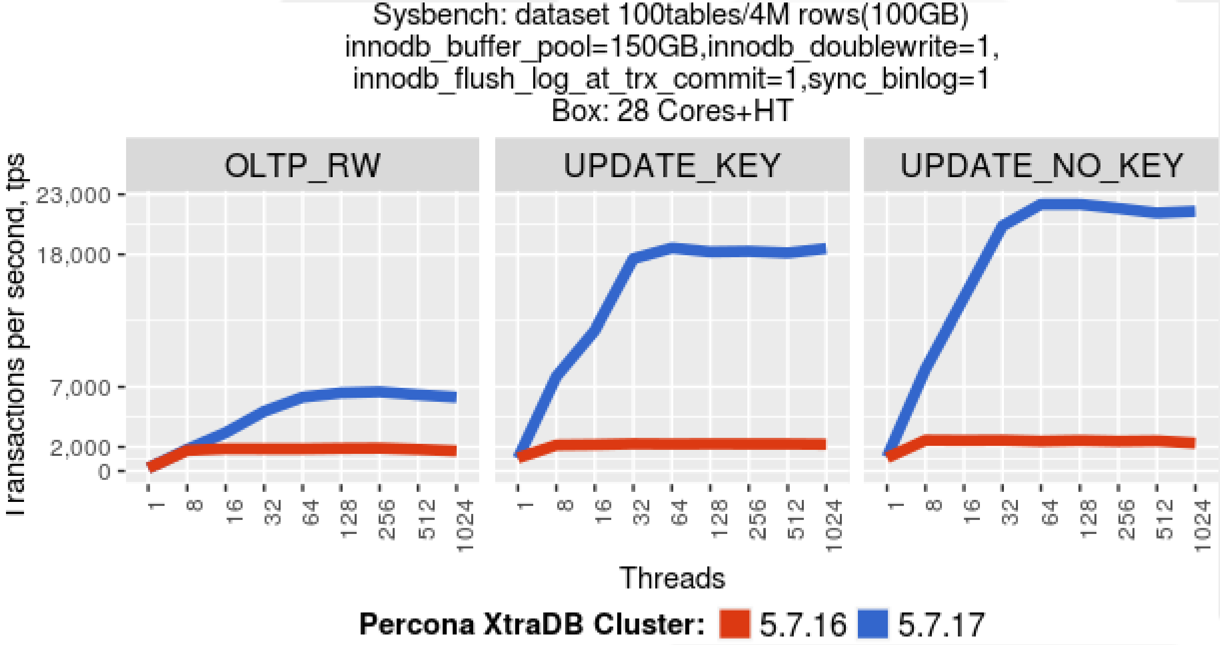
PXC includes ProxySQL as load balancer:

- **`proxysql-admin`** configuration tool
- **ProxySQL schedulers** :
  - Health Checks
  - Reconfigures Nodes
- **PXC Maintenance Mode**
  - Tell load balancer to rebalance load

Performance Enhancements

- Several Scalability Fixes in PXC 5.7.17
- New Defaults:

```
gcs.fc_limit=100
evs.send_window=10
evs.user_send_window=4
```



Sysbench: dataset 100tables/4M rows(100GB)
innodb_buffer_pool=150GB,innodb_doublewrite=1,
innodb_flush_log_at_trx_commit=1,sync_binlog=1
Box: 28 Cores+HT

be aware of

Limitations

Limitations - Galera Cluster/PXC

Does not work as expected:

- InnoDB/XtraDB Only
- `tx_isolation=SERIALIZABLE`
- `GET_LOCK()`
- `LOCK TABLES`
- `SELECT ... FOR UPDATE`
- Careful with `ALTER TABLE ... IMPORT/EXPORT`.
- Capped maximum transaction size
-
- XA transactions

Limitations - Group Replication

Does not work as expected:

- InnoDB/XtraDB Only
- `tx_isolation=SERIALIZABLE`
- `GET_LOCK()`
- `LOCK TABLES`
- `SELECT ... FOR UPDATE`
- Careful with `ALTER TABLE ... IMPORT/EXPORT`.
- Careful with large transactions
-
- no support for tables with multi-level foreign key dependencies, can create inconsistencies

nothing is perfect

Known Issues

Galera Cluster/PXC - Issues

- Crashes due to **background thread handling trx processing**
  - `mysql-wsrep#306`: stored procedure aborts
  - `mysql-wsrep#305`: event scheduler
  - `mysql-wsrep#304`: local scope functions such as `CURRENT_USER()`
- Various crashes **related to DDL**:
  - `mysql-wsrep#301`: running `SHOW CREATE TABLE` in multiple nodes with DDL can cause crash.
  - `mysql-wsrep#275`: Aborting trx leaves behind open tables in cache can cause crash

Galera Cluster/PXC - Issues

- **Concurrent DDLs** using `wsrep_OSU_method=RSU` crash/inconsistency issues
  - `mysql-wsrep#283` & `mysql-wsrep#282`
- **Shutdown issues**:
  - `mysql-wsrep#303`: cleanup during shutdown fails to clear the EXPLICT MDL locks (FTWRL)
  - `mysql-wsrep#273`: Not getting clean shutting down message if we start the server with unknown `variable`
  - `mysql-wsrep#279`: Trying to access stale binlog handler leads to crash

Group Replication - Issues

Partition Tolerance issues, split brain cannot be prevented:

- `#84727`: **partitioned nodes still accept writes: queries hang**
- `#84728`: **GR failure at start still starts MySQL**
- `#84729`: block reads on partitioned nodes
- `#84733`: not possible to start with `super_read_only=1` (Fixed in 8.0)
- `#84784`: Nodes Do Not Reconnect
- `#84795`: **STOP GROUP_REPLICATION sets super_read_only=off**
- `#84574`: DDL execute on partitioned node leads to split brain

Group Replication

Reduce impact on applications:

- #84731: mysql client connections get stuck during GR start

Stability:

- #84785: **Prevent Large Transactions in Group Replication**
- #84792: Member using 100% CPU in idle cluster
- #84796: GR Member status is wrong

Group Replication

Usability:

- #84674: unresolved hostnames block GR from starting (Fixed in 5.7.18)
- #84794: **cannot kill query that is stuck inside GR**
- #84798: Group Replication can use some verbosity in the error log

but we try to make it perfect

Quality Assurance

MySQL Test Suite

- MySQL Group Replication has an extensive MTR test suite, which covers member join primitives and recovery, member state change, query handling, concurrency, stress etc.
- Galera as well as Percona XtraDB Cluster uses its own MTR testsuite (not as extensive as mysql group replication) to test recovery, member state change, query handling, concurrency, stress etc.

pquery

- **pquery** is an open-source (GPLv2 licensed) multi-threaded test program created for stress testing the MySQL server (in any flavor), either randomly or sequentially, for QA purposes.

- To test Group Replication, Percona XtraDB Cluster and Galera we improved our existing pquery cluster framework. This framework will start a 3 node cluster and run pquery against these cluster nodes.

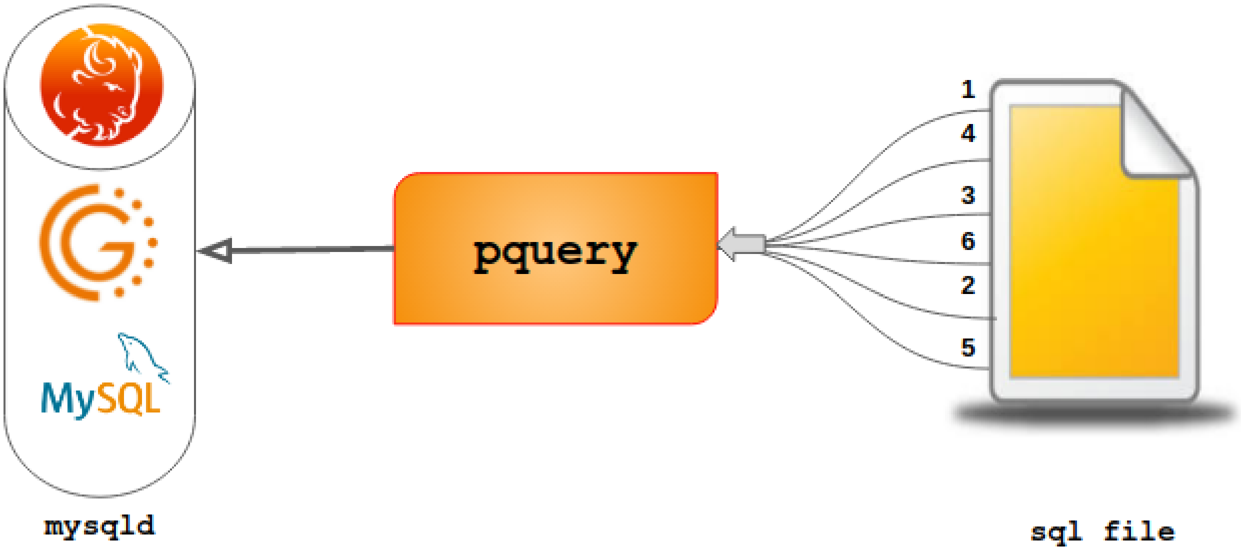
pquery

- **pquery** is mainly used for "Random Spread Testing" using a rich set of sql statement combinations. We have extracted these SQL statements from the MTR testsuite using a MTR to SQL convertor script (**`mtr_to_sql.sh`**)

pquery features

- 20+ Coredumps (crashes/asserts) per hour
- Fully automatic testcase creation
- C++ core
- 120 Seconds per trial run time
- Thousands of SQL lines executed per trial
- Compatible with sporadic issues
- High end automation
- Ultra fast testcase reduction
- Full framework

pquery framework



pquery framework

Group Replication pquery run resultset

```
$ ~/percona-qa/pquery-results.sh
================ [Run: 466282] Sorted unique issue strings
                  (1000 trials executed, 1557 remaining reducer scripts)
head->variables.gtid_next.ty                (Seen  16 times: reducers [107-2,3]
. is_set                                    (Seen  40 times: reducers [9-1] ...
key .= 64U                                  (Seen   1 times: reducers [268-1] )
.length % 4                                 (Seen  47 times: reducers [7-1] ...
m_pos.m_index_1 < mi->rli->                 (Seen   3 times: reducers [257-1] ..
rem0rec.cc line 867                         (Seen   1 times: reducers [515-1] )
.slen % 2                                   (Seen  24 times: reducers [6-1]  ...
.slen % 4                                   (Seen  12 times: reducers [116-1] ..
sort_field->length >= length                (Seen   5 times: reducers [30-1] ...
..thd                                       (Seen 283 times: reducers [1-1]  ...
.thd->is_error                              (Seen   1 times: reducers [393-1] )
thd->lex->sql_command == SQLCOM_XA_COMMIT  (Seen   2 times: reducers [576-1] )
thd->mdl_con...                             (Seen   2 times: reducers [194-1] ..
.tlen % 2                                   (Seen  57 times: reducers [55-1]  ..
.tlen % 4                                   (Seen  24 times: reducers [34-1] ...
Z10read_tokenPK18sql_digest_storagejPj      (Seen   6 times: reducers [173-1]  .
```

Group Replication pquery run sample testcase

- Out of 1000 pquery trials GR crashed 283 times with similar assertion message:
    - `handle_fatal_signal (sig=6)` in `Gtid_table_access_context::init`
        - https://bugs.mysql.com/bug.php?id=85364
- Generated reduced testcase using `reducer.sh`

```
DROP DATABASE test;
ALTER t t0ADD c c0CHAR exist;
XA START 'xid0';
SET @@GLOBAL.binlog_checksum=NONE;
```

Percona XtraDB Cluster pquery run resultset:

```
$ ~/percona-qa/pquery-results.sh
================ [Run: 987219] Sorted unique issue strings
                  (1000 trials executed, 724 remaining reducer scripts)
false                                         (Seen  26 times: reducers [60-1,2
get_state                                     (Seen   7 times: reducers [513-3]
. is_set                                      (Seen  25 times: reducers [36-1]
.length % 4                                   (Seen  31 times: reducers [23-2]
.mdl_context.has_locks                        (Seen   4 times: reducers [122-2]
.thd->is_current_stmt_binlog_format_row       (Seen  12 times: reducers [163-1]
thd->mdl_context.owns_equal...                (Seen   3 times: reducers [4-2] .
thd->security_context                         (Seen  69 times: reducers [45-2,3
.tlen % 2                                     (Seen  30 times: reducers [21-3]
.tlen % 4                                     (Seen  25 times: reducers [56-2]
trx0sys.cc line 354                           (Seen   3 times: reducers [617-2]
trx0trx.cc line 389                           (Seen  41 times: reducers [40-3]
ut0ut.cc line 917                             (Seen   6 times: reducers [455-3]
ZN12ha_myisammrg18append_create_infoEP6String (Seen 230 times: reducers [9-1] .
ZN3THD21send_statement_statusEv               (Seen   6 times: reducers [697-3]
ZN8MDL_lock28has_pendi                        (Seen   7 times: reducers [519-3]
```

Percona XtraDB Cluster pquery run sample testcase

- Out of 1000 pquery trials PXC crashed 69 times with similar assertion message:
  Assertion failed

  ```
  thd->security_context()->user().str
  ```

  - https://github.com/codership/mysql-wsrep/issues/304

- Reduced testcase

  ```
  Start 2 node cluster
  Execute following on one of the node
  "ALTER USER CURRENT_USER() IDENTIFIED BY 'abcd2';"
  ```

Startup scripts

- As part of QA testing we have made some handy scripts to start multiple Percona XtraDB Cluster/Galera/Group Replication nodes on the fly.

- These scripts are available in the **`Percona-QA/percona-qa`** github project. Currently these scripts supports binary tarball distributions only.

PXC/Galera startup script

- For PXC/Galera run **percona-qa/pxc-startup.sh** script from the Percona XtraDB Cluster basedir. It will generate a PXC startup script called **start_pxc**

```
$ git clone \
      https://github.com/Percona-QA/percona-qa
$ cd <PXC_BASE>
$ ~/percona-qa/pxc-startup.sh
Adding script: ./start_pxc
./start_pxc will create ./stop_pxc | ./*node_cli
| ./wipe scripts
$ ./start_pxc 5
Starting PXC nodes..
$
```

Group Replication startup script

- For Group Replication run **`percona-qa/startup.sh`** script from Group Replication basedir. It will generate a GR startup script called **`start_group_replication`**

```
$ cd <MySQL BASE>
$ ~/percona-qa/startup.sh
Adding scripts: start | start_group_replication |
start_valgrind | start_gypsy | stop | kill |
setup | cl | test | init | wipe | all | prepare |
run | measure | tokutek_init
Setting up server with default directories
[..]
$
```

Group Replication startup script

```
$ ./start_group_replication 3
Starting 3 node Group Replication, please wait...
Started node1.
Started node2.
Started node3.
Added scripts:  | 1cl  | 2cl  | 3cl
 | wipe_group_replication | stop_group_replication
Started 3 Node Group Replication.
[..]
$
```

benchmark-eating

Performance Tests

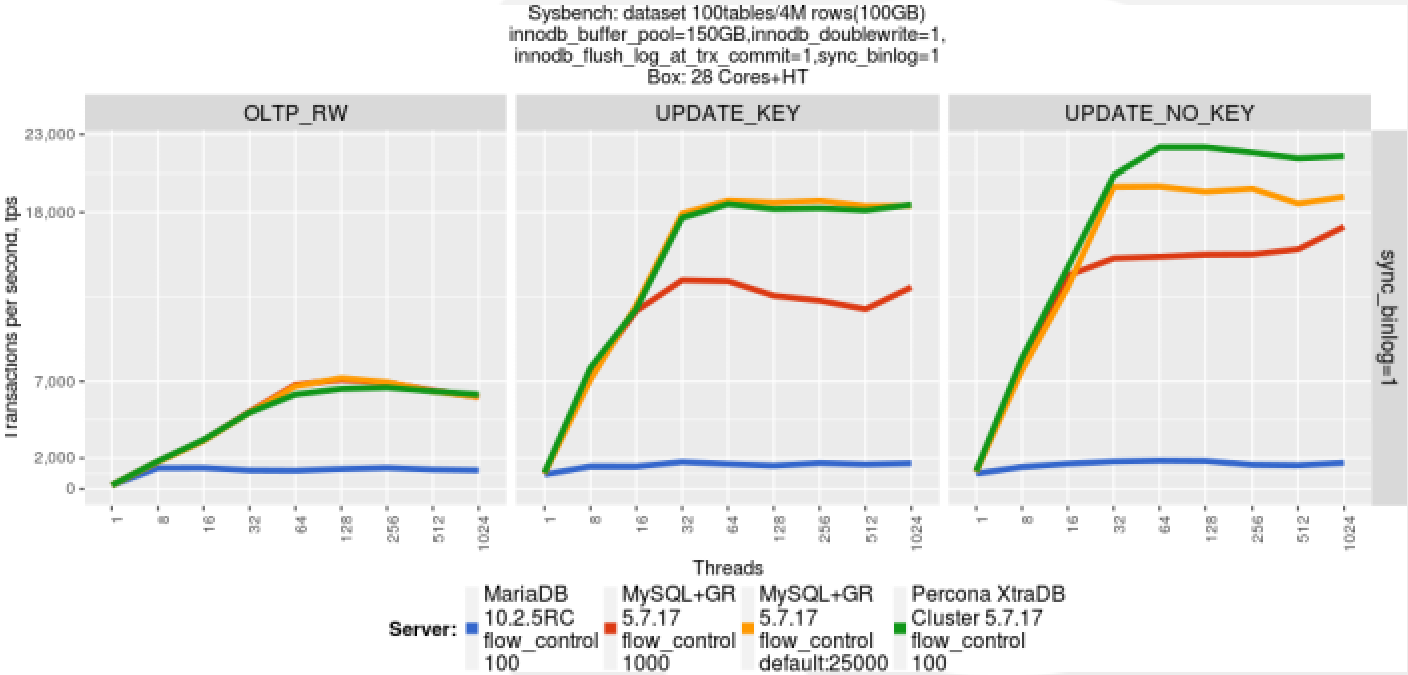
Performance Tests

- Performance comparison between Percona XtraDB Cluster, Galera and Group Replication.
    - Workload : Sysbench `OLTP_RW`, `UPDATE_KEY` and `UPDATE_NOKEY`
    - Table count : 100 (4 millions rows each)
    - Data Size : 100GB
    - Cluster : 3 Node

https://www.percona.com/blog/2017/04/19/performance-improvements-percona-xtradb-cluster-5-7-17/

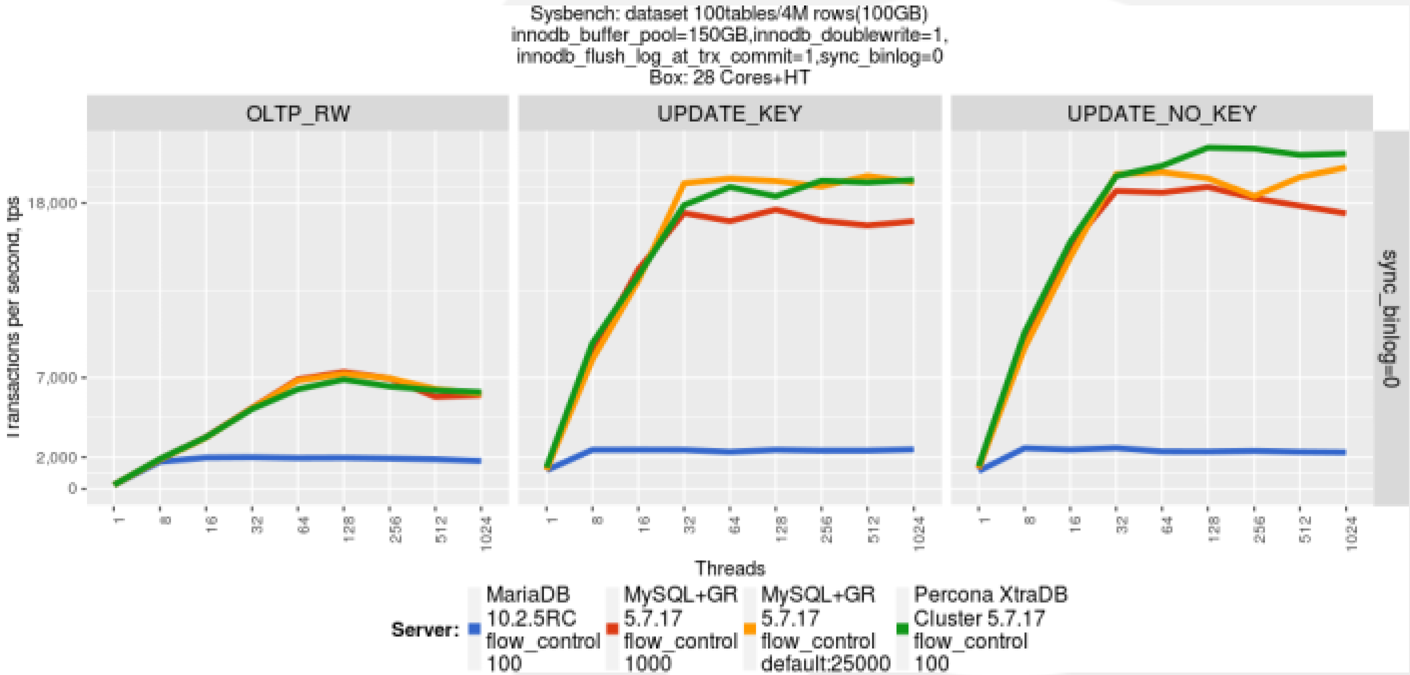
Performance Tests

Sysbench `OLTP_RW`, `UPDATE_KEY` and `UPDATE_NOKEY` workloads with 100 tables (`sync_binlog=1`)



Performance Tests

Sysbench `OLTP_RW`, `UPDATE_KEY` and `UPDATE_NOKEY` workloads with 100 tables (`sync_binlog=0`)



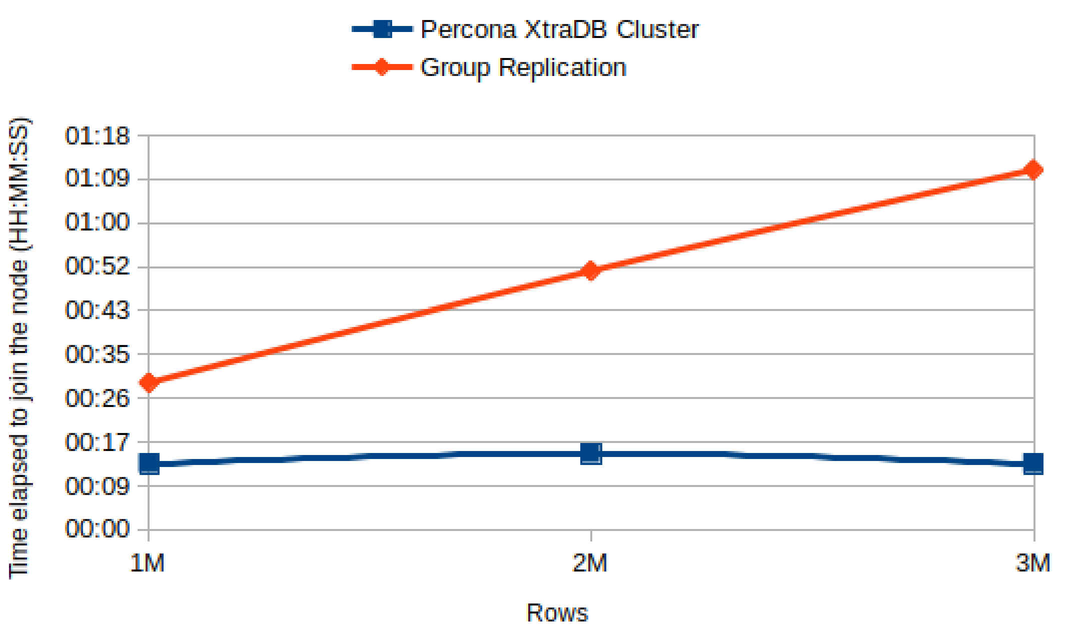
Performance Tests

Cluster node joining speed performance.

- Testcase
  - Shutdown one node from 3 node cluster
  - Sysbench run (create single table)
  - Start the node which was offline.
  - Check node status
    - With PXC script will check **`wsrep_local_state_comment`** status
    - With Group Replication script will check replication group **`ONLINE`** status.

Performance Tests

Cluster node joining speed performance graph.



(smaller is better)

Summary

	Galera	PXC	GR/MIC
Automatic Node Provisioning	✓	✓	
Load Balancer Integration		✓	✓
Enforcing Best Practices		✓	✓
Partition Handling	✓	✓	
Mature Technology	✓	✓	
Multi-Master	✓	✓	✓
WAN Support	✓	✓	
OS Support	✓	✓	✓
Performance		✓	✓
Supported By Percona	✓	✓	✓

just a few

Questions?

Ramesh Sivaraman

Kenny Gryp