Galera Replication

Synchronous Multi-Master Replication for InnoDB
...well, why not for any other DBMS as well

Seppo Jaakola – Alexey Yurchenko
Contents

1. Galera Cluster
2. Replication API
3. Benchmarking
4. Installation & Management
5. Galera Project
Replication for Transactional DBMS
Replication API

DBMS

repl API

Interface for replication system
- Calls for replication
- Callbacks from replication

Plugin framework
Pluggable Replicator

Provider can be loaded at DBMS start
Galera Cluster

For MySQL/InnoDB
Galera Cluster

For MySQL/InnoDB

wsrep extension implements replication API

Galera Replication

InnoDB

wsrep
Galera Cluster

InnoDB

wsrep

Galera Replication

For MySQL/InnoDB

wsrep extension implements replication API

dynamically loaded library
Galera Cluster

Clients

Transparent connections

Galera Replication

InnoDB

wsrep
Multi Master

Clients

Transparen connections

Multi-master

InnoDB

wsrep

Galera Replication
Multi Master

Clients

InnoDB

wsrep

InnoDB

wsrep

Transparent connections

Multi-master

Galera Replication
Multi Master

Clients

InnoDB
wsrep
InnoDB
wsrep
InnoDB
wsrep

Transparent connections

Multi-master

Galeria Replication
Synchronous Replication

Clients

InnoDB
wsrep

Transparent connections
Multi-master

Synchronous replication

G a l e r a R e p l i c a t i o n

Galera Replication
Galera Replication

- Synchronous multi-master replication
  - High Availability

- No middle-ware, connections directly to DBMS
  - Transparency

- Row events, row level locking
  - Write scalability

- Certification based replication method
Synchronous Replication

Client

commit

trx

wsrep

Galera Replication
Synchronous Replication

Transaction is replicated to all nodes => HA
**Synchronous Replication**

Transaction is applied at later time => virtual synchrony

**Galera Replication**
Certification Based Replication

- Transactions process independently in each cluster node
- Transaction write sets will be replicated at commit time
- Cluster wide conflicts resolved by certification test
Client

MySQL

Query Processing

write set population

WS extract

certification test

replication

MySQL

write set applier

certification test

replication

Group Communication

April 14, 2010

Codership @ MySQL Conference 2010
Client

commit

MySQL
Commit Processing
write set population
write set
replication

WS extract

MySQL
write set applier

certification
test
certification
test
replication

Group Communication

April 14, 2010

Codership @ MySQL Conference 2010
Client

MySQL

Commit Processing

write set population

WS extract

certification test

replication

MySQL

write set applier

certification test

replication

Group Communication
Client

MySQL

Commit Processing

commit
rollback

write set population
WS extract
certification test
replication
WS

MySQL

certification test
replication

write set applier

Group Communication
Replication API
Replication API

- Galera integrates closely in DBMS transaction processing
  - There must be an interface between DBMS and replication system
Other Replication APIs

- MySQL's API cooking up:
  ➔ http://forge.mysql.com/wiki/MySQL_Replication:_Walk-through_of_the_new_5.1_and_6.0_features

- Drizzle's API, already there:

- MariaDB specifying new API
  ➔ https://lists.launchpad.net/maria-developers/msg01998.html
wsrep API

- Codership's replication API
- DBMS agnostic replication interface
- Defines:
  - Write Set replication for transactions
  - TO isolation for replicating DDL
- Suitable for different replication modes (sync/async, multi-master, master/slave, PITR...)

- [https://launchpad.net/wsrep](https://launchpad.net/wsrep)
wsrep API Implementation

- Replication provider library load/unload
- Write set population calls
- Write set replication calls (at commit)
- Prioritized transactions
  - Lock queue modified
  - Aborting local victims
- Configuration hooks
- Status hooks
- TO isolation for DDL queries
Galera Library

wsrep API

wsrep hooks

wsrep provider

certification

replication

GCS framework

vsbes

spread

gcomm

dlopen

Galera
Benchmarking

- Tested with several benchmarks
  - Sysbench, dbt2, DOTS, osdb, jmeter, sqlgen...
- Tested with 'physical hardware' and with Amazon EC2 instances
  
  → In general, shows good scalability even with write intensive work loads
SysBench Benchmarks

- SysBench OLTP mode test
- 1M rows
- EC2 Large instances
Synchronous WAN Replication

- SysBench OLTP
- 1M rows
- EC2 large instances
- EU → US
- Distance: ~3000 miles
- Ping RTT: ~88 ms
Installation
Installing MySQL/Galera

Download from www.codership.com

Distributions choices:
1. Pre-built RPM or Debian package
2. demo tar distribution
3. Source build
Demo Distribution

- Pre-built 32/64 bit linux binaries
- Installs in one directory path
- Contains a sample database
- Good for testing/evaluation
Demo Distribution

- Install as regular user (not root)
  
  $ tar xzf mysql-5.1.43-galera-0.7.3-x86_64.tgz

- Node startup by: mysql-galera script
  - Commands: start | stop | check

- Specify cluster_address
  - Start first node with address: gcomm://
  - Start other nodes with gcomm://<first-node-ip>

  $ mysql-galera -g gcomm:// start
  $ mysql-galera -g gcomm://<other-IP> start
Galera in Cloud

- **VPS.net**
  - Nice new cloud computing solution
  - MySQL/Galera images available

- **Amazon EC2**
  - Extensively tested in EC2
  - Deploy e.g. Ubuntu node and install MySQL/Galera manually
  - Pre-built image underway
Cluster Topologies

- Use 3 or more nodes for HA
- Application load balancing gives best performance
- Use load balancer if a single connection point is needed
- Reference node can help in joining
Dedicated Replication Interconnection

Public connections

Min 1 Gb/sec replication network

SW

192.168.0.1

10.0.0.1

192.168.0.2

10.0.0.2

SW
Application Load Balancing

- + Gives best performance
- - Application must react to cluster changes
Load Balancer

in order of performance:

- HW balancers
- IP dispatching in kernel e.g. LVS
- TCP/IP load balancers e.g. GLB, in user land
- Proxy (e.g. MySQL Proxy)
Reference Node

- Works as donor for joining nodes
- Backups by xtrabackup

Galera Replication

C l i e n t s

No client connections
Reference Node as MySQL Master

Clients

Galera Replication

MySQL master

MySQL slave
Management
### wsrep Variables

```sql
mysql> show variables like 'wsrep%';
+-----------------+------------------------------------------+
| Variable_name    | Value                                    |
+-----------------+------------------------------------------+
| wsrep_auto_increment_control | ON                                       |
| wsrep_cluster_address       | gcomm://                                 |
| wsrep_cluster_name          | my_wsrep_cluster                         |
| wsrep_convert_LOCK_to_trx   | OFF                                      |
| wsrep_data_home_dir         | /home/galera/mysql-5.1.42-2957,1439/mysql/var/ |
| wsrep_debug_option          | NULL                                     |
| wsrep_debug              | OFF                                      |
| wsrep_drupal_282555_workaround | ON                                       |
| wsrep_local_cache_size     | 20971520                                 |
| wsrep_node_incoming_address | 10.0.0.121:3306                          |
| wsrep_node_name            | abyssinian                               |
| wsrep_on                  | ON                                       |
| wsrep_provider            | /home/galera/mysql-5.1.42-2957,1439/galera/lib/libmmgalera.so |
| wsrep_provider_options     | NULL                                     |
| wsrep_retry_autocommit     | ON                                       |
| wsrep_slave_threads        | 1                                        |
| wsrep_sst_auth            | root:rootpass                            |
| wsrep_sst_donor           | NULL                                     |
| wsrep_sst_method          | mysqlldump                               |
| wsrep_sst_receive_address  | AUTO                                     |
| wsrep_start_position      | NULL                                     |
| wsrep_ws_persistency      | OFF                                      |
+-----------------+------------------------------------------+
22 rows in set (0.00 sec)
```
wsrep Variables

- **wsrep_provider**
  - Path to provider library
- **wsrep_cluster_address**
  - Tells the connection point where node can join
  - 'gcomm://' for first node
  - 'gcomm://<IP address>', for joining nodes
### wsrep Status

```sql
mysql> show status like 'wsrep%';
+----------------+----------------+
| Variable_name   | Value           |
+----------------+----------------+
| wsrep_local_state_uuid | 0eedf650-1694-11df-0800-6227ab0639e3 |
| wsrep_last_committed     | 3               |
| wsrep_replicated        | 0               |
| wsrep_replicated_bytes  | 0               |
| wsrep_received          | 0               |
| wsrep_received_bytes    | 0               |
| wsrep_local_commits     | 0               |
| wsrep_local_cert_failures| 0             |
| wsrep_flow_control_waits| 0               |
| wsrep_local_status      | Joined (5)      |
| wsrep_cluster_conf_id   | 1               |
| wsrep_cluster_size      | 1               |
| wsrep_cluster_state_uuid| 0eedf650-1694-11df-0800-6227ab0639e3 |
| wsrep_cluster_status    | Primary         |
| wsrep_local_index       | 0               |
| wsrep_ready             | ON              |
+----------------+----------------+
17 rows in set (0.00 sec)
```
wsrep Status

- `wsrep_last_committed`
  - Tells which transaction has committed last

- `wsrep_local_cert_failures`

- `wsrep_local_bf_aborts`
  - How much cluster caused rollbacks

- `wsrep_flow_control_waits`
  - How much wait for flow control
Backups

- No direct backup method in 0.7 release :(  

- To get a backup
  - Join/depart a node in a cluster
  - *Use reference node as MySQL master and fan out to a backup slave*
  - *Use xtrabackup in reference node to get hot backup*
Joining New Nodes

Clients

MySQL
10.0.0.1

MySQL
10.0.0.2

SST Request

Clients

MySQL
10.0.0.3

wsrep_cluster_address= 10.0.0.2

Active cluster
Joining node
Joining New Nodes

Clients

Donor node

1. mysql dump
2. load
Joining New Nodes

Active cluster

Clients

MySQL

10.0.0.1

MySQL

10.0.0.2

MySQL

10.0.0.3
Galera Project
Galera Project

- **Kick off** (2008)
- **First public releases** (2009)
- **0.7 release**
  - Fully open source
  - 0.7.1
  - 0.7.2
  - 0.7.3
Release 0.7

- Current release 0.7.3
  - Stable release
  - Production readiness
  - Open source
- Simple management & installation utilities
- State transfer by mysqldump
- “Reasonably” good performance
Road Map

- **Stability milestone**
  - 0.7 releases
    - 0.7.4...

- **Optimization milestone**
  - Incremental backups
  - Xtrabackup
  - UDP multicast

- **Management milestone**
  - Cluster commands
  - Management console
Summary

- Certification based replication turns out effective
  - High Availability
  - Transparency
  - Good scalability even with high write rates
- wsrep API is “not too hard” to implement
- Any (transactional) DBMS can leverage this replication possibility
Codership – The Saga

- Founders Seppo Jaakola, Alexey Yurchenko, Teemu Ollakka
- Fin-Rus community working from Finland
- Experts in distributed systems & DBMS development, information security
- Set Sail Oct 2007
- Projects:
  - Galera
  - GLB (Debian ITP)
  - Cluster testing framework (in-house)
Get in Touch!

- R&D consulting services
- Support subscriptions

- Downloads available: http://www.codership.com
- info@codership.com
- Mailing list: codership-team@googlegroups.com