MySQL synchronous replication in practice with Galera

FOSDEM MySQL and Friends Devroom
February 5, 2012, ULB Brussels

Oli Sennhauser
Senior MySQL Consultant, FromDual
oli.sennhauser@fromdual.com
Content

- Galera Cluster
- Why Galera?
- Characteristics
- Set-up
- Configuration
- Starting / stopping
- SST
- Information
Galera Cluster

- Synchronous Multi-Master Replication

- Scaling reads (and writes)
Galera Cluster

• If one node fails:

  ![Diagram showing Galera Cluster setup]

  - Load balancing (LB)
  - Galera replication
  - Node 1, Node 2, Node 3

• High Availability (HA)
Why Galera?

- Master-Slave Replication
  - Not multi-Master, asynchronous, inconsistencies
- Master-Master Replication
  - Some kind of multi-Master, asynchronous, inconsistencies, conflicts
- MHA, MMM (v1, v2), Tungsten
  - Bases on MySQL Replication
- MySQL Cluster
  - Not like InnoDB, Know-How, Network-DB!
- Active/passive Failover Cluster
  - Operations, resources idling
- Schooner
  - Expensive, not sure what technology (Memcached, Replication, ...)
Characteristics

- Synchronous replication
- Based on InnoDB SE (other SE theoretically possible)
- Active-active real multi-master topology
- Read and write to any cluster node
- Automatic membership control
- True parallel replication, on row level
- No slave lag
- No lost transactions
- Read AND write scalability (Read Scale-Out!)
- Patch off MySQL binaries (Codership provides binaries)
- Be aware of Hot Spots on rows
- Higher probability of dead locks
- Full sync blocks for writing → 3 nodes
- Initial sync for very big databases (>>50 Gbyte) with mysqldump → rsync, xtrabackup
Set-up

- 3 nodes is recommended
  - Or 2 + 1 (2 mysqld + garbd) → SST!!!
  - 2 nodes → split brain!
- Codership MySQL + Galera Plug-in (wsrep)
- User for SST is root!
  - We recommend to use your own user.
  - On all nodes:

  ```sql
  GRANT ALL PRIVILEGES ON *.* TO 'sst'@'%' IDENTIFIED BY 'secret';
  GRANT ALL PRIVILEGES ON *.* TO 'sst'@'localhost' IDENTIFIED BY 'secret';
  ```
## Configuration

- **my.cnf (galera.conf)**

```plaintext
<table>
<thead>
<tr>
<th>Configuration Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>default_storage_engine</td>
<td>InnoDB</td>
</tr>
<tr>
<td>binlog_format</td>
<td>row</td>
</tr>
<tr>
<td>innodb_autoinc_lock_mode</td>
<td>2</td>
</tr>
<tr>
<td>innodb_locks_unsafe_for_binlog</td>
<td>1</td>
</tr>
<tr>
<td>innodb_flush_log_at_trx_commit</td>
<td>0</td>
</tr>
<tr>
<td>innodb_doublewrite</td>
<td>0</td>
</tr>
<tr>
<td>query_cache_size</td>
<td>0</td>
</tr>
<tr>
<td>query_cache_type</td>
<td>0</td>
</tr>
<tr>
<td># wsrep_provider</td>
<td>none</td>
</tr>
<tr>
<td>wsrep_provider</td>
<td>../lib/plugin/libgalera_smm.so</td>
</tr>
<tr>
<td># wsrep_cluster_address</td>
<td>gcomm://192.168.42.3</td>
</tr>
<tr>
<td>wsrep_cluster_address</td>
<td>gcomm://</td>
</tr>
<tr>
<td>wsrep_sst_method</td>
<td>mysqlldump</td>
</tr>
<tr>
<td>wsrep_sst_auth</td>
<td>sst:secret</td>
</tr>
</tbody>
</table>
```
Start nodes

- First node:
  
  /etc/init.d/mysql start

- Other nodes:

  wsrep_cluster_address = "gcomm://192.168.42.1"
  
  /etc/init.d/mysql start

- Check with

  - MySQL error log (on this AND remote node!)
  - SHOW GLOBAL STATUS;
Checks

120131  07:37:17  mysqld_safe Starting mysqld daemon
...
120131  7:37:18 [Note] WSREP: wsrep_load(): loading provider library 'libgalera_smm.so'
120131  7:37:18 [Note] WSREP: Start replication
...
120131  7:37:18 [Note] WSREP: Shifting CLOSED -> OPEN (TO: 0)
...
120131  7:37:23 [Note] WSREP: Quorum results:
    conf_id   = 2,
    members   = 3/3 (joined/total)

SHOW GLOBAL STATUS LIKE 'wsrep*';
+----------------------------+--------------------------------------+
| Variable_name              | Value                                |
+----------------------------+--------------------------------------+
| wsrep_local_state_comment  | Synced (6)                           |
| wsrep_cluster_conf_id      | 3                                    |
| wsrep_cluster_size         | 3                                    |
| wsrep_cluster_status       | Primary                              |
| wsrep_connected            | ON                                   |
| wsrep_local_index          | 0                                    |
| wsrep_ready                | ON                                   |
+----------------------------+--------------------------------------+
Snapshot State Transfer (SST)

- Initial full sync between 1st and other nodes
  → mysqldump, rsync, (xtrabackup?, LVM?)
- Blocks the Donor! (→ 3 nodes)
- With v2.0 there is an Incremental State Transfer (IST)

```
120131 16:26:42 [Note] WSREP: Quorum results:
    conf_id    = 4,
    members    = 2/3 (joined/total)
120131 16:26:44 [Note] WSREP: Node 2 (Node 3) requested state transfer from '*any*'.
    Selected 0 (Node 1)(SYNCHED) as donor.
120131 16:26:44 [Note] WSREP: Shifting SYNCHED -> DONOR/DESYNCHED (TO: 2695744)
120131 16:27:10 [Note] WSREP: 2 (Node 3): State transfer from 0 (Node 1) complete.
120131 16:27:10 [Note] WSREP: Member 2 (Node 3) synced with group.
120131 16:27:10 [Note] WSREP: Shifting DONOR/DESYNCHED -> JOINED (TO: 2695744)
120131 16:27:10 [Note] WSREP: Member 0 (Node 1) synced with group.
120131 16:27:10 [Note] WSREP: Shifting JOINED -> SYNCHED (TO: 2695744)
120131 16:27:10 [Note] WSREP: Synchronized with group, ready for connections
```
Restarting a node

- 2\textsuperscript{nd} and 3\textsuperscript{rd} node → no problem
- 1\textsuperscript{st} node:

```
# wsrep_cluster_address = "gcomm://192.168.42.3"
wsrep_cluster_address = "gcomm://"
```

→ This is IMHO non optimal because we have 2 different situations:

- Initial 1\textsuperscript{st} node start
- 1\textsuperscript{st} node restart
### Variables

#### Currently (1.1) 27 Variables

<table>
<thead>
<tr>
<th>Variable_name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>wsrep_cluster_address</td>
<td>gcomm://</td>
</tr>
<tr>
<td>wsrep_cluster_name</td>
<td>Galera-1.0 wsrep-21</td>
</tr>
<tr>
<td>wsrep_max_ws_rows</td>
<td>131072</td>
</tr>
<tr>
<td>wsrep_max_ws_size</td>
<td>1073741824</td>
</tr>
<tr>
<td>wsrep_node_incoming_address</td>
<td>192.168.42.1:3306</td>
</tr>
<tr>
<td>wsrep_node_name</td>
<td>Node 1</td>
</tr>
<tr>
<td>wsrep_notify_cmd</td>
<td></td>
</tr>
<tr>
<td>wsrep_on</td>
<td>ON</td>
</tr>
<tr>
<td>wsrep_provider</td>
<td>.../plugin/libgalera_smm.so</td>
</tr>
<tr>
<td>wsrep_retry_autocommit</td>
<td>1</td>
</tr>
<tr>
<td>wsrep_slave_threads</td>
<td>1</td>
</tr>
<tr>
<td>wsrep_sst_auth</td>
<td>******</td>
</tr>
<tr>
<td>wsrep_sst_donor</td>
<td></td>
</tr>
<tr>
<td>wsrep_sst_method</td>
<td>mysqldump</td>
</tr>
<tr>
<td>wsrep_sst_receive_address</td>
<td>AUTO</td>
</tr>
</tbody>
</table>
wsrep_provider_options

- evs.debug_log_mask = 0x1;
- evs.inactive_check_period = PT0.5S
- evs.inactive_timeout = PT15S;
- evs.info_log_mask = 0;
- evs.install_timeout = PT15S;
- evs.join_retrans_period = PT0.3S;
- evs.keepalive_period = PT1S;
- evs.max_install_timeouts = 1;
- evs.send_window = 4;
- evs.stats_report_period = PT1M;
- evs.suspect_timeout = PT5S;
- evs.use_aggregate = true;
- evs.user_send_window = 2;
- evs.version = 0;
- evs.view_forget_timeout = PT5M;
- gcache.dir = ...
- gcache.keep_pages_size = 0;
- gcache.mem_size = 0;
- gcache.name = .../galera.cache;
- gcache.page_size = 128M;
- gcache.size = 128M;
- gcs.fc_debug = 0;
- gcs.fc_factor = 0.5;
- gcs.fc_limit = 16;
- gcs.fc_master_slave = NO;
- gcs.max_packet_size = 64500;
- gcs.max_throttle = 0.25;
- gcs.recv_q_hard_limit = 922372036854775807;
- gcs.recv_q_soft_limit = 0.25;
- gmcast.listen_addr = tcp://127.0.0.1:4567;
- gmcast.mcast_addr = ;
- gmcast.mcast_ttl = 1;
- gmcast.peer_timeout = PT3S;
- gmcast.time_wait = PT5S;
- gmcast.version = 0;
- pc.checksum = true;
- pc.ignore_quorum = false;
- pc.ignore_sb = false;
- pc.linger = PT2S;
- pc.npvo = false;
- pc.version = 0;
- protonet.backend = asio;
- protonet.version = 0;
- replicator.commit_order = 3
Status

- Currently (1.1) 38 Status information
- SHOW GLOBAL STATUS LIKE 'wsrep%';
  - Cluster status
  - Performance metrics
  - General information

+----------------------------+--------------------------------------+
| Variable_name              | Value                                |
+----------------------------+--------------------------------------+
| wsrep_last_committed       | 2695744                              |
| wsrep_replicated           | 1                                    |
| wsrep_replicated_bytes     | 576                                  |
| wsrep_received             | 9                                    |
| wsrep_received_bytes       | 1051                                 |
| wsrep_local_commits        | 1                                    |
| wsrep_local_send_queue     | 0                                    |
| wsrep_local_recv_queue     | 0                                    |
| wsrep_flow_control_sent    | 0                                    |
| wsrep_flow_control_recv    | 0                                    |
| wsrep_provider_version     | 22.1.1(r95)                          |
+----------------------------+--------------------------------------+
Load Balancing

- In your Application (on your own)
- Connectors
  - Connector/J
  - PHP: MySQLnd replication and load balancing plug-in
- SW Load Balancer
  - GLB, Pen, LVS, HAProxy, MySQL Proxy, SQL Relay,
- HW Load Balancer
Operations

• 2 Modes:
  • Master-Master
  • Master-Slave

• Initial configuration (do not mess it up)

• SST (DB size, NW bandwidth (WAN))

• Start / restart

• Deadlocks and hot spots
Galera Replication

- Graph from Vadim Tkachenko (Percona):
  http://www.mysqlperformanceblog.com/2012/01/19/percona-xtradb-cluster-feature-2-multi-master-replication/
Questions?

Discussion?

We have some time for face-to-face talks...

- FromDual provides neutral and independent:
  - Consulting
  - Remote-DBA
  - Support for MySQL, Galera, Percona Server
  - Training

www.fromdual.com