Galera in MariaDB 10.4

State of the Art and Plans

Seppo Jaakola
Codership
➢ Seppo Jaakola
➢ One of the Founders of Codership

➢ Codership – Galera Replication developers
➢ Partner of MariaDB for developing and supporting MariaDB Galera Cluster
➢ Galera releases since 2009
Agenda

- Galera in 10.4 Status
- Galera Cluster Upgrading
- Streaming Replication
- XA Transaction Support
- Spider Cluster
Galera in 10.4 and Beyond

Galera 4.0

- Group Commit Support
- Non Blocking DDL
- Huge transactions by streaming replication
- Inconsistency Voting Protocol

MariaDB 10.4

- Gcache Encryption
- MariaDB GTID Compatibility

Galera 4.1

- XA transaction Support
- Spider Cluster
Galera Upgrade

wsrep API Change
Galera Rolling Upgrades

Upgrade with API #26
Galera Rolling Upgrades

One node upgraded to API #26.

Upgrade with API #26.

MariaDB API 25

MariaDB API 25

MariaDB API 26

Galera Replication API 25
Galera Rolling Upgrades

Upgrade with API #26

MariaDB API 25

read & write

MariaDB API 26

read & write

MariaDB API 26

One node upgraded To API #26

Galera Replication API 25
Galera Rolling Upgrades

All nodes upgraded to API #26

API #26 features now enabled in replication
Streaming Replication

Huge Transaction Support
Huge Transaction Demo Setup

1. Two nodes
2. Steady load of pure autocommit updates to measure trx throughput
3. A huge table with ~1.5M rows
4. Run update on huge table to modify all rows
   • → monitor trx/sec rate in the cluster when the huge transaction kicks in
Impact of Huge Transaction

Huge Transaction Slave Lag

Trx in master 24 secs

Trx in slave 9 secs
Streaming Replication

- Transaction is replicated, gradually in small fragments, during transaction processing
  - i.e. before actual commit, we replicate a number of small scale fragments
- Size threshold for fragment replication is configurable
- Replicated fragments are applied in slave transactions in all cluster nodes
  - Fragments hold locks in all nodes and cannot be conflicted later
Streaming Replication

Huge transaction

Update, update, update....

Node A

Huge trx

Node B

Galera Replication
Streaming Replication

Huge transaction

Node A

Huge trx

Node B

Update, update, update....

Galera Replication
Streaming Replication

Huge transaction

Node A

Node B

Huge trx

Update, update, update....

Galera Replication
Streaming Replication

Huge transaction

commit

Node A

Huge trx

Node B

Galera Replication

WS
Fragment Transaction

SR transaction pool

applier

applier

applier

WS
SR trx :2
CF: 0

certification
Fragment Transaction

SR transaction pool

applier

applier

WS
SR trx : 2
CF: 0

SR#2 THD

ev->apply_event()
...

SR#1 THD

SR#n THD

ev->apply_event()

wsrep_SR_store->append_frag_apply()
Fragment Transaction

WS
SR trx :2
CF: 1
certification

SR transaction pool

SR#1 THD
SR#2 THD
SR#n THD

applier
applier
applier
Fragment Transaction

applier

applier

WS
SR trx:2
CF: 1

SR#2 THD

trans_commit()

wsrep_SR_store->append_frag_commit()
# Configuring Streaming Replication

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>wsrep_trx_fragment_unit</code></td>
<td>Unit metrics for fragmenting, options are:</td>
</tr>
<tr>
<td></td>
<td>• bytes  WS size in bytes</td>
</tr>
<tr>
<td></td>
<td>• events  # of binlog events</td>
</tr>
<tr>
<td></td>
<td>• rows    # of rows modified</td>
</tr>
<tr>
<td></td>
<td>• statements  # of SQL statements issued</td>
</tr>
<tr>
<td><code>wsrep_trx_fragment_size</code></td>
<td>• Threshold size (in units), when fragment will be replicated</td>
</tr>
<tr>
<td></td>
<td>• 0 = no streaming</td>
</tr>
</tbody>
</table>
Streaming Replication Demo Setup

1. Same scenario as before

2. Configure node1 to fragment huge transaction in 10K batches
   - `wsrep_trx_fragment_unit = bytes`
   - `wsrep_trx_fragment_size = 10000`

monitor trx/sec rate in the cluster when streaming replication progresses
Streaming Replication

![Graph showing streaming replication performance over time with a peak of 70 secs.]
Streaming Replication
XA Transactions with Galera 3
XA Transaction Support

Node A

XA trans

smith

Node B

smith

XA Start
XA Insert into persons 'jones'

Node A

XA trans
insert

Smith

Node B

Smith
XA Transaction Support

XA Prepare

Node A

XA trans
insert

smith

Node B

smith
XA Transaction Support

XA Prepare

Node A

XA trans

insert

smith

Node B

smith

WS
XA Transaction Support
XA Transaction Support

Node A

XA trans

insert

smith

Node B

smith

jones
XA Transaction Support

Node A

smith

Node B

smith
jones
XA by Streaming Replication
XA Transaction Support

Node A

XA trans

smith

Node B

smith

WS
XA Transaction Support

- XA Insert into persons 'jones'
- Node A
  - XA trans
  - insert
  - smith
- Node B
  - SR trans
  - insert
  - smith
- WS
XA Transaction Support

XA Prepare

Node A
- XA trans insert
- smith

Node B
- SR trans insert
- smith
XA Transaction Support

Node A
- XA trans
- insert
- smith

Node B
- SR trans
- insert
- smith
- jones

XA Rollback

WS rollback
XA Transaction Support

Node A

Node B

smith

smith
Spider Cluster
Insert into t values....

Spider SE

XA

Table t

A

B

Shard A

Shard B
Insert into t values....

Spider SE

Table t

A
B

Node 1
Node 2
Node 3

Node 1
Node 2
garbd

Galera

Galera
Insert into t values....
Insert into t values (1),(2)

shard A

XA start

Insert (1)

XA prepare

XA Commit

shard B

XA start

Insert (2)

XA prepare

XA Commit
Insert into t values (1),(2)

shard A

XA start

Insert (1)

XA prepare

XA Commit

shard B

Select * from t

XA start

Insert (2)

XA prepare

XA Commit

(1) = (1)

(1) = (1), (2)
backport fix for mysql
bug#12161 (XA and binlog)

Details

Type: Task  Status: IN PROGRESS
Priority: Major
Resolution: Unresolved
Component/s: None  Fix Version/s: 10.4
Labels: upstream-fixed
Epic Link: Replication Enhancements

Description

5.7 finally fixes the 10-yr-old bug#12161 — a.k.a. prepared XA transactions are lost on disconnect. They solved it by introducing a new XA_prepare_log_event. As we'll need to be able to read this event, we can as well merge the whole fix for this bug.

Issue Links

is duplicated by

- [MDEV-742](https://example.com/mdev-742) LP:803649 - Xa recovery failed on client disconnect...

links to

- Bug #12161 Xa recovery and client disconnection

Activity

- [Elena Stepanova](https://example.com/elenastepanova) added a comment - 2017-01-24 12:04

  Is it still possible to do it in 10.2? I'll set it to 10.2-ga to get on the radar, but feel free to unset it if it can't be done.

People