Pitfall! Replication Hazards

Choosing the right replication strategy for your apps

Version 1.0
Updated May 2019

Get Comfortable!

Demo Starts

3
Get Comfortable!

Demo Starts

2
Get Comfortable!

Demo Starts

1
What Just Happened?

**Same App, Different Results**
- Galera handles deadlocks differently than single-node
- This talk is about replication complexity
  - And why simpler is often better

**Andrew Garner**
- 9+ years, DBA and Data consultant
- Software Engineer

**Marco Nicosia**
- Product Manager
- 4 years on MySQL for PCF
Agenda

- Demo 1: Deadlocks
- Goal of this Talk
- Demo 2: Galera Strengths
- Demo 3: Troubles with DDLs
- Demo 4: Troubles with DDLs
- Demo 5: Eventually Consistent
- Demo 6: Divergent Data
- Mitigations
CONVINCE YOU THAT REPLICATION IS GREAT

... when used sparingly
Why Show Us These Videos?

All MySQL replication strategies have trade-offs

- We’re here to demonstrate those effects
- So that you can be informed when you make choices
- You can partner with your Developers to build good software
Why Intermediate?

- Not how to install & use
- Not advanced tuning
- “Make an informed decision”
- Share our experiences with MySQL replication

Not an advertisement
Agenda

- Demo 1: Deadlocks
- Goal of this Talk
- Demo 2: Galera Strengths
- Demo 3: Troubles with DDLs
- Demo 4: Troubles with DDLs
- Demo 5: Eventually Consistent
- Demo 6: Divergent Data
- Mitigations

Pivotal
Demo #2
Galera HA

Galera Multi-Master
Galera Clusters

Advertised as Multi-Master MySQL

Can be a great HA solution

... at the cost of complexity.
Why Love Galera?

- Fully automates replication
  - No slave set up
- Sub-second failover
- Rolling upgrades!
- Automatic node recovery
Agenda

✓ Demo 1: Deadlocks
✓ Goal of this Talk
✓ Demo 2: Galera Strengths

❏ Demo 3: Troubles with DDLs
❏ Demo 4: Troubles with DMLs
❏ Demo 5: Eventually Consistent
❏ Demo 6: Crashing Galera

❏ Mitigations

Pivotal
Demo #3
Large DDLs

Concepts:

- DDL modes
- TOI vs. RSU

Learn more:
MariaDB [ddl.db]>
MariaDB [ddl.db]> alter table ddl_table add column (thirdCol CHAR(100) DEFAULT 'yyy')
Query OK, 0 rows affected (4 min 0.32 sec)
Records: 0  Duplicates: 0  Warnings: 0
MariaDB [ddl.db]>
What Just Happened?

A moderate DDL blocked all writes to the cluster.

● One writer issued a large (~4min) DDL
● All other writers fully blocked
● Total Order Isolation serializes all DDLs
● AND blocks all commits
Agenda

✓ Demo 1: Deadlocks
✓ Goal of this Talk
✓ Demo 2: Galera Strengths
✓ Demo 3: Troubles with DDLs
☑ Demo 4: Troubles with DMLs
☑ Demo 5: Eventually Consistent
☑ Demo 6: Divergent Data
☑ Mitigations
Demo #4
Large DMLs

Concepts:

- Replication Lag
- Flow Control

Learn more:
http://galeracluster.com/documentation-webpages/nodestates.html
```sql
mysql> DELETE FROM t LIMIT 10000;
Query OK, 10000 rows affected (0.21 sec)
```

### Direct Reader

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>25</td>
</tr>
<tr>
<td>name</td>
<td>John</td>
</tr>
<tr>
<td>age</td>
<td>30</td>
</tr>
<tr>
<td>name</td>
<td>Mary</td>
</tr>
<tr>
<td>age</td>
<td>28</td>
</tr>
<tr>
<td>name</td>
<td>Lisa</td>
</tr>
</tbody>
</table>

### Direct Reader

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Smith</td>
</tr>
<tr>
<td>age</td>
<td>20</td>
</tr>
<tr>
<td>name</td>
<td>Jones</td>
</tr>
<tr>
<td>age</td>
<td>22</td>
</tr>
<tr>
<td>name</td>
<td>Brown</td>
</tr>
<tr>
<td>age</td>
<td>24</td>
</tr>
</tbody>
</table>

What Just Happened?

A moderate DML caused a follower to return stale results.

- We deleted 10,000 rows on a “simple” table
- The follower has to do unnecessary work
  - Transactions on the Leader carry on
- Applying on the Follower can take longer than the Leader
- Stale data continues even after the Follower has applied, because it has fallen behind
What Just Happened?

A moderate DML blocked all other writers on the master.

- One active writer committed a large DML, adding 1 million rows
- Two nodes fell significantly behind
- All other writers were blocked
- Cluster snaps back to normal
Agenda

- Demo 1: Deadlocks
- Goal of this Talk
- Demo 2: Galera Strengths
- Demo 3: Troubles with DDLs
- Demo 4: Troubles with DMLs
- Demo 5: Eventually Consistent
- Demo 6: Crashing Galera
- Mitigations
Demo #5
Eventual Consistency

Concepts:

- Replication Lag
- Flow Control

Learn more:
http://galeracluster.com/documentation-webpages/nodestates.html
mysql> ALTER TABLE t1 ADD COLUMN id int PRIMARY KEY AUTO_INCREMENT;
Query OK, 0 rows affected (1 min 46.15 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SELECT ts FROM SampleData ORDER BY ts DESC LIMIT 1;
+----+
| ts |
+----+
+----+
1 row in set (0.49 sec)

mysql> SELECT ts FROM SampleData ORDER BY ts DESC LIMIT 1;
+----+
| ts |
+----+
| 2019-05-27 20:30:58 |
+----+
1 row in set (0.70 sec)

MySQL Writers Viewer

Follower Lag Viewer
What Just Happened?

A moderate DDL caused a follower to return stale results.

- Adding a single column
- Follower falls behind Leader
- Stale data even after the Follower has applied the DDL
What Just Happened?

A DDL caused inconsistent reads & writes across a Galera cluster

- Single writer issued large DDL
- Another node falls behind

```
2018-04-25 7:43:43 140546106472192 [Note] WSREP: Member 2.0 (mysql/2) desyncs itself from group
```

```
```

- Subsequent DDLs take time to propagate
- We’ve observed times greater than 1 hour
Agenda

✓ Demo 1: Deadlocks
✓ Goal of this Talk
✓ Demo 2: Galera Strengths
✓ Demo 3: Troubles with DDLs
✓ Demo 4: Troubles with **DMLs**
✓ Demo 5: Eventually Consistent

☐ Demo 6: Divergent Data
☐ Mitigations
Demo #6
Divergent Data
mysql> select @global.gtid_executed;
    
    +-----------------------------+
    | @global.gtid_executed      |
    | 7450158e-7681-11e9-9a19-006bd50538b8:1-913782 |
    +-----------------------------+
    1 row in set (0.00 sec)

mysql>

Master_SSL_Certpath:
  @BB: 9b202ca-9132804
  Executed_Gtid_Set: 7450158e-7681-11e9-9a19-006bd50538b8:1-913284
  @BB: 1-913284
  Auto_Position: 1
  Replicate_Rewrite_DB:
    Channel_Name:
      Master_TLS_Version: TLSv1.2
  1 row in set (0.00 sec)

mysql>

mysql> FLUSH STATUS;
Query OK, 0 rows affected (0.00 sec)

mysql> select ;
What Just Happened?

Running a seemingly innocent command caused GTID divergence

● Innocent queries can generate transactions on the follower.
● The implication is that switchover won’t work because the leader is now missing a transaction, so I can’t just trivially promote my follower.
● Looking at gtids the follower obviously now has some inconsistency.
● In a month I may not have binlogs to really know what that change even was. Was it flush logs or did something change data I actually care about?
● Lesson: be careful about queries run on the follower.
Synch vs. Certification

Myth: Multiple Copies of Data

Fact:
Guarantees a writeset will commit
Apply happens after certification
Variable_name | Value
--- | ---
wsrep_cluster_status | Primary

1 row in set (0.02 sec)

MariaDB [ddl_db] > show global status like 'wsrep_cluster_size';

<table>
<thead>
<tr>
<th>Variable_name</th>
<th>Value</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>wsrep_cluster_size</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

1 row in set (0.00 sec)

MariaDB [ddl_db] > 

20 Minutes Pass...

MySQL1

MySQL2

MySQL0

Net Secure: https://0-proxy-a-mysql.com:15000

3 out of 3 nodes are unhealthy.

NODES | STATUS | CURRENT_SESSIONS | IP ADDRESS
--- | --- | --- | ---
backend-0 | HEALTHY | 2 | 10.0.4.56
backend-1 | UNHEALTHY | 0 | 10.0.4.57
backend-2 | 0 | 10.0.4.58

State transfer to 1.0 (mysql/0) complete.
2017-09-23 4:45:50 140837667805952 [Note] WSREP: 2.0 (mysql/0): State transfer from "any" to "mysql/0" complete.
2017-09-23 4:45:50 140837667805952 [Note] WSREP: Member 2.0 (mysql/0) synced with group.
2017-09-23 4:45:50 140837667805952 [Note] WSREP: Member 2.0 (mysql/0) synced with group.
2017-09-23 4:44:45 1400012494505728 [Note] WSREP: Member 2.0 (mysql/0) requested state transfer from "any". Selected 2.0 (mysql/0) (SYNCED) as donor.
2017-09-23 4:45:50 1400012494505728 [Note] WSREP: Member 2.0 (mysql/0): State transfer to 1.0 (mysql/0) complete.
2017-09-23 4:45:50 1400012494505728 [Note] WSREP: Member 2.0 (mysql/0) synced with group.
What Just Happened?

A failed DDL caused data divergence

- Two nodes had nearly full disks
- Single writer issued large DDL
  - DDL immediately failed on other two nodes
  - Leaving incompatible schema
- Single writer eventually issued incompatible DML
- Two nodes immediately crash and SST
  - Single copy of DB for duration of SST
Conclusion

Mitigations
mysql>
mysql> DELETE FROM t5 LIMIT 10000;
Query OK, 10000 rows affected (8.21 sec)

mysql> ALTER TABLE t2 ADD COLUMN id INT PRIMARY KEY AUTO_INCREMENT;
Query OK, 0 rows affected (0.29 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SELECT id, data FROM t2
Query OK, 52244 rows affected (8.71 sec)
Records: 52244  Duplicates: 0  Warnings: 0

mysql> DELETE FROM t2 ORDER BY data DESC LIMIT 10000;

Every 1.0s: mysql --default-file=/var/vcap/jobs/mysql/...
Sun May 25 00:45:15 2013

Seconds_Behind_Master: 0
Mitigations

Technology
- Restrict Galera traffic to 1 node!
- Dedicated clusters
- Force InnoDB only
- Disable table locks
- Force primary keys
- Tune Galera
- Maintain headroom

Education
- TOI vs. RSU
- Max txn size
- Deadlocks
- Primary keys
- Critical Reads
- Flow control
# What Have We Learned

## World’s most advanced features and Un-seen benefits

- True Multi-master, Active-Active Cluster: Read and write to any node at any time.
- Synchronous Replication: No slave lag, no data is lost at node crash.
- Tightly Coupled: All nodes hold the same state. No diverged data between nodes allowed.
- Multi-threaded Slave: For better performance. For any workload.
- No Master-Slave Failover Operations or Use of VIP.
- Hot Standby: No downtime during failover (since there is no failover).
- Automatic Node Provisioning: No need to manually back up the database and copy it to the new node.
- Supports InnoDB.
- Transparent to Applications: Required no (or minimal changes) to the application.
- No Read and Write Splitting Needed.
- Easy to Use and Deploy.

## Things to be aware of

- Multi-Master means eventually consistent.
- No slave lag: true, but eventually consistent for reads.
- No VIP: If you use a Load Balancer.
- Supports InnoDB: nasty surprises.
- No R/W split: if you know what you are doing.

Conclusions

- Galera is powerful
- Best when single-tenant
- Not standard MySQL
- Developers must be Galera-aware
Special Thanks

- Our Customers
- Laine Campbell
- The Percona support team
- Shatarupa Nandi
- Ben Laplanche
- Andrew Garner
- Joseph Palermo

- Samuel Serrano
- Morgan Fine
- Urvashi Reddy
- Difan Zhao
- Rob Dimsdale
- Christopher Hendrix
- Andrew Crump
Thank You!

Cloud Foundry is OSS

https://www.cloudfoundry.org/

GitHub: https://github.com/cloudfoundry/cf-deployment

Cloud Foundry MySQL: https://github.com/cloudfoundry/cf-mysql-deployment

PXC release:
https://github.com/cloudfoundry-incubator/pxc-release

Presenters: Marco Nicosia, Andrew Garner

- Pivotal Labs, Data and Cloud Foundry
- Over 2,000 employees in more than 20 locations globally
- Key customers include Comcast, Allstate, Ford, Citi, GE, Southwest, Verizon.
- Investors include GE, Dell, Ford, and Microsoft.

We’re Hiring:
San Francisco, New York, Toronto, Dublin, and London