Global Transaction ID (GTID) Replication Implementation and Troubleshooting

Percona Live London 2014

Abdel-Mawla Gharieb
MySQL Support Engineer
FromDual GmbH

abdel-mawla.gharieb@fromdual.com
About FromDual GmbH

- FromDual provides neutral and independent:
  - Consulting for MySQL, Galera Cluster, MariaDB and Percona Server
  - Support for all MySQL and Galera Cluster
  - Remote-DBA Services for all MySQL
  - MySQL Training
- Open Source Business Alliance (OSBA)
- Member of SOUG, DOAG, /ch/open

www.fromdual.com
GTID Replication

➢ What is GTID?
➢ GTID benefits
➢ GTID Important Variables.
➢ GTID Replication Implementation.
➢ GTID fail-over practical example.
➢ Migration to GTID Replication.
➢ GTID Replication Troubleshooting.
What is GTID?
What is GTID?

GTID is a global transaction identifier which consists of two parts separated by a column:

{source_id:transaction_id}

source_id: server's UUID.

transaction_id: sequence number.

b9b4712a-df64-11e3-b391-60672090eb04:3
GTID Benefits
GTID Benefits

- Setting up MySQL replication is so simple now!
- Consistency is guaranteed between master and slaves.
- Simple to determine inconsistency.
- Simple to do Point in Time Recovery (PiTR).
- Fail-over process become much easier.
- Automatic fail-over scripts are easy to implement now.
GTID Benefits

Master A

Master B

Slave

Binlog.298 9837462

Binlog.169 176534

Binlog.298 9837462

Master_log_file=??
Master_log_pos=??

Binlog.17 837465
GTID Benefits

Master A

Master_Auto_Position=1

Slave

Master B

UUID_A:252

UUID_A:252

UUID_A:252
GTID Important Variables
GTID Important Variables

- **gtid-mode**: ON|OFF

- **enforce-gtid-consistency**: Prevent executing the non-transactionally safe statements.
  - CREATE TABLE ... SELECT statements.
  - CREATE TEMPORARY TABLE statements inside transactions.
  - Transactions or statements that update both transactional and nontransactional tables.
GTID Important Variables

- **gtid-purged**: transactions have been purged from the binary logs.
- **gtid-executed**: transactions already executed on the server.
- **gtid-next**: GTID for the next transaction.
GTID Replication Implementation (Fresh installation)
GTID Replication Implementation

- Master's side configurations.
- Slave's side configurations.
Master's side configurations

- Add the following variables to my.cnf:

```
[mysqld]
server-id = 1
log-bin = mysql-bin

gtid_mode = on
enforce_gtid_consistency
log_slave_updates
```
Master's side configurations

- Restart MySQL to apply the configuration changes:

  shell> service mysql restart

- Create a MySQL user to be used by the slave:

  SQL> GRANT REPLICATION SLAVE ON *.* TO 'slave_user_name'@'slave_ip' IDENTIFIED BY 's3cret';
Slave's side configurations

- Add the following variables to my.cnf:

```
[mysqld]
server_id = 2
log_bin = mysql-bin

gtid_mode = on
enforce_gtid_consistency
log_slave_updates
```
Slave's side configurations

- Restart MySQL to apply the configuration changes:

  shell> service mysql restart

- Execute the CHANGE MASTER TO command:

  SQL> CHANGE MASTER TO
  MASTER_HOST='master_ip',
  MASTER_PORT=3306,
  MASTER_USER='slave_user_name',
  MASTER_PASSWORD='s3cret',
  MASTER_AUTO_POSITION=1;
Slave's side configurations

- Start the replication:

```sql
SQL> START SLAVE;
```
Checking the replication!!

- Check the replication status:

```sql
SQL> SHOW SLAVE STATUS\G
Slave_IO_State: Waiting for master to send event
  Master_Host: 127.0.0.1
  Master_User: gtid_repl
  Master_Port: 3320
  ...
  Slave_IO_Running: Yes
  Slave_SQL_Running: Yes
  ...
Retrieved_Gtid_Set: b9b4712a-df64-11e3-b391-60672090eb04:1-2
Executed_Gtid_Set: b9b4712a-df64-11e3-b391-60672090eb04:1-2
  Auto_Position: 1
```
Adding new slave to a GTID Replication
Adding new slave to GTID Replication

- Backup the master server:

```shell
shell> mysqldump -u root -p --all-databases --flush-privileges --single-transaction --flush-logs --triggers --routines --events --hex-blob > /path/to/backupdir/full_backup-$TIMESTAMP.sql
```

- Start MySQL service having GTID variables added as described earlier.
- Restore the master backup file on the slave.
- Use change master to with `MASTER_AUTO_POSITION=1`
- Start the slave.
Adding new slave to GTID Replication

- mysqldump knows about GTID !!

```
SET @@GLOBAL.GTID_PURGED='b9b4712a-df64-11e3-b391-60672090eb04:1-7';
```
Migration to GTID Replication
Migration to GTID

- Make the Master server read only: `SET GLOBAL read_only=ON`;
- Allow all slaves to catch up with the master.
- Shutdown MySQL service on ALL servers.
- Add the GTID variables.
- Add `read_only` in the master's and `skip_slave_start` in the slave's.
- Start MySQL service on ALL servers.
- Issue `CHANGE MASTER .. MASTER_AUTO_POSITION = 1;` on all slaves and start them.
- Make the master writeable `SET GLOBAL read_only=OFF;`
Migration to GTID

Is Online migration from Classic Replication to Transaction-based Replication available?

NOT yet!!

Why?

- GTID_MODE is a read only variable.
- Different GTID_MODE values among replication servers is not acceptable.
Replication Troubleshooting
Replication Troubleshooting

- Skip or ignore a transaction.
- Re-initialize or re-build a broken slave.
Skip a transaction
Sample errors (from SHOW SLAVE STATUS output):

Last_SQL_Error: Could not execute **Write_rows event** on table test.t1; Duplicate entry '4' for key 'PRIMARY', Error_code: 1062; handler error HA_ERR_FOUND_DUPP_KEY; the event's master log mysql-bin.000304, end_log_pos 285

Last_SQL_Error: Could not execute **Update_rows event** on table test.t1; Can't find record in 't1', Error_code: 1032; handler error HA_ERR_KEY_NOT_FOUND; the event's master log mysql-bin.000304, end_log_pos 492

Last_SQL_Error: Could not execute **Delete_rows event** on table test.t1; Can't find record in 't1', Error_code: 1032; handler error HA_ERR_KEY_NOT_FOUND; the event's master log mysql-bin.000304, end_log_pos 688
Skip a transaction

How to solve the problem?

- Check which transaction is causing the problem:

  SQL> SHOW SLAVE STATUS\G
  .
  Retrieved_Gtid_Set: b9b4712a-df64-11e3-b391-60672090eb04:1-7
  Executed_Gtid_Set : b9b4712a-df64-11e3-b391-60672090eb04:1-6
  Auto_Position: 1

- Inject an empty transaction:

  SQL> SET GTID_NEXT='b9b4712a-df64-11e3-b391-60672090eb04:7';
  SQL> BEGIN;COMMIT;
  SQL> SET GTID_NEXT='AUTOMATIC';
  SQL> START SLAVE;
Re-initialize a broken slave
Re-initialize a slave

Sample error:

```
Last_IO_Errno: 1236
Last_IO_Error: Got fatal error 1236 from master when reading data from binary log:
'The slave is connecting using CHANGE MASTER TO MASTER_AUTO_POSITION = 1,
but the master has purged binary logs containing GTIDs that the slave requires.'
```
Re-initialize a slave

How to solve the problem?

• Backup the master's database:

```
shell> mysqldump -u root -p --all-databases --flush-privileges --single-transaction --master-data=2 --flush-logs --triggers --routines --events --hex-blob /path/to/backupdir/full_backup-$TIMESTAMP.sql
```

```
shell> head -n 50 /path/to/backupdir/full_backup-$TIMESTAMP.sql | grep PURGED
SET @@GLOBAL.GTID_PURGED='b9b4712a-df64-11e3-b391-60672090eb04:1-8';
```

• Restore the backup on the broken slave:

```
shell> mysql -u root -p < /path/to/backupdir/full_backup-$TIMESTAMP.sql
ERROR 1840 (HY000): @@GLOBAL.GTID_PURGED can only be set when @@GLOBAL.GTID_EXECUTED is empty
```

Oppppps!!!
Re-initialize a slave

- Empty the GTID_EXECUTED on the slave:

  SQL> RESET MASTER;

- Restore the backup again:

  shell> mysql -u root -p < /path/to/backupdir/full_backup-$TIMESTAMP.sql

- Make sure that the value of GTID_EXECUTED is correct:

  SQL> SHOW GLOBAL VARIABLES LIKE 'gtid_executed';
  +---------------+------------------------------------------+
  | Variable_name | Value                                    |
  +---------------+------------------------------------------+
  | gtid_executed | b9b4712a-df64-11e3-b391-60672090eb04:1-8 |
  +---------------+------------------------------------------+
  1 row in set (0.00 sec)
Re-initialize a slave

• Now we can start the slave:

SQL> START SLAVE;
Q & A

Questions?

Discussion?

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

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

www.fromdual.com/presentations