DBAHire.com

MySQL Fabric: High Availability Solution for Connector/Python

Jaime Crespo

PyConES 2014 Zaragoza

-8 Nov 2014-

dbahire.com

Table of Contents

MySQL Fabric: High Availability Solution for Connector/Python

WHAT IS MYSQL FABRIC?

This is Me Fighting Python Programmers that Write Poor SQL Queries

- MySQL Consultant at DBAHire.com
- Used to work for Oracle (MySQL), Percona
- Loves MySQL query optimization and HA

Raise Your Hands

- Who uses here MySQL for some of his/her applications?
- Who has had performance problems with his/her database before?
- Who had had suffered availability problems because hardware/software failures?

There Are Many Solutions for MySQL HA

- DRBD and other active-passive, shared-storage solutions
- Standard Master-Slave replication
- MySQL NDB Cluster
- Galera/Percona XtraDB Cluster

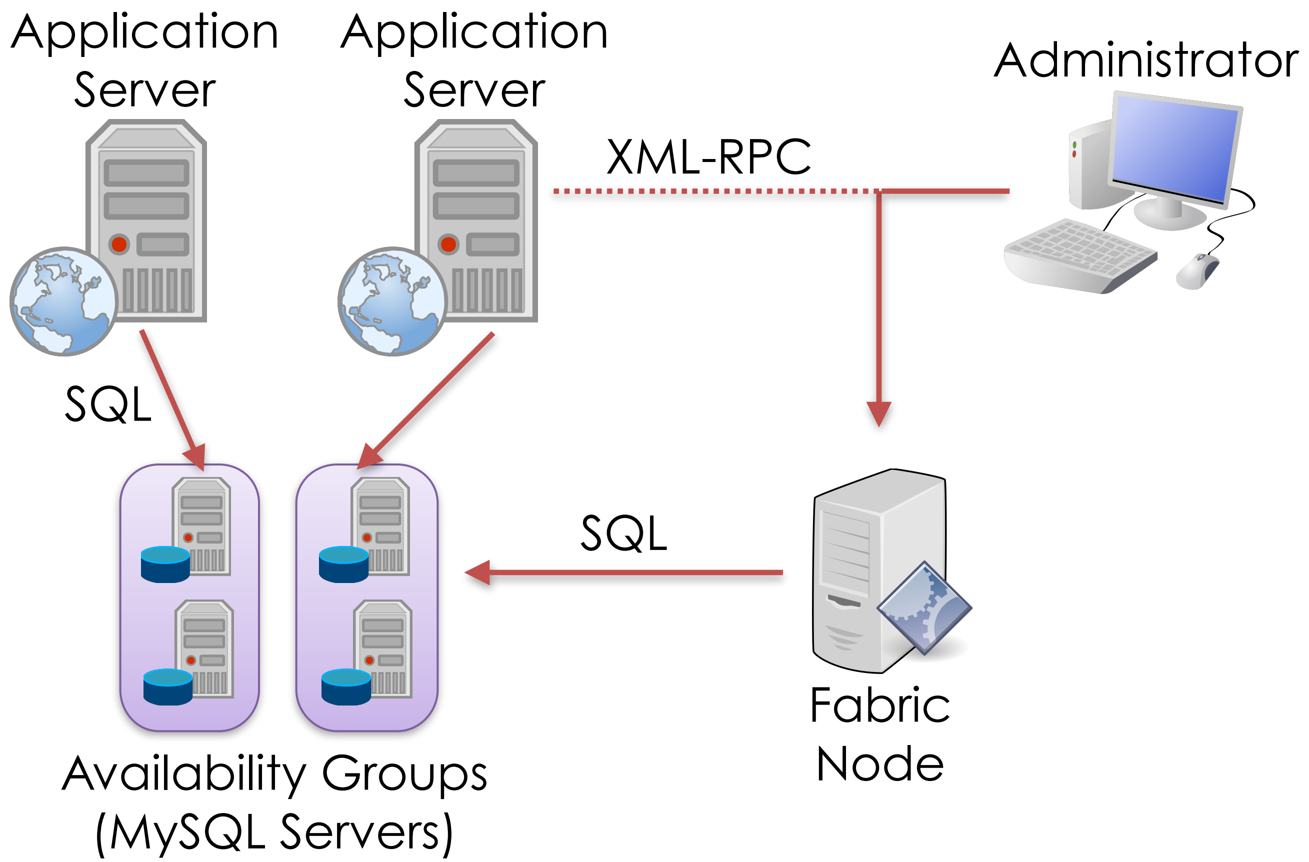
Problems of Other Solutions

- Passive nodes are a waste of resources
- Some of them are not shared-nothing
- No integrated sharding (write scaling)
- Complex to setup and administrate
- Requires application rewrites due to the usage of different storage engines/clustering limitations
- Not reliable (easy to break)
- Requires learning new technologies

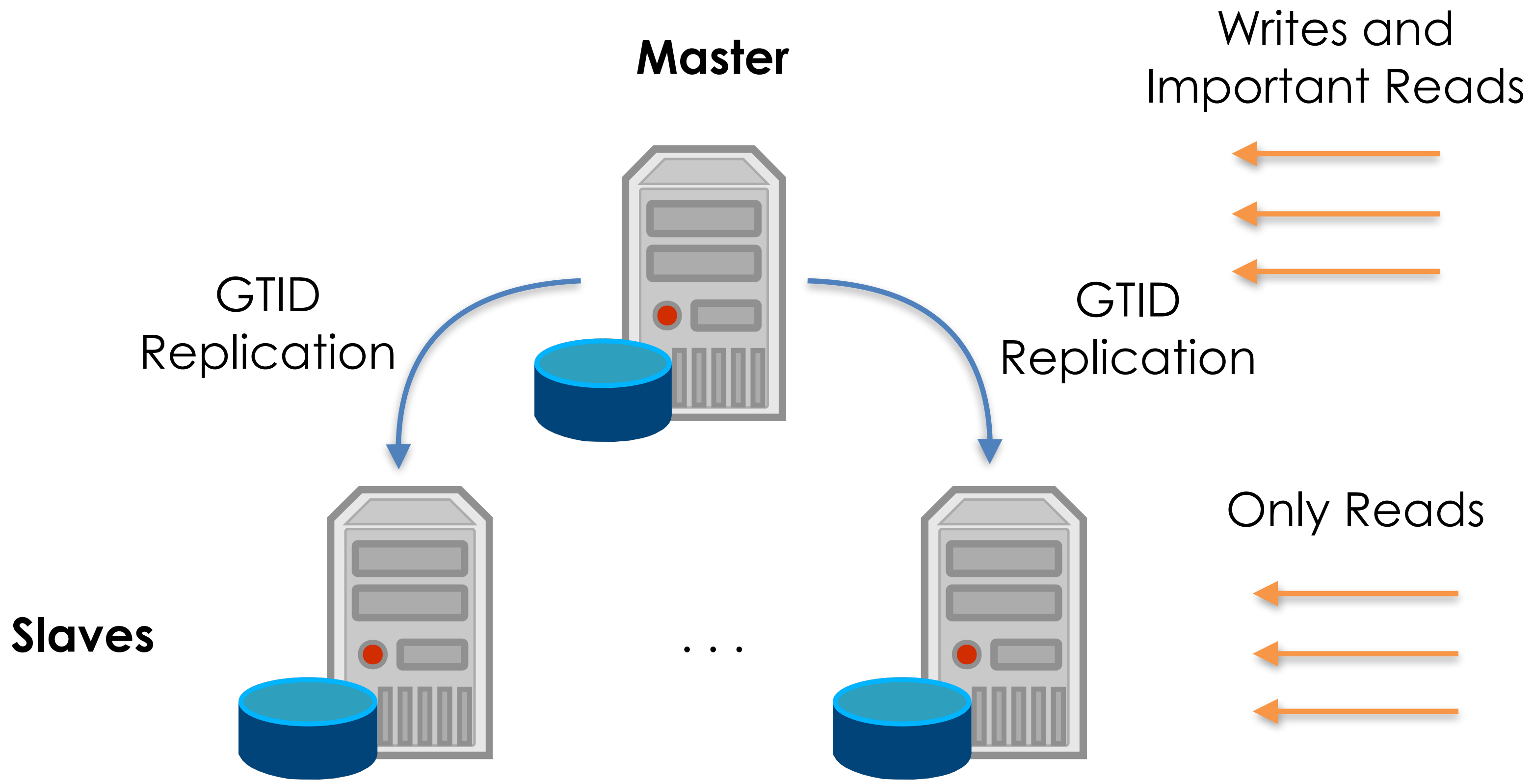
MySQL Fabric Introduction

- Distributed framework/middleware for managing farms of MySQL servers
  - Written in Python
  - Highly extensible
  - Manages high availability
  - “Semiautomatic” sharding
  - No extra latency
  - Based on GTID replication
  - Fully open source
  - Fabric aware connectors: Python, Java and PHP

High Level Architecture



Availability Group



MySQL Fabric: High Availability Solution for Connector/Python

INSTALLATION AND SETUP

MySQL Servers

- It requires MySQL 5.6
- For Ubuntu >=14.04:
  ```
  sudo aptitude install mysql-client-core-5.6
  mysql-server-5.6
  ```
- For other distributions, use the mysql community repo: http://dev.mysql.com/downloads/repo/
  - `sudo yum install mysql-community-server`

Python Connector

- Official Connector/Python http://dev.mysql.com/downloads/connector/python/
  - Required to get full advantage of the framework

MySQL Fabric

- MySQL Fabric is part of the “MySQL Utilities”: http://dev.mysql.com/downloads/utilities/
  - All of them are written in Python

Basic Usage

```
[ec2-user@jynus_com ~]$ mysqlfabric
Usage: mysqlfabric [--param, --config] <grp> <cmd> [arg, ...].

MySQL Fabric 1.5.3 - MySQL server farm management framework

Options:
  --version              show program's version number and exit
  -h, --help             show this help message and exit
  --param=CONFIG_PARAMS
                         Override a configuration parameter.
  --config=FILE          Read configuration from FILE.

Basic commands:
    help <grp> <cmd>   Show help for command
    help commands      List all commands
    help groups        List all groups
```

Getting Help

```
[ec2-user@jynus_com ~]$ mysqlfabric help group

Commands available in group 'group' are:
    group activate group_id  [--synchronous]
    group description group_id  [--description=NONE] [--synchronous]
    group deactivate group_id  [--synchronous]
    group create group_id  [--description=NONE] [--synchronous]
    group remove group_id server_id  [--synchronous]
    group add group_id address  [--timeout=NONE] [--update_only] [--
synchronous]
    group health group_id
    group lookup_servers group_id  [--server_id=NONE] [--status=NONE]
[--mode=NONE]
    group destroy group_id  [--synchronous]
    group demote group_id  [--update_only] [--synchronous]
    group promote group_id  [--slave_id=NONE] [--update_only] [--
synchronous]
    group lookup_groups  [--group_id=NONE]
```

/etc/mysql/fabric.cfg

```
[DEFAULT]
prefix =
sysconfdir = /etc
logdir = /var/log

[statistics]
prune_time = 3600

[logging]
url = file:///var/log/fabric.log
level = INFO

[storage]
auth_plugin =
mysql_native_password
database = fabric
user = fabric
address = localhost:3306
connection_delay = 1
connection_timeout = 6
password =
connection_attempts = 6

[failure_tracking]
notification_interval = 60
notification_clients = 50
detection_timeout = 1
detection_interval = 6
notifications = 300
detections = 3
failover_interval = 0
prune_time = 3600
```

/etc/mysql/fabric.cfg (cont.)

```
[servers]
password =
user = fabric
unreachable_timeout = 5

[connector]
ttl = 1

[client]
password =

[protocol.xmlrpc]
disable_authentication = no
ssl_cert =
realm = MySQL Fabric
ssl_key =
ssl_ca =
threads = 5
user = admin
address = localhost:32274
password =

[executor]
executors = 5

[sharding]
prune_limit = 10000
mysqldump_program = /usr/bin/mysqldump
mysqlclient_program = /usr/bin/mysql

[protocol.mysql]
disable_authentication = no
ssl_cert =
ssl_key =
ssl_ca =
user = admin
address = localhost:32275
password =
```

Setting Up the MySQL Store

```
[ec2-user@jynus_com ~]$ mysql -u root

mysql> CREATE USER fabric@localhost IDENTIFIED BY 'fabric';
Query OK, 0 rows affected (0.09 sec)

mysql> GRANT ALL ON fabric.* TO fabric@localhost;
Query OK, 0 rows affected (0.02 sec)
```

Setting Up the MySQL Store (cont.)

```
[ec2-user@jynus_com ~]$ mysqlfabric manage setup
[INFO] 1415176616.193902 - MainThread - Initializing
persister: user (fabric), server (localhost:3306), database
(fabric).
Finishing initial setup
=======================
Password for admin user is not yet set.
Password for admin/xmlrpc:
Repeat Password:
Password set.
```

Setting up the MySQL Servers

```
[ec2-user@pycones1 ~]$ vim /etc/my.cnf
gtid-mode=ON
enforce-gtid-consistency
server-id = 2
log-bin = /var/lib/mysql/binlog
relay-log = /var/lib/mysql/relay
binlog-format = ROW
log-slave-updates
expire-logs-days = 15
sync-binlog = 0
max_binlog_size = 50M

mysql> grant all on *.* to fabric@<fabric_host>;
Query OK, 0 rows affected (0.01 sec)
```

Service Start

```
[ec2-user@jynus_com ~]$ mysqlfabric manage start
[INFO] 1415178211.952550 - MainThread - Initializing persister: user (fabric), server
(localhost:3306), database (fabric).
[WARNING] 1415178211.962863 - MainThread - Provider error: No module named novaclient.
[INFO] 1415178211.963233 - MainThread - Loading Services.
[INFO] 1415178211.979758 - MainThread - MySQL-RPC protocol server started, listening on
localhost:32275
[WARNING] 1415178211.980087 - MainThread - Authentication disabled
[INFO] 1415178211.993972 - MainThread - Fabric node starting.
[INFO] 1415178211.995830 - MainThread - Starting Executor.
[INFO] 1415178211.996073 - MainThread - Setting 5 executor(s).
[INFO] 1415178211.996911 - Executor-0 - Started.
[INFO] 1415178211.999560 - Executor-1 - Started.
[INFO] 1415178212.002178 - Executor-2 - Started.
[INFO] 1415178212.003786 - Executor-3 - Started.
[INFO] 1415178212.005955 - MainThread - Executor started.
[INFO] 1415178212.007485 - Executor-4 - Started.
[INFO] 1415178212.036155 - MainThread - Starting failure detector.
[INFO] 1415178212.038245 - XML-RPC-Server - XML-RPC protocol server ('127.0.0.1', 32274)
started.
[INFO] 1415178212.038924 - XML-RPC-Server - Setting 1 XML-RPC session(s).
[INFO] 1415178212.039532 - XML-RPC-Session-0 - Started XML-RPC-Session.
```

MySQL Fabric: High Availability Solution for Connector/Python

HIGH AVAILABILITY

New Availability Group

```
[ec2-user@jynus_com ~]$ mysqlfabric group create pycones
Fabric UUID:  5ca1ab1e-a007-feed-f00d-cab3fe13249e
Time-To-Live: 1

                                uuid finished success result
------------------------------------ -------- ------- ------
81271531-e1b7-40e6-8dc2-e444752db7c2        1       1      1

state success          when                                                      description
----- ------- ------------- ------------------------------------------------------------------
    3       2   1.41518e+09 Triggered by <mysql.fabric.events.Event object at 0x1ecea90>.
    4       2   1.41518e+09                            Executing action (_create_group).
    5       2   1.41518e+09                             Executed action (_create_group).
```

Adding a Node to the Group

```
[ec2-user@jynus_com ~]$ mysqlfabric group add pycones pycones1
Fabric UUID:  5ca1ab1e-a007-feed-f00d-cab3fe13249e
Time-To-Live: 1

                                uuid finished success result
------------------------------------ -------- ------- ------
5071369c-88d1-4f95-af2a-2a4828a00756        1       1      1

state success          when                                              description
----- ------- ------------- ------------------------------------------------------------
    3       2   1.41518e+09 Triggered by <mysql.fabric.events.Event object at 0x1e4c450>.
    4       2   1.41518e+09                              Executing action (_add_server).
    5       2   1.41518e+09                               Executed action (_add_server).
```

Both Servers Are Still Read-Only

```
[ec2-user@jynus_com ~]$ mysqlfabric group lookup_servers pycones
Fabric UUID:  5ca1ab1e-a007-feed-f00d-cab3fe13249e
Time-To-Live: 1

                         server_uuid  address    status       mode weight
------------------------------------ -------- --------- ---------- ------
4c50e85f-64cf-11e4-998e-0a07078f4ec7 pycones1 SECONDARY READ_ONLY     1.0
50f30034-64cf-11e4-998e-0a3081f4545c pycones2 SECONDARY READ_ONLY     1.0
```

Promoting a Node

```
[ec2-user@jynus_com ~]$ mysqlfabric group promote pycones \
                        --slave_id=4c50e85f-64cf-11e4-998e-0a07078f4ec7
Fabric UUID:  5ca1ab1e-a007-feed-f00d-cab3fe13249e
Time-To-Live: 1

                                uuid finished success result
------------------------------------ -------- ------- ------
4c50e85f-64cf-11e4-998e-0a07078f4ec7        1       1      1

state success          when                                                description
----- ------- ------------- ------------------------------------------------------------
    3       2   1.41519e+09 Triggered by <mysql.fabric.events.Event object at 0x1f73f10>.
    4       2   1.41519e+09                              Executing action (_define_ha_operation).
    5       2   1.41519e+09                               Executed action (_define_ha_operation).

[ec2-user@jynus_com ~]$ mysqlfabric group lookup_servers pycones
Fabric UUID:  5ca1ab1e-a007-feed-f00d-cab3fe13249e
Time-To-Live: 1

                         server_uuid  address    status       mode weight
------------------------------------ -------- --------- ---------- ------
4c50e85f-64cf-11e4-998e-0a07078f4ec7 pycones1   PRIMARY READ_WRITE    1.0
50f30034-64cf-11e4-998e-0a3081f4545c pycones2 SECONDARY READ_ONLY     1.0
```

Example Read-only Code

```
import mysql.connector
from mysql.connector import fabric

conn = mysql.connector.connect(
        fabric={"host" : "localhost", "port" : 32274,
                "username": "admin", "password" : "",
                'report_errors': True },
        user="root", password="", database="test",
        autocommit=True)

conn.set_property(group="pycones", mode=fabric.MODE_READONLY)

cursor = conn.cursor()
query = """SELECT @@global.server_uuid, Name,
           District, Population FROM City WHERE id = 657"""
cursor.execute(query)

for (server, name, district, population) in cursor:
        print("server " + server + ": " + name + "(" + district + "), pop. " +
str(population))

cursor.close()
conn.close()
```

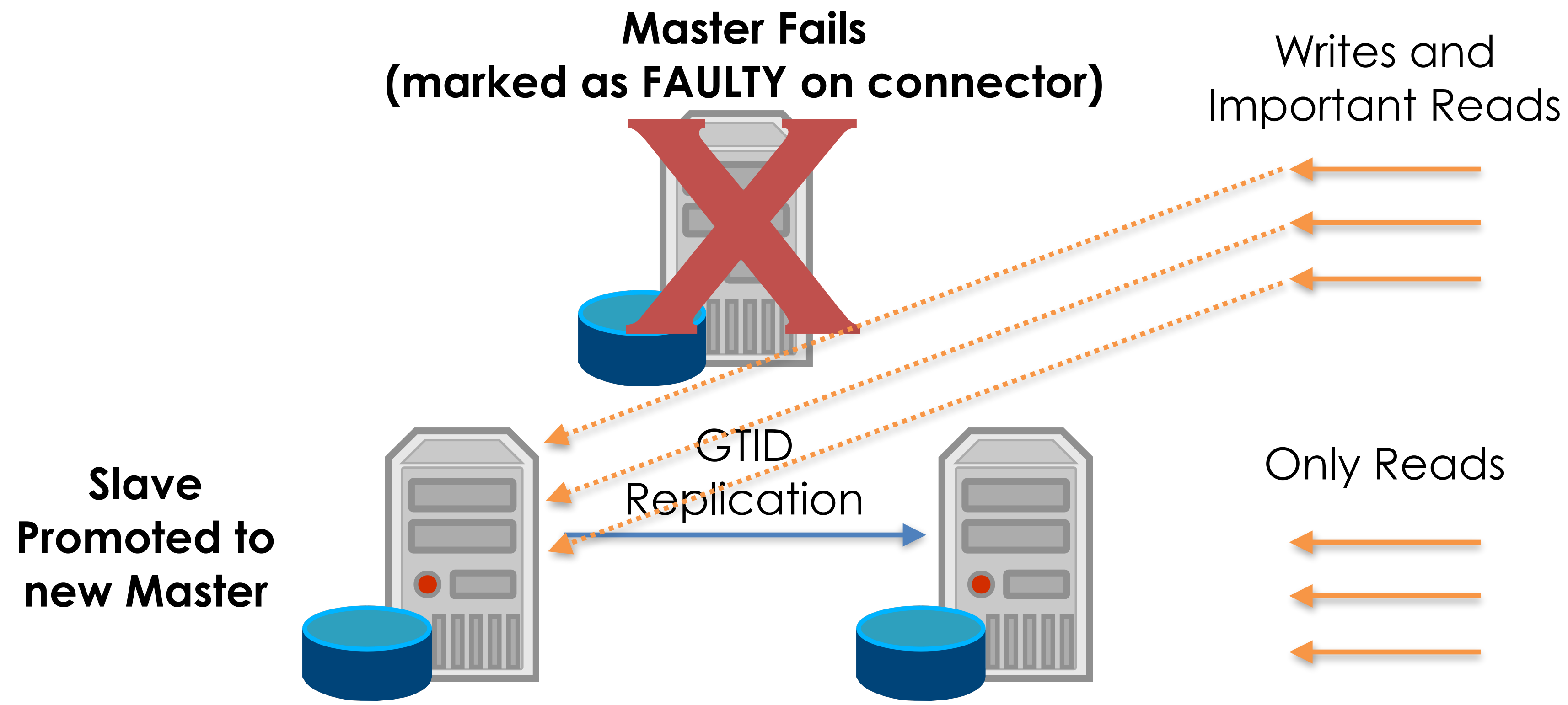
Reads can be sent to any server (but preferably to a slave), read/writes, only to the master.

Result

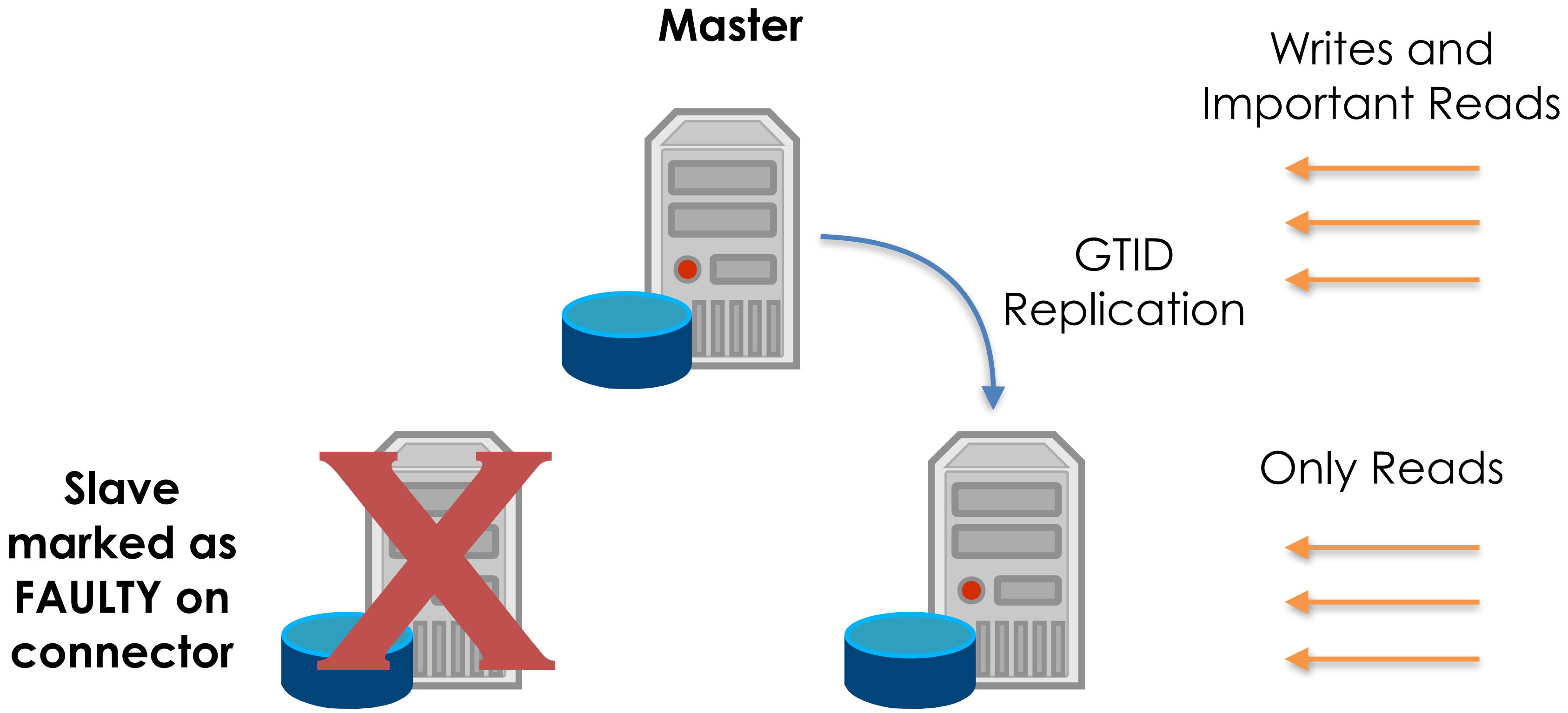
```
[ec2-user@jynus_com ~]$ while true; do python fabric_test.py; sleep 1; done
server 4c50e85f-64cf-11e4-998e-0a07078f4ec7: Zaragoza(Aragonia), pop. 603367
server 4c50e85f-64cf-11e4-998e-0a07078f4ec7: Zaragoza(Aragonia), pop. 603367
server 4c50e85f-64cf-11e4-998e-0a07078f4ec7: Zaragoza(Aragonia), pop. 603367
server 4c50e85f-64cf-11e4-998e-0a07078f4ec7: Zaragoza(Aragonia), pop. 603367
server 4c50e85f-64cf-11e4-998e-0a07078f4ec7: Zaragoza(Aragonia), pop. 603367
server 4c50e85f-64cf-11e4-9                                    pop. 603367
server 4c50e85f-64cf-11e4-9                                    pop. 603367
[...]
server 50f30034-64cf-11e4-9                                    pop. 603367
server 50f30034-64cf-11e4-9                                    pop. 603367
server 50f30034-64cf-11e4-9                                    pop. 603367
server 50f30034-64cf-11e4-9                                    pop. 603367
server 50f30034-64cf-11e4-9                                    pop. 603367
server 50f30034-64cf-11e4-9                                    pop. 603367
server 50f30034-64cf-11e4-9                                    pop. 603367
server 50f30034-64cf-11e4-998e-0a3081f4545c: Zaragoza(Aragonia), pop. 603367
```

Queries are sent to the SECONDARY. When we force it to crash, it failovers transparently to the other server after a brief timeout. If it was a master, it also triggers a failover. The server is marked as FAULTY to the connector

Master Fails



Slave Fails


Master
Writes and Important Reads
GTID Replication
Slave marked as FAULTY on connector
Only Reads

MySQL Fabric: High Availability Solution for Connector/Python

SHARDING

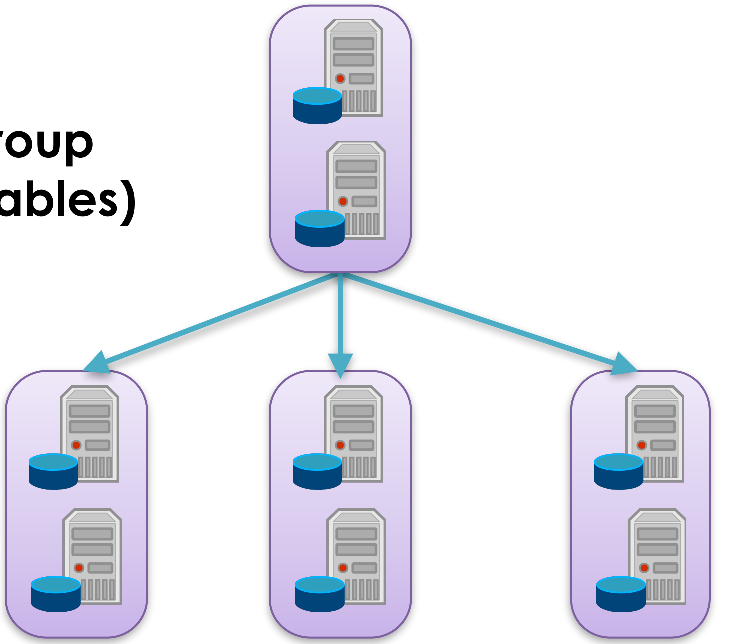
Setup of Local Availability Groups

- By default, AG are global (contain all data)
  - We can create local groups with only a portion of it
  - HASH or RANGE partitioning is allowed

Sharding Schema

Global Group
(common tables)

Local Groups
(tables sharded
by key)



id=1, 4, 7 id=2, 5, 8 id=3, 6, 9

Defining the Partitioning

```
[ec2-user@jynus_com ~]$ mysqlfabric sharding create_definition HASH pycones
Fabric UUID:  5ca1ab1e-a007-feed-f00d-cab3fe13249e
Time-To-Live: 1

                                uuid finished success result
------------------------------------ -------- ------- ------
9668bc06-6fd2-43e6-9afb-13203d61bb01        1       1      2

state success          when                                             description
----- ------- ------------- -------------------------------------------------------
    3       2    1.4152e+09 Triggered by <mysql.fabric.events.Event object at 0x20f26d0>.
    4       2    1.4152e+09                    Executing action (_define_shard_mapping).
    5       2    1.4152e+09                     Executed action (_define_shard_mapping).
```

Defining the Partitioning (cont.)

```
[ec2-user@jynus_com ~]$ mysqlfabric dump shard_maps
Fabric UUID:  5ca1ab1e-a007-feed-f00d-cab3fe13249e
Time-To-Live: 1

mapping_id type_name global_group_id
---------- --------- ---------------
         2      HASH         pycones
```

Defining the Partitioning (cont.)

```
[ec2-user@jynus_com ~]$ mysqlfabric sharding add_table 2 test.City ID
Fabric UUID:  5ca1ab1e-a007-feed-f00d-cab3fe13249e
Time-To-Live: 1

                                uuid finished success result
------------------------------------ -------- ------- ------
abcf6a48-c0c5-47ab-820e-280d5484fee2        1       1      1

state success          when
description
----- ------- ------------
------------------------------------------------------------
    3       2    1.4152e+09 Triggered by <mysql.fabric.events.Event
object at 0x20f2650>.
    4       2    1.4152e+09                       Executing action
(_add_shard_mapping).
    5       2    1.4152e+09                        Executed action
(_add_shard_mapping).
```

Creating the Sharded Groups

```
[ec2-user@jynus_com ~]$ mysqlfabric group create pycones-shard1
[...]
[ec2-user@jynus_com ~]$ mysqlfabric group create pycones-shard2
[...]
[ec2-user@jynus_com ~]$ mysqlfabric sharding add_shard 2 pycones-shard1,pycones-shard2
Fabric UUID:  5ca1ab1e-a007-feed-f00d-cab3fe13249e
Time-To-Live: 1

                                uuid finished success result
------------------------------------ -------- ------- ------
170da72d-e6a8-4317-892b-67c40b362b83        1       1      1

state success          when                                                  description
----- ------- ------------- ------------------------------------------------------------
    3       2    1.4152e+09 Triggered by <mysql.fabric.events.Event object at 0x21aa490>.
    4       2    1.4152e+09                                Executing action (_add_shard).
    5       2    1.4152e+09                                 Executed action (_add_shard).
```

Using the Global Scope

- `conn.set_property(tables=["test.City"], scope=fabric.SCOPE_GLOBAL, mode=fabric.MODE_READWRITE)`
- Then insert your table, it will be replicated and split correctly between the shards

Sharding By Key

- `conn.set_property(tables=["test.City"], key=my_id, mode=fabric.MODE_READWRITE, scope=fabric.SCOPE_LOCAL)`
- You can now select and modify the data using the right shard group

Example Code Using Sharding by Key

```
import mysql.connector
from mysql.connector import fabric
import random

conn = mysql.connector.connect(
        fabric={"host" : "localhost", "port" : 32274,
                "username": "admin", "password" : "",
                'report_errors': True },
        user="root", password="", database="test", autocommit=True)
my_id = str(random.randint(1, 1000))
conn.set_property(tables=["test.City"], key=my_id,
                  scope=fabric.SCOPE_LOCAL, mode=fabric.MODE_READWRITE)

cursor = conn.cursor()
query = """SELECT @@global.server_uuid, Name,
           District, Population FROM City WHERE id = %s"""
cursor.execute(query, (my_id,))

for (server, name, district, population) in cursor:
        print("server " + server + ": " + name +
              "(" + district + "), pop. " + str(population))

cursor.close()
conn.close()
```

Result

```
[ec2-user@jynus_com ~]$ while true; do python fabric_test2.py; sleep 1; done
server 4c50e85f-64cf-11e4-998e-0a07078f4ec7: Lubao(Central Luzon), pop. 125699
server 4c50e85f-64cf-11e4-998e-0a07078f4ec7: Abaetetuba(Pará), pop. 111258
server 86f95e3f-5e41-11e4-aed1-0800273d6990: Concepción(Bíobío), pop. 217664
server 4c50e85f-64cf-11e4-998e-0a07078f4ec7: Emmen(Drenthe), pop. 105853
server 4c50e85f-64cf-11e4-998e-0a07078f4ec7: Huambo(Huambo), pop. 163100
server 4c50e85f-64cf-11e4-998e-0a07078f4ec7: Ozamis(Northern Mindanao), pop. 110420
server 4c50e85f-64cf-11e4-998e-0a07078f4ec7: Amersfoort(Utrecht), pop. 126270
server 86f95e3f-5e41-11e4-aed1-0800273d6990: Florencio Varela(Buenos Aires), pop. 315432
server 86f95e3f-5e41-11e4-aed1-0800273d6990: Cagayan de Oro(Northern Mindanao), pop. 461877
server 4c50e85f-64cf-11e4-998e-0a07078f4ec7: Depok(West Java), pop. 365200
server 4c50e85f-64cf-11e4-998e-0a07078f4ec7: Pilar(Buenos Aires), pop. 113428
server 4c50e85f-64cf-11e4-998e-0a07078f4ec7: Kupang(Nusa Tenggara Timur), pop. 129300
server 4c50e85f-64cf-11e4-998e-0a07078f4ec7: Ciomas(West Java), pop. 187400
server 4c50e85f-64cf-11e4-998e-0a07078f4ec7: Franca(São Paulo), pop. 290139
server 86f95e3f-5e41-11e4-aed1-0800273d6990: Bayawan (Tulong)(Central Visayas), pop. 101391
server 4c50e85f-64cf-11e4-998e-0a07078f4ec7: Almere(Flevoland), pop. 142465
server 86f95e3f-5e41-11e4-aed1-0800273d6990: Araguaína(Tocantins), pop. 114948
server 86f95e3f-5e41-11e4-aed1-0800273d6990: Foz do Iguaçu(Paraná), pop. 259425
server 4c50e85f-64cf-11e4-998e-0a07078f4ec7: Silang(Southern Tagalog), pop. 156137
server 86f95e3f-5e41-11e4-aed1-0800273d6990: Leiden(Zuid-Holland), pop. 117196
server 86f95e3f-5e41-11e4-aed1-0800273d6990: Olongapo(Central Luzon), pop. 194260
```

Obtaining Sharding Information

```
[ec2-user@jynus_com ~]$ mysqlfabric dump sharding_information
Fabric UUID:  5ca1ab1e-a007-feed-f00d-cab3fe13249e
Time-To-Live: 1

schema_name table_name column_name lower_bound shard_id type_name group_id       global_group
----------- ---------- ----------- ----------- -------- --------- -------------- ------------
test        City       ID          E6416...    1        HASH      pycones-shard1 pycones
test        City       ID          DC3AD...    2        HASH      pycones-shard2 pycones

[ec2-user@jynus_com ~]$ mysqlfabric sharding lookup_servers test.City 657
Fabric UUID:  5ca1ab1e-a007-feed-f00d-cab3fe13249e
Time-To-Live: 1
                         server_uuid  address    status       mode weight
------------------------------------ --------- --------- ---------- ------
4c50e85f-64cf-11e4-998e-0a07078f4ec7 pycones3    PRIMARY READ_WRITE    1.0
```

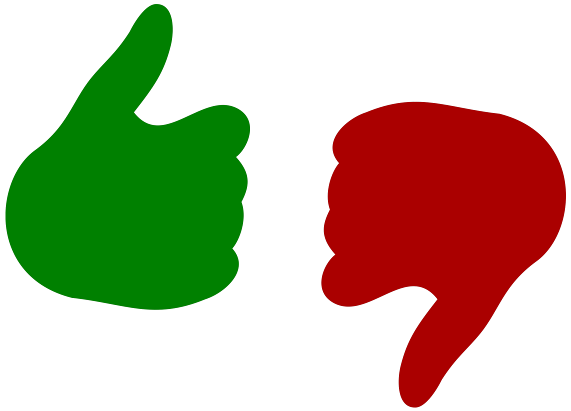
Common Operations Supported

- Splitting a sharded group into other 2
- Moving the shard to a different group
- Fully automatic provisioning is only available through plugins
  - There an existing one for OpenStack

MySQL Fabric: High Availability Solution for Connector/Python

CONCLUSION

Pros and Cons



- Pros:
  - Easy to setup and configure
  - Uses a well know protocol (standard replication) with standard on-disk engines (InnoDB)
  - Secondary nodes provide transparent HA and read scalability
  - Sharding provides write scalability
  - No extra latency spent on load balancers/proxies
  - Extensible for extra functionality and backend support
- Cons:
  - The Fabric node is itself a SPOF- it should be made redundant (e.g. pacemaker)
  - No multi-master/synchronous support (coming in 5.7?): I recommend Galera for now as an alternative
  - Relatively new development history (in comparison)
  - Designed for large MySQL farms (7+ nodes)

Q&A

Knowing More about Database Optimization and High Availability

- My blog: http://dbahire.com
- Course: *"Optimization, Administration and High Availability with MySQL 5.6"* on 15th December in Zaragoza
  - More dates & places coming soon