MySQL Group Replication & MySQL InnoDB Cluster

Production Ready?

Kenny Gryp

🇧🇪 🧇 productions

Table of Contents

MySQL Group Replication

MySQL InnoDB Cluster

MySQL Group Replication

- Developed by Oracle
- Generally Available in MySQL 5.7.17 on December 2016
- MySQL InnoDB Cluster as Solution

MySQL Group Replication is a MySQL Server plugin that **provides distributed state machine replication** with strong coordination between servers. Servers coordinate themselves automatically, when they are part of the same replication group. **Any server in the group can process updates**. **Conflicts are detected and handled automatically**. There is a **built-in membership service** that keeps the view of the group consistent and available for all servers at any given point in time. **Servers can leave and join the group** and the view will be updated accordingly.

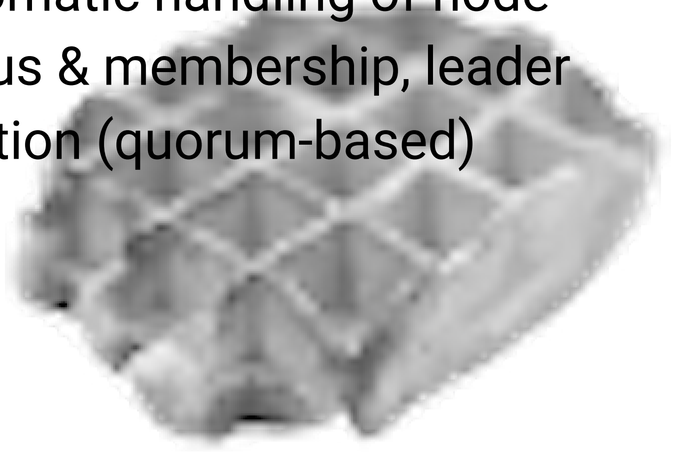
Asynchronous Replication vs. GR

Async

- Async delivery
- Master -> Replica(s)
- Replica 'fetches' binlogs and executes
- external scripts required for automatic failover, split brain prevention...

GR

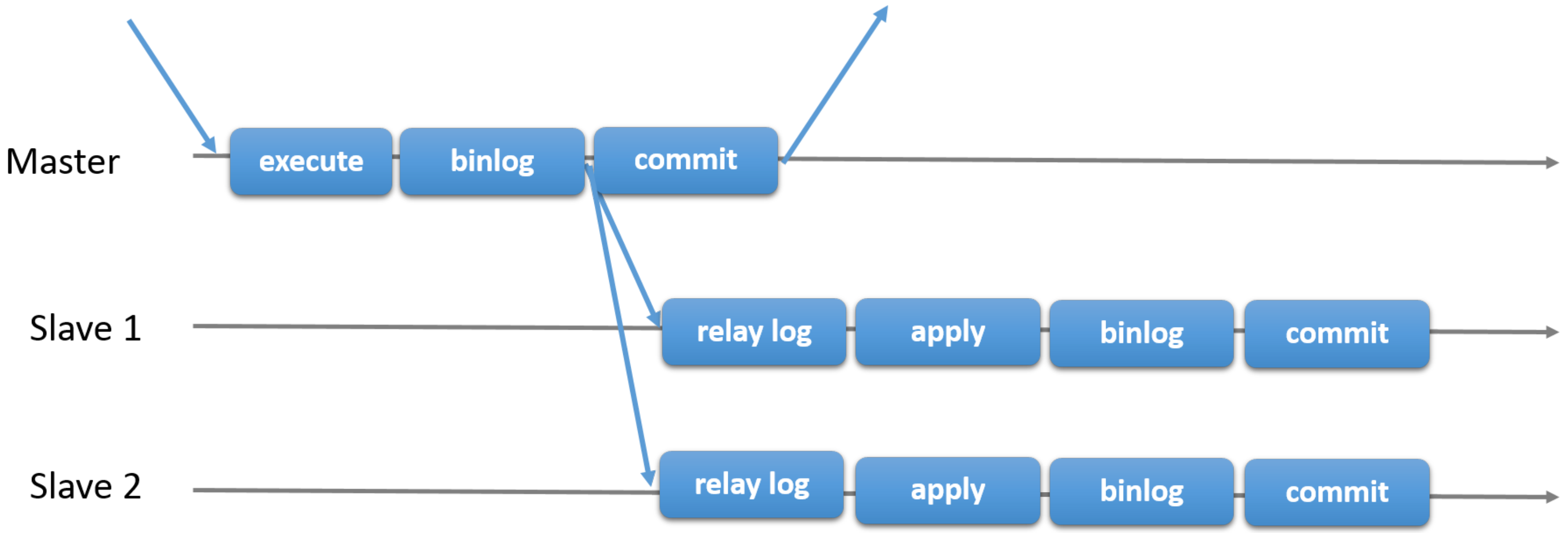
- Sync delivery
  (at TRX Commit)
- Members <-> Members
- Majority of members receive TRX (PAXOS)
- Automatic handling of node status & membership, leader election (quorum-based)



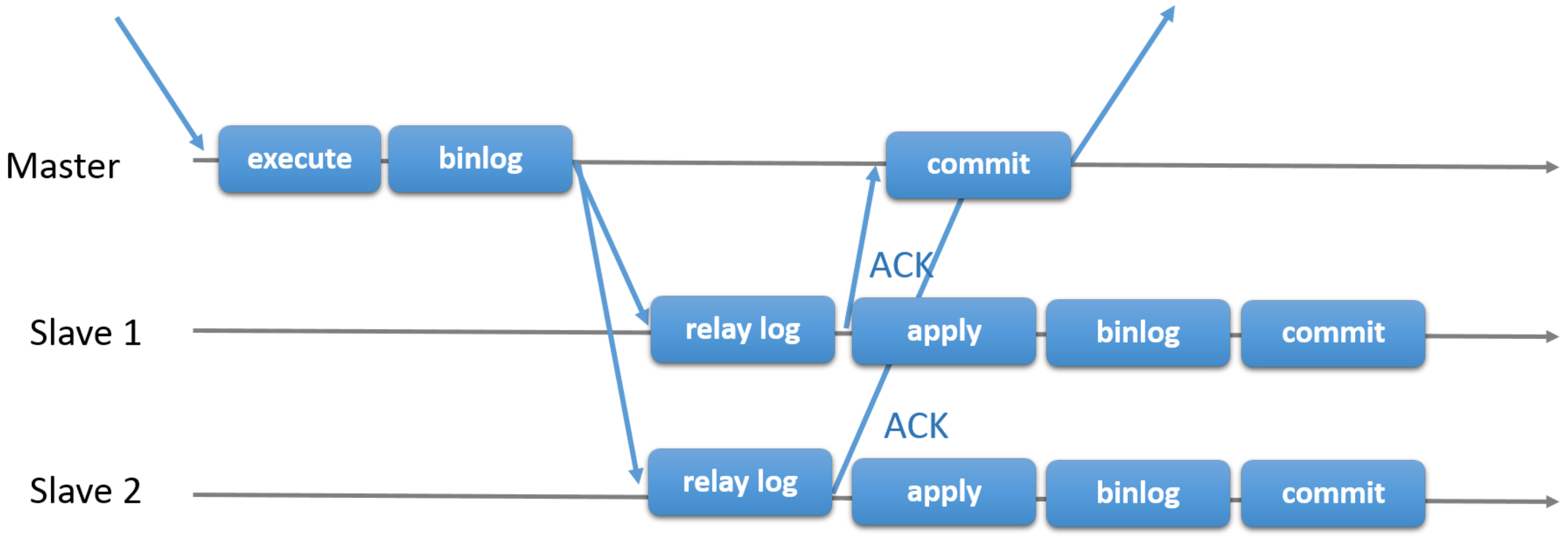
Group Replication

- GR uses a PAXOS protocol to ensure all nodes receive data
  - Increased `COMMIT` time
    similar to PXC (& semi-sync replication)
- Easy to configure/setup (easier than Async GTID Setups)
- (Integrated multi-node conflict detection)

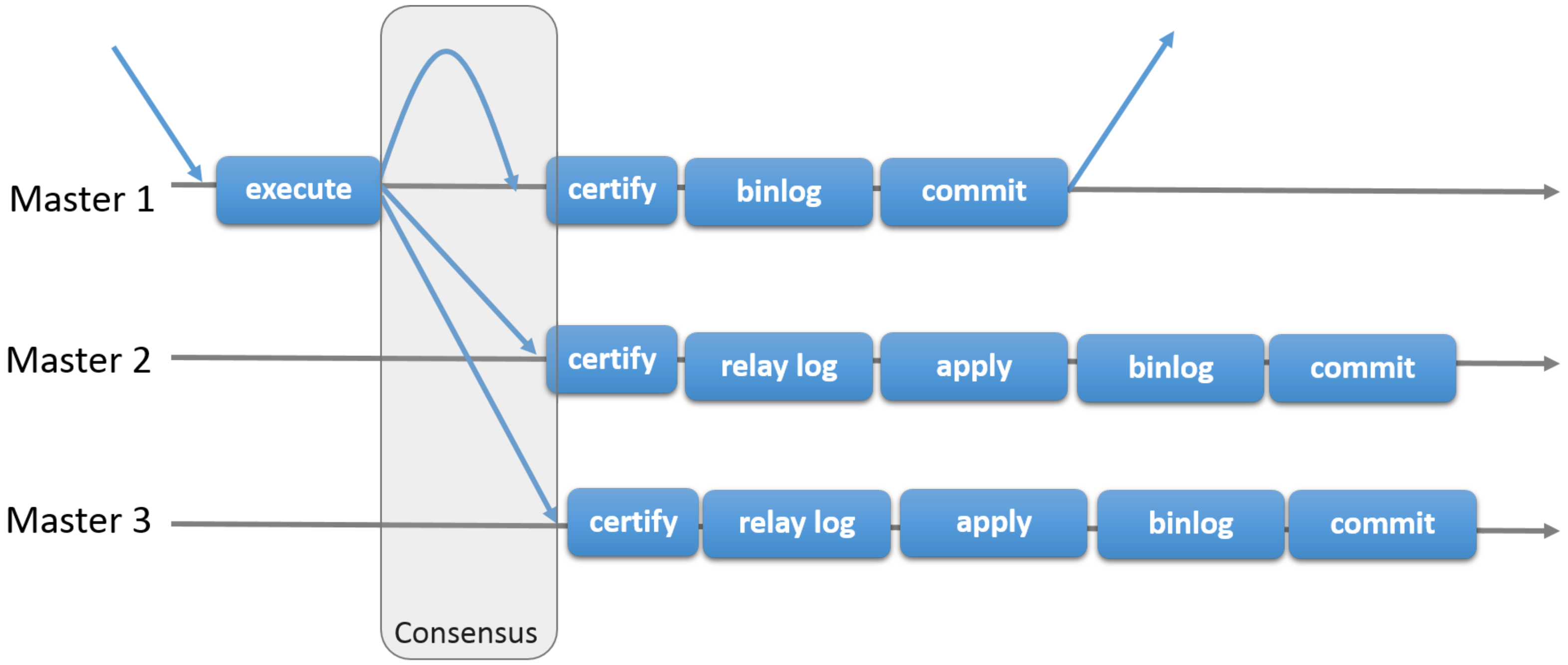
Asynchronous Replication



Semi-Sync Replication



Group Replication

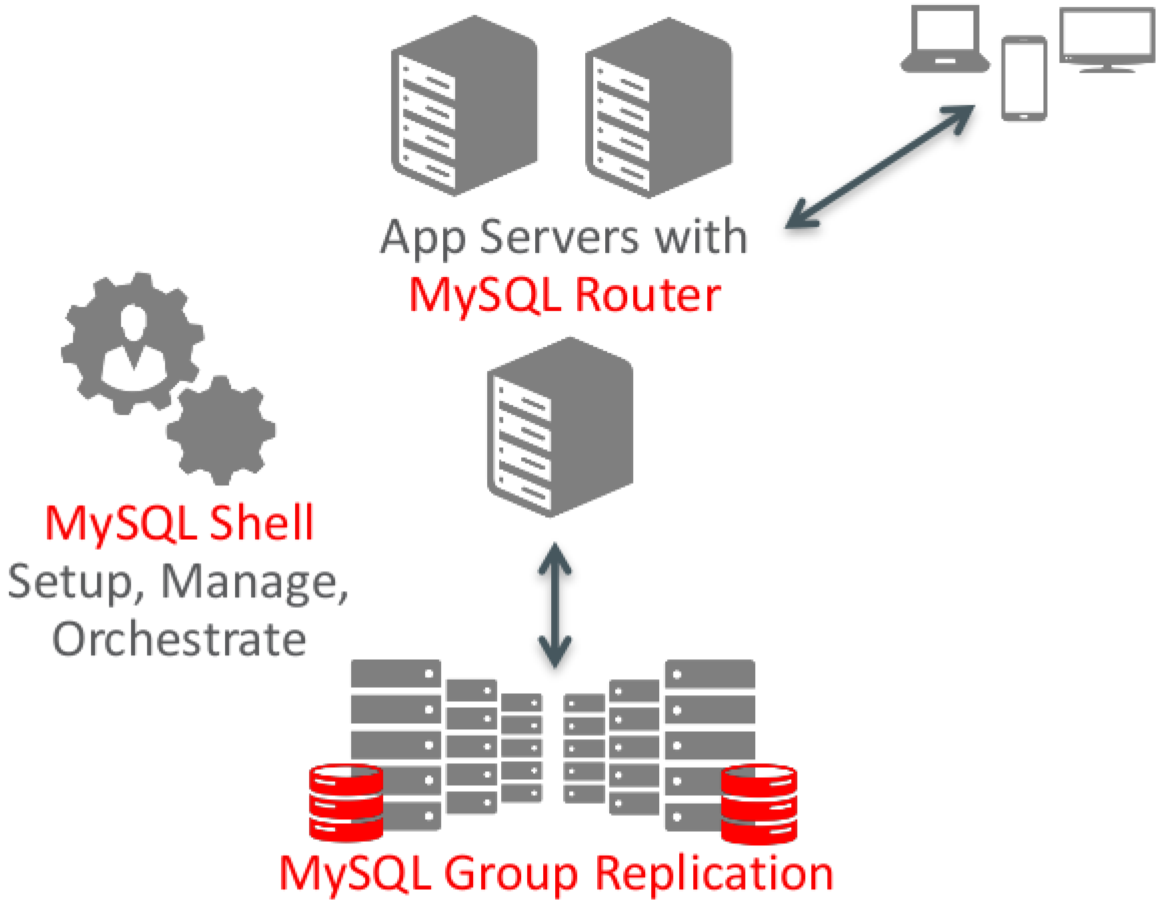


Use Cases

Environments Requiring:

- Strict **Durability** requirements
  - no data loss when a database node fails (0 RPO master failure):
  - **Consistency**: integrated split-brain prevention (Quorum based)
- Faster Failover than standard async (better RTO master failure)
- (**Write to multiple nodes simultaneously**)

MySQL InnoDB Cluster



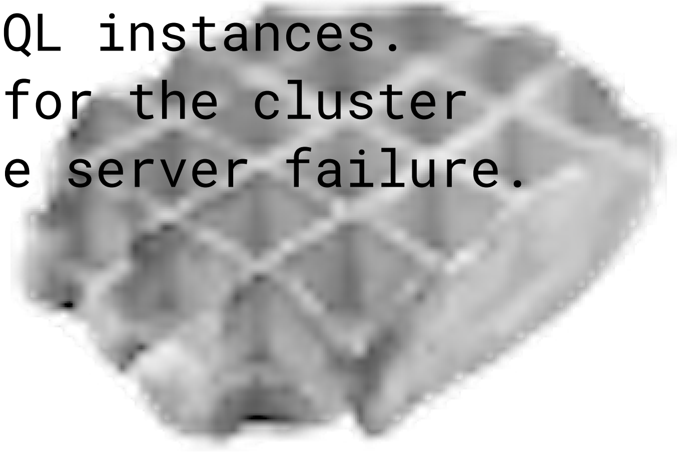
MySQL InnoDB Cluster

Admin API

MySQL Shell

MySQL Shell

```
mysql-js> var cluster
            = dba.createCluster('prodCluster')

A new InnoDB cluster will be created
on instance 'ic@ic-1:3306'.

Creating InnoDB cluster 'prodCluster' on 'ic@ic-1:3306'
Adding Seed Instance...

Cluster successfully created. Use
Cluster.addInstance() to add MySQL instances.
At least 3 instances are needed for the cluster
to be able to withstand up to one server failure.
```



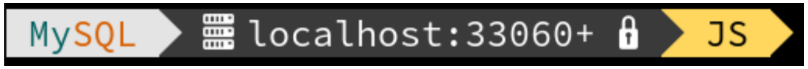
MySQL Shell

"Makes Group Replication Configuration Easy"

- Not really 5.7.21 & <= 8.0.4:
  - #90439: AdminAPI does not change my.cnf
  - #90438: AdminAPI fails to rejoin instances

MySQL Shell

"Makes Group Replication Configuration Easy"

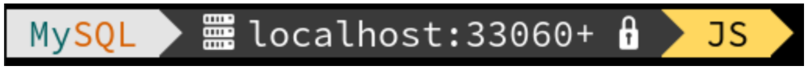
- Not really 5.7.21 & <= 8.0.4:
  - #90439: AdminAPI does not change my.cnf
  - #90438: AdminAPI fails to rejoin instances
- MySQL 8.0.11 (GA)
  - 

    ```
    MySQL  localhost:33060+  JS
    ```

    (great unicode support)

MySQL Shell

"Makes Group Replication Configuration Easy"

- Not really 5.7.21 & <= 8.0.4:
  - #90439: AdminAPI does not change my.cnf
  - #90438: AdminAPI fails to rejoin instances
- MySQL 8.0.11 (GA)
  - 
    MySQL localhost:33060+ JS (great unicode support)
  - Config is saved (`SET PERSIST`)
  - All actions can be done from a remote `mysqlsh`

MySQL Group Replication

MySQL Group Replication

- Split Brain Prevention
- Data Consistency
- Usability
- Stability
- Performance

Split Brain Prevention

MySQL Group Replication

Split Brain Prevention

No known split brain issues anymore!

Big improvement over 5.7.17
(first GA)

Data Consistency

MySQL Group Replication

Data Consistency

Multi Writer

```
I have read the MySQL InnoDB cluster manual and
I understand the requirements and limitations
of advanced Multi-Master Mode.
Confirm [y/N]: NO
```

Data Consistency

Multi Writer

```
I have read the MySQL InnoDB cluster manual and
I understand the requirements and limitations
of advanced Multi-Master Mode.
Confirm [y/N]: NO
```

Multi-Master is not recommended

Data Consistency

Multi Writer

```
I have read the MySQL InnoDB cluster manual and
I understand the requirements and limitations
of advanced Multi-Master Mode.
Confirm [y/N]: NO
```

Multi-Master is not recommended

- #89194: Wrong certification lead to data inconsistency and GR breakage. (Multi-Master, should be fixed in 5.7.22 and 8.0.11)
- #89938: Rejoin old primary node may duplicate key when recovery

Usability

MySQL Group Replication

Usability

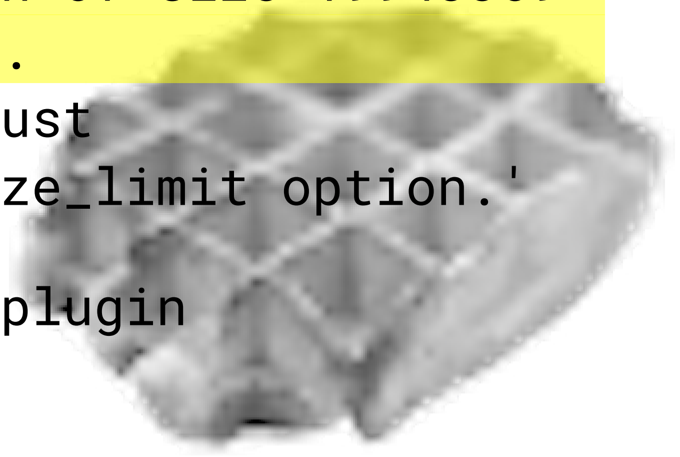
```
mysql> INSERT INTO maurage
       SELECT null FROM chez_lefred
              WHERE dim0s_office IS NULL;
ERROR 3100 (HY000): Error on observer while
       running replication hook 'before_commit'.
```

Usability

```
mysql> INSERT INTO maurage
       SELECT null FROM chez_lefred
              WHERE dim0s_office IS NULL;
ERROR 3100 (HY000): Error on observer while
       running replication hook 'before_commit'.
```

Error Log:

```
Plugin group_replication reported:
'Error on session 75. Transaction of size 19943309
exceeds specified limit 15000000.
To increase the limit please adjust
group_replication_transaction_size_limit option.'

Run function 'before_commit' in plugin
'group_replication' failed
```



Usability

```
mysql> COMMIT;
ERROR 1180 (HY000): Got error 149
- 'Lock deadlock; Retry transaction' during COMMIT
```

Usability

```
mysql> COMMIT;
ERROR 1180 (HY000): Got error 149
- 'Lock deadlock; Retry transaction' during COMMIT
```

- Nothing in the error log!
- Cannot troubleshoot
- (Only happens in multi-writer mode)

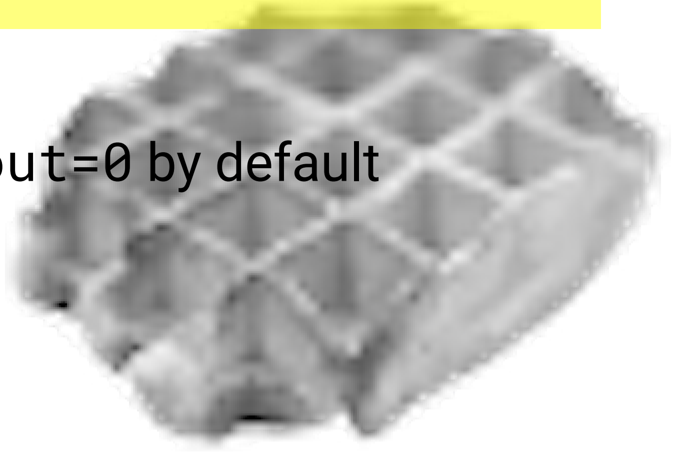
Usability

```
mysql> show processlist\G

     Id: 25
   User: root
   Host: localhost
     db: NULL
Command: Query
    Time: 131
   State: checking permissions
    Info: create database node2
```

Usability

```
mysql> show processlist\G

     Id: 25
   User: root
   Host: localhost
     db: NULL
Command: Query
    Time: 131
   State: checking permissions
    Info: create database node2
```

- no Quorum
- gr_unreachable_majority_timeout=0 by default :(



Usability

Features:

- No automatic node provisioning
- #84730: Cannot troubleshoot Transaction Rollbacks
- #90461: Changing replication mode cannot happen online
- #84729: Impossible to block reads on partitioned nodes
- #90484: No (easy) way to know if a GR node is writable or not
- #90485: Ignore group_replication_group_seeds nodes if they are not primary/active

Bug:

- #90483: DNS based seeds resolving to itself causes GR to not start

Usability

Features & Bugs from Jean-François Gagné:

- #89147: ... error messages is ambiguous.
- #89145: Provide relay log details in case of Group Replication applier failure.
- #89197: When GR fails, the error message says to "START SLAVE".

Stability

MySQL Group Replication

Stability

Feature:

- #84784: Nodes do not reconnect back to the group replication once they got disconnected, causing nodes to drop from the cluster (except last 2 nodes)

Bug:

- #90457: mysqld crash with ctrl-c/z'ed `START GROUP_REPLICATION`

Performance

MySQL Group Replication

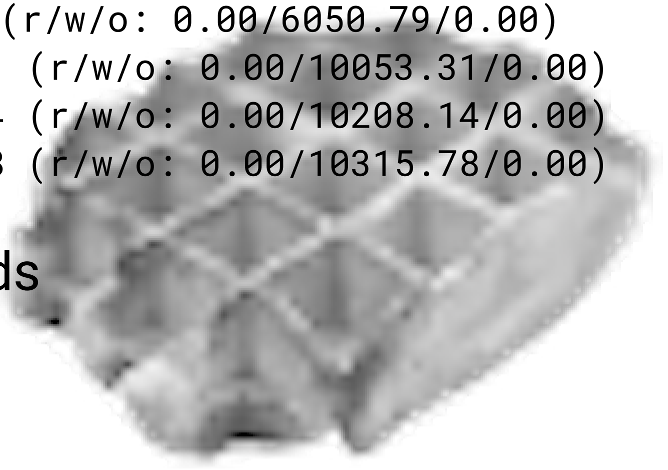
Performance

```
[ 220s] threads: 16 tps: 10599.99 qps: 10598.99 (r/w/o: 0.00/10598.99/0.00)
[ 221s] threads: 16 tps: 10571.71 qps: 10571.71 (r/w/o: 0.00/10571.71/0.00)
[ 222s] threads: 16 tps: 10307.88 qps: 10307.88 (r/w/o: 0.00/10307.88/0.00)
[ 223s] threads: 16 tps: 8220.26 qps: 8220.26 (r/w/o: 0.00/8220.26/0.00)
[ 224s] threads: 16 tps: 6381.09 qps: 6381.09 (r/w/o: 0.00/6381.09/0.00)
[ 225s] threads: 16 tps: 10348.85 qps: 10348.85 (r/w/o: 0.00/10348.85/0.00)
[ 226s] threads: 16 tps: 9383.95 qps: 9383.95 (r/w/o: 0.00/9383.95/0.00)
[ 227s] threads: 16 tps: 10528.06 qps: 10528.06 (r/w/o: 0.00/10528.06/0.00)

[ 280s] threads: 16 tps: 10335.09 qps: 10335.09 (r/w/o: 0.00/10335.09/0.00)
[ 281s] threads: 16 tps: 10372.06 qps: 10372.06 (r/w/o: 0.00/10372.06/0.00)
[ 282s] threads: 16 tps: 10237.61 qps: 10237.61 (r/w/o: 0.00/10237.61/0.00)
[ 283s] threads: 16 tps: 8206.20 qps: 8206.20 (r/w/o: 0.00/8206.20/0.00)
[ 284s] threads: 16 tps: 6050.79 qps: 6050.79 (r/w/o: 0.00/6050.79/0.00)
[ 285s] threads: 16 tps: 10053.31 qps: 10053.31 (r/w/o: 0.00/10053.31/0.00)
[ 286s] threads: 16 tps: 10208.14 qps: 10208.14 (r/w/o: 0.00/10208.14/0.00)
[ 287s] threads: 16 tps: 10315.78 qps: 10315.78 (r/w/o: 0.00/10315.78/0.00)
```

Performance

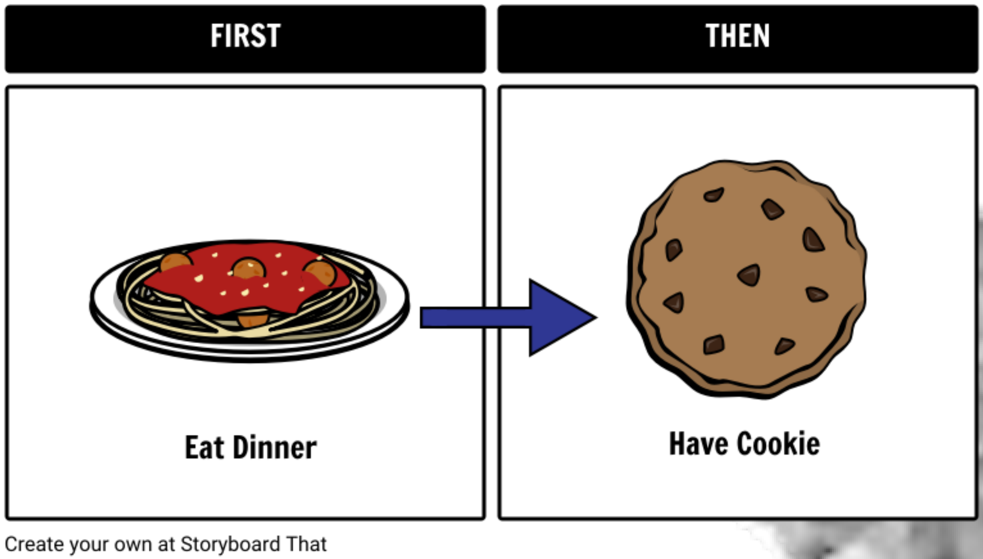
```
[ 220s] threads: 16 tps: 10599.99 qps: 10598.99 (r/w/o: 0.00/10598.99/0.00)
[ 221s] threads: 16 tps: 10571.71 qps: 10571.71 (r/w/o: 0.00/10571.71/0.00)
[ 222s] threads: 16 tps: 10307.88 qps: 10307.88 (r/w/o: 0.00/10307.88/0.00)
[ 223s] threads: 16 tps: 8220.26 qps: 8220.26 (r/w/o: 0.00/8220.26/0.00)
[ 224s] threads: 16 tps: 6381.09 qps: 6381.09 (r/w/o: 0.00/6381.09/0.00)
[ 225s] threads: 16 tps: 10348.85 qps: 10348.85 (r/w/o: 0.00/10348.85/0.00)
[ 226s] threads: 16 tps: 9383.95 qps: 9383.95 (r/w/o: 0.00/9383.95/0.00)
[ 227s] threads: 16 tps: 10528.06 qps: 10528.06 (r/w/o: 0.00/10528.06/0.00)

[ 280s] threads: 16 tps: 10335.09 qps: 10335.09 (r/w/o: 0.00/10335.09/0.00)
[ 281s] threads: 16 tps: 10372.06 qps: 10372.06 (r/w/o: 0.00/10372.06/0.00)
[ 282s] threads: 16 tps: 10237.61 qps: 10237.61 (r/w/o: 0.00/10237.61/0.00)
[ 283s] threads: 16 tps: 8206.20 qps: 8206.20 (r/w/o: 0.00/8206.20/0.00)
[ 284s] threads: 16 tps: 6050.79 qps: 6050.79 (r/w/o: 0.00/6050.79/0.00)
[ 285s] threads: 16 tps: 10053.31 qps: 10053.31 (r/w/o: 0.00/10053.31/0.00)
[ 286s] threads: 16 tps: 10208.14 qps: 10208.14 (r/w/o: 0.00/10208.14/0.00)
[ 287s] threads: 16 tps: 10315.78 qps: 10315.78 (r/w/o: 0.00/10315.78/0.00)
```

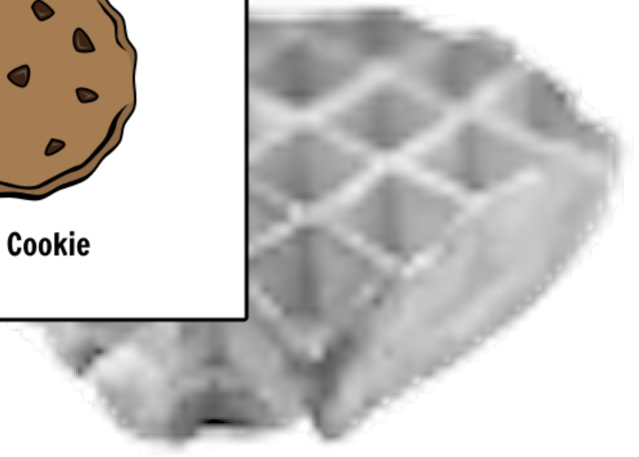
#84774 Performance drop every 60 seconds



Performance

Split-Brain Consistency & Usability first





MySQL Router

MySQL Router

- Quite simple load balancer:
    - TCP port for Writes & Reads
    - TCP port for Reads
- Routing Strategies (almost only valuable configuration setting)

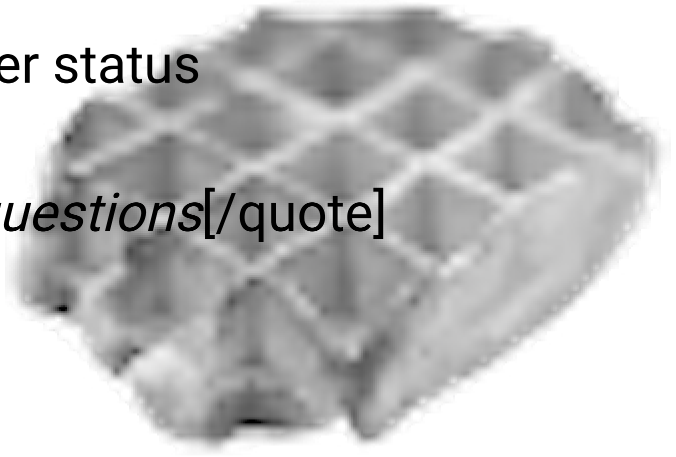
```
first-available
next-available
round-robin
round-robin-with-fallback
```

MySQL Router

- Quite simple load balancer:
  - TCP port for Writes & Reads
  - TCP port for Reads
- Routing Strategies (almost only valuable configuration setting)

  ```
  first-available
  next-available
  round-robin
  round-robin-with-fallback
  ```

- #83236: Not possible to see mysqlrouter status
  [quote]that's by design
  bugs.mysql.com is not a place to ask questions[/quote]



MySQL Router

Limitations:

- No transparent read write splitting
- No query caching
- No connection multiplexing
- No way to get the router status
- No query rules
- No traffic mirroring
- No firewall

MySQL Router

Limitations:

- No transparent read write splitting
- No query caching
- No connection multiplexing
- No way to get the router status
- No query rules
- No traffic mirroring
- No firewall

Use ProxySQL!

Best Practices

Best Practices - Architecture

- Uneven amount of nodes
- Not recommended for WAN
    - => important timeouts are not configurable yet
- Use an intelligent Load Balancer
    - => #84729 Impossible to block reads on partitioned nodes

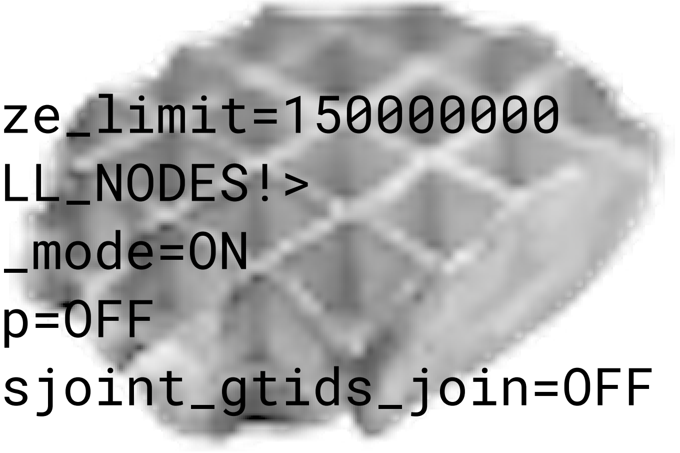
Best Practices - Configuration Settings

```
hostname=RESOLVABLE
super_read_only=ON
group_replication_unreachable_majority_timeout=20
log_error_verbosity=3
group_replication_ssl_mode=REQUIRED
disabled_storage_engines="MyISAM,BLACKHOLE,FEDERATED,
                          ARCHIVE,MEMORY"
group_replication_auto_increment_increment=1
```

Best Practices - Configuration Settings

```
hostname=RESOLVABLE
super_read_only=ON
group_replication_unreachable_majority_timeout=20
log_error_verbosity=3
group_replication_ssl_mode=REQUIRED
disabled_storage_engines="MyISAM,BLACKHOLE,FEDERATED,
                          ARCHIVE,MEMORY"
group_replication_auto_increment_increment=1
```

extra when using **5.7 & < 8.0.11**



```
group_replication_transaction_size_limit=150000000
group_replication_group_seeds=<ALL_NODES!>
group_replication_single_primary_mode=ON
group_replication_bootstrap_group=OFF
group_replication_allow_local_disjoint_gtids_join=OFF
```

Best Practices

```
hostname=VALID_RESOLVABLE_HOSTNAME
```

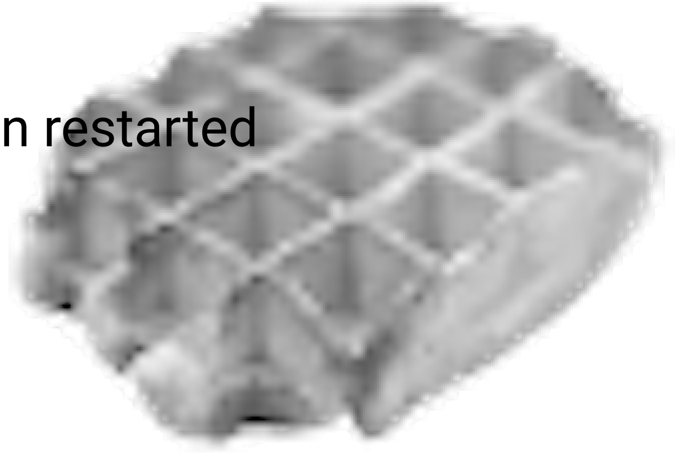
other GR nodes will resolve the hostname to setup connections

Best Practices

```
super_read_only=ON
```

Avoid PEBCAK split brain!

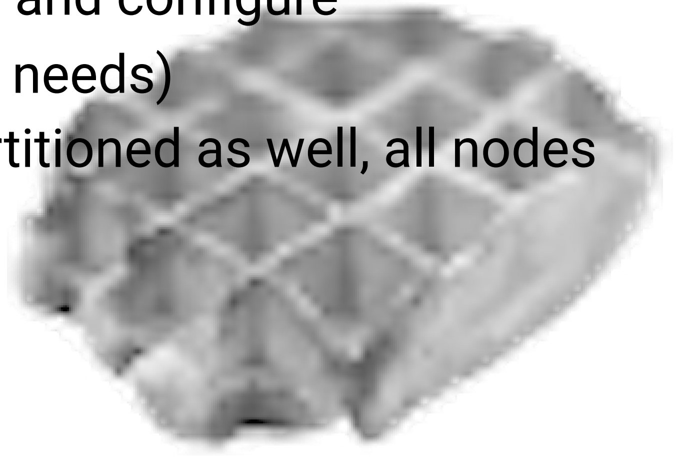
- Using `mysqlsh` with `< 8.0.11` does not persist configuration and GR does not start on boot
  - => writeable single mysql node when restarted



Best Practices

```
gr_unreachable_majority_timeout=20
```

- Applications will get an error instead of hanging forever (Default `0`)
- `20` seconds will abort group replication and configure `super_read_only=ON` (adapt to your needs)
- Drawback: if remaining 2 nodes get partitioned as well, all nodes go in ERROR and bootstrap is required



Best Practices

```
log_error_verbosity=3
```

In MySQL 8, output is scarce, configure verbosity level 3 to allow better troubleshooting.

Best Practices

`gr_ssl_mode=REQUIRED`

- `DISABLED` (default)
- Similar to client `ssl-mode=REQUIRED`
- `mysqlsh` (py): `dba.create_cluster('maurage', (memberSslMode='REQUIRED'))`

Best Practices

```
disabled_storage_engines=
"MyISAM,BLACKHOLE,FEDERATED,
ARCHIVE,MEMORY"
```

Only InnoDB is supported!

Best Practices

```
gr_auto_increment_increment=1
```

- Default 7
- Single-Primary/Writer is recommended
- No need for >1

Best Practices

```
gr_transaction_size_limit=150000000
```

- `< 8.0.2` default: unlimited maximum size of transactions
- `>= 8.0.2` default: 143,0511474609MB
- Keep Memory available for GR

Best Practices

```
gr_group_seeds=<ALL_NODES!>
```

- `< 8.0.11`: with `mysqlsh` configured cluster does not properly configure seeds causing nodes not to rejoin #90438
- Configure IP Addresses, not hostnames #90483

Best Practices

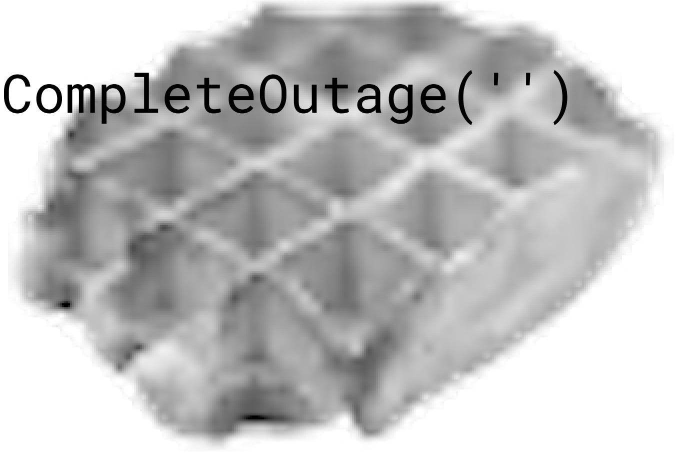
```
gr_single_primary_mode=ON
```

```
I have read the MySQL InnoDB cluster manual and
I understand the requirements and limitations
of advanced Multi-Master Mode.
Confirm [y/N]: NO
```

Best Practices

`gr_bootstrap_group=OFF`

- Do not set this to `ON`, ONLY when creating a cluster.
  - does not go to `OFF` automatically
  - set back to `OFF` immediately
  - => use `dba.rebootClusterFromCompleteOutage('')` in some scenarios



Best Practices

```
gr_allow_local_disjoint_gtids_join=OFF
```

- Don't even try to live with errant transactions
  - Big concern for data consistency
- Removed in 8.0.4

Limitations

Limitations

GR does not support:

- `GET_LOCK()`
- `binlog_format=STATEMENT`
- Large transactions
- `SELECT FOR UPDATE` ([#85998](#))
- IPv6 ([#90217](#))
- Non InnoDB Storage Engines
- Consistent reads on all nodes
- No PK on all tables

Production Ready?

Production Ready?

Good

- Solid split brain prevention
- `mysqlsh` in 8.0.11 really starts to show it's power!

Production Ready?

Not So Good

- Many of the features listed in this presentation

Bad

- #84729: Impossible to block reads on partitioned nodes
- #90484: No (easy) way to know if a GR node is writable
- Compared to Percona XtraDB Cluster/Galera Cluster:
  - No automatic node provisioning
  - Not possible to have synchronous reads

Ugly

- #84784: Nodes do not reconnect

Production Ready? - My Opinion

(for the masses)

Component	MySQL 5.7 GA	MySQL 8.0 GA (+)
MySQL Shell	NO	YES
MySQL Router	NO (#)	NO (#)
Group Replication	NO (*)	NO (*)

Production Ready? - My Opinion

(for the masses)

Component	MySQL 5.7 GA	MySQL 8.0 GA (+)
MySQL Shell	NO	YES
MySQL Router	NO (#)	NO (#)
Group Replication	NO (*)	NO (*)

(+) MySQL 8.0 is new, expect early adoption issues

(#) Use ProxySQL!

(*) **Early Adopters** required, much needed feedback to make the product better.

Production Ready? - My Opinion

(for the masses)

Component	MySQL 5.7 GA	MySQL 8.0 GA (+)
MySQL Shell	NO	YES
MySQL Router	NO (#)	NO (#)
Group Replication	NO (*)	NO (*)

(+) MySQL 8.0 is new, expect early adoption issues

(#) Use ProxySQL!

(*) **Early Adopters** required, much needed feedback to make the product better.

- Best Practices!
- Test thoroughly!

Ready For Production? (2018-04)

Ready For Production? (2018-04)

- bottled end of 2016

Ready For Production? (2018-04)

- bottled end of 2016
- delicious gem, still youthful

Ready For Production? (2018-04)

- bottled end of 2016
- delicious gem, still youthful
- already very enjoyable for connoisseurs

Ready For Production? (2018-04)

- bottled end of 2016
- delicious gem, still youthful
- already very enjoyable for connoisseurs
- great legs

Ready For Production? (2018-04)

- bottled end of 2016
- delicious gem, still youthful
- already very enjoyable for connoisseurs
- great legs
- nice structure

Ready For Production? (2018-04)

- bottled end of 2016
- delicious gem, still youthful
- already very enjoyable for connoisseurs
- great legs
- nice structure
- needs some decanting to become top-knotch

Ready For Production? (2018-04)

- bottled end of 2016
- delicious gem, still youthful
- already very enjoyable for connoisseurs
- great legs
- nice structure
- needs some decanting to become top-knotch
- KG: 90 points