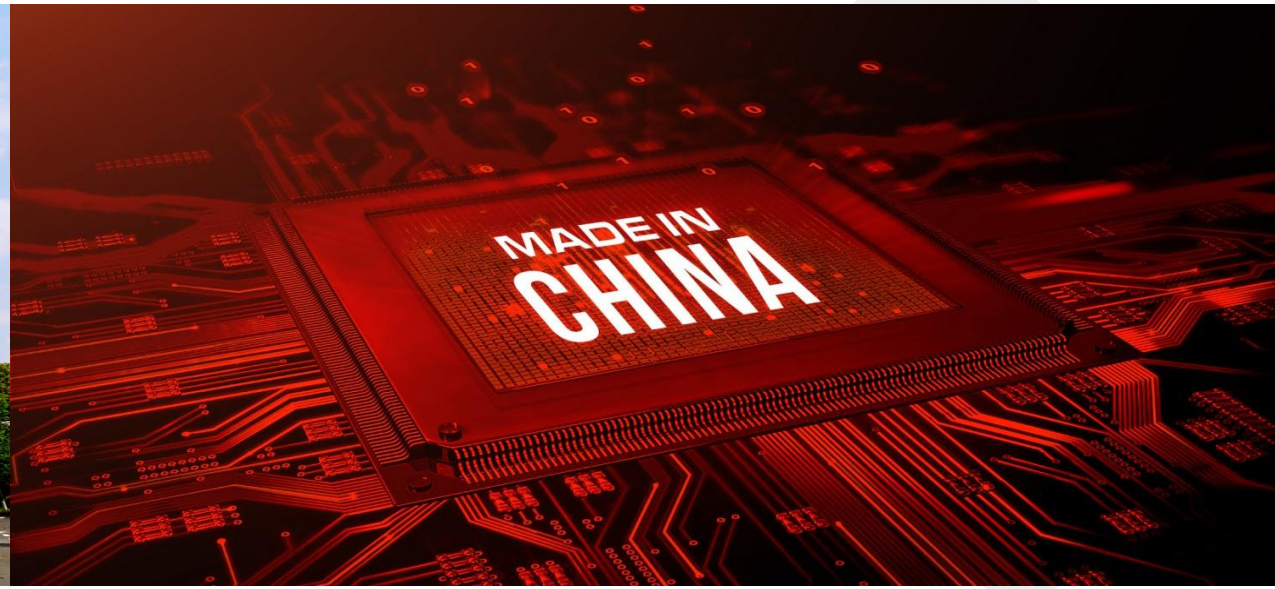
Great DB
万里数据库



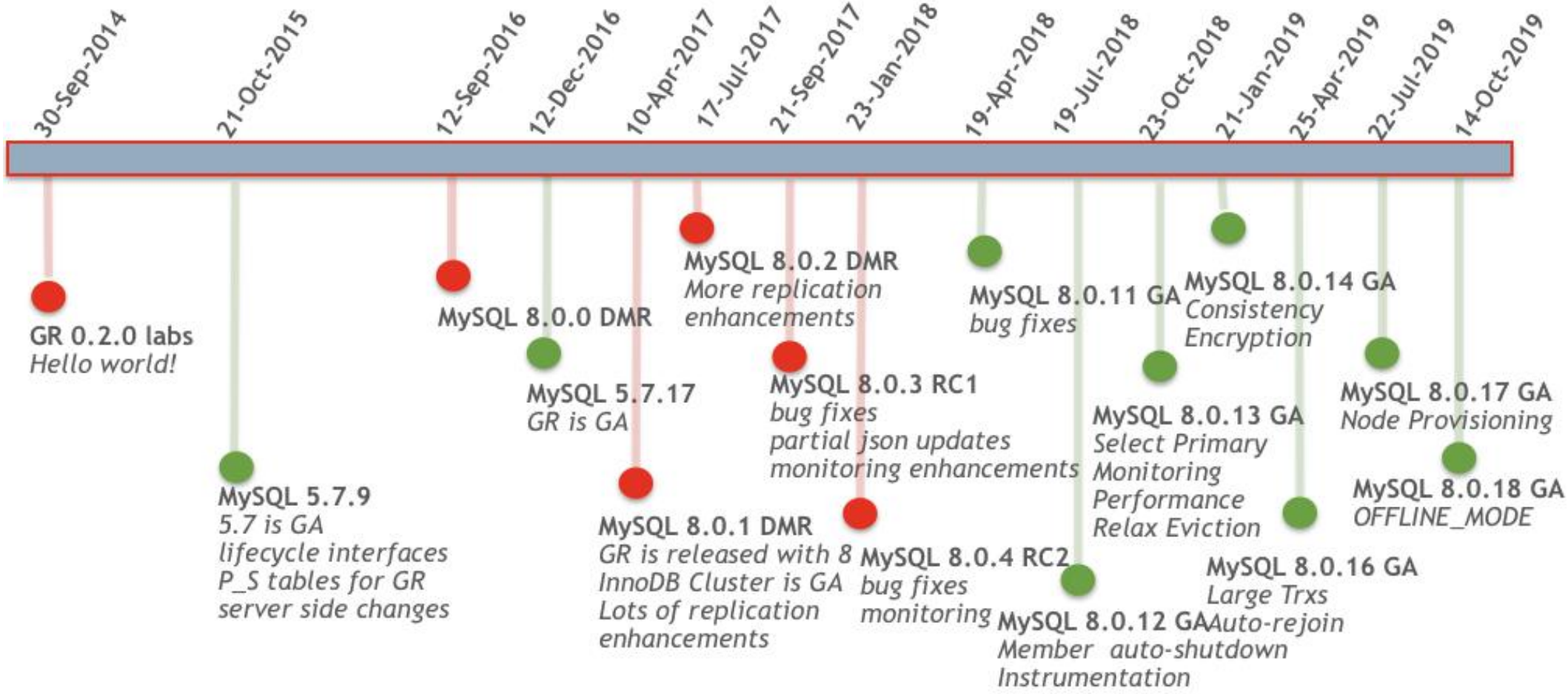
MGR Best Practice

万里数据库 娄帅

2021.5.22

MySQL Group Replication

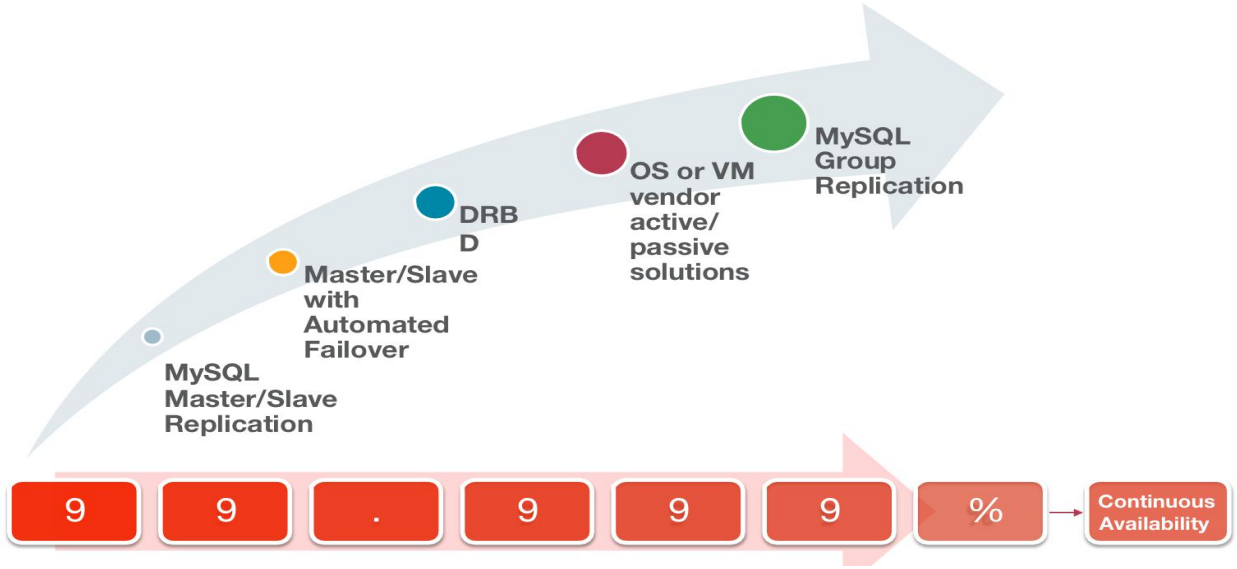
- Developed by Oracle
- GA in MySQL 5.7.17 on December 2016
- InnoDB Cluster作为整体解决方案



Asynchronous Replication vs. MGR

Async

- 异步传送
- 主 -> 从
- 从库通过binlog dump抽取主库binlog并执行
- 人工或者外部脚本进行故障切换



GR

- 全同步传送
- 节点间传递
- 多数派节点收到事务消息(PAXOS)
- 自动处理故障切换，集群成员变化，选主

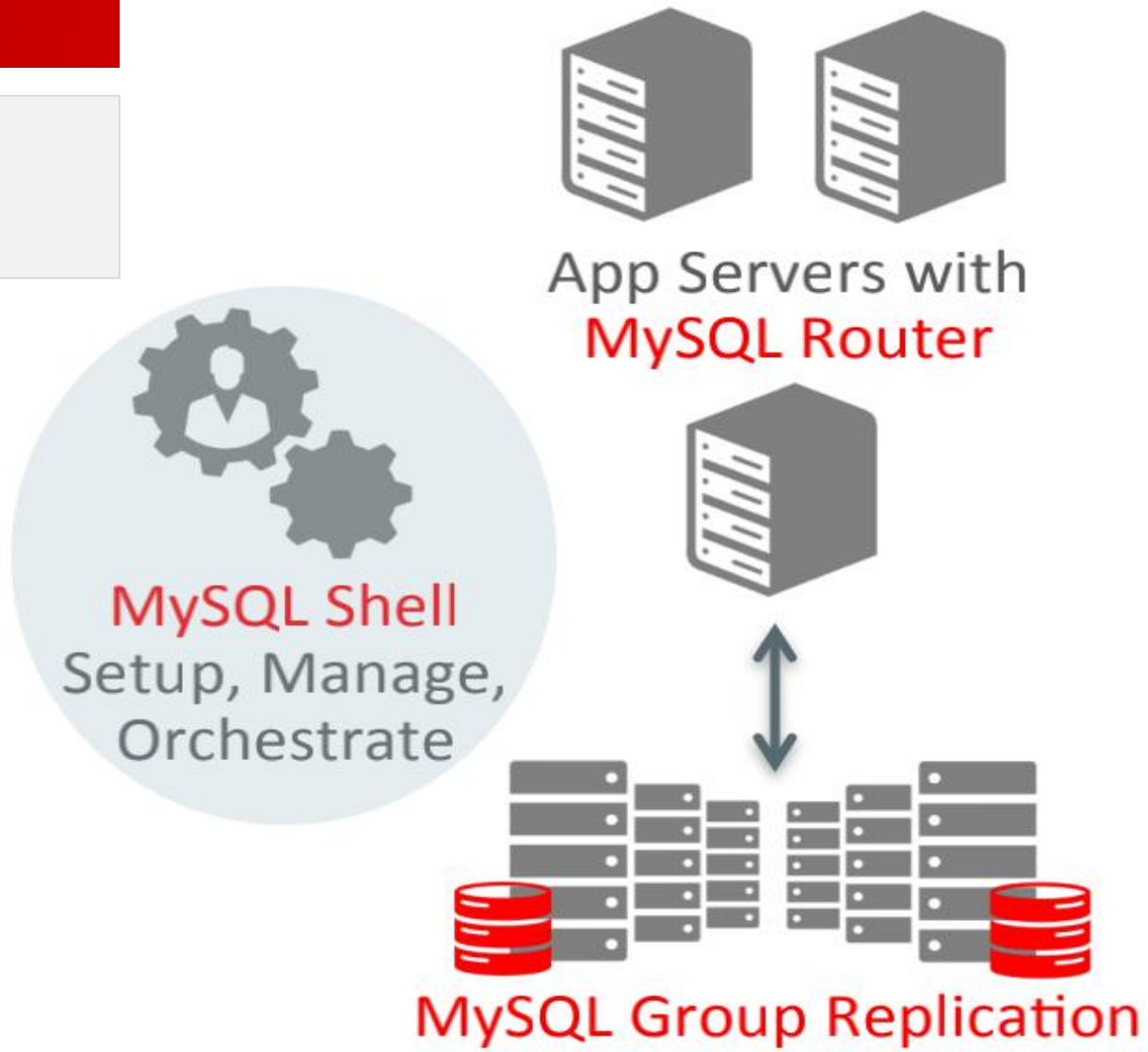
适用场景

- 数据强一致，故障时数据0丢失，RPO=0
- 更快的故障切换，无需第三方切换脚本
- 防止脑裂

InnoDB Cluster —— A Solution

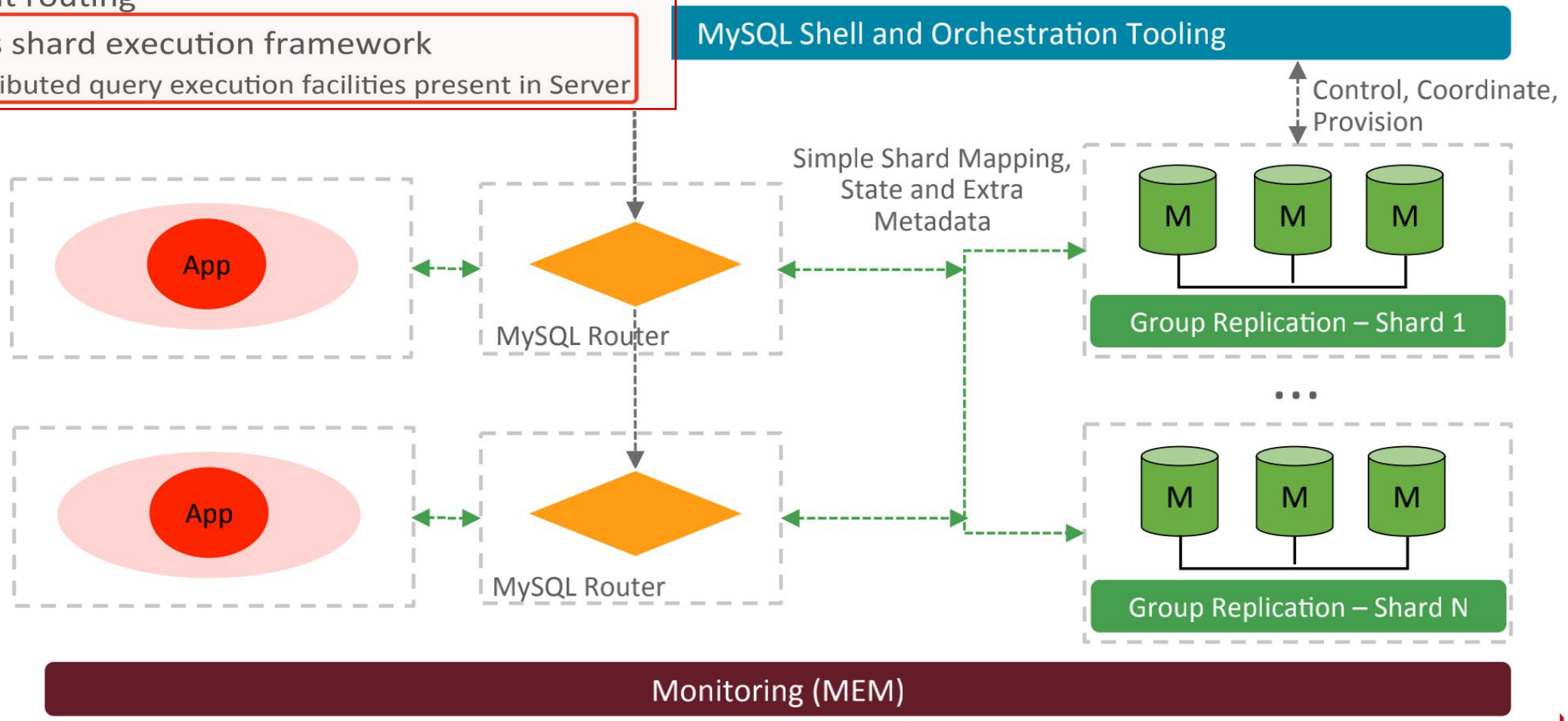
InnoDB Cluster解决方案

- MySQL Router：路由客户端连接
- MySQL Shell: 集群搭建、管理工具



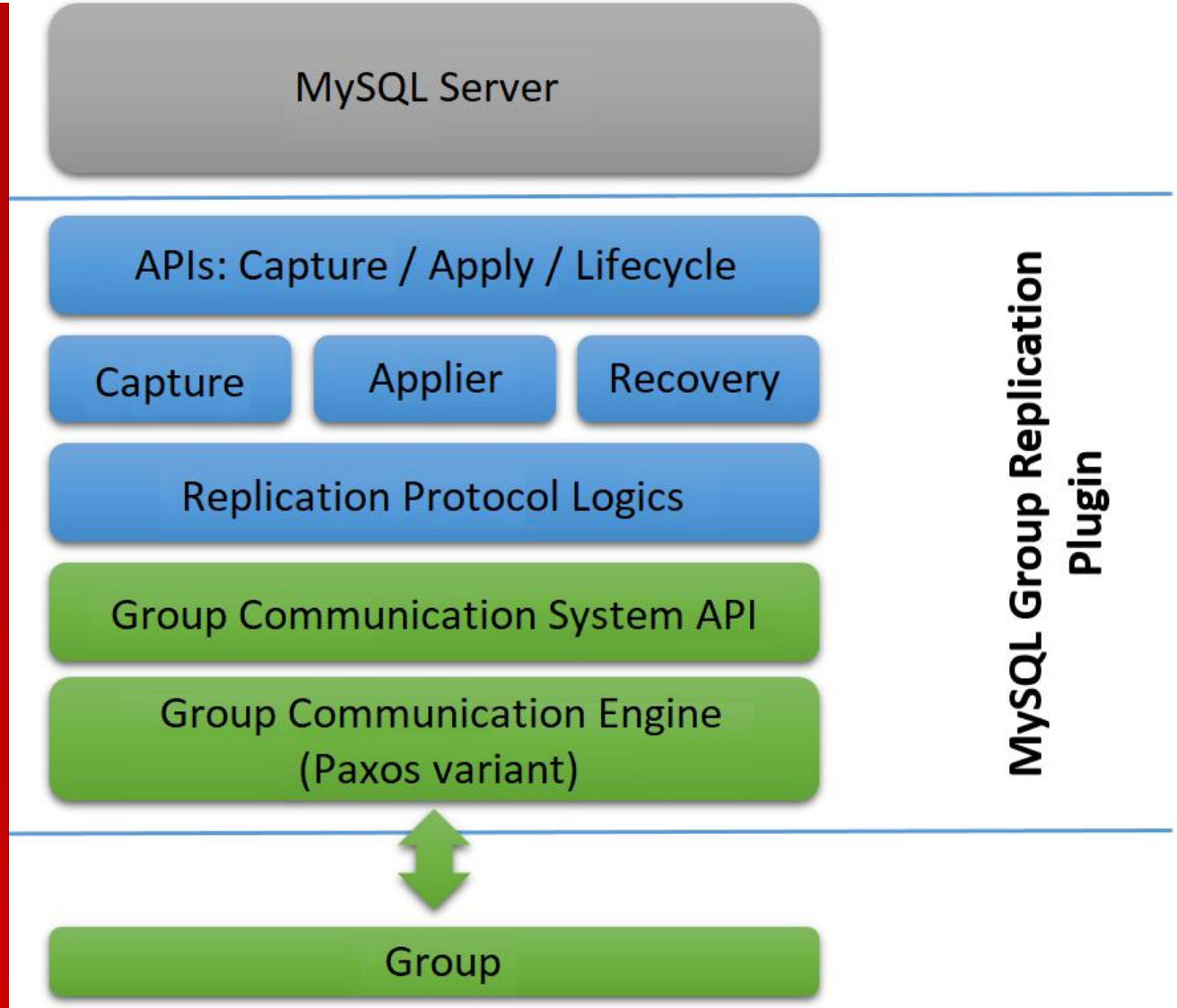
InnoDB Cluster: The End Goal

- MySQL Router
  - Manages shard mappings and related metadata
  - Manages client routing
  - Provides cross shard execution framework
    - On top of distributed query execution facilities present in Server

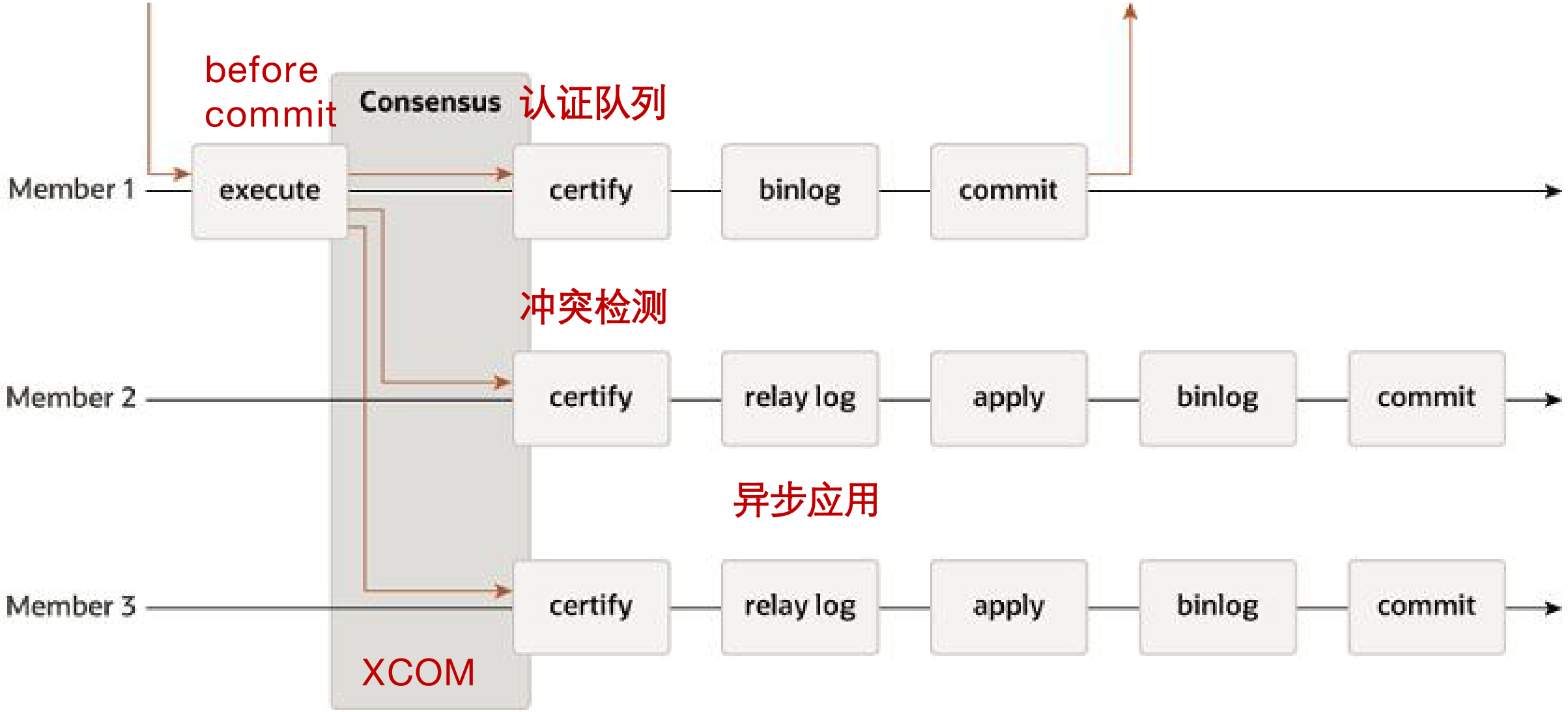


MGR Architecture

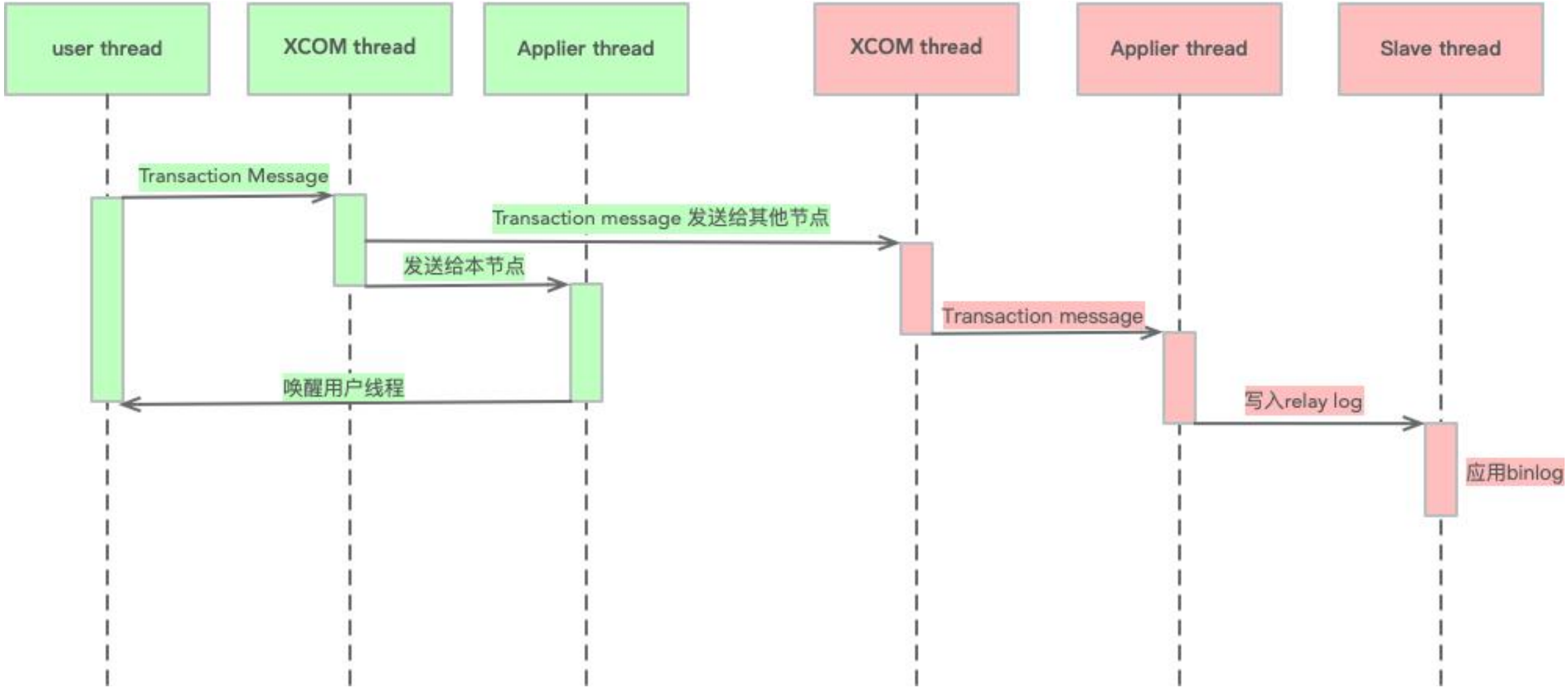
- **Capture:** 跟踪本节点的事务相关信息
- **Applier:** 执行其它节点的远程事务
- **Recovery:** 负责故障恢复时，选择donar节点，catch up binlog等
- **Replication Protocol Logics:** 消息的封装、接收XCOM返回的消、发送本节点消息给XCOM、冲突检测等
- **GCE:** GCS的具体实现Xcom (eXtended COMmunications)



Life of a Transaction Commit in MGR



Thread Viewport of Commit in MGR



目录

Architecure

Great DB 万里数据库

- 使用奇数个节点
- 网络稳定，延迟低，尽量避免WAN部署
- 使用单主模式
- 表必须有主键
- 必须使用InnoDB引擎
- BINLOG_FORMAT=ROW

Basic Configurations

Great DB 万里数据库

- group_replication_single_primary_mode=ON
  - 单主模式
- log_error_verbosity=3
  - 默认为2，只输出Error，Warning
  - 3包含Information，能够输出更多GR的日志
- group_replication_group_seeds=<ALL_NODES!>
  - 填写所有节点
- group_replication_bootstrap_group=OFF
  - 只有在搭建新集群的过程中打开
  - 新集群搭建完成后立刻关闭
  - 避免节点重启后，搭建新的集群

Large Transaction & Flow Control Configurations

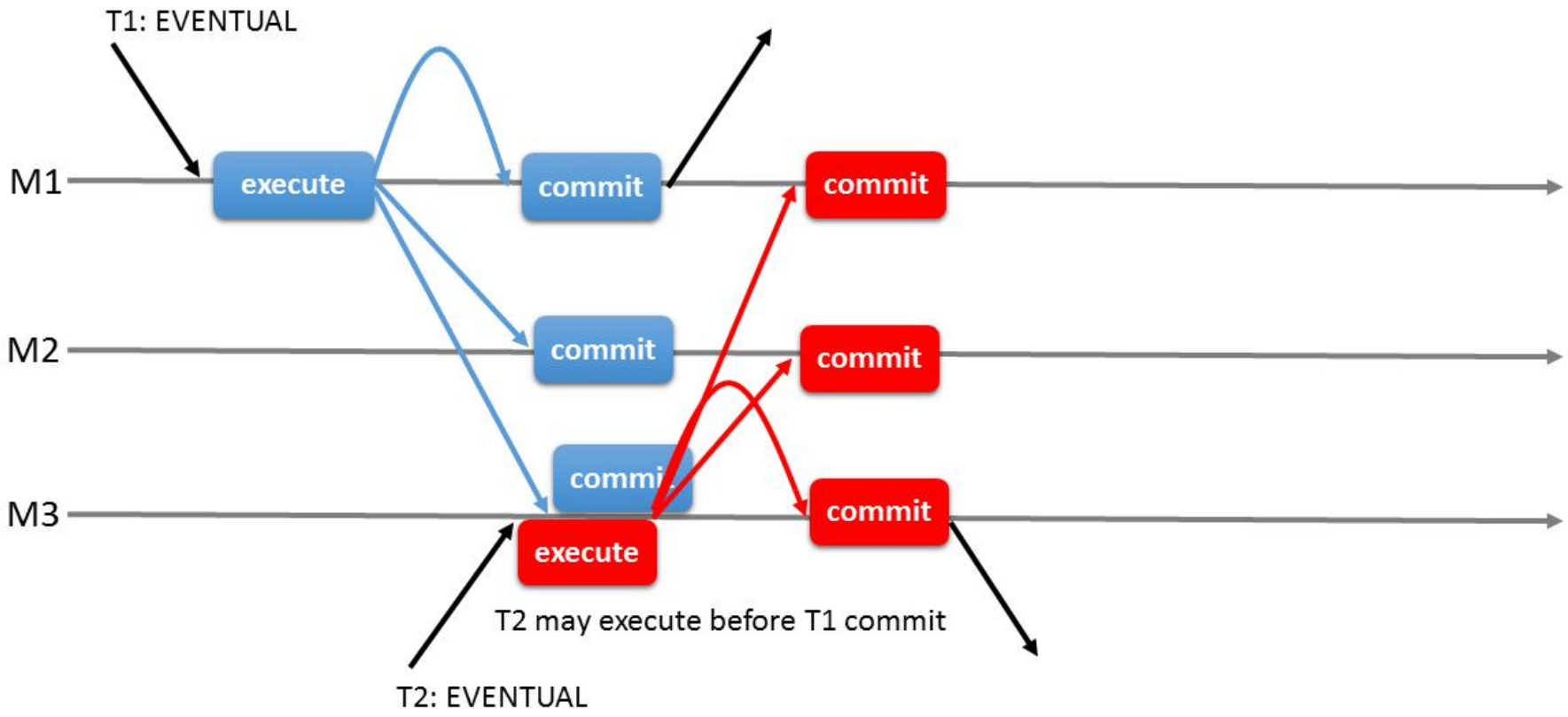
Great DB 万里数据库

- group_replication_transaction_size_limit=150000000
  - 单个事务大小上限
  - 尽量避免大事务出现
  - **只会在commit时报错，事务执行过程中即使超过阈值也无感知**
- group_replication_communication_max_message_size=10M
  - 将大事务切分成小包，进行paxos传递
- group_replication_flow_control_mode=ON
  - 流控开关
  - **触发流控后，只会延迟等待1s**
- group_replication_flow_control_certifier_threshold
  - 触发流控的待认证的队列长度
- group_replication_flow_control_applier_threshold
  - 触发流控的待执行的队列长度

Consistency Level —— EVENTUAL (Default)

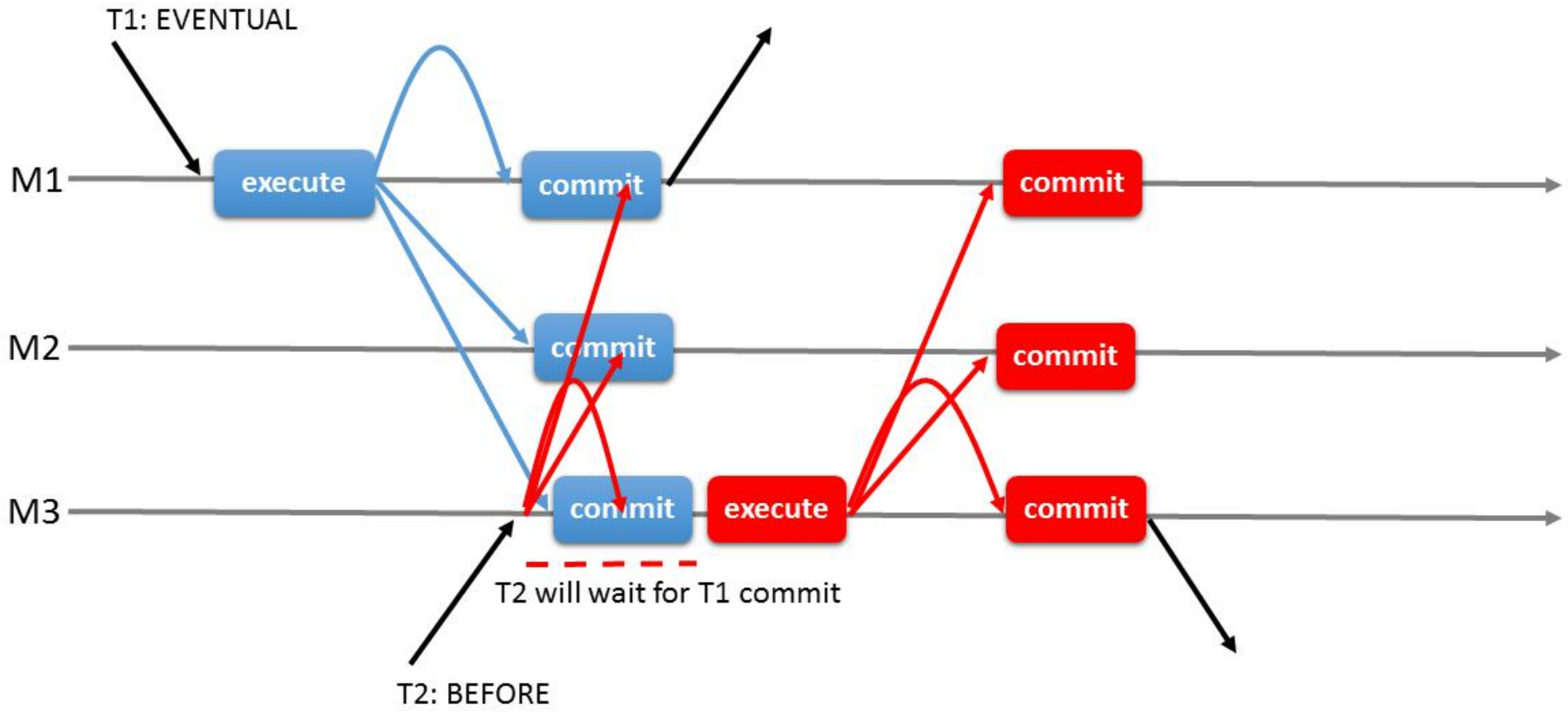
T2不关心是否有待应用的事务，不做任何等待操作。

存在读不到最新数据的可能。



Consistency Level —— BEFORE

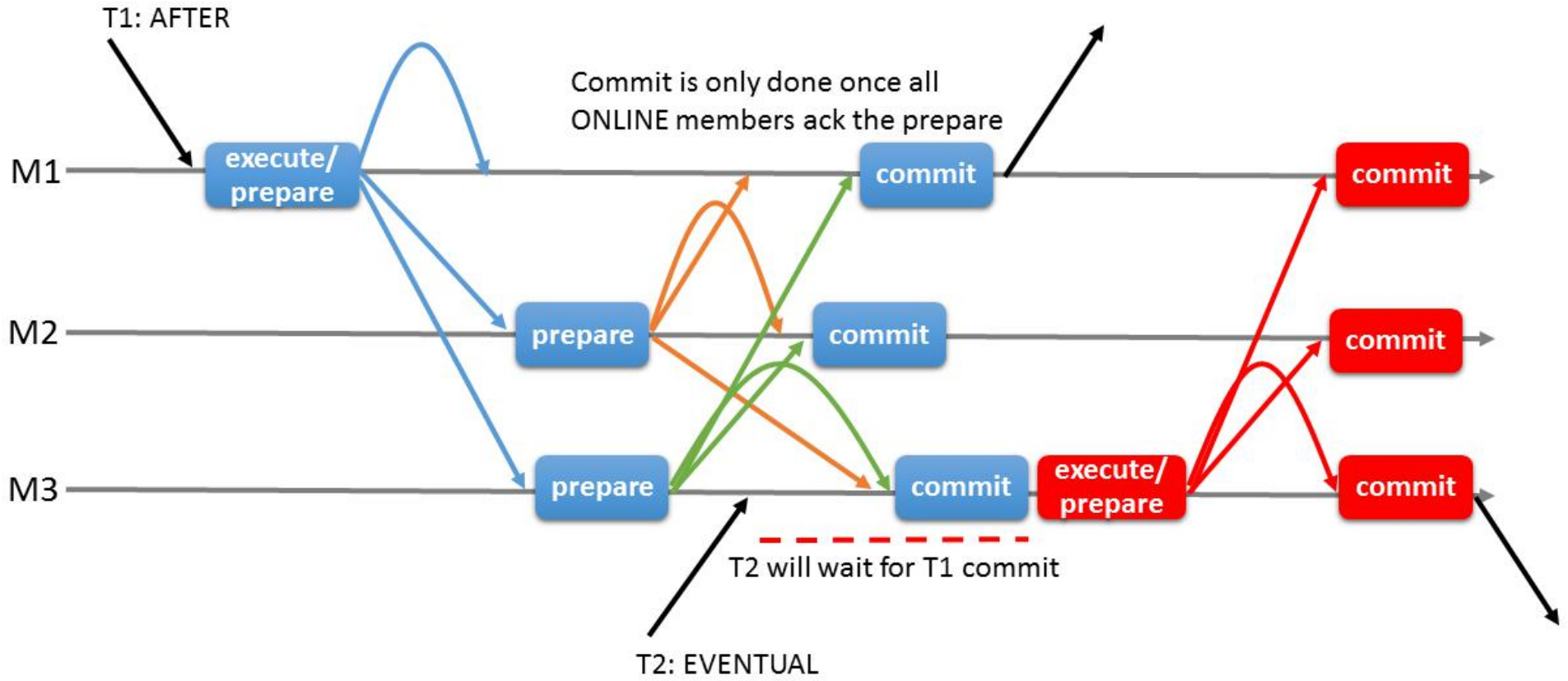
- T2等待待应用的事务执行完成，保证T2可以读到最新的数据。
- 增加T2的响应延迟，如果applier速度过慢，会导致延迟较大。
- BEFORE_ON_PRIMARY_FAILOVER: 发生选主后，事务会被阻塞，直到新主应用完堆积的日志。



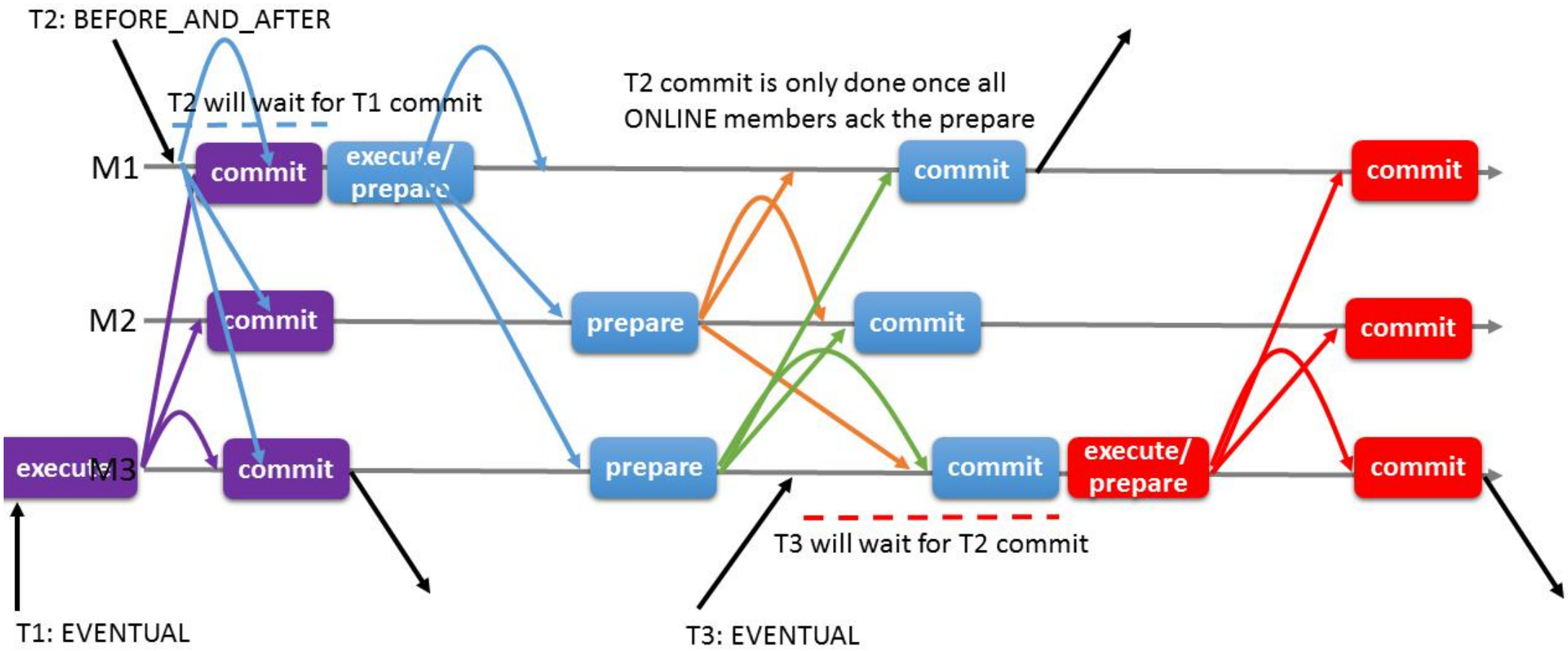
Consistency Level —— AFTER

- T1等待其它节点进入commit阶段后，才真正返回给客户。
- T2如果发现有prepare事务，需要等待prepare事务执行完成。

- 会增加T1的响应延迟。如果远端应用过慢，会导致T1等待时间过长。
- T2也需要等待T1执行完成。



Consistency Level —— BEFORE_AND_AFTER



Consistency Level —— How to Choose

Great DB 万里数据库

如果只在PRIMARY读写

- 要求写事务在其它节点同时应用，使用AFTER
- 否则使用BEFORE_ON_PRIMAY_FAILOVER

如果load balance读，不希望读到历史数据

- 大量写入场景，使用BEFORE
- 少量写入场景，使用AFTER
- 指定特定需要的事务使用BEFORE

Consistency Level —— How to Choose

- **目前不太建议使用AFTER隔离级别，机制比较复杂，存在几种已知未修复bug，会导致主节点或者其它节点异常退出。**

- **导致主节点退出，8.0.26将会修复。**
  - 2020-09-29T06:40:09.508840Z 17 [ERROR] [MY-013309] [Repl] Plugin group_replication reported: 'Transaction '1:247' does not exist on Group Replication consistency manager while receiving remote transaction prepare.'

- **新加入的节点直接退出集群**
- 2021-01-22T11:53:52.710397Z 28 [ERROR] [MY-013304] [Repl] Plugin group_replication reported: 'Error releasing transaction '2:7' for commit on session '31' after being prepared on all group members.'

- 2021-01-29T11:39:16.767485Z 29 [ERROR] [MY-013309] [Repl] Plugin group_replication reported: 'Transaction '2:7' does not exist on Group Replication consistency manager while receiving remote transaction prepare.'

Replication Configuration

- binlog_transaction_dependency_tracking=WRITESET
  - 提升slave执行的并发度
- slave_checkpoint_period=2
  - BEFORE级别下，提升从库读性能
- slave_parallel_workers
  - 合理配置，提升从库执行效率
- slave_parallel_type=LOGICAL_CLOCK
- slave_preserve_commit_order=ON

Membership Change Configurations

- group_replication_unreachable_majority_timeout=10
  - 网络分区时，少数派状等待此时长后，状态变为Error，回滚pending事务
- group_replication_autorejoin_tries=3
  - 自动尝试连入集群的次数，尝试间隔5min。
- group_replication_exit_state_action=READ_ONLY
  - 退出集群后，server的状态设置
  - 配合自己的路由软件进行合理设置
- group_replication_member_expel_timeout=5
  - 将suspicious节点踢出集群的等待时长
  - 如果网络环境一般，可以适当调大30-60，不要太大

Other Configuration Recommendations

Great DB 万里数据库

避免foreign key 的使用

- https://bugs.mysql.com/bug.php?id=97836
- 尤其避免foreign key部分引用的情况 pk(a,b,c)，fk ref (a)
- 导致secondary节点应用relay失败

自增值相关

group_replication_auto_increment_increment

- **自增步长，替代auto_increment_increment**
- 可以配置成集群节点个数
- 多主情况下尤其要注意

server_id

- **MGR情况下, server_id 作为auto_increment_offset**
- 每个节点的server_id尽量从1开始，1，2，3
- 多主情况下尤其要注意

Other Configuration Recommendations

- group_replication_member_weight
  - 选主过程中的节点权重
  - 可以将主机房的节点权重加大
- clone_valid_donor_list
  - 新加入节点，clone全量数据时，选择的donar节点
  - 尽量使用非主节点作为donar节点

Limitations

- binlog_format=ROW
- 使用InnoDB表
- 表需要有主键
- 节点数限制，上限9
- 事务大小限制
- 不要不同版本混合使用

多主模式下的限制

- LOCK
  - LOCK TABLE
  - GET_LOCK
  - GAP LOCK
- 不支持SERIALIZABLE
- DDL和DML在不同节点并发执行
- SELECT FOR UPDATE导致DEAD LOCK

Performance_Schema

```
mysql3> SELECT * FROM performance_schema.replication_group_members\G
*************************** 1. row ***************************
  CHANNEL_NAME: group_replication_applier
     MEMBER_ID: ade14d5c-9e1e-11e7-b034-08002718d305
   MEMBER_HOST: mysql4
   MEMBER_PORT: 3306
  MEMBER_STATE: ONLINE
   MEMBER_ROLE: SECONDARY
MEMBER_VERSION: 8.0.16
*************************** 2. row ***************************
  CHANNEL_NAME: group_replication_applier
     MEMBER_ID: b9d01593-9dfb-11e7-8ca6-08002718d305
   MEMBER_HOST: mysql3
   MEMBER_PORT: 3306
  MEMBER_STATE: ONLINE
   MEMBER_ROLE: PRIMARY
MEMBER_VERSION: 8.0.16
```

成员状态

成员角色

Performance_Schema

```
mysql> select * from performance_schema.replication_group_member_stats\G
```

```
*************************** 1. row ***************************
                           CHANNEL_NAME: group_replication_applier
                                VIEW_ID: 15059231192196925:2
                              MEMBER_ID: ade14d5c-9e1e-11e7-b034-08002...
            COUNT_TRANSACTIONS_IN_QUEUE: 0
             COUNT_TRANSACTIONS_CHECKED: 27992
               COUNT_CONFLICTS_DETECTED: 0
      COUNT_TRANSACTIONS_ROWS_VALIDATING: 0
      TRANSACTIONS_COMMITTED_ALL_MEMBERS: 8fc848d7-9e1c-11e7-9407-08002...
b9d01593-9dfb-11e7-8ca6-08002718d305:1-21,
da2f0910-8767-11e6-b82d-08002718d305:1-164741
          LAST_CONFLICT_FREE_TRANSACTION: 8fc848d7-9e1c-11e7-9407-08002...
COUNT_TRANSACTIONS_REMOTE_IN_APPLIER_QUEUE: 0
       COUNT_TRANSACTIONS_REMOTE_APPLIED: 27992
       COUNT_TRANSACTIONS_LOCAL_PROPOSED: 0
```

等待冲突检测的队列中的事务数

等待slave线程apply的事务数

```
************ 2. row ***************************
                           CHANNEL_NAME: group_replication_applier
                                VIEW_ID: 15059231192196925:2
                              MEMBER_ID: b9d01593-9dfb-11e7-8ca6-08002...
            COUNT_TRANSACTIONS_IN_QUEUE: 0
             COUNT_TRANSACTIONS_CHECKED: 28000
               COUNT_CONFLICTS_DETECTED: 0
      COUNT_TRANSACTIONS_ROWS_VALIDATING: 0
      TRANSACTIONS_COMMITTED_ALL_MEMBERS: 8fc848d7-9e1c-11e7-9407-08002...
b9d01593-9dfb-11e7-8ca6-08002718d305:1-21,
da2f0910-8767-11e6-b82d-08002718d305:1-164741
          LAST_CONFLICT_FREE_TRANSACTION: 8fc848d7-9e1c-11e7-9407-08002...
COUNT_TRANSACTIONS_REMOTE_IN_APPLIER_QUEUE: 0
       COUNT_TRANSACTIONS_REMOTE_APPLIED: 1
       COUNT_TRANSACTIONS_LOCAL_PROPOSED: 28000
       COUNT_TRANSACTIONS_LOCAL_ROLLBACK: 0
```

Status During Recovery

Great DB 万里数据库

```
mysql> SHOW SLAVE STATUS FOR CHANNEL 'group_replication_recovery'\G
```

```
*************************** 1. row ***************************
           Slave_IO_State:
              Master_Host: <NULL>
              Master_User: gr_repl
              Master_Port: 0
                    ...
           Relay_Log_File: mysql4-relay-bin-group_replication_recovery.000001
                    ...
            Slave_IO_Running: No
           Slave_SQL_Running: No
                  ...
           Executed_Gtid_Set: 5de4400b-3dd7-11e6-8a71-08002774c31b:1-814089,
afb80f36-2bff-11e6-84e0-0800277dd3bf:1-5718
                  ...
             Channel_Name: group_replication_recovery
```

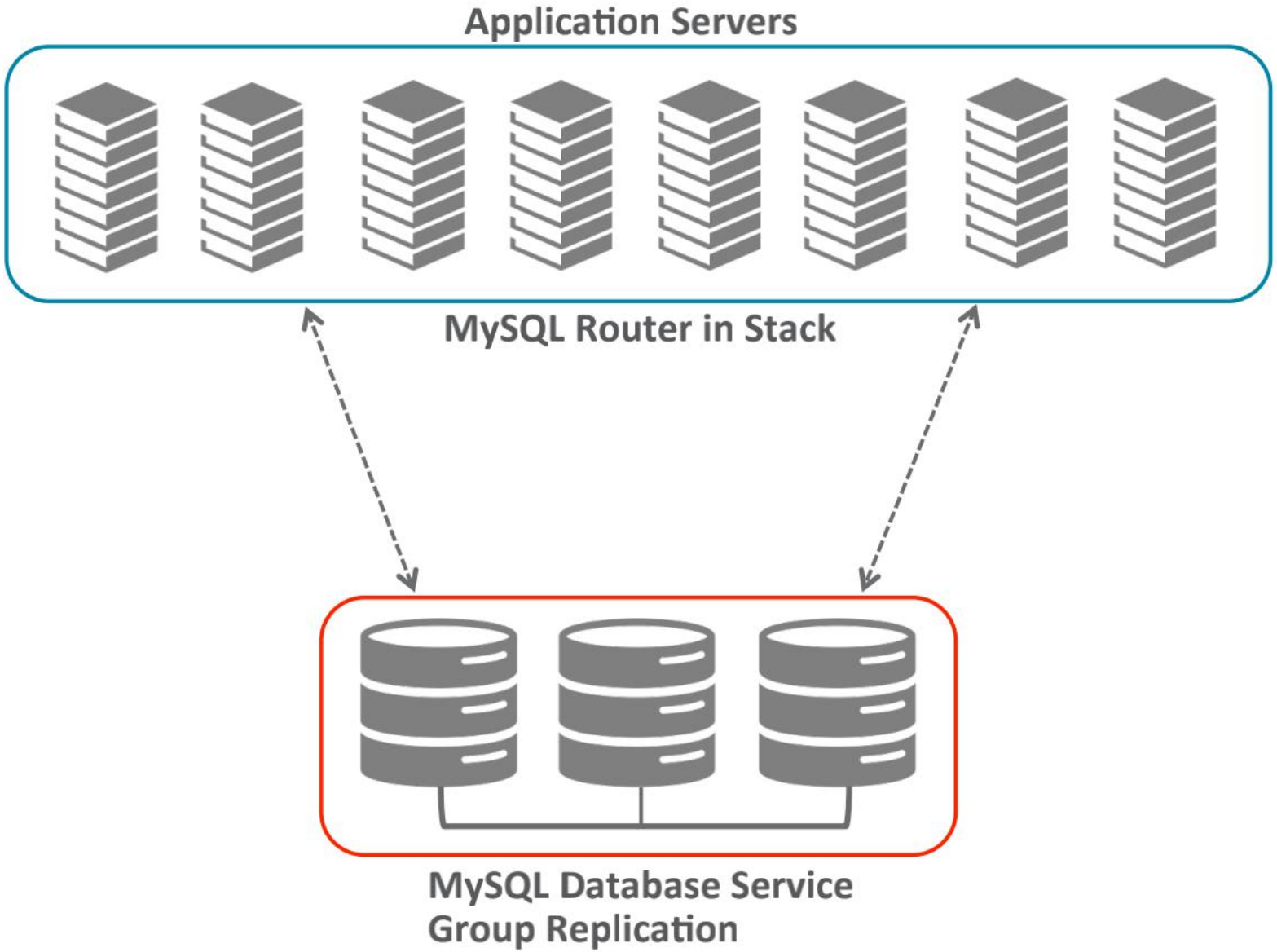
Change Primary Member

```
+---------------+--------------+-------------+----------------+
| member_host   | member_state | member_role | member_version |
+---------------+--------------+-------------+----------------+
| 172.28.128.15 | ONLINE       | SECONDARY   | 8.0.22         |
| 172.28.128.14 | ONLINE       | PRIMARY     | 8.0.22         |
| 172.28.128.13 | ONLINE       | SECONDARY   | 8.0.22         |
+---------------+--------------+-------------+----------------+
3 rows in set (0.00 sec)
```

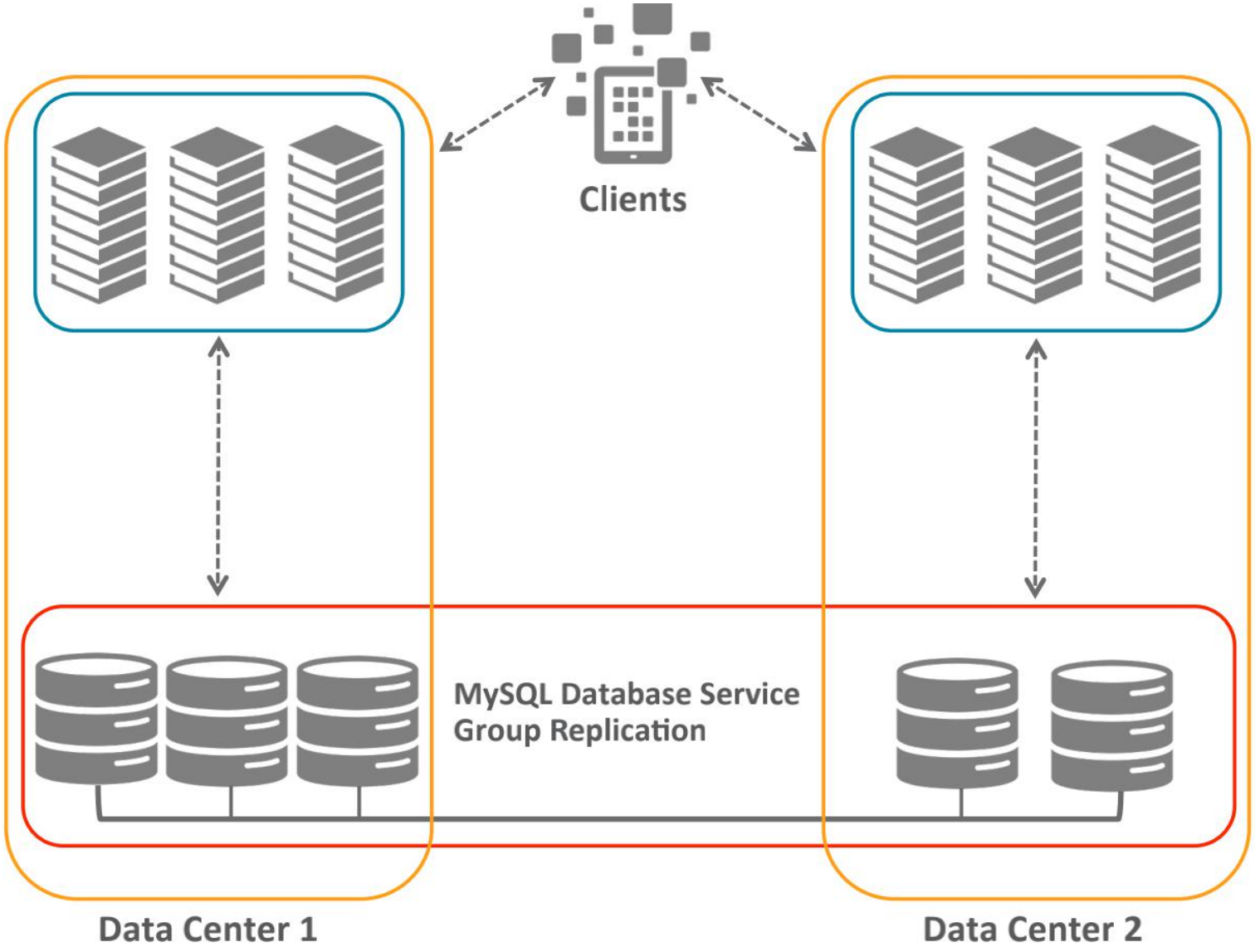
```
mysql> select group_replication_set_as_primary('c5aed435-d58d-11ea-bb26-5254004d77d3');
+-----------------------------------------------------------------------------+
| group_replication_set_as_primary('c5aed435-d58d-11ea-bb26-5254004d77d3') |
+-----------------------------------------------------------------------------+
| Primary server switched to: c5aed435-d58d-11ea-bb26-5254004d77d3         |
+-----------------------------------------------------------------------------+
1 row in set (1.03 sec)

mysql> select member_host,member_state,member_role,member_version from performance_schema.replication_group_members;
+---------------+--------------+-------------+----------------+
| member_host   | member_state | member_role | member_version |
+---------------+--------------+-------------+----------------+
| 172.28.128.15 | ONLINE       | PRIMARY     | 8.0.22         |
| 172.28.128.14 | ONLINE       | SECONDARY   | 8.0.22         |
| 172.28.128.13 | ONLINE       | SECONDARY   | 8.0.22         |
+---------------+--------------+-------------+----------------+
3 rows in set (0.00 sec)
```

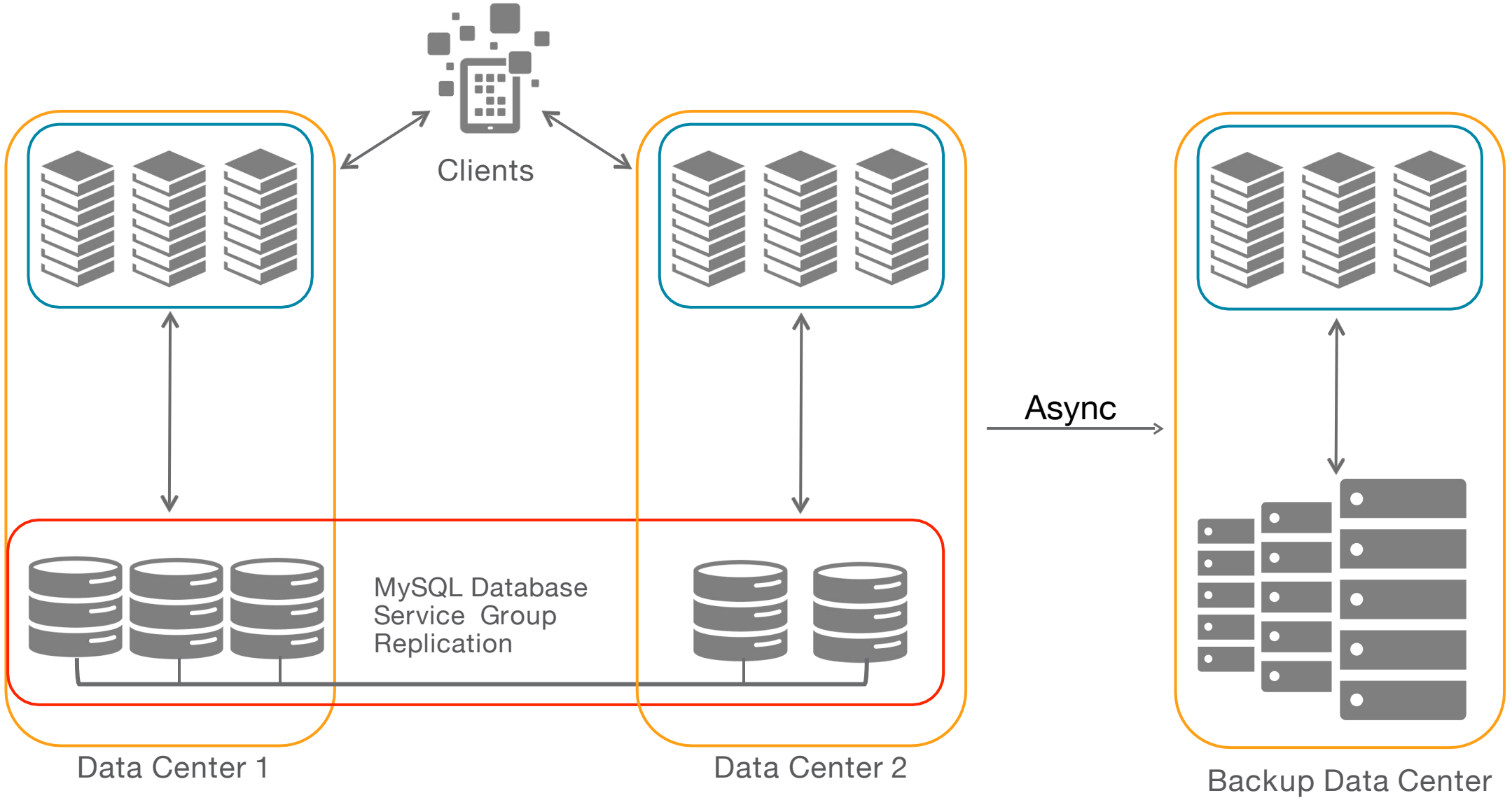
Single Data Center



Two Data Centers In One City



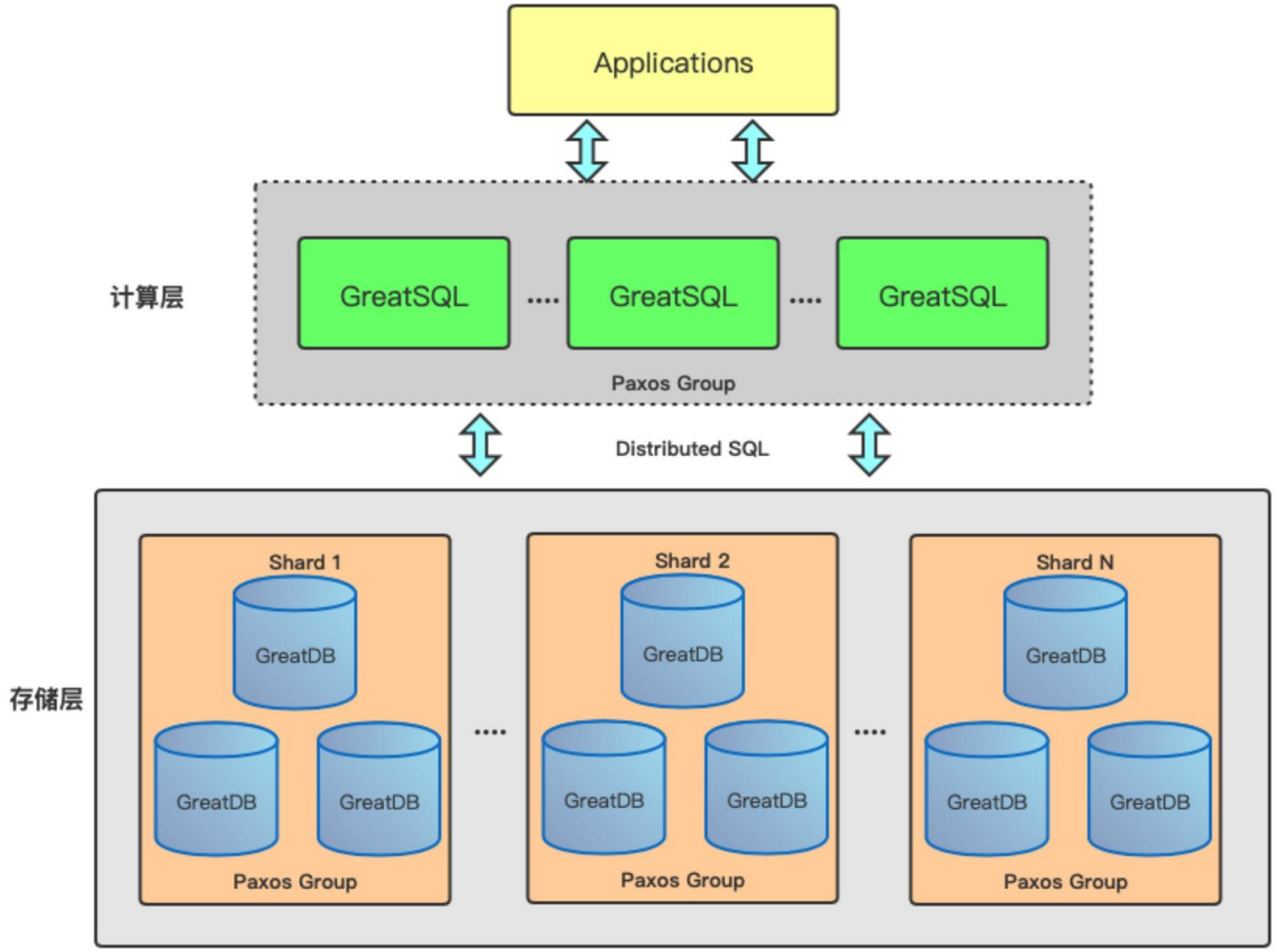
Three Data Centers in Two Cities



References

1. Why MySQL High Availability Matters
2. MySQL High Availability with Group Replication
3. MySQL High Availability with InnoDB Cluster
4. MySQL InnoDB Cluster - New Features in 8.0 Releases - Best Practices
5. MySQL Group Replication & MySQL InnoDB Cluster
6. MySQL InnoDB Cluster in a Nutshell

GreatDB Cluster




互动答疑环节

不忘DB初心，牢记万里使命