

# MySQL Group Replication

## 使用群组复制实现MySQL高可用性

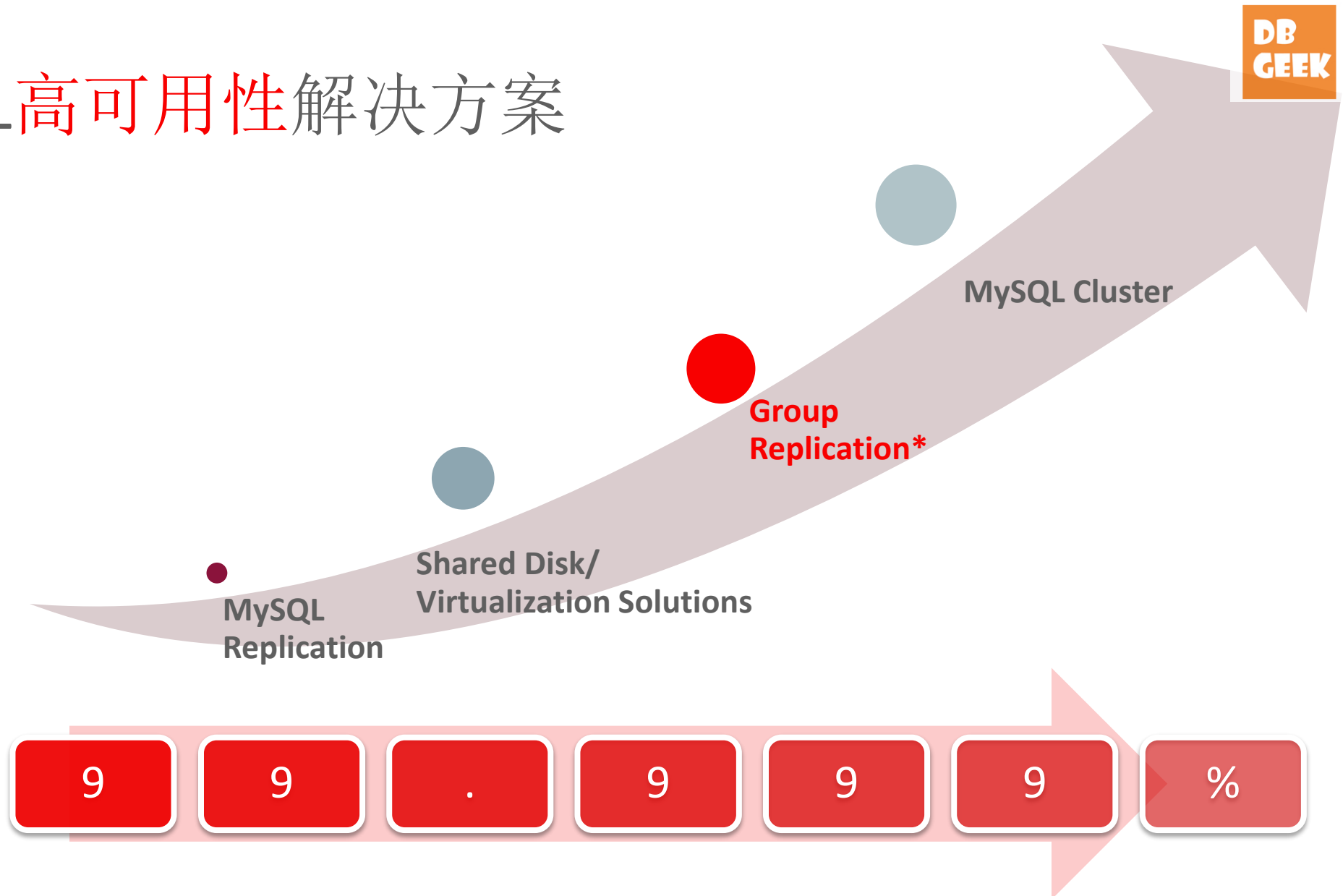
徐轶韬

MySQL Senior Sales Consultant

## Safe Harbor Statement /协议

以下内容旨在概述产品的总体发展方向, 它的目的是仅供参考, 不得纳入任何合同中。这不是一个承诺提供任何材料, 代码或功能, 不应该作为制定购买决策的依赖。开发, 发布, 以及为Oracle产品的任何特性或功能的安排均由Oracle自行决定。

# MySQL高可用性解决方案

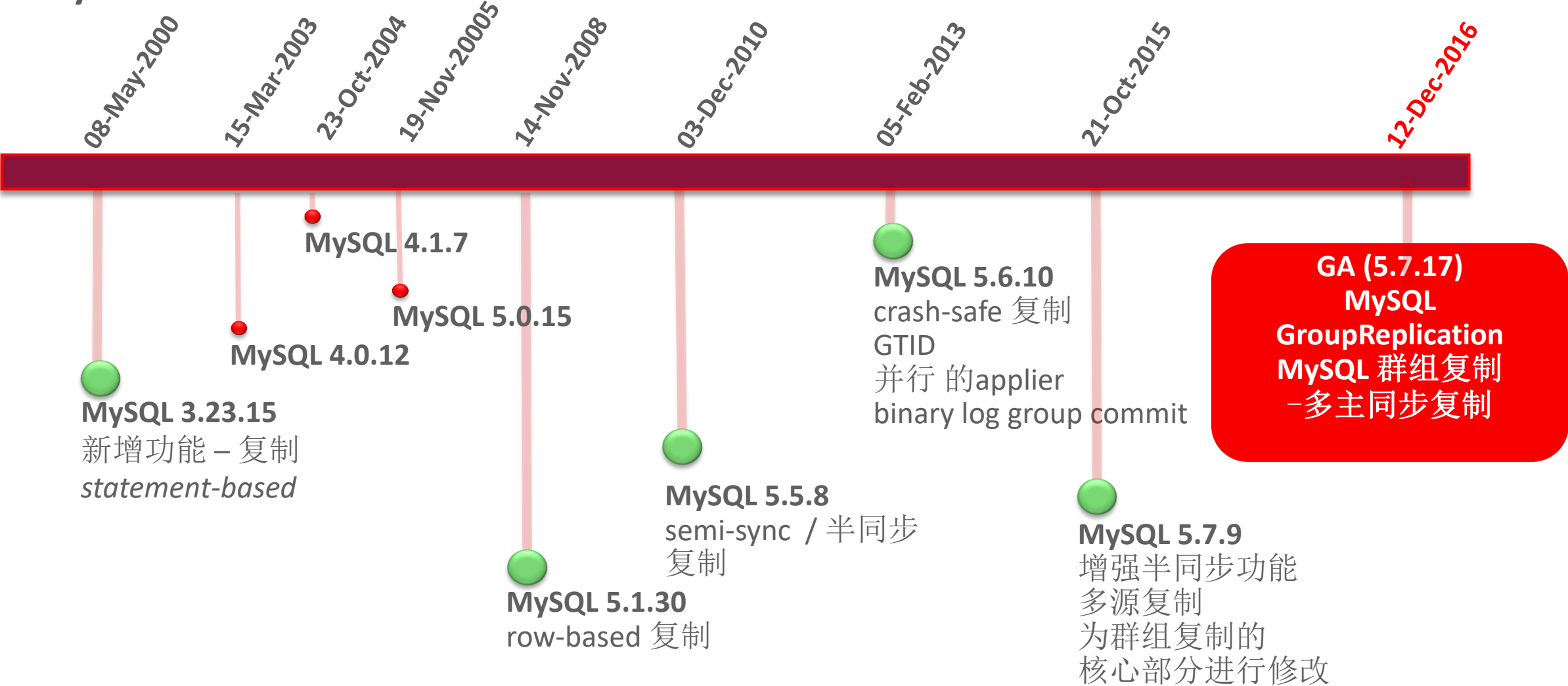


# 议程

- 1 MySQL复制的演变
- 2 MySQL群组复制
- 3 MySQL群组复制特性
- 4 MySQL群组复制性能
- 5 MySQL InnoDB Cluster – 路线发展

# 1 MySQL复制的演变

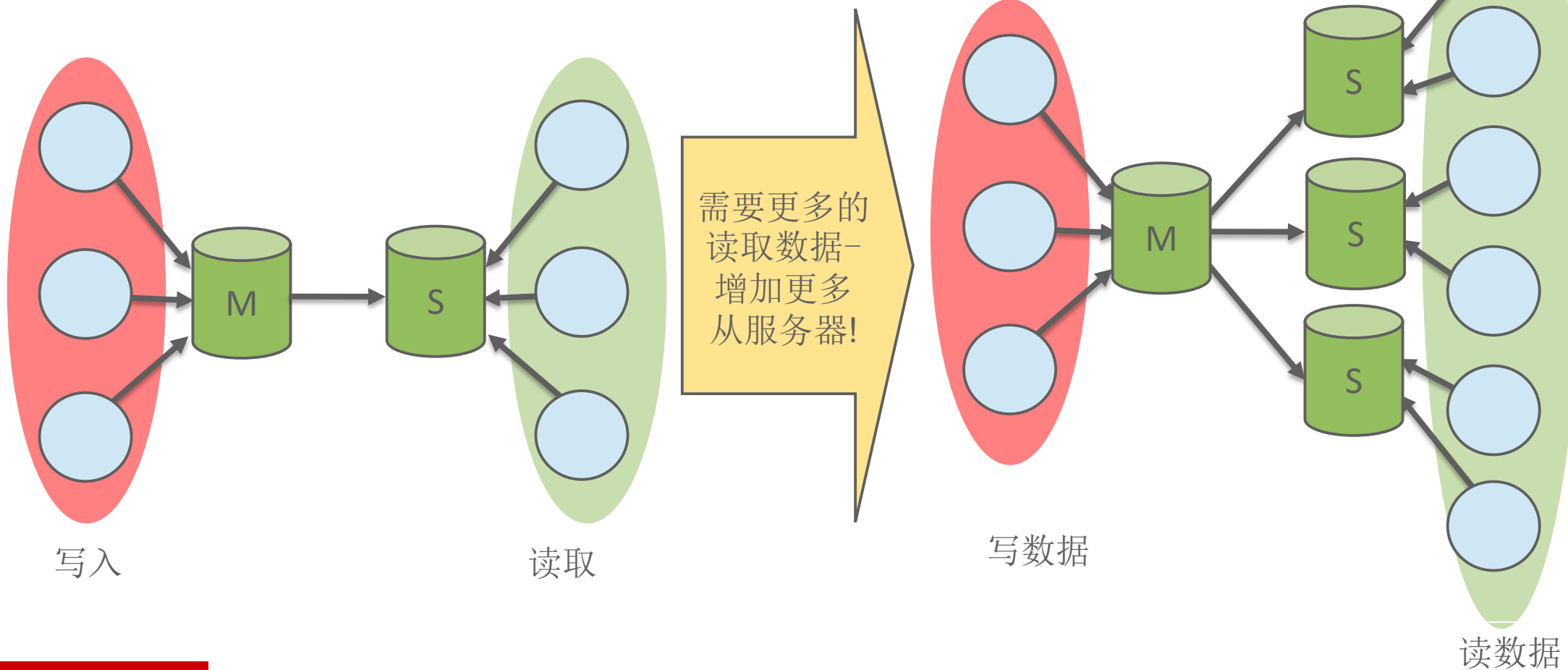
# MySQL复制功能的历史





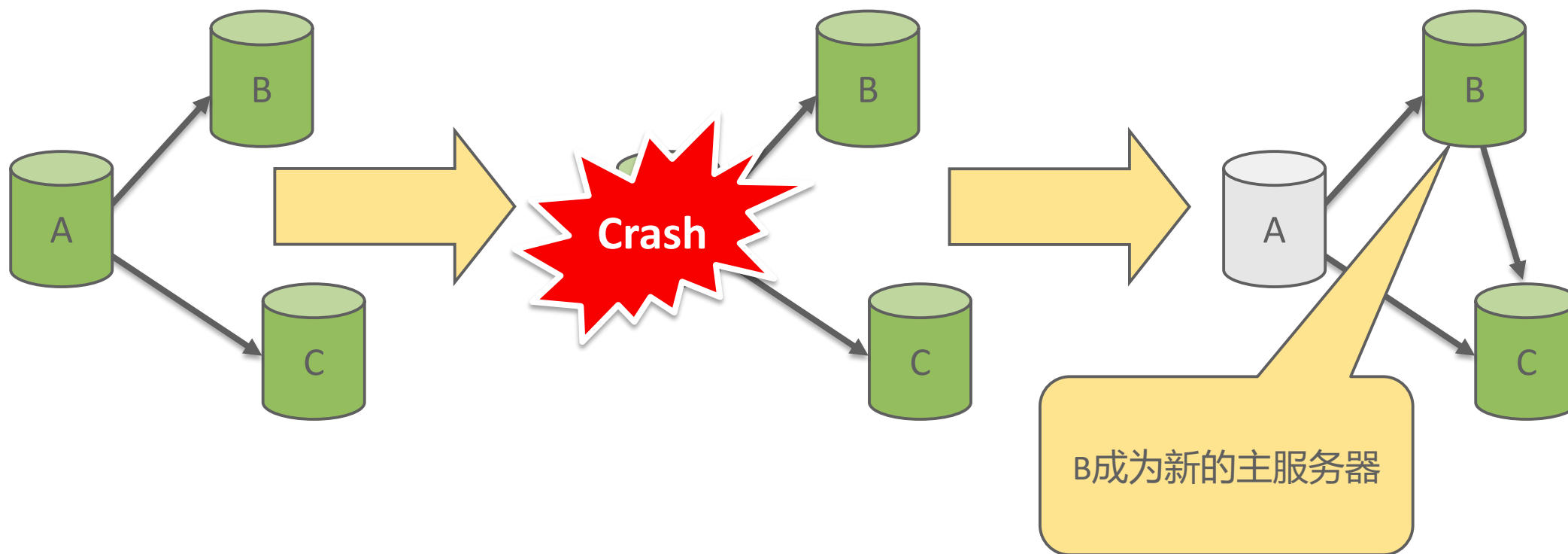
# 什么情况下使用复制功能？

数据读取的增加



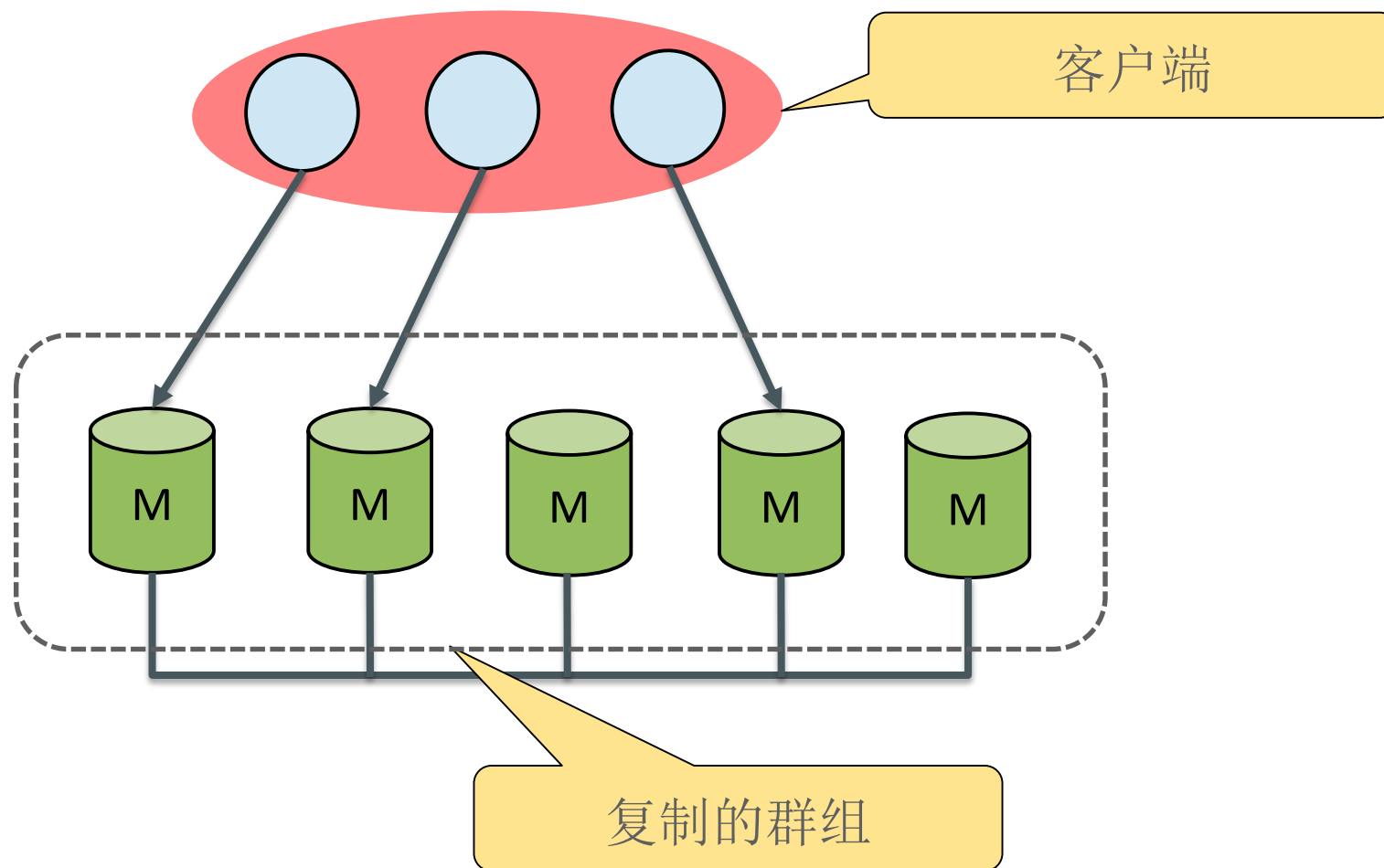
# 什么情况下使用复制功能？

备份: 主机停机时备份代替主服务器





# MySQL 群组复制

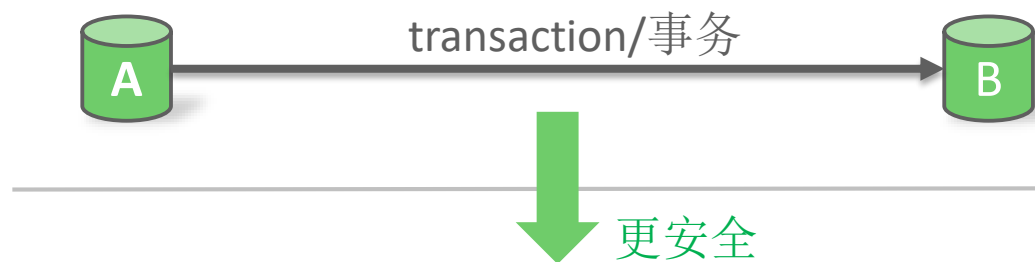


# MySQL高可用性的演变



Since 3.23  
异步(native)

# MySQL高可用性的演变

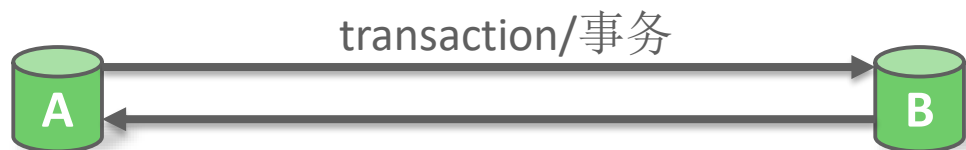


Since 3.23  
异步(native)

# MySQL高可用性的演变

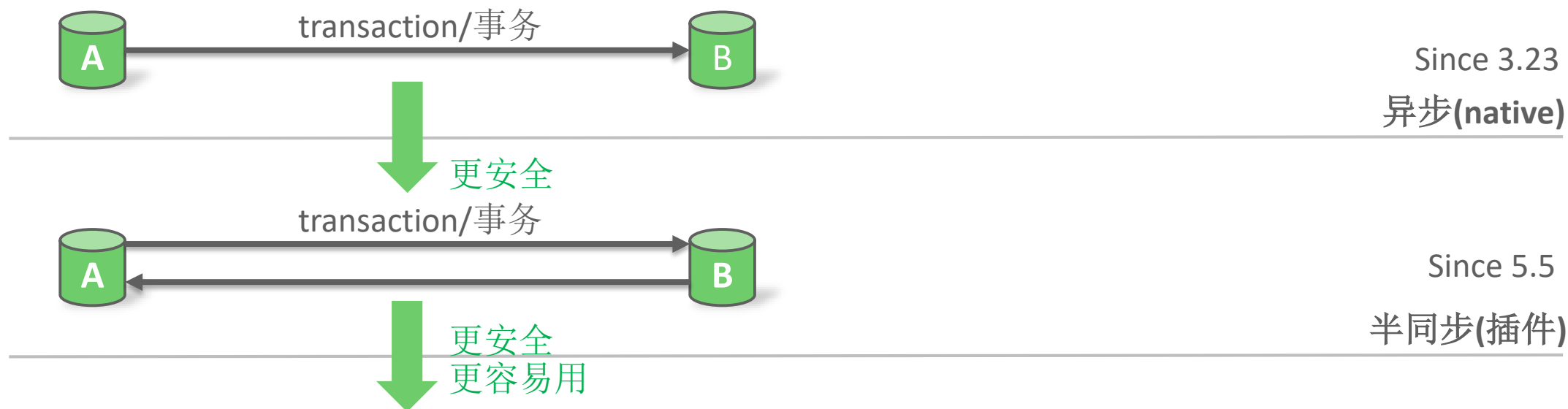


Since 3.23  
异步(native)



Since 5.5  
半同步(插件)

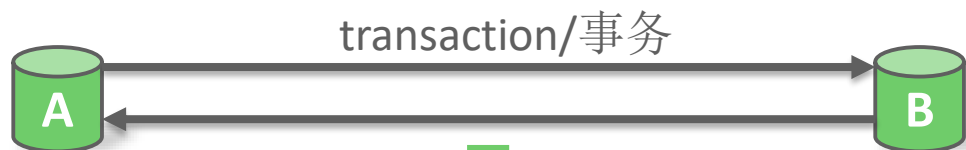
# MySQL高可用性的演变



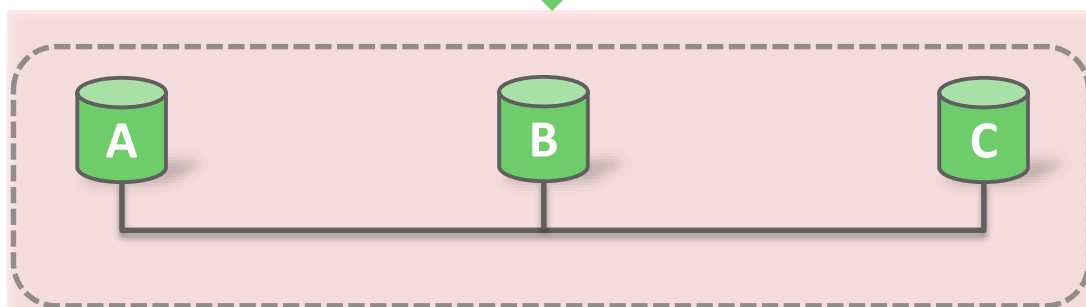
# MySQL高可用性的演变



Since 3.23  
异步(native)



Since 5.5  
半同步(插件)



Since 5.7  
群组复制(插件)



# 更安全、更容易用、可扩展

- 更安全

- 多份数据、数据同步、没有数据丢失

- 更容易使用，可扩展

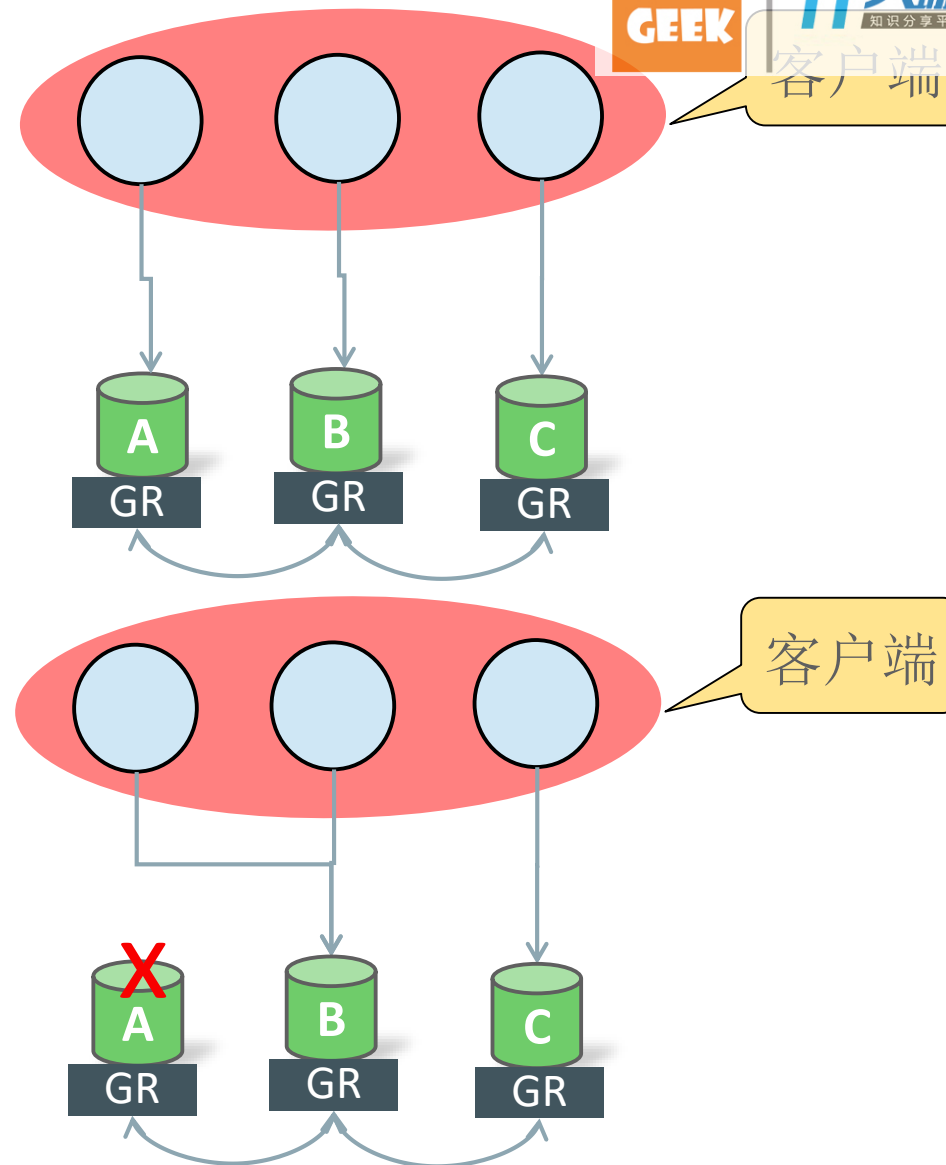
- 可以增加数据库实例

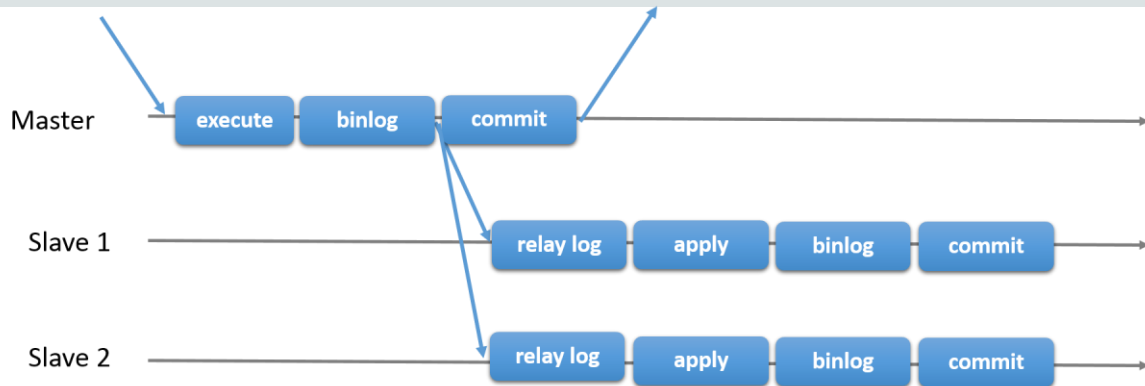
- 提供数据同步

- 对读写分离, 提供数据一致性

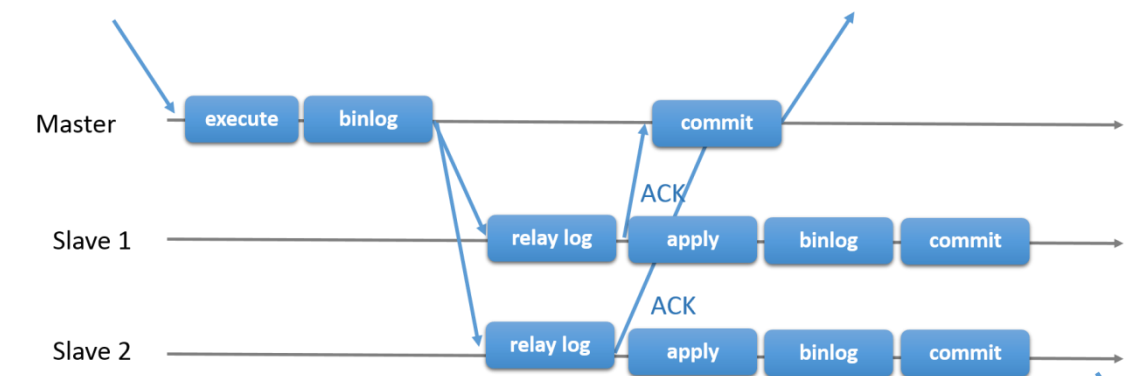
- 高可用性

- 多台实例群组，提供高可用操作



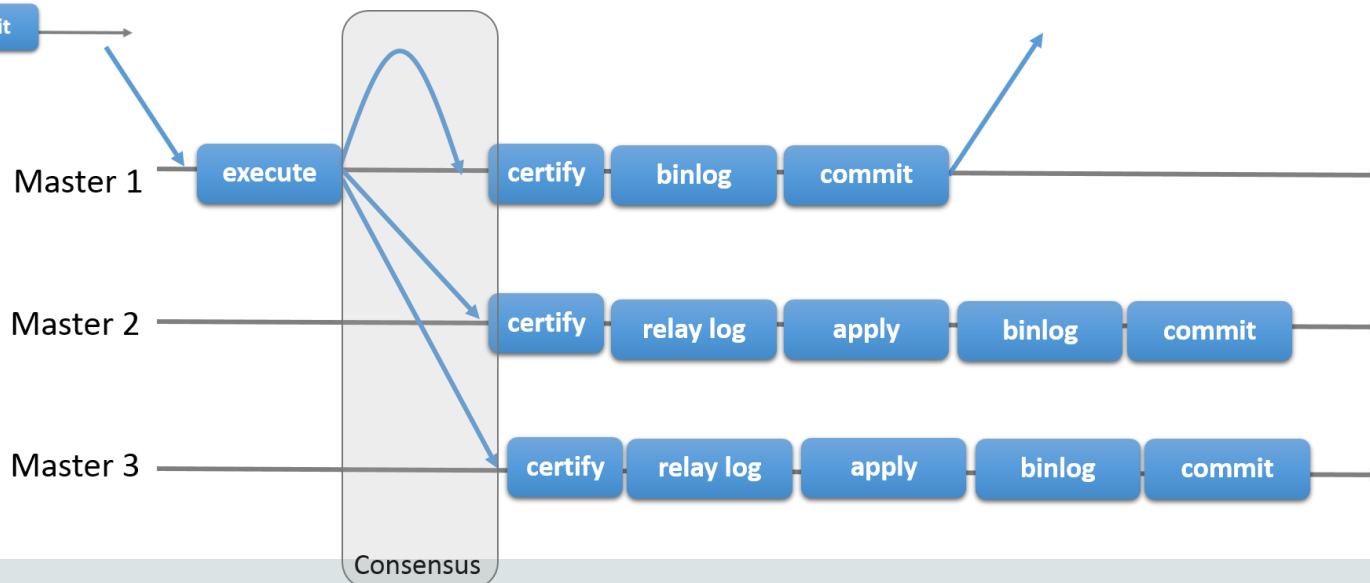


MySQL 异步复制



MySQL 半同步复制

MySQL 群组复制

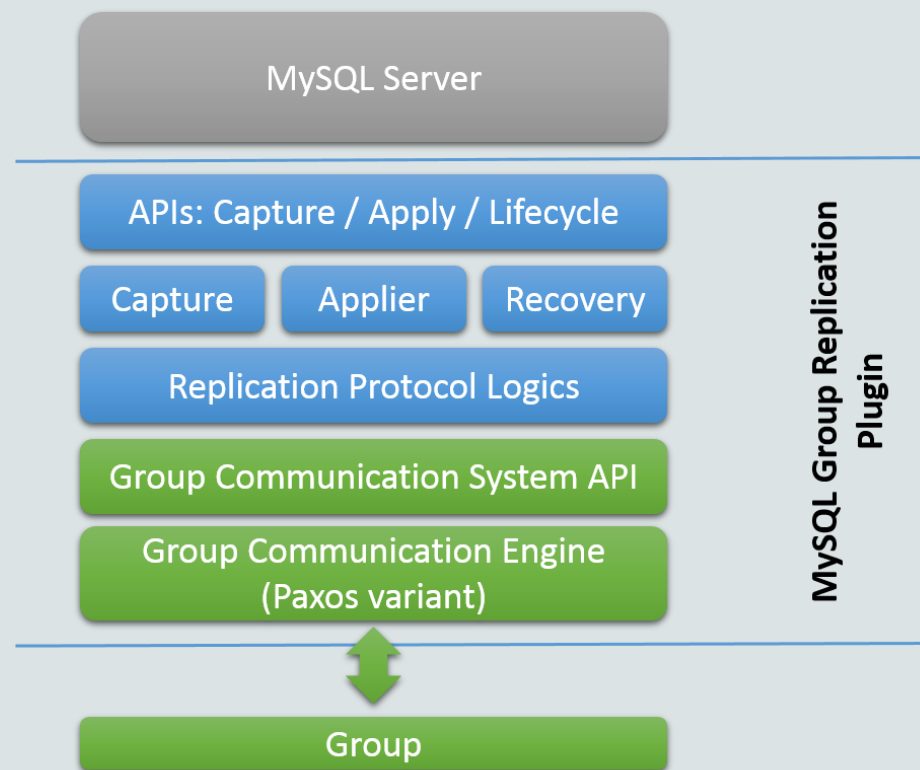


## 2 MySQL 群组复制基础

### 2.1 什么是 Group Replication?

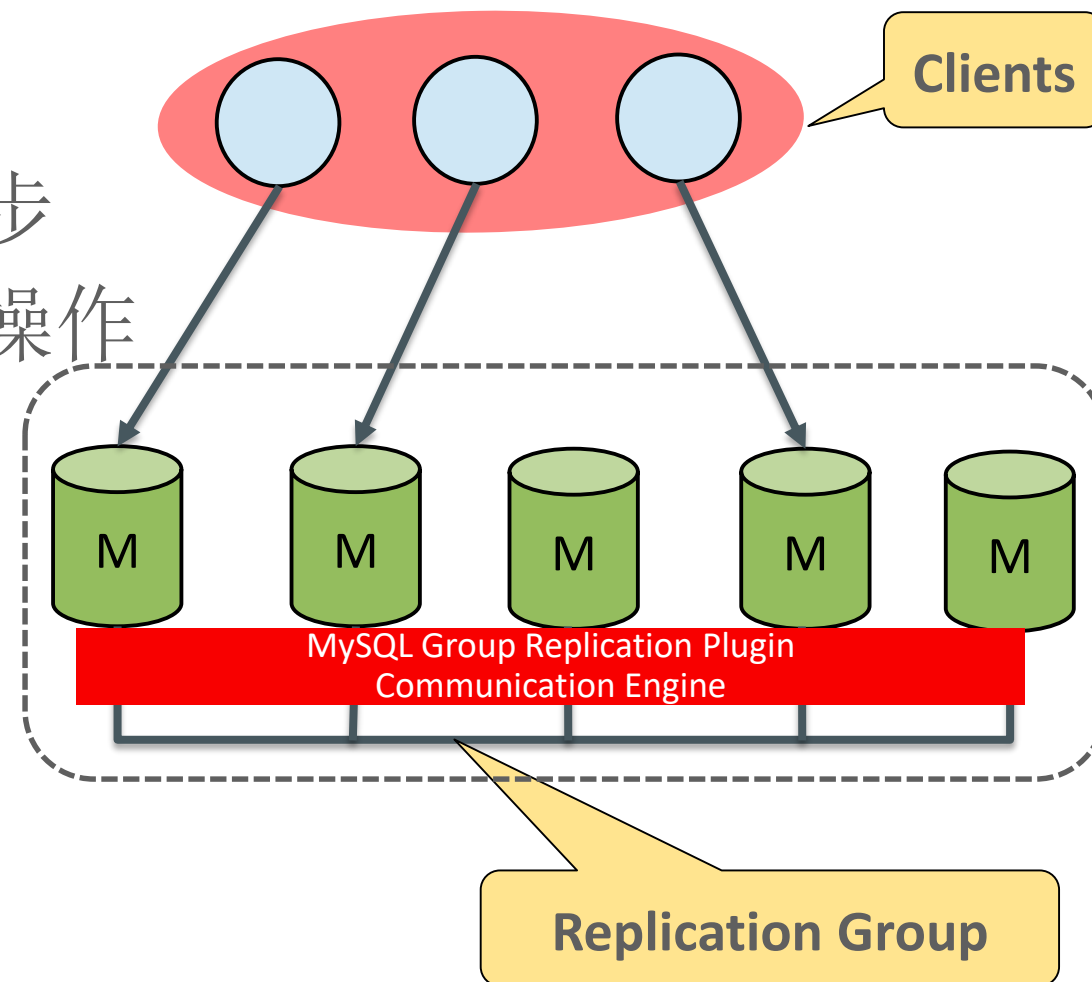
### 2.2 设置群组复制 Group Replication\*

[http://mysqlhighavailability.com/mysqlha/gr/doc/getting\\_started.html](http://mysqlhighavailability.com/mysqlha/gr/doc/getting_started.html)



# 什么是 MySQL Group Replication?

- MySQL 插件
- 非共享、多数据库实例进行同步
- 多个数据库可以并行数据变更操作
- 自动管理群组成员



# 群组复制的高可用性

## 更好的容错度

- 容忍故障（F）所需的服务器数量（N）为  $N = 2F + 1$ .
- 最多支持9个成员
  - 允许4个成员故障。
- 没有脑裂的问题
  - 仅当大多数成员在线时，群组才是可用

Group Size	Majority	Instant Failures Tolerated
1	1	0
2	2	0
3	2	1
4	3	1
5	3	2
6	4	2
7	4	3
8	5	3
9	5	4

3

# MySQL Group Replication 功能

- 3.1 Single Primary Mode (Default) - 单主模式(默认)
- 3.2 Multi-Master Update Everywhere - 多主模式
- 3.3 Parallel Appliers Support - 支持并行执行
- 3.4 Full GTID Replication Support - 支持GTID
- 3.5 Group Replication Monitor - 监视

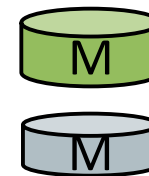


3

# MySQL Group Replication 功能

- 3.1 **Single Primary Mode (Default)** - 单主模式
- 3.2 **Multi-Master Update Everywhere** - 多主模式
- 3.3 **Parallel Appliers Support** - 支持并行执行
- 3.4 **Full GTID Replication Support** - 支持GTID
- 3.5 **Group Replication Monitor** - 监视

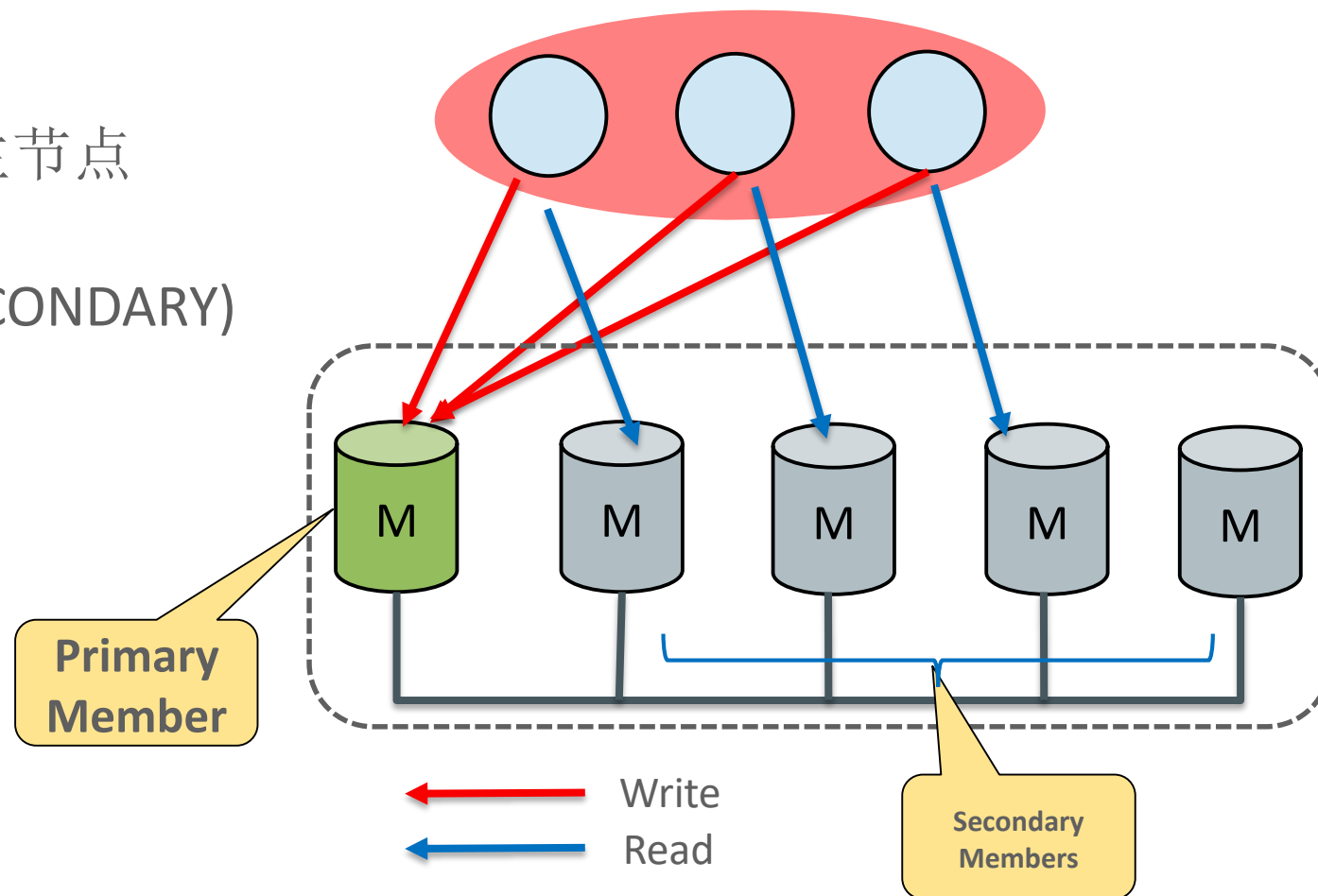
# Single Primary Mode / 单主模式

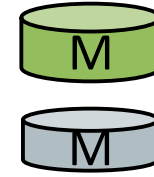


READ-WRITE  
DB GEEK  
SUPER-READ-ONLY

IT大咖说  
知识分享平台

- 单个MySQL实例充当可写的主节点 (PRIMARY)
- 其它的群组成员为热备用(SECONDARY)





READ-WRITE  
DB GEEK  
SUPER-READ-ONLY

IT大咖说  
知识分享平台

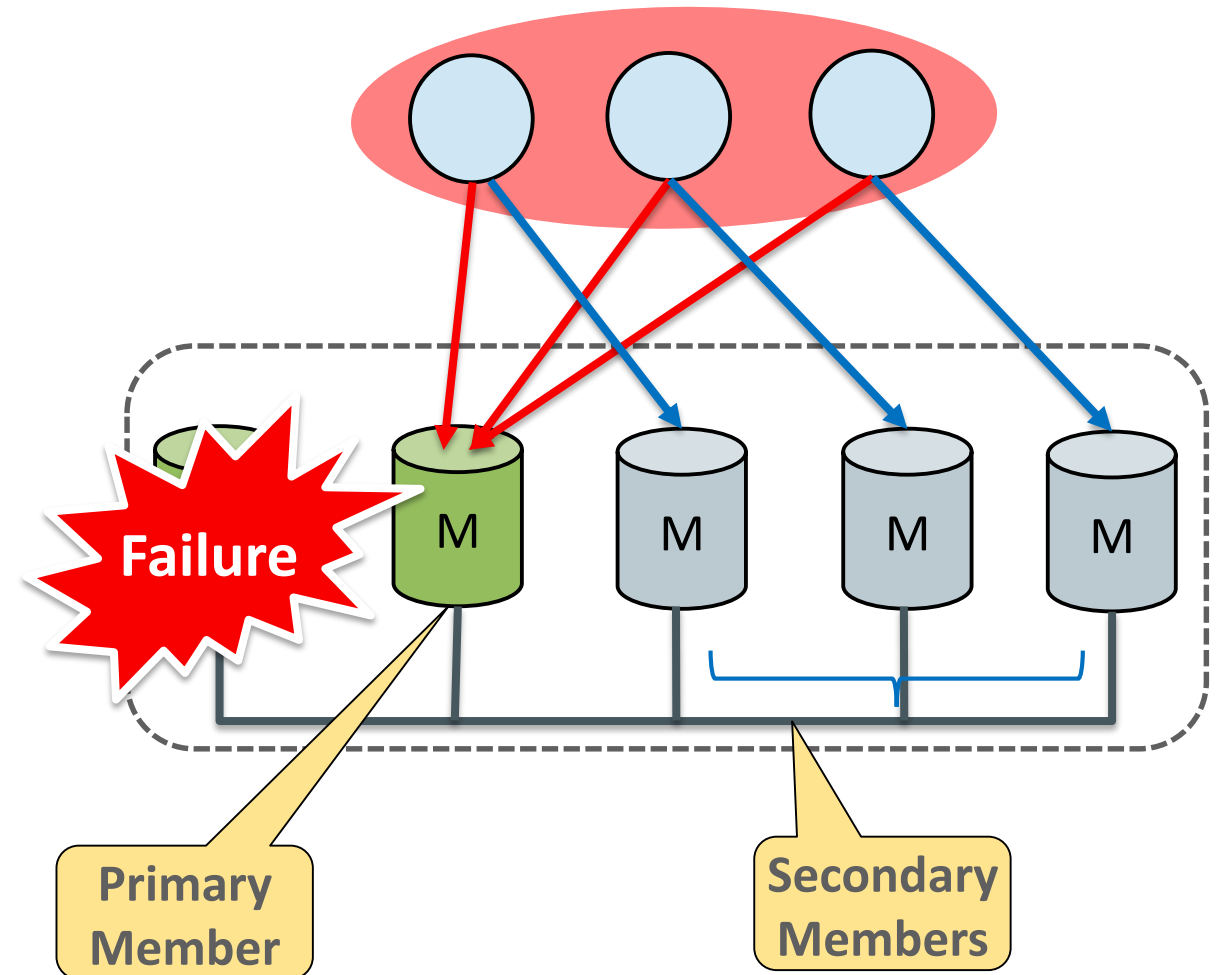
# Single Primary Mode / 单主模式

## 主成员自动启动机制

- 当发生故障或主成员离开时，自动选择主成员

## 默认模式

- Group\_replication\_single\_primary\_mode = ON
- 现有的应用更容易采用此模式。
  - 主备部署（热备）
  - 读写分离
- 避免多主模式的限制。



3

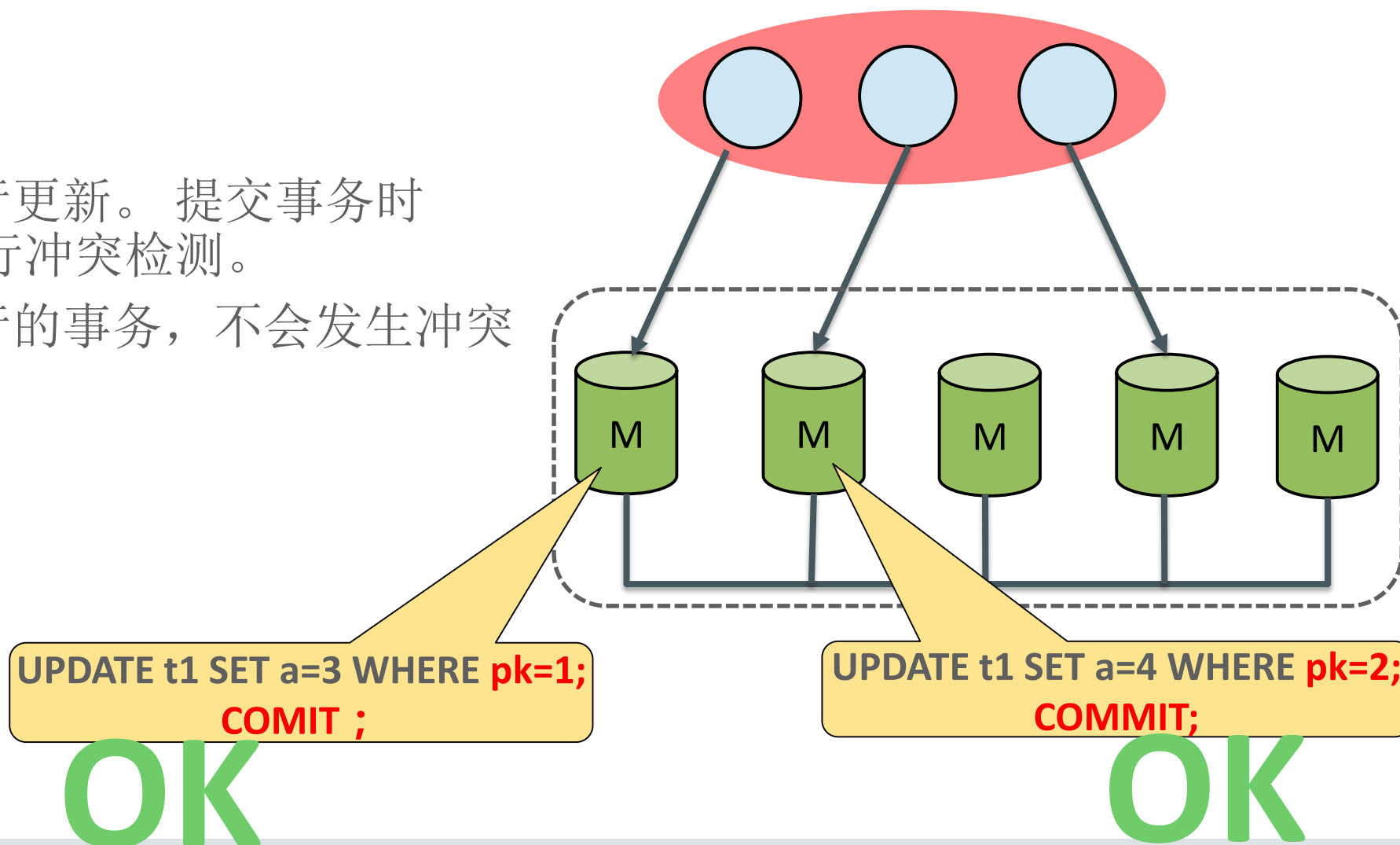
# MySQL Group Replication 功能

- 3.1 Single Primary Mode (Default) - 单主模式
- 3.2 **Multi-Master Update Everywhere** - **多主模式**
- 3.3 Parallel Appliers Support - 支持并行执行
- 3.4 Full GTID Replication Support - 支持GTID
- 3.5 Group Replication Monitor - 监视

# 多主同步模式

## 冲突检测

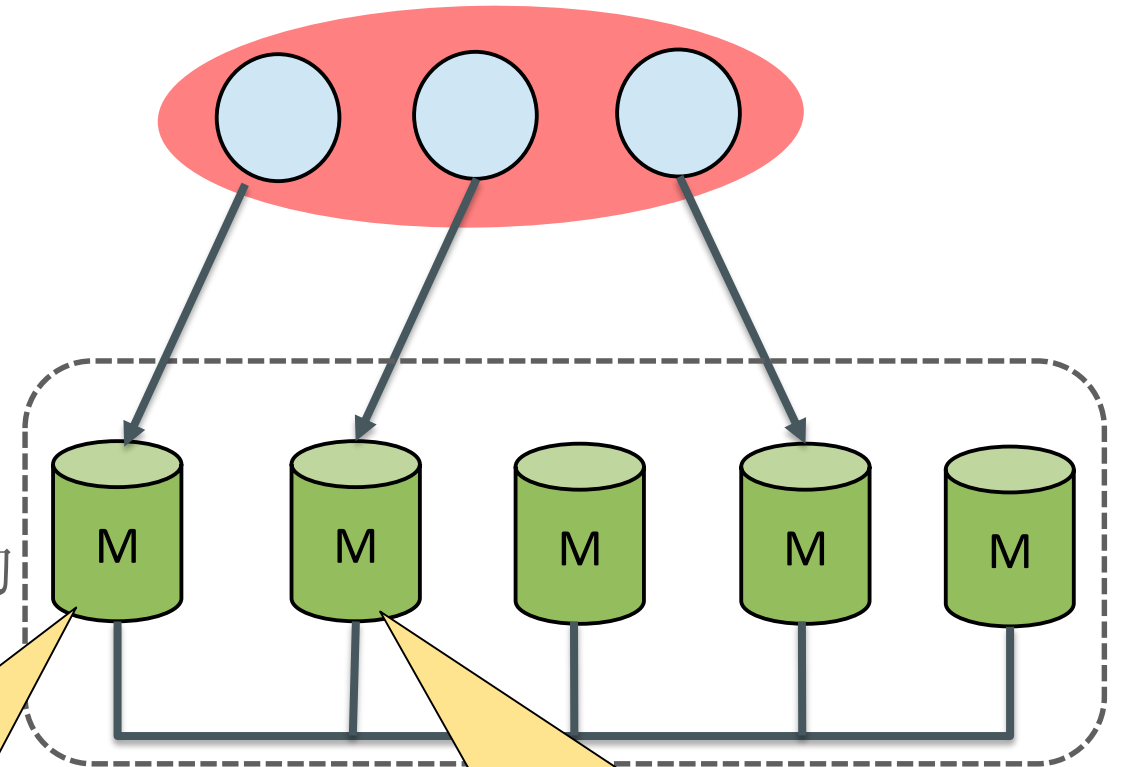
- 使用主键检测
- 多主上并行执行更新。提交事务时 (COMMIT)，进行冲突检测。
- 并行更新不同行的事务，不会发生冲突



# 多主同步模式

## 冲突检测

- 使用主键检测
- 多主上并行执行更新。在提交事务 (COMMIT), 检查冲突。
- 并行更新不同行的事务, 不会发生冲突
- 在不同的服务器上, 并行更新相同行数数据, 只有一个事务可以 **COMMIT**, 其他的都必须回滚



UPDATE t1 SET a=3 WHERE pk=1  
COMMIT (Later)



UPDATE t1 SET a=4 WHERE pk=1  
COMMIT (First)

OK



# 多主同步模式

**MySQL Group Replication** <http://mysqlhighavailability.com/mysqlha/gr/doc/limitations.html>

- 限制和规则
  - 仅InnoDB Engine(Transactional and row level lock)
  - 表必须有主键
  - gtid-mode=ON
  - binlog格式 要Row-based
  - 对于同一个对象执行DDL和DML应在同一成员上执行;不支持在不同服务器上执行DDL
  - 不支持具有多级外键依赖关系的表，特别是已定义CASCADING外键约束的表
  - 不支持“serializable”隔离级别

3

# MySQL Group Replication 功能

- 3.1 Single Primary Mode (Default) - 单主模式
- 3.2 Multi-Master Update Everywhere - 多主模式
- 3.3 **Parallel Appliers Support** - **支持并行执行**
- 3.4 Full GTID Replication Support - 支持GTID
- 3.5 Group Replication Monitor - 监视

# Parallel Appliers Support

- 群组复制的基础架构充分利用二进制日志
  - 有自带的“channel”：group\_replication\_applier
  - 减少延迟，提高复制性能
  - 与异步复制采用同样的配置方法

```
--slave_parallel_workers = NUMBER  
--slave_parallel_type = logical_clock  
--slave_preserve_commit_order = ON
```

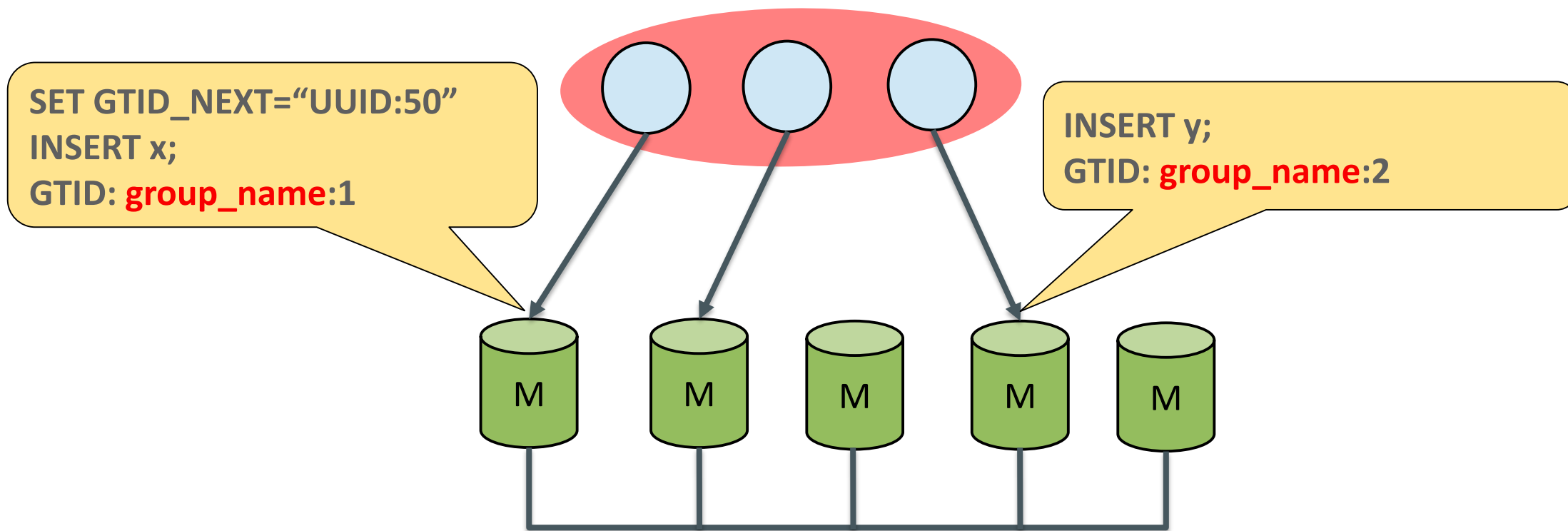
3

# MySQL Group Replication 功能

- 3.1 Single Primary Mode (Default) - 单主模式
- 3.2 Multi-Master Update Everywhere - 多主模式
- 3.3 Parallel Appliers Support - 支持并行applier
- 3.4 **Full GTID Replication Support** - **支持GTID**
- 3.5 Group Replication Monitor - 监视

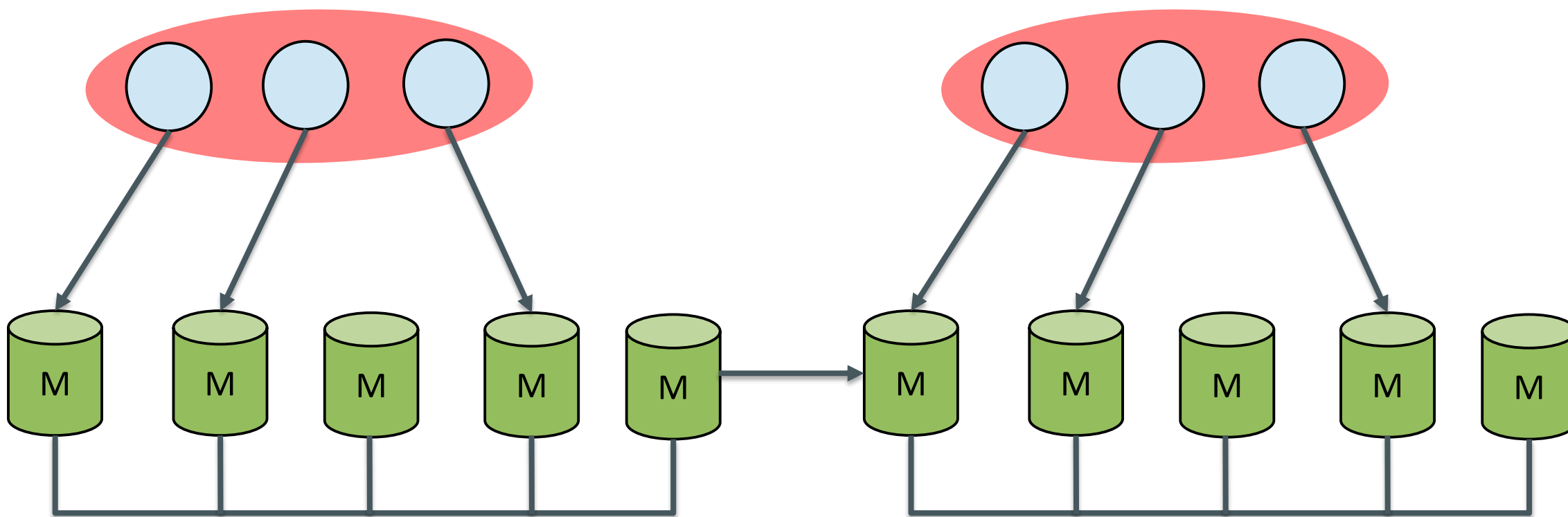
# 完全支持GTID

- 所有群组成员具有相同UUID, 共享一个group\_name, 可指定GTID。



# 完全支持GTID

- 支持跨群组复制，保留GTID



3

# MySQL Group Replication 功能

- 3.1 Single Primary Mode (Default) - 单主模式
- 3.2 Multi-Master Update Everywhere - 多主模式
- 3.3 Parallel Appliers Support - 支持并行执行
- 3.4 Full GTID Replication Support - 支持GTID
- 3.5 **Group Replication Monitor** - **监视**

# Group Replication Monitor-监视

- 两个新的performance\_schema表

- replication\_group\_members

```
mysql@virtual-41:~/demo
[mysql@virtual-41 demo]$ mysql -uroot -h127.0.0.1 -e "SELECT * FROM performance_schema.replication_group_members;
> "
+-----+-----+-----+-----+-----+
| CHANNEL_NAME | MEMBER_ID | MEMBER_HOST | MEMBER_PORT | MEMBER_STATE |
+-----+-----+-----+-----+-----+
| group_replication_applier | 79d8f230-c5a1-11e6-a0c0-080027867cd7 | virtual-41.localhost | 3306 | ONLINE |
| group_replication_applier | 7cc78540-c5a1-11e6-a1f4-080027867cd7 | virtual-41.localhost | 3316 | ONLINE |
+-----+-----+-----+-----+-----+
[mysql@virtual-41 demo]$
```

- replication\_group\_member\_stats

本地成员状态

```
mysql@virtual-41:~/demo
mysql@virtual-41 demo]$ mysql -t -uroot -h127.0.0.1 -e "SELECT * FROM performance_schema.replication_group_member_stats\G"
***** 1. row *****
CHANNEL_NAME: group_replication_applier
VIEW_ID: 14821385998969119:2
MEMBER_ID: 79d8f230-c5a1-11e6-a0c0-080027867cd7
COUNT_TRANSACTIONS_IN_QUEUE: 0
COUNT_TRANSACTIONS_CHECKED: 4
COUNT_CONFLICTS_DETECTED: 0
COUNT_TRANSACTIONS_ROWS_VALIDATING: 0
TRANSACTIONS_COMMITTED_ALL_MEMBERS: 8a94f357-aab4-11df-86ab-c80aa9429562:1-16
LAST_CONFLICT_FREE_TRANSACTION: 8a94f357-aab4-11df-86ab-c80aa9429562:16
mysql@virtual-41 demo]$
```

- 扩展Replication performance\_schema 表

- group\_replication\_recovery channel information

- group\_replication\_applier channel information

- 新的全局变量

- group\_replication\_primary\_member

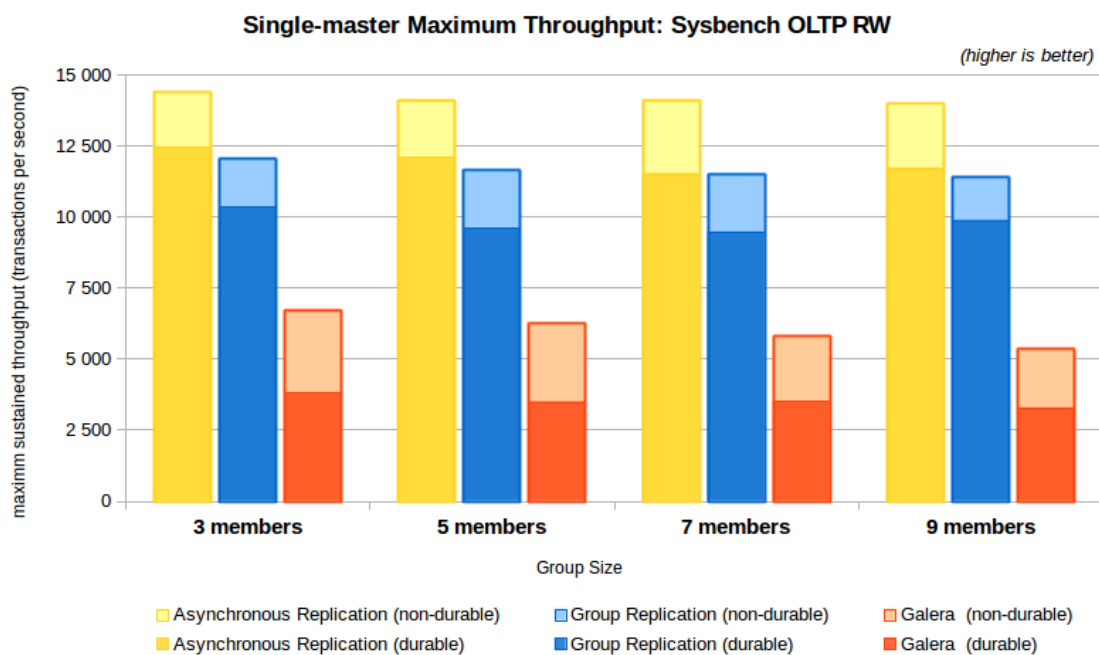
```
mysql@virtual-41:~/demo
mysql@virtual-41 demo]$ mysql -t -uroot -h127.0.0.1 -e "show global status like 'group_replication_primary_member';"
+-----+-----+
| Variable_name | Value |
+-----+-----+
| group_replication_primary_member | 79d8f230-c5a1-11e6-a0c0-080027867cd7 |
+-----+-----+
mysql@virtual-41 demo]$
```



# 4 Performance / 性能

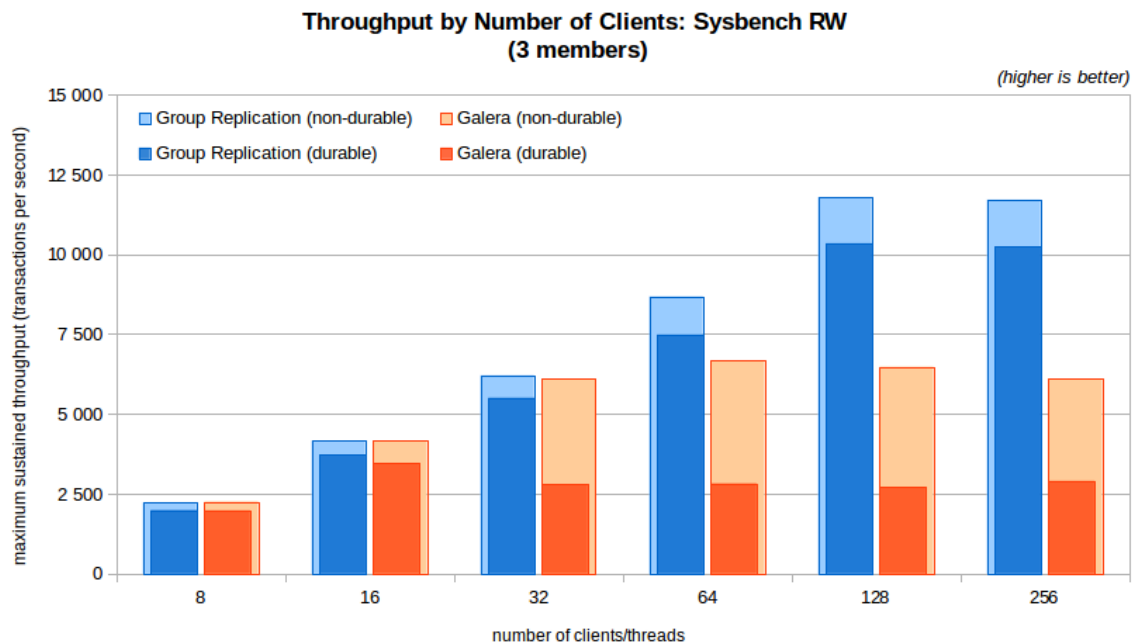
<http://mysqlhighavailability.com/performance-evaluation-mysql-5-7-group-replication/>

# 吞吐量/组成员个数



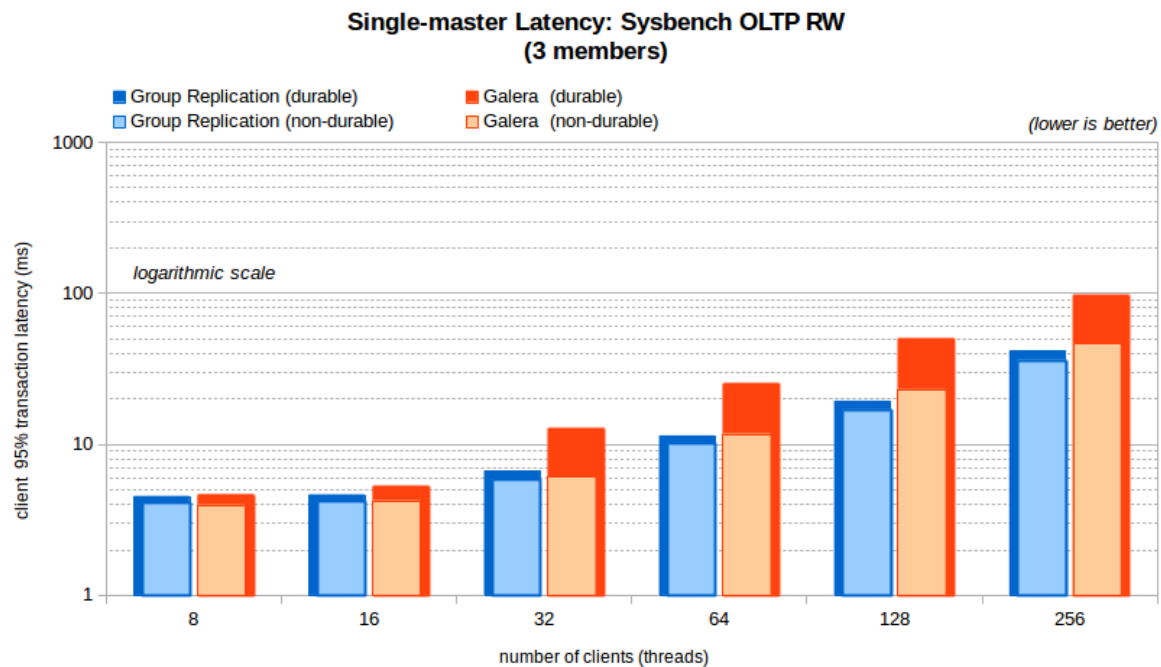
- 群组复制的吞吐量可达异步复制的80%，而Galera分别为46%（3成员），38%（9成员）
- 持久性配置对Galera的影响比较大，仅达到30%。
- 群组复制成员增加到9个，吞吐量表现很稳定，可以扩张至更大的群组。
- 群组复制的吞吐量可以达到Galera的2倍（持久性配置时的3倍）
- `innodb_flush_log_at_trx_commit=1`  
`sync_binlog=1`

# 吞吐量/并发连接数



- 并发连接数最低时，相差不大
- Galera在并发连接在16和32时，相对群组复制明显减少
- 群组复制的并发连接数可以达到128，并发连接数达到256时，也表现得很稳定
- Galera在并发连接数16时，表现最好的吞吐量，之后逐渐变低。

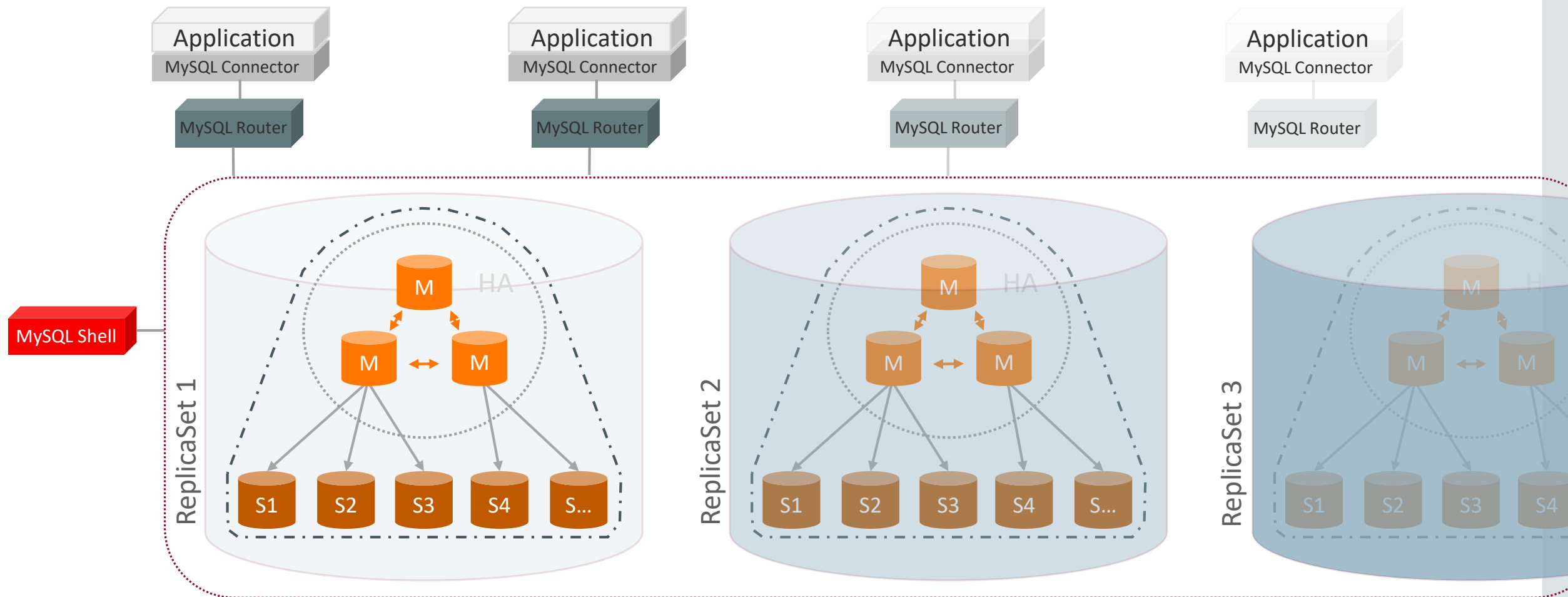
# 延迟/并发连接数



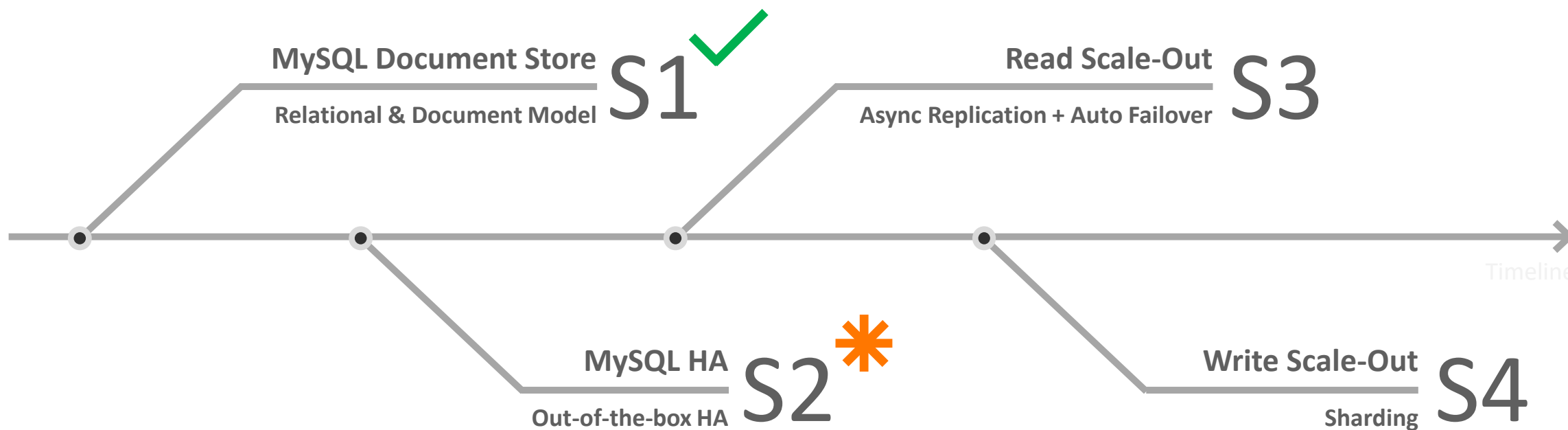
- 群组复制的延迟性随着组成员的增加比Galera要低

# 5 MySQL InnoDB Cluster路线发展

# MySQL InnoDB Cluster: 最终目标



# 路线发展



# MySQL Group Replication 总结

- **更安全，易用的高可用性MySQL数据库平台**
  - 通过稳定和成熟的核心API/插件来制定MySQL Group Replication
  - 通过 GTIDs, row based replication, performance schema tables 来搭建
  - 成员数据修复：服务器故障转移处理。
  - 提供容错，实现多主机更新和可靠的MySQL服务
  - 提供单主模式，多主模式 – 方便应用采用！
- **Cloud Friendly**
  - 用于需要弹性技术的部署，如基于云平台的基础
  - 不使用tcp广播，可用于云平台
- **Group Replication 文档**
  - <http://mysqlhighavailability.com/gr/doc/>



ORACLE®