



MySQL高可用演进及实践

数据技术嘉年华 · ON LINE, 2021.11.26

- 叶金荣
- 万里数据库开源生态负责人
- Oracle MySQL ACE Director
- 腾讯云TVP



高可用面临的挑战

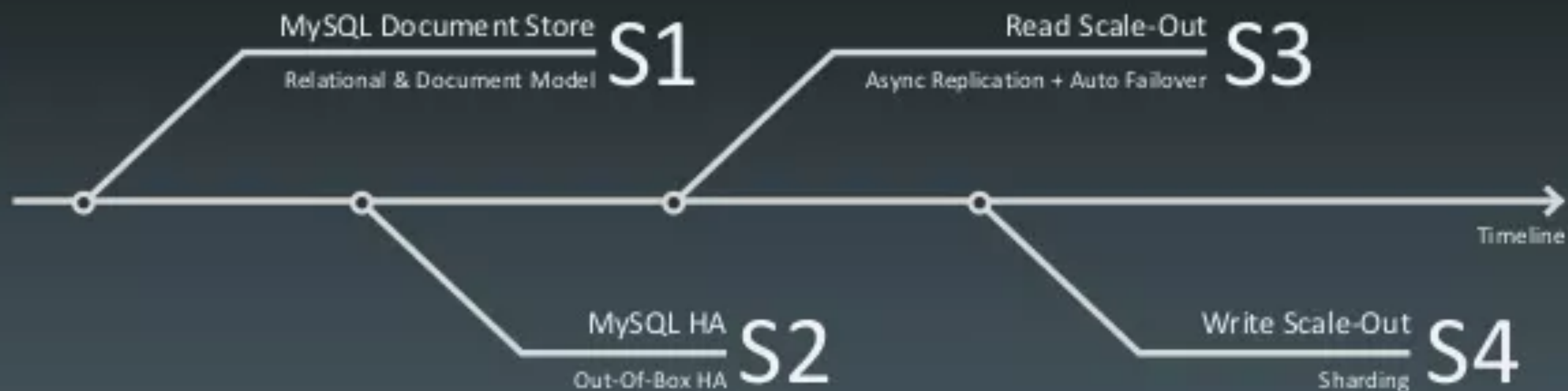


- RPO, 数据一致性
- RTO, 服务连续性

可用性	建议
99%	3天15小时36分
99.9%	8小时46分
99.99%	52分34秒
99.999%	5分15秒
99.9999%	32秒

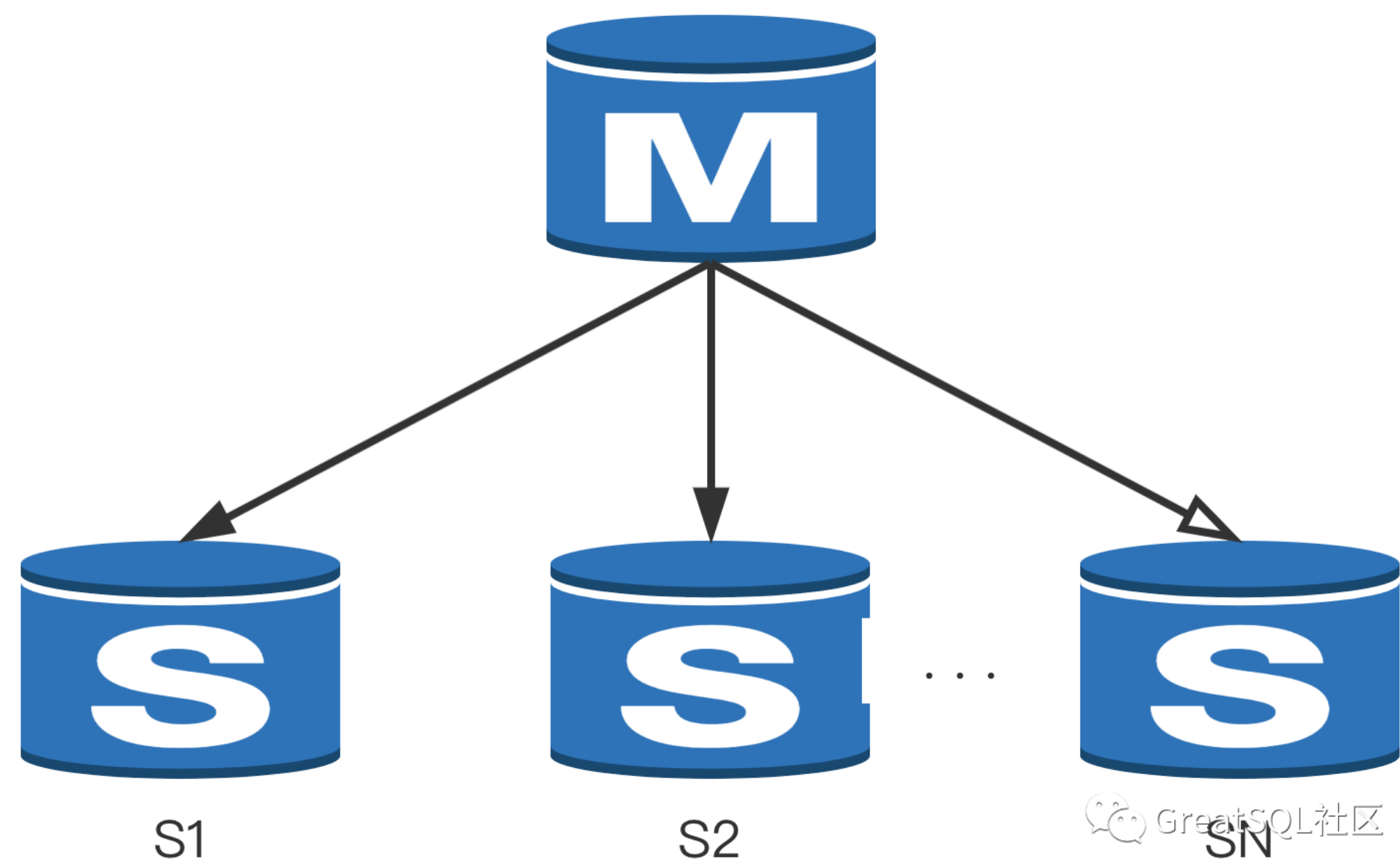
- 远古，Replication
- 古代，MHA+Replication
- 近代，Semisynchronous Replication
- 现代，InnoDB Cluster

MySQL Vision – 4 Steps

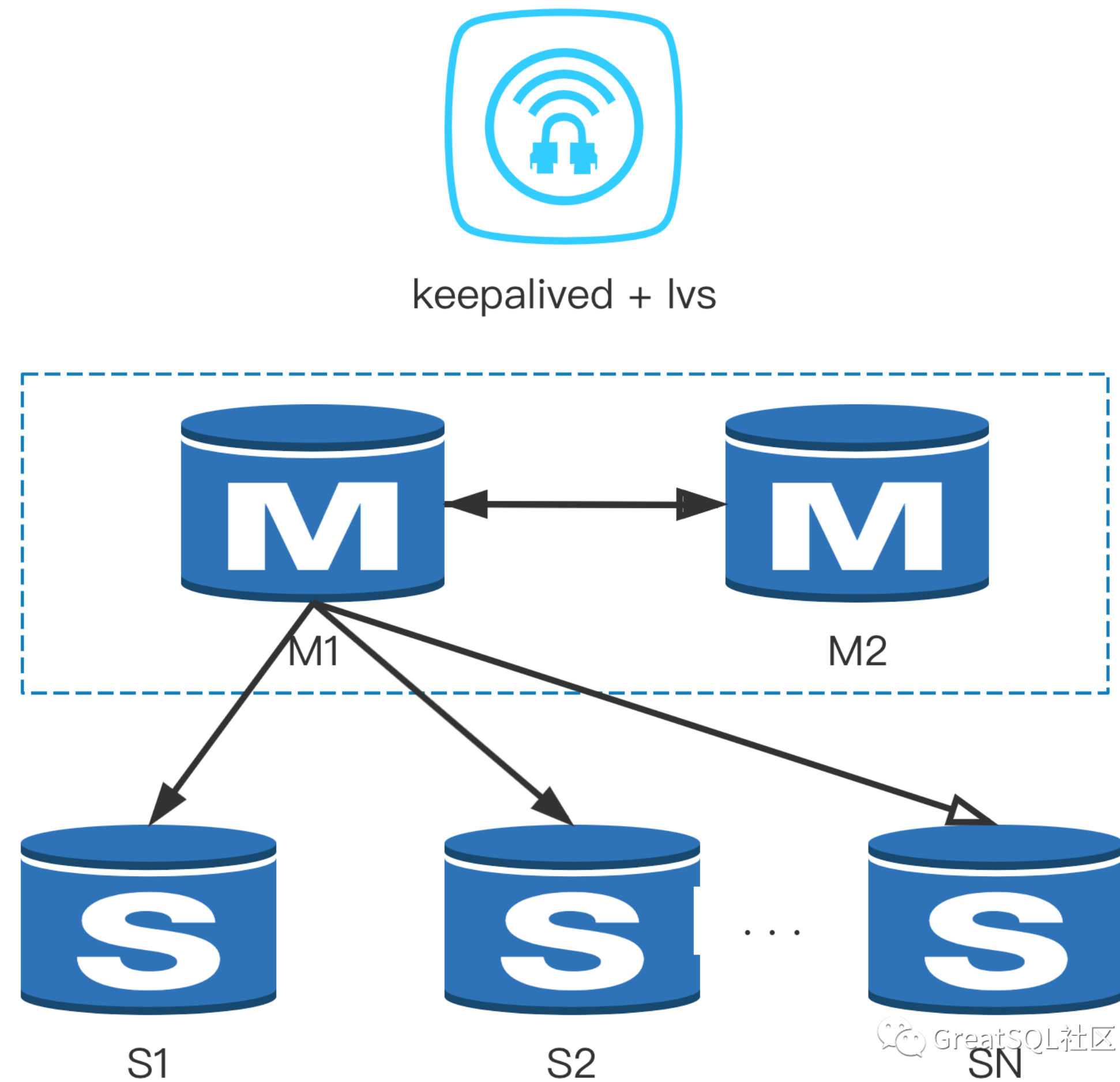


- MySQL 5.5之前，基于异步复制
- 互联网发展早期的高可用+读写分离方案
- 基本上可以保障99%可用性
- 常见架构方案
 - 一主一从、一主多从、级联
 - 单向、双向复制（很容易冲突）

- 存在的问题
 - 异步
 - 主从数据一致性无法保证, 尤其是早期只支持SBR
 - 没有可靠的高可用切换机制, 需要自行实现
 - 从节点单线程, 大事务、无主键等容易造成复制延迟

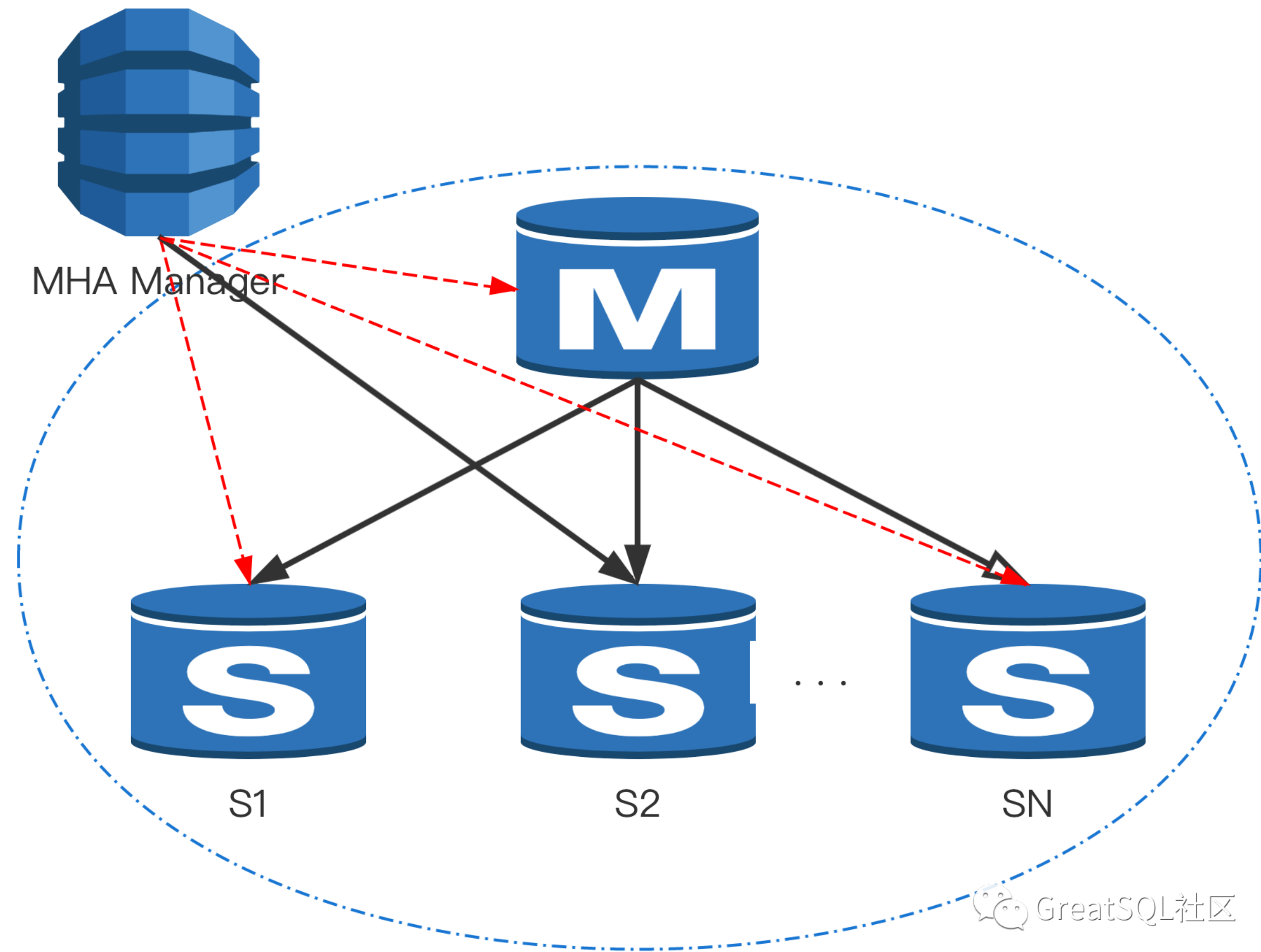


一主多从



双主多从

- MHA配合异步复制
- 利用MHA实现高可用管理
- 切换前尝试补数据
- 尽可能实现RPO=0, 降低RTO
- 基本上可以保障99.9%可用性



- 存在的问题
 - 对5.7后的并行复制支持不好, 基本上废弃
 - 需要自行开发切换脚本
 - 需要构建ssh信任, 存在安全风险
 - 项目已处于不更新状态

- 5.6引入GTID、binlog server、半同步复制等特性
 - 引入新选项slave-rows-search-algorithms, 对无主键的表支持更友好
- 5.7后引入增强半同步
 - 半同步线程独立, 减少瓶颈点
 - 主库group commit, 从库writeset, 提高效率
 - 可设置最少多少个从库ack, 提高可靠性
- 配合ProxySQL、Orchestrator等工具
- 基本上可以保障99.9%可用性

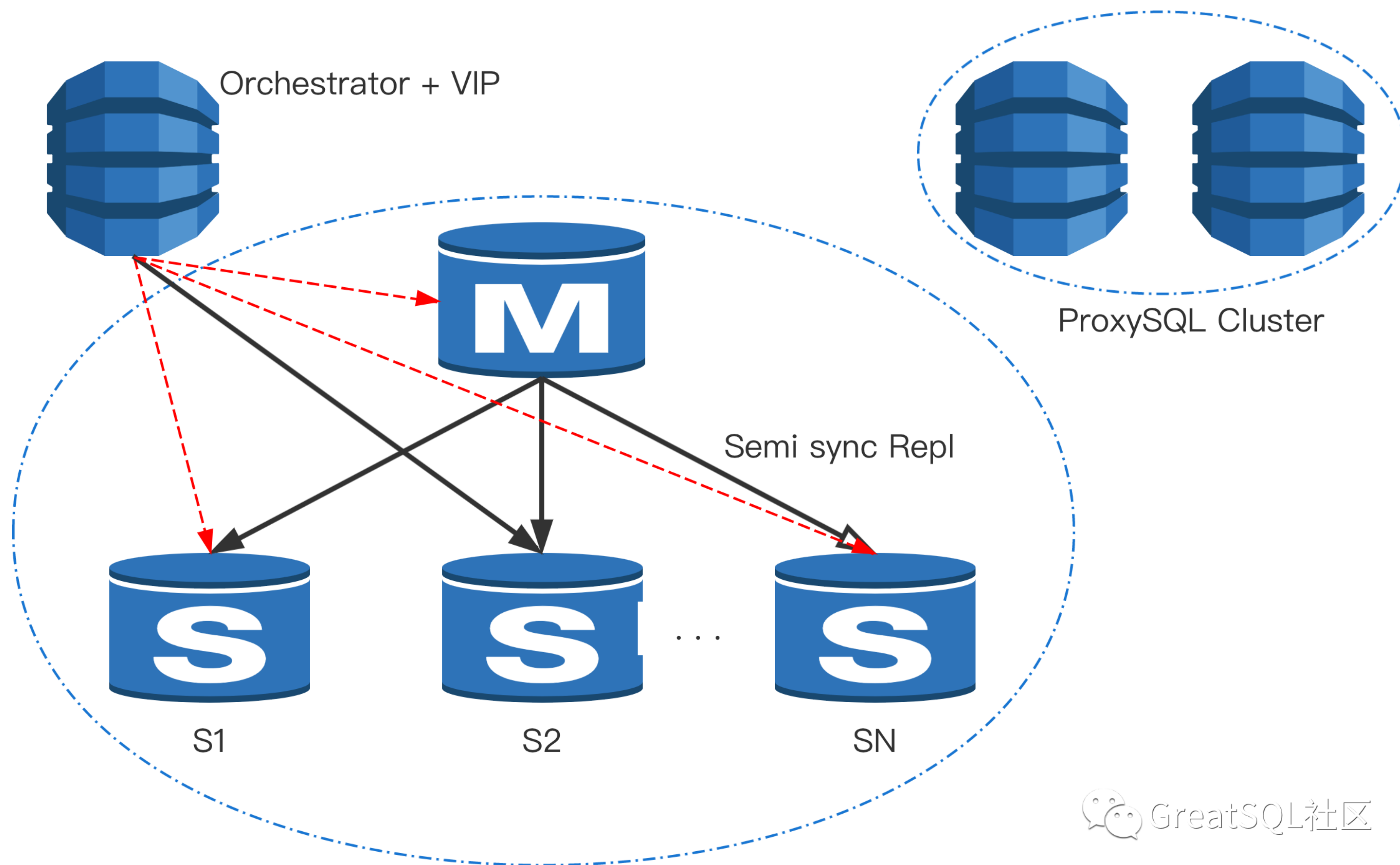
- Master

```
mysql> INSTALL PLUGIN rpl_semi_sync_master SONAME 'semisync_master.so';  
mysql> SET GLOBAL rpl_semi_sync_master_enabled = 1;  
mysql> SET GLOBAL rpl_semi_sync_master_timeout = 9999999;  
mysql> semi_sync_master_wait_for_slave_count = N  
mysql> semi_sync_master_wait_point = AFTER_SYNC
```

- Slave

```
mysql> INSTALL PLUGIN rpl_semi_sync_slave SONAME 'semisync_slave.so';  
mysql> SET GLOBAL rpl_semi_sync_slave_enabled = 1;
```

近代, Semisynchronous Replication



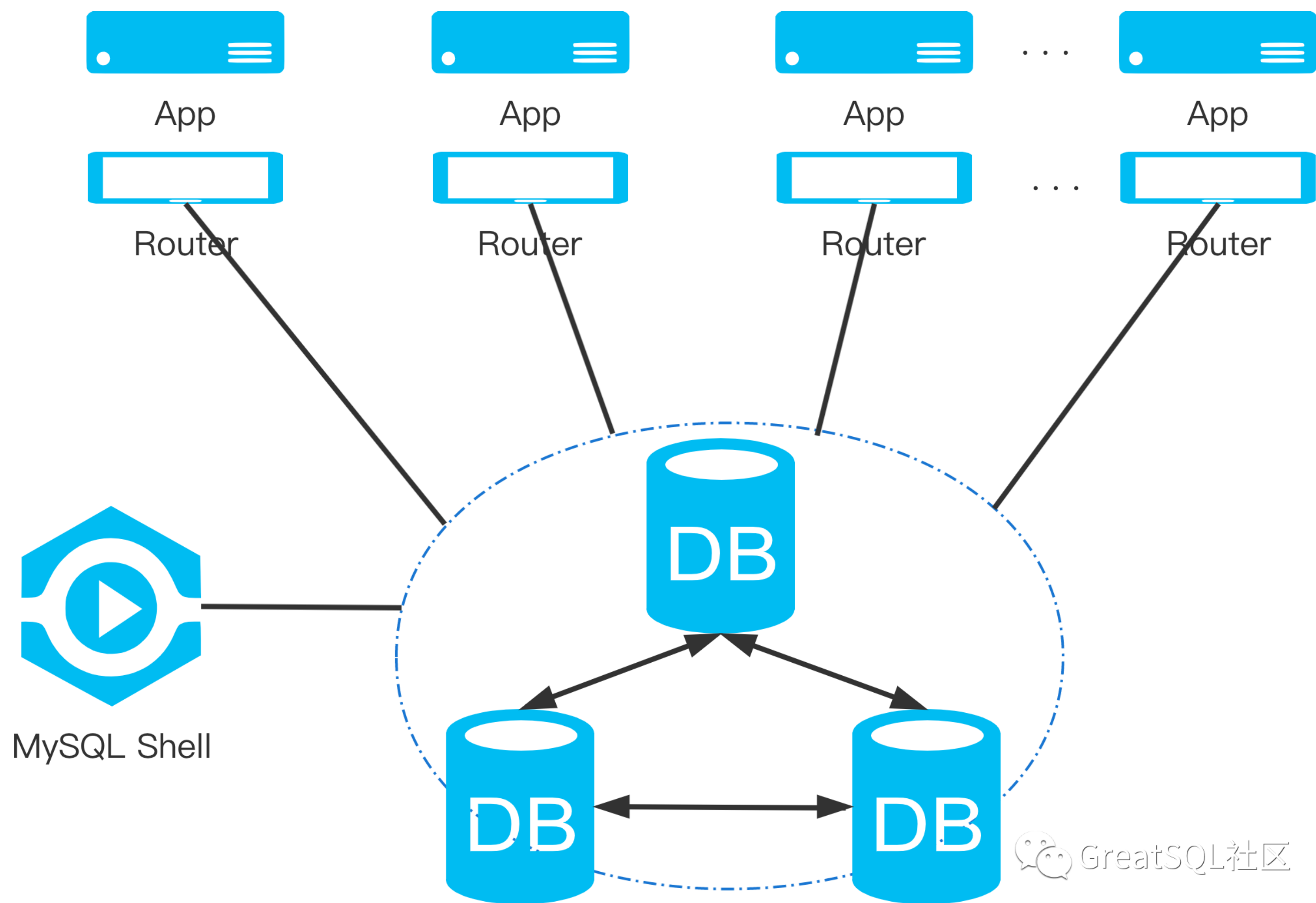
- 问题、风险
 - 5.6版本的半同步复制存在幻读问题
 - 5.6版本的ack确认在dump thread中, 很容易造成瓶颈
 - 半同步退化选择
 - 特殊情况下主从数据不一致

现代，MySQL原生高可用，InnoDB Cluster



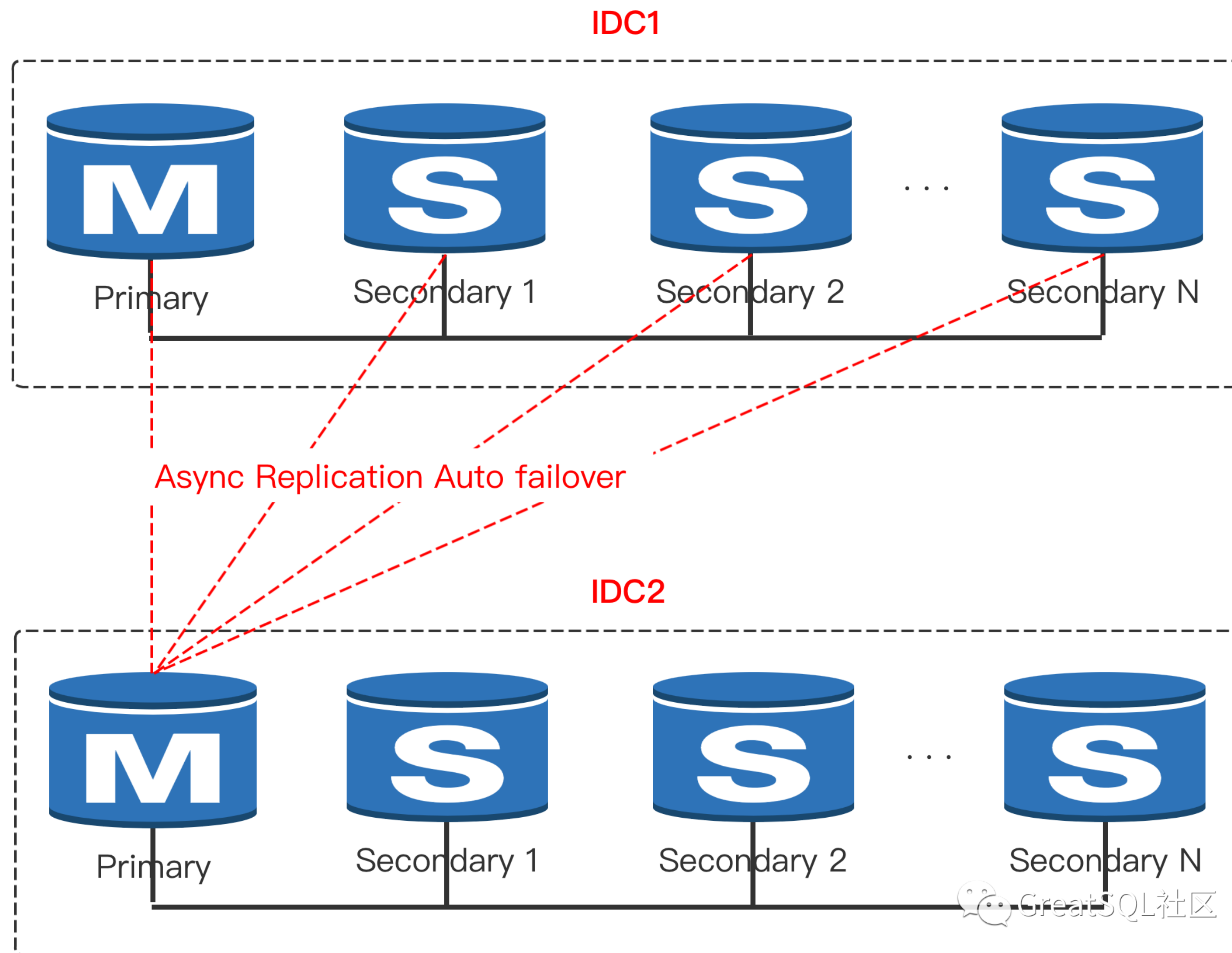
- MGR是MySQL官方推出的新一代原生高可用方案
 - 数据一致性有保障
 - 故障自动转移，无需人工干预
- MySQL InnoDB Cluster，简称MIC
- 包含MySQL Server、Shell、Router等组件
- MySQL Shell包含丰富的接口、指令集
- MySQL Router自带读写分离及MGR状态感知
- 基本上可以保障99.99%可用性

现代，MySQL原生高可用，InnoDB Cluster



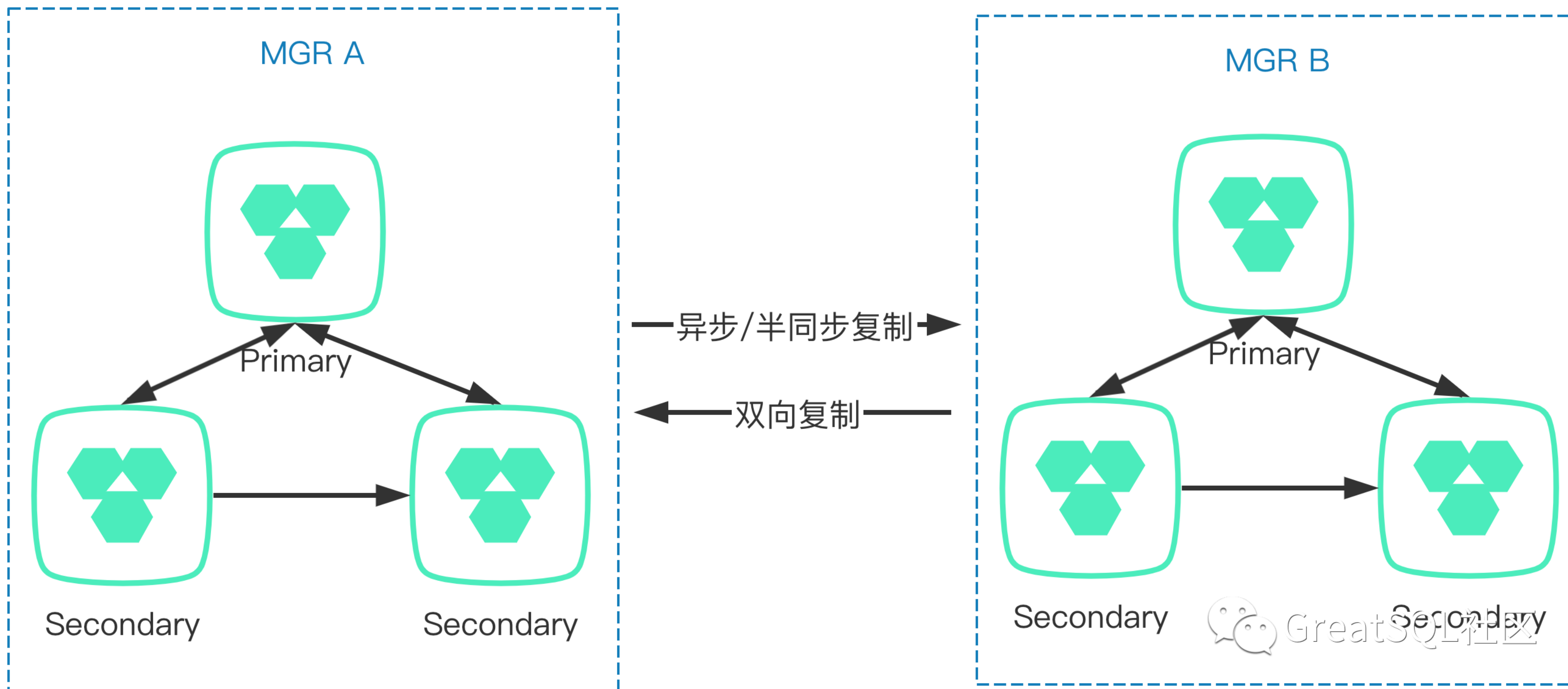
- MySQL 8.0.22后推出Async Replication Auto failover特性
- 异步复制、半同步复制、MGR均支持
- 主节点发生切换时，从节点无需做额外工作
- 更进一步降低了RTO
- 部署时，增加几个选项
 - SOURCE_CONNECTION_AUTO_FAILOVER = 1
 - MASTER_RETRY_COUNT(3)
 - MASTER_CONNECT_RETRY(10)

现代，MySQL原生高可用，InnoDB Cluster



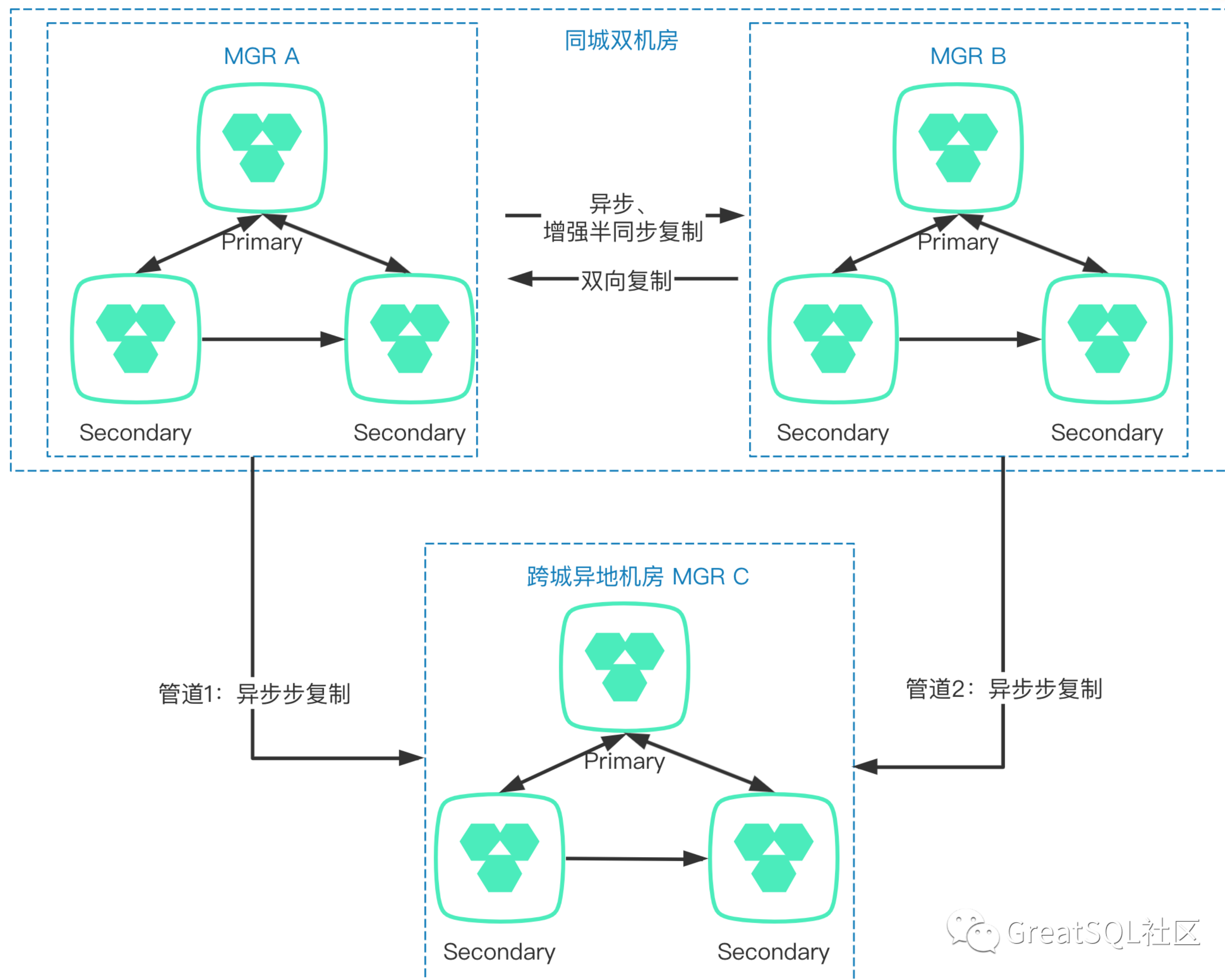
业务场景	可用方案	建议
【一般业务】不要求高一致性、高可靠性，以只读为主	<ul style="list-style-type: none">异步复制半同步复制MGR	5.7以上版本
【重要业务】对数据一致性、高可用保障都有较高要求	<ul style="list-style-type: none">半同步复制MGR	8.0以上版本
【核心业务】对数据一致性、高可用保障都有较高要求	<ul style="list-style-type: none">增强半同步复制MGR	8.0以上版本 多IDC、数据多副本 配合MySQL Shell、Router原生方案

• 同城跨IDC架构



MySQL高可用架构方案选择建议

• 跨城多IDC架构



MGR最佳实践

- 使用MySQL最新版本（8.0+），或者GreatSQL
- 奇数节点（最多9个）
- InnoDB引擎，且要有主键
- 单主模式
- 开启并行复制
- （尽量）不要使用外键
- 不要跑大事务
- 低延迟网络，避免WAN部署


```
[root@GreatSQL][(none)]> SELECT MEMBER_ID AS id,  
COUNT_TRANSACTIONS_IN_QUEUE AS trx_tobe_verified,  
COUNT_TRANSACTIONS_REMOTE_IN_APPLIER_QUEUE AS trx_tobe_applied,  
COUNT_TRANSACTIONS_CHECKED AS trx_chkd,  
COUNT_TRANSACTIONS_REMOTE_APPLIED AS trx_done,  
COUNT_TRANSACTIONS_LOCAL_PROPOSED AS proposed FROM  
performance_schema.replication_group_member_stats;
```

id	trx_tobe_verified	trx_tobe_applied	trx_chkd	trx_done	proposed
4b2b46e2-3b13-11ec-9800-525400fb993a	0	326	21384	40	21349
4b51849b-3b13-11ec-a180-525400e802e2	0	7	21370	21374	0
4b7b3b88-3b13-11ec-86e9-525400e2078a	1	0	21255	21255	0

等待冲突检测队列

等待apply队列

```
[root@GreatSQL][(none)]> select RECEIVED_TRANSACTION_SET as gtid_lag from
performance_schema.replication_connection_status
  where channel_name = 'group_replication_applier'
 union all
  select variable_value from performance_schema.global_variables
  where variable_name = 'gtid_executed';
```

gtid_lag
aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa1:1-612197303
aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa1:1-612196721

已获取的事务GTID

本地已apply的事务GTID

- GreatSQL是由万里数据库维护的MySQL分支
- 专注于提升MGR可靠性及性能，支持InnoDB并行查询特性
- 是适用于金融级应用的MySQL分支版本
- 地理标签，提升多机房架构数据可靠性
- 全新流控机制，让MGR运行更流畅不频繁抖动
- 相对官方社区版，MGR运行更稳定、可靠
- InnoDB表也支持并行查询，让CPU资源不再浪费
- 其他...

不忘DB初心，牢记万里使命

GreatSQL，更流畅

