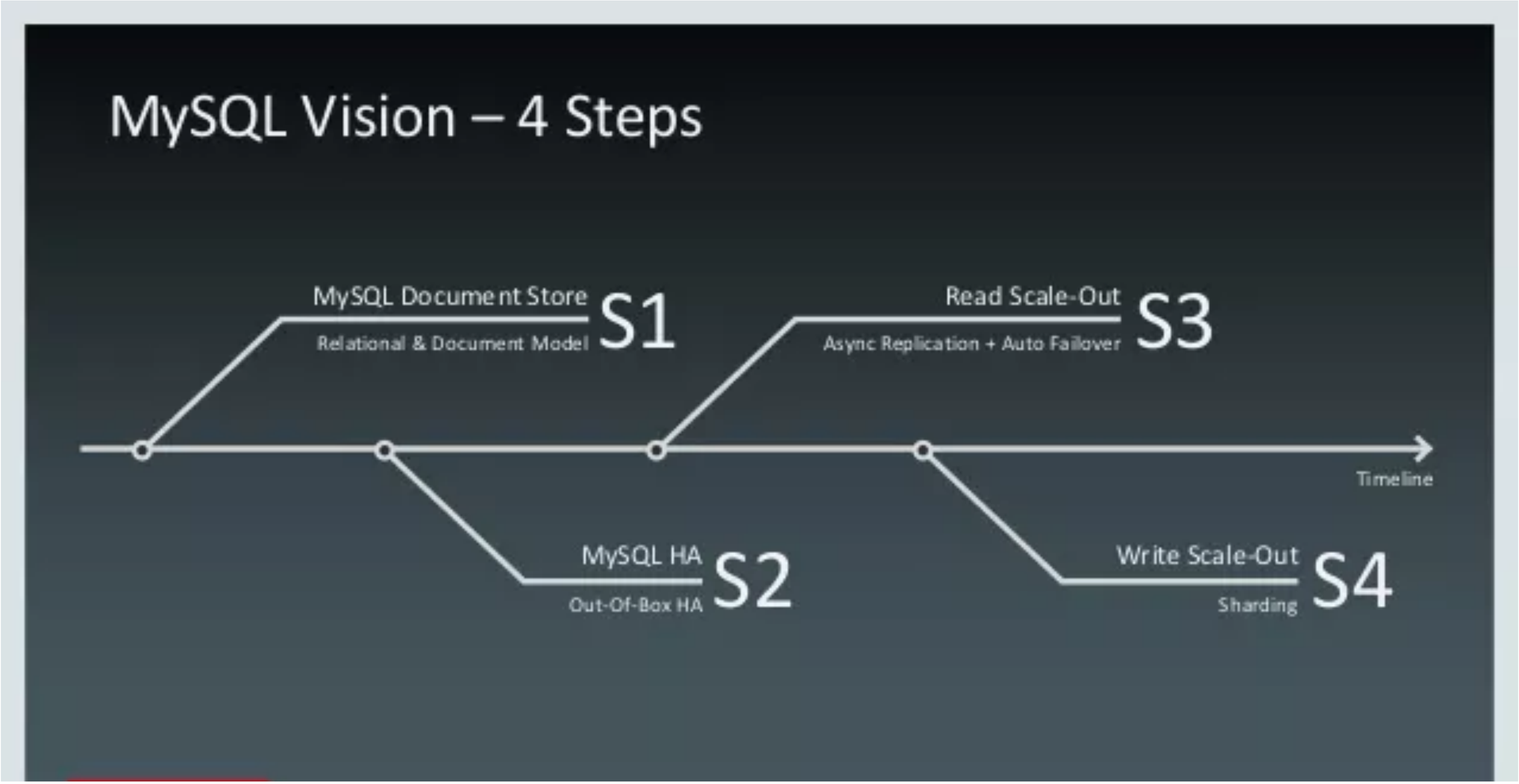
MySQL高可用之MGR

叶金荣
2025.7.31

01 About MGR

About MGR - Roadmap



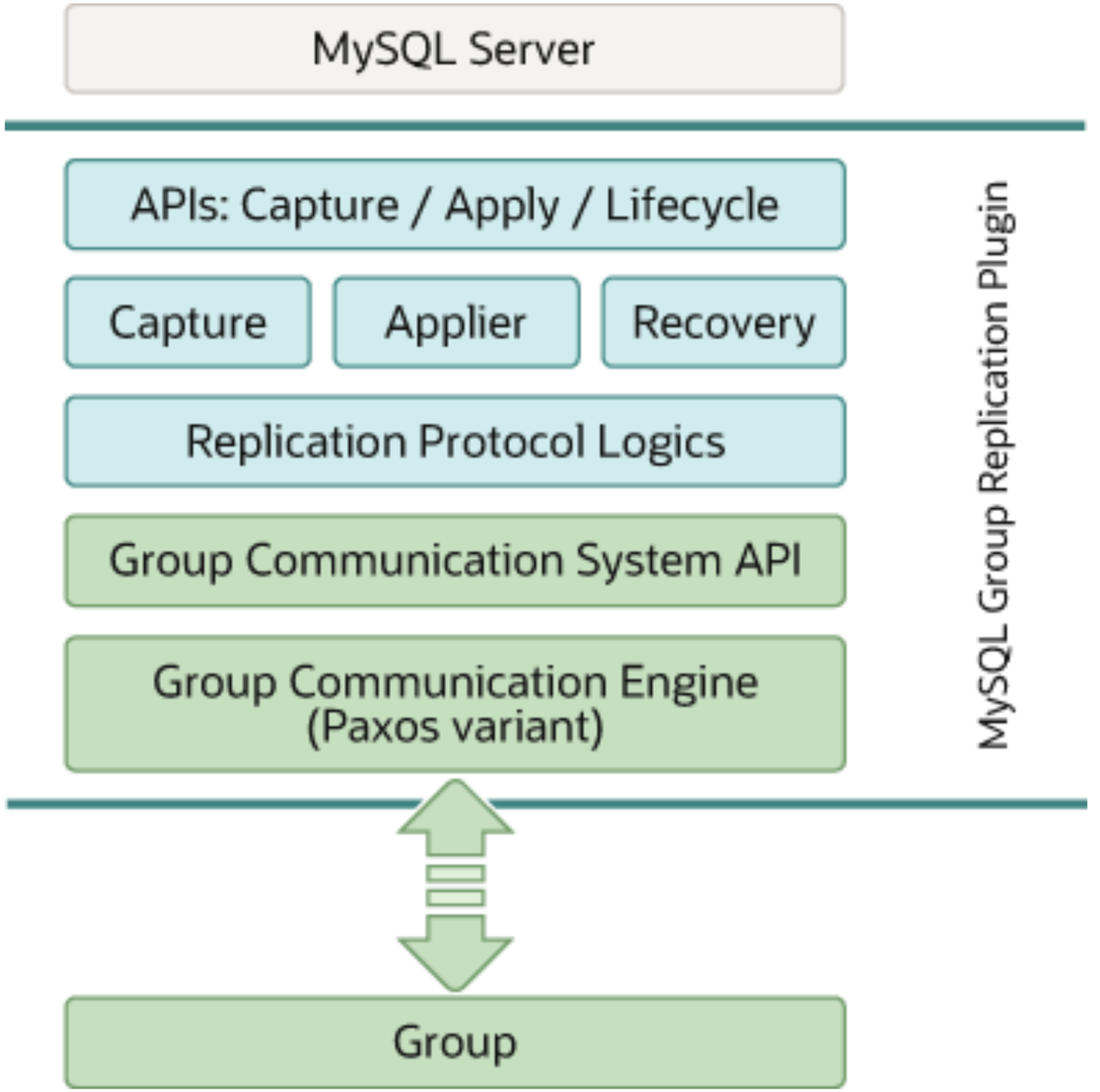
About MGR

GreatSQL

- MySQL Group Replication，简称MGR/GR，组复制
- 高可用、强一致的分布式数据库解决方案
  - 强一致性：事务提交需多数派节点确认，保证数据可靠
  - 高可用性：自动故障检测与恢复
  - 可扩展性：动态添加、删除成员
  - 低耦合：shared-nothing
  - 灵活：单主、多主

About MGR - 工作机制

- 主要模块
  - Capture，跟踪本地节点产生的事务
  - Applier，执行远程节点传输到本地的事务
  - Recovery，故障恢复时选donor节点、catch up binlog等
  - Replication Protocol Logics，消息封装、收发Xcom消息、冲突检测等
  - GCE，在该层具体实现XCom机制



About MGR - 多数派原则

GreatSQL

- 多数派原则
  - 要求达成多数派一致，即超过半数节点形成共识，事务才可以被提交
  - 想要形成多数派，则最少3个节点，最多9个节点

总节点数	多数派节点数	最大容忍故障节点数
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

About MGR - 数据同步、事务认证

- 事务数据同步、认证过程（简化后）
  - 事务写Binlog前先进入到GR层
  - 将事务封装后全局排序，发送给各成员
  - 各成员进行事务认证（冲突检测）
  - 认证通过后本地写Binlog完成提交，认证失败则回滚事务
  - 远程成员将事务转储Relay Log后回放
- 关键点
  - 不同节点同时更新同一行数据（根据主键判定）时，有可能产生冲突
  - 不同节点同时更新不同数据行时，不会产生冲突
  - 暂不支持DDL冲突检测

About MGR - 选主机制

- 何时触发选主
  - 主节点离线：管理员关闭主节点、主节点宕机、网络中断或主动退出组
  - 健康检测超时：组内节点未在设定时间内收到主节点心跳
  - 手动切换：主动发起切换
- 选主优先原则
  - 版本号：低版本优先
  - 节点权重：权重值越高越优先
  - 节点ID排序

About MGR - 事务认证队列清理

- 何时清理
  - 认证通过且已被各个节点都提交的事务
- 如何清理
  - 后台线程每隔60s（硬编码，目前无法调整）清理
  - 若等待被清理的事务特别多，可能出现经典的性能抖动问题
- 优化建议
  - 多使用小事务，避免事务堵塞
  - 不要在业务高峰期执行DDL

About MGR - 事务一致性保障

GreatSQL

- 根据不同业务场景选择事务一致性保障等级
  - 相关参数group_replication_consistency

可选值	作用描述	适用场景
EVENTUAL（默认）	读操作无需等待事务应用完成，可保证最终一致性	高吞吐低延迟场景，对一致性要求不高
BEFORE_ON_PRIMARY_FAILOVER	新主需先应用完积压事务才上线，避免读到旧事务	主节点切换时避免读取旧事务
BEFORE	读操作前等待所有先前事务应用完成，确保读一致性	对读一致性要求高的场景（如实时查询）
AFTER	事务提交后等待所有节点应用后才允许后续读操作	写后一致性要求高的场景（如支付系统）
BEFORE_AND_AFTER	结合BEFORE和AFTER，确保读写操作前后数据完全一致	强一致性场景（如核心交易系统）

- 多数场景使用默认EVENTUAL就可以
- 担心切主时读旧事务数据选BEFORE_ON_PRIMARY_FAILOVER
- 在从节点也总能读到最新数据选BEFORE
- AFTER及以上虽提高一致性等级，但可能引发性能问题，需谨慎（为什么MGR一致性模式不推荐AFTER）

About MGR - 约束限制

GreatSQL

- 使用MGR的一些约束限制
  - 必须是有主键的InnoDB表
  - 超过150MB（默认值，最大2GB）的大事务会报错
  - 不支持复制过滤（Replication Filters）
  - 不支持table lock 及 name lock（即 GET_LOCK() 函数 ）
  - 在不同节点上对同一个表分别执行DML和DDL时，可能造成数据丢失或节点报错退出
  - 在多主模式下不支持串行（SERIALIZABLE）隔离级别和多层级联外键表
  - 在多主模式下，如果多个节点都执行SELECT ... FOR UPDATE后提交事务会造成死锁
  - 各节点选择ROW格式Binlog，并转储Relay Log
  - 各节点的server_id, server_uuid不能相同
  - 各节点的lower_case_table_names参数值一致

MGR vs PXC

GreatSQL

■类比产品

- MariaDB Galera Cluster
- Percona XtraDB Cluster (PXC)

■关键差异点

- Group Communication System
- Binlogs & Gcache
- Node Provisioning
- Partition
- Flow Control
- Cross plateform
- DDL

MGR的优势与挑战

GreatSQL

■优势

- 自动化容错：节点故障自动恢复，无人工干预
- 弹性扩展：支持单主/多主，并可根据业务需求动态调整集群规模
- 数据安全：基于Paxos协议避免数不一致风险

■挑战

- 性能开销：组内事务广播，同步复制与冲突检测，可能增加事务延迟
- 网络分区：多数派节点发生网络分区时需人工介入
- 网络开销：要确保网络低延迟与高带宽

MGR的经典应用场景

GreatSQL

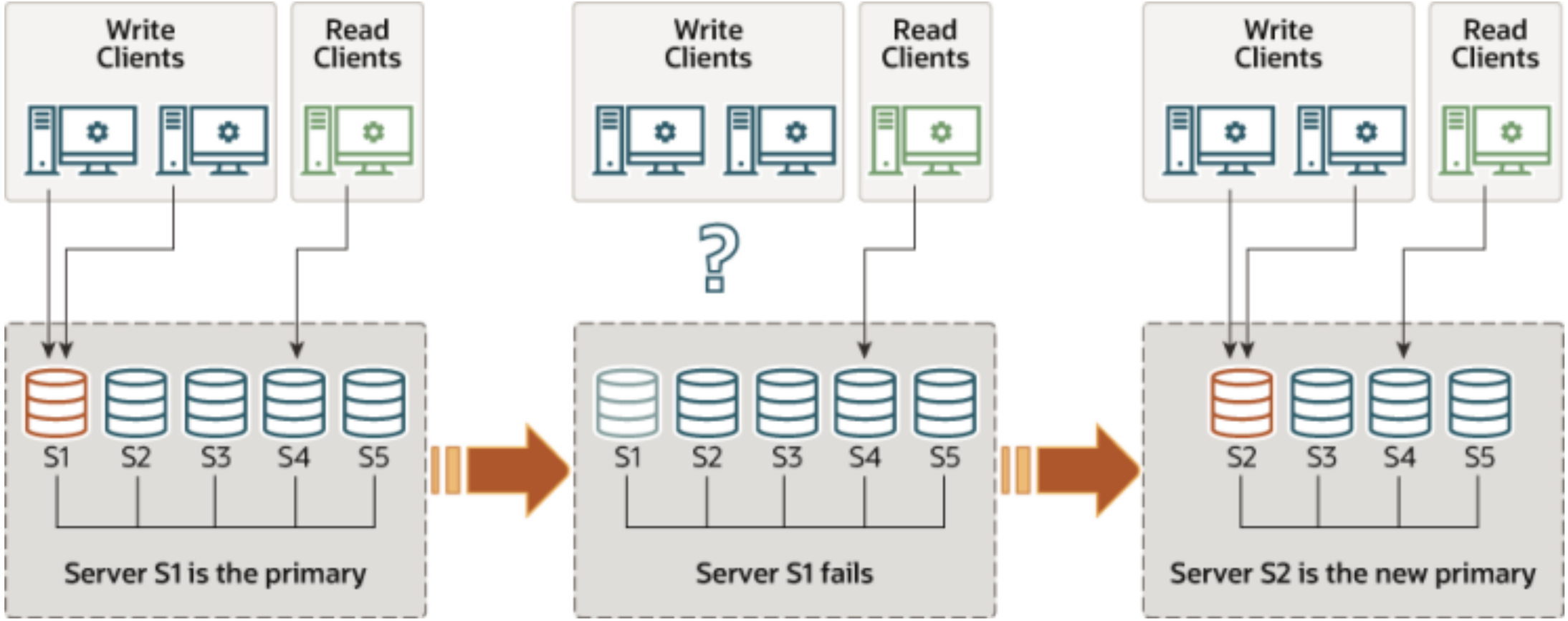
- 数据库基本高可用
- 数据库异地多活
- 数据库灾备系统

02 MGR高可用架构

MGR高可用

GreatSQL

- 写入模式（1）：单主
  - 开始时S1是Primary，提供读写服务
  - 当S1故障时，S2-S5节点投票选举S2作为新Primary

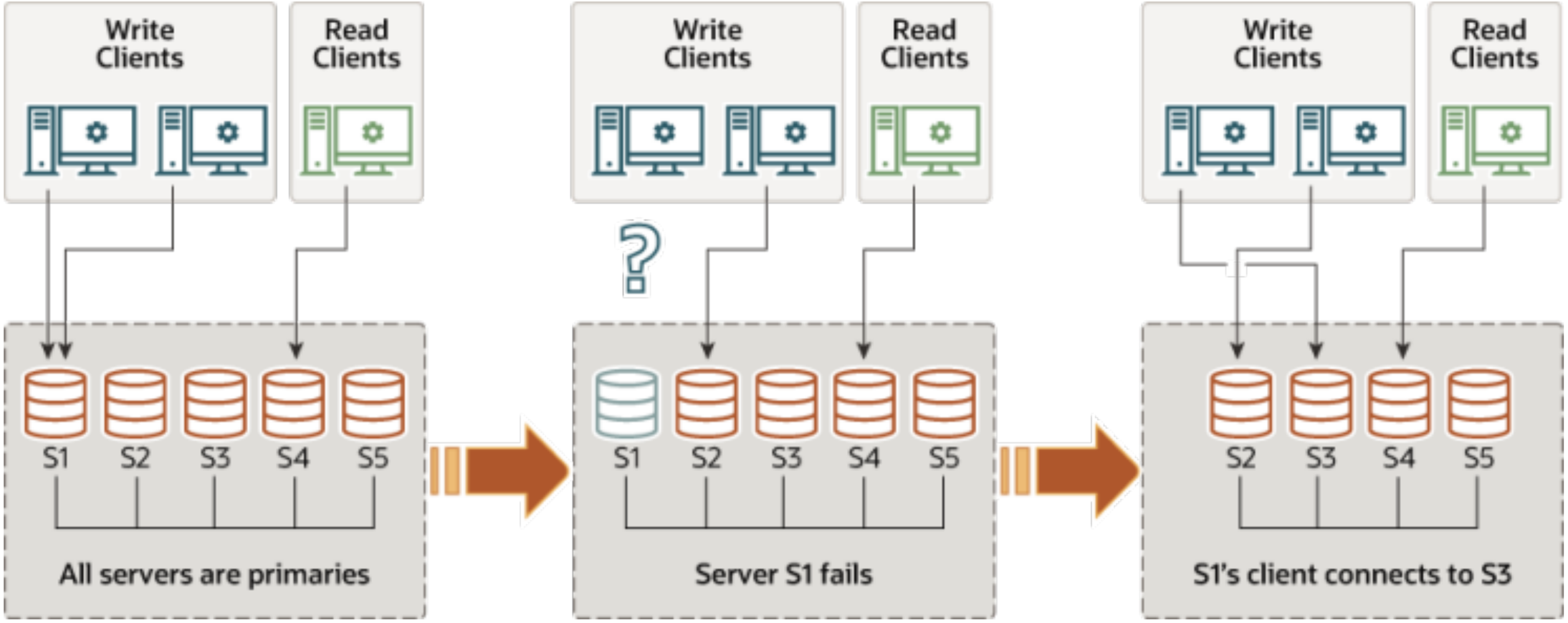


MGR高可用

GreatSQL

▪写入模式（2）： 多主

- 所有节点都是Primary，都可以提供读写服务
- 任何一个节点发生故障时，只需要把指向这个节点的流量切换走就行



MGR高可用

GreatSQL

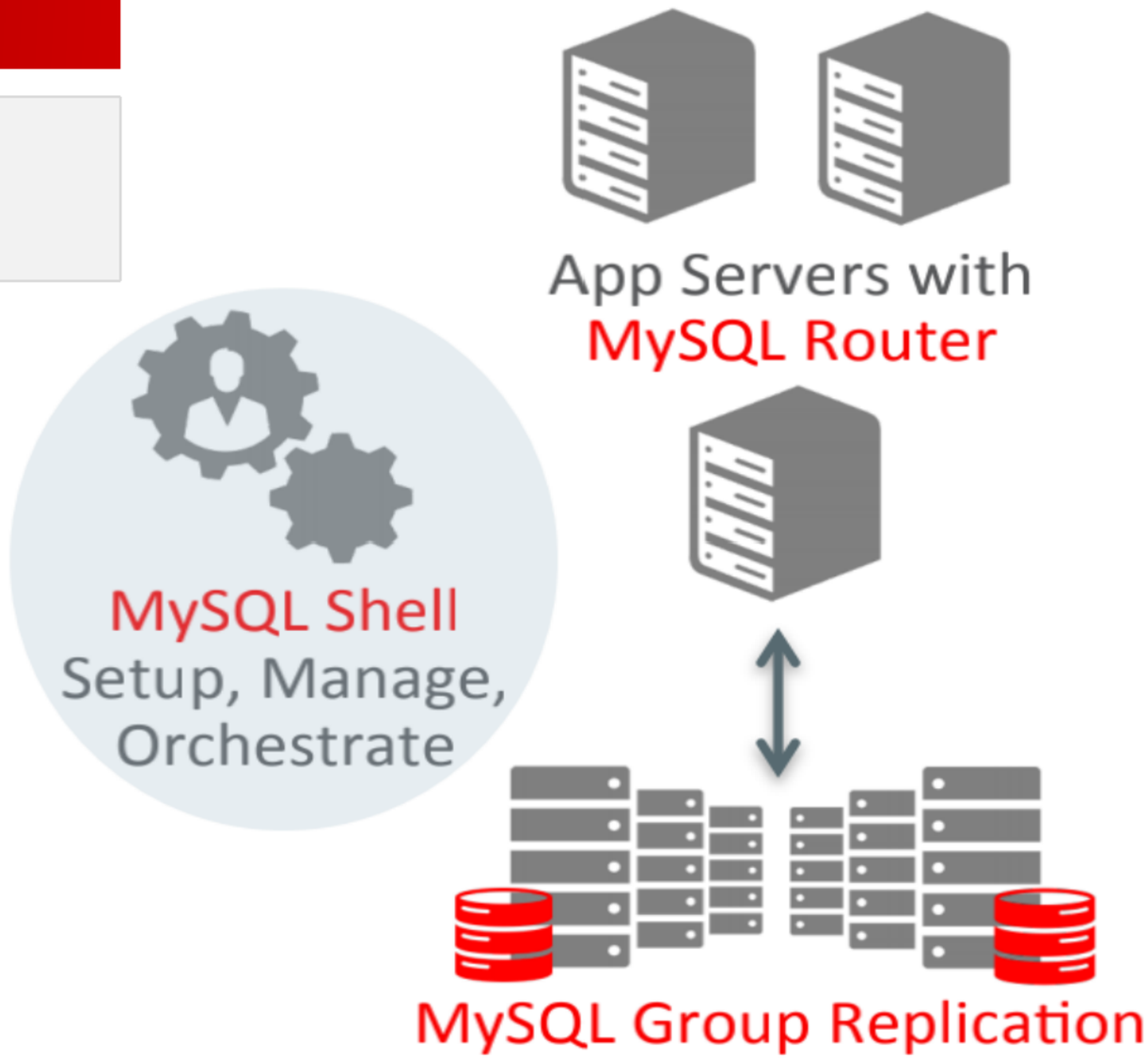
- 写入模式（3）：透明Proxy + MGR
  - 从架构及业务复杂度等方面考虑，建议选择单主模式
  - 应用端通过MySQL Router（或ProxySQL）连接后端组复制成员节点
  - 当后端节点发生切换时，Proxy层自动感知
  - 对应用端来说几乎是透明的，架构上也更灵活，扩展性更好

MGR高可用 - MIC

InnoDB Cluster解决方案

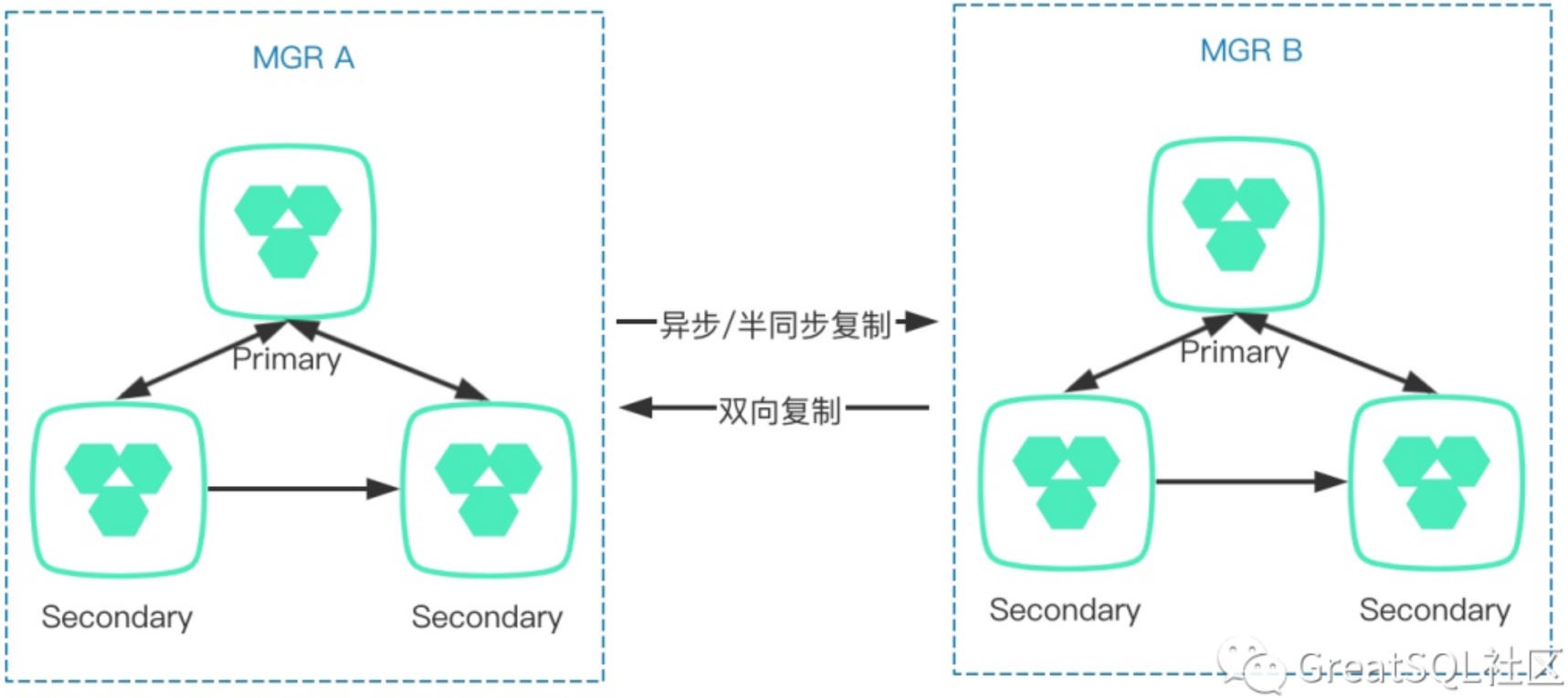
- MySQL Router：路由客户端连接
- MySQL Shell: 集群搭建、管理工具

- MIC
  - M：MySQL
  - I：InnoDB
  - C：Cluster



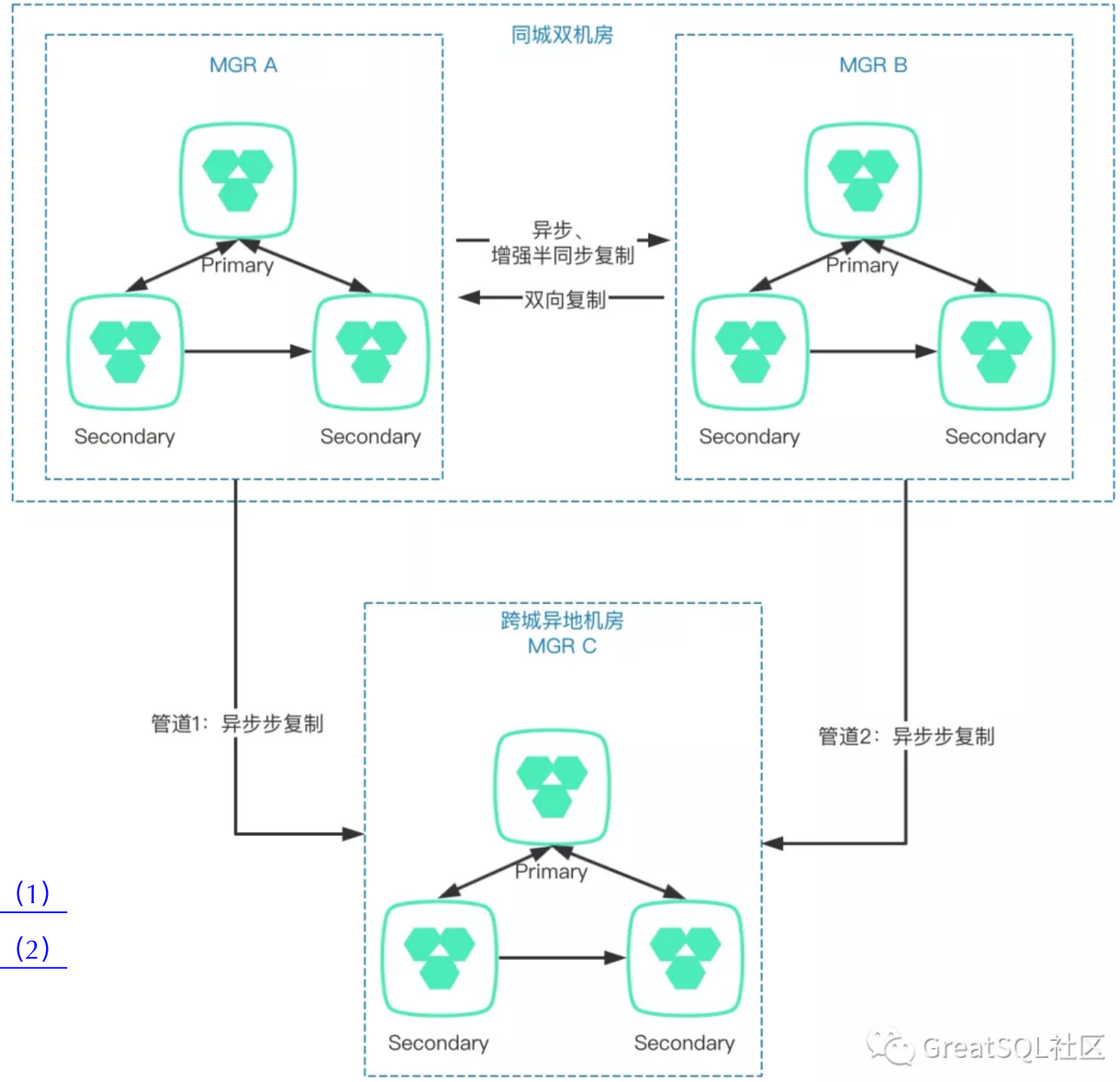
MGR高可用 - 跨IDC高可用

- 同城跨IDC架构



MGR高可用 - 跨城高可用

- 跨城多IDC架构



参考：

金融应用场景下跨数据中心的MGR架构方案（1）

金融应用场景下跨数据中心的MGR架构方案（2）

03 MGR最佳实践

MGR最佳实践

- 奇数节点（最多9个）
- 低延迟网络，避免WAN部署
- 单主模式
- 表要有主键
- InnoDB引擎
- 不要用到外键
- 不超过10MB的小事务
- 对同一个表DDL和DML不要在不同节点进行

MGR常见性能瓶颈及原因

- 大事务造成延迟，甚至节点退出
- 网络成为瓶颈，导致消息延迟大
- 组中个别节点存在性能瓶颈
- 不恰当的流控阈值，导致性能受限
- 其他常见性能瓶颈导致

MGR性能优化参考（1）

- 各节点服务器性能基本一致，不存在明显短板
- 各节点服务器分摊的业务压力基本一致，不让某个节点承担过重负载
- 事务保障级别选择最终一致性，不选择BEFORE、AFTER级别
- 避免产生大事务，把大事务拆分成小事务
- 适当调低 group_replication_transaction_size_limit 阈值，限制事务大小
- 避免跨VLNA运行组复制，节点间网络延迟最好低于5ms
- 关闭通常不必要的流控机制

MGR性能优化参考（2）

- 采用单主模式，并启用快速单主模式

```
group_replication_single_primary_mode=ON
```

- 适当调低事务组复制事务上限阈值，避免产生大事务

```
group_replication_transaction_size_limit=20M
```

- 对大事务消息进行分片处理，避免网络延迟

```
group_replication_communication_max_message_size=10M
```

- 关闭流控机制

```
group_replication_flow_control_mode="DISABLED"
```

- 适当调整节点广播异常超时阈值，如果网络环境不好，可以适当调高

```
group_replication_member_expel_timeout=5
```

MGR监控

```
mysql> SELECT MEMBER_ID, COUNT_TRANSACTIONS_IN_QUEUE AS trx_que,
  COUNT_TRANSACTIONS_REMOTE_IN_APPLIER_QUEUE AS app_que,
  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;
+--------------------------------------+----------+---------+-----------+-----------+-----------+
| MEMBER_ID                            | cert_que | app_que |  trx_chkd |  trx_done | proposed  |
+--------------------------------------+----------+---------+-----------+-----------+-----------+
| 4d10a9db-e53e-11eb-aeaf-525400fb993a |        0 |     326 | 459419476 | 459419154 |         0 |
| 5f031f98-e53e-11eb-bd31-525400e802e2 |     2936 |  238391 | 459290196 | 459290198 |         0 |
| ab884cc1-e095-11eb-876c-525400e2078a |     2976 |  238519 | 459673911 |         2 | 459673915 |
+--------------------------------------+----------+---------+-----------+-----------+-----------+
```

等待冲突检测队列

等待apply队列

04 About GreatSQL

GreatSQL开源数据库简介

- GreatSQL是MySQL分支，在**高可用性**、**性能提升**、**安全性**和**兼容性**等多个维度深度优化，专为严苛金融级业务场景打造。其基于Rapid和Turbo引擎的HTAP解决方案，满足金融等行业的高要求，是MySQL/Percona Server的免费且理想替代方案
- 源码、文档、下载等资源
  - 官网：https://greatsql.cn
  - 文档：https://greatsql.cn/docs/
  - 源码：https://gitee.com/GreatSQL/GreatSQL
  - 下载：https://gitee.com/GreatSQL/GreatSQL/releases

GreatSQL优势特性

GreatSQL优势特性

- **服务高可用性提升**：在MGR中增加地理标签、仲裁节点、快速单主、智能选主等新特性，并优化MGR事务认证和流控机制

- **HTAP性能大幅提升**：新增适用于OLAP业务需求的Rapid和Turbo引擎，优化事务锁机制，TPC-H和TPC-C性能分别最高提升约**200**多倍和**30%**

- **良好的SQL语法兼容性**：**100%**兼容MySQL，支持常见 Oracle 用法

- **金融级安全性保证**：支持数据存储和通信国密算法、备份加密、审计、数据脱敏、账号密码增强等特性

- 详情参见：**GreatSQL 优势特性**

GreatSQL

更流畅，更安心

成为中国广受欢迎的
开源数据库