



第十一届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2020

架构革新 高效可控



北京国际会议中心 | 2020/12/21-12/23

官方MySQL在高可用演进之路

知数堂 吴炳锡





大纲

1. 什么是高可用
2. 第一代传统复制： MHA及使用架构
3. 第二代基于GTID复制： GTID+Binlog server At Booking & Facebook
4. 第三代增强半同步复制： GTID+增强半同步及多IDC架构及使用架构
5. 第四代MySQL原生高可用 ： MySQL InnoDB Cluster
6. MySQL Replication Auto Failover
7. 高可用选择建议&趋势

什么是高可用

高可用： High availability 缩写HA ， IT术语，，指系统无中断地执行其功能的能力，代表系统的可用性程度。

可用性	年故障时间
99.99999%	32秒
99.999%	5分15秒
99.99%	52分34秒
99.9%	8小时46分
99%	3天15小时36分

高可用版

采用一主一备的经典高可用架构，适合80%以上的用户场景。

- 大中型企业的生产数据库
- 互联网、物联网、零售电商、物流、游戏等行业的数据库

高可用性（多可用区）

最近更新时间：2019-11-19 17:01:15

[编辑](#) [查看pdf](#)

多可用区部署可保护数据库，以防数据库实例发生故障或可用区中断，请参见 [地域和可用区](#)。

云数据库 MySQL 多可用区部署为数据库实例提供高可用性和故障转移支持。多可用区是在单可用区的级别上，将同一地域的多个单可用区组合成的物理区域。

引：<https://zh.wikipedia.org/wiki/%E9%AB%98%E5%8F%AF%E7%94%A8%E6%80%A7>
<https://cloud.tencent.com/document/product/236/8459>



什么是高可用

数据库高可用又面临

数据一致RPO

业务连接续性RTO

容易混淆的概念：数据库可用区灾备

复制一致

性能 & 一致性

半同步

幻读

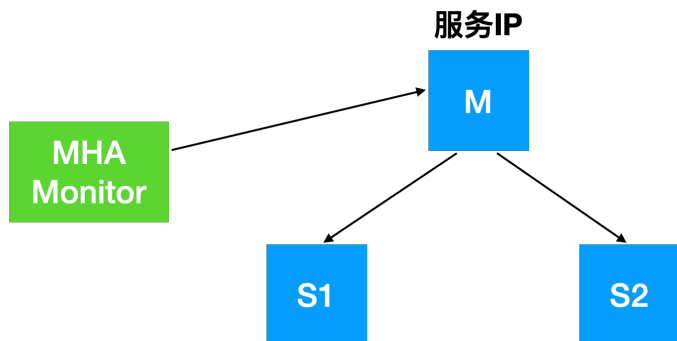
增强半同步

幽灵事务

MGR

?

MHA及使用架构



适用场景：

- MySQL版本 ≤ 5.5 ，异步复制 一主多从环境
- 基于传统复制的高可用

解决问题：

- 尽最大能力做数据补偿（并不保证）
- 最大努力实现 RPO，有RTO概念支持

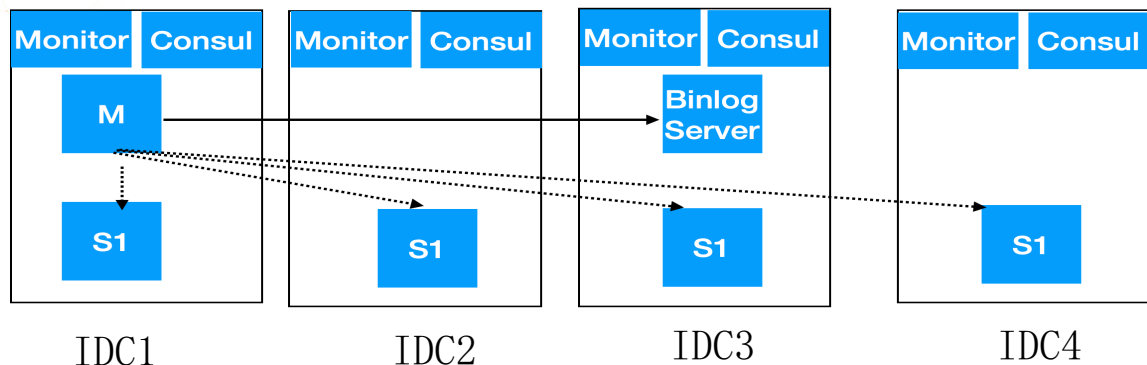
存在的问题：

- GTID模型强依赖binlog server
- 对于5.7后binlog不能识别，对并行复制支持不好；
- 服务IP切换依赖自行编写的脚本，也可以和DNS结合使用
- 运维上需要做SSH信任，切换判断，现在项目基本无维护

MHA及使用架构

- 优点
 - MySQL \leq 5.5, MySQL高可王者
 - 可以利用原始主库上binlog, slave上的relay log进行数据补偿
 - 为了绝对保证RPO, 需要把binlog放到共享存储上
- 缺点
 - 运行中不在管理从库是不是复制
 - 如果使用了共享存储存binlog, 会把幽灵事务补偿到从库上
 - 对于 MySQL 5.7的Binlog解析报错 & MySQL 8.0不用考虑

第二代基于GTID复制



————→ binlog半同步，保证数据安全，一般至少两个，保证RP0

.....→ 异步复制，保障性能，通过监控延迟，保证RT0

Monitor 多个IDC中Monitor组成分布式监控，把健康的MySQL注册到consul中，同时对于从库复制延迟情况，也同步到consul中

Consul 配置中心，对外提供健康的MySQL服务



第二代基于GTID复制

•特性

- 基于MySQL 5.6 + GTID
- binlog server使用半同步（是否允许半同步退化）

•缺点

- 存在幻读问题
- MySQL 5.6本身半同步ack确认在Dump_thread中，dump_thread存在IO瓶颈问题



第三代增强半同步复制

这代产品对于MySQL版有强依赖： MySQL 5.7以后版本

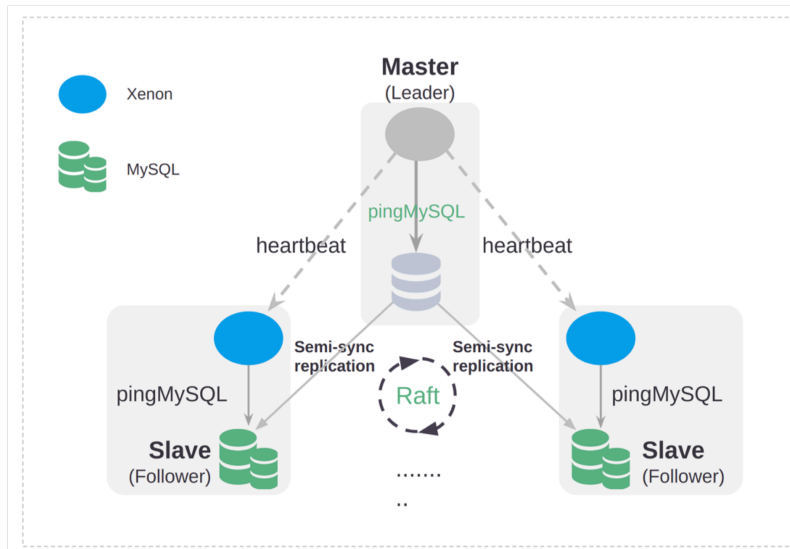
代表产品： MySQL Replication Manager / github orchestrator

国内青云开源的： Xenon (MySQL Plus)

MySQL 5.7增强点：

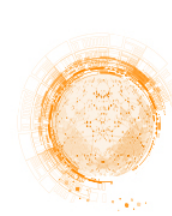
1. 增强半同步，主从为了增强半同步都独立出来线程
2. 主库binlog group commit
3. 从库： 并行复制，基于事务的并行复制， sql_thread: writeset 从库上的并行提升

第三代增强半同步复制

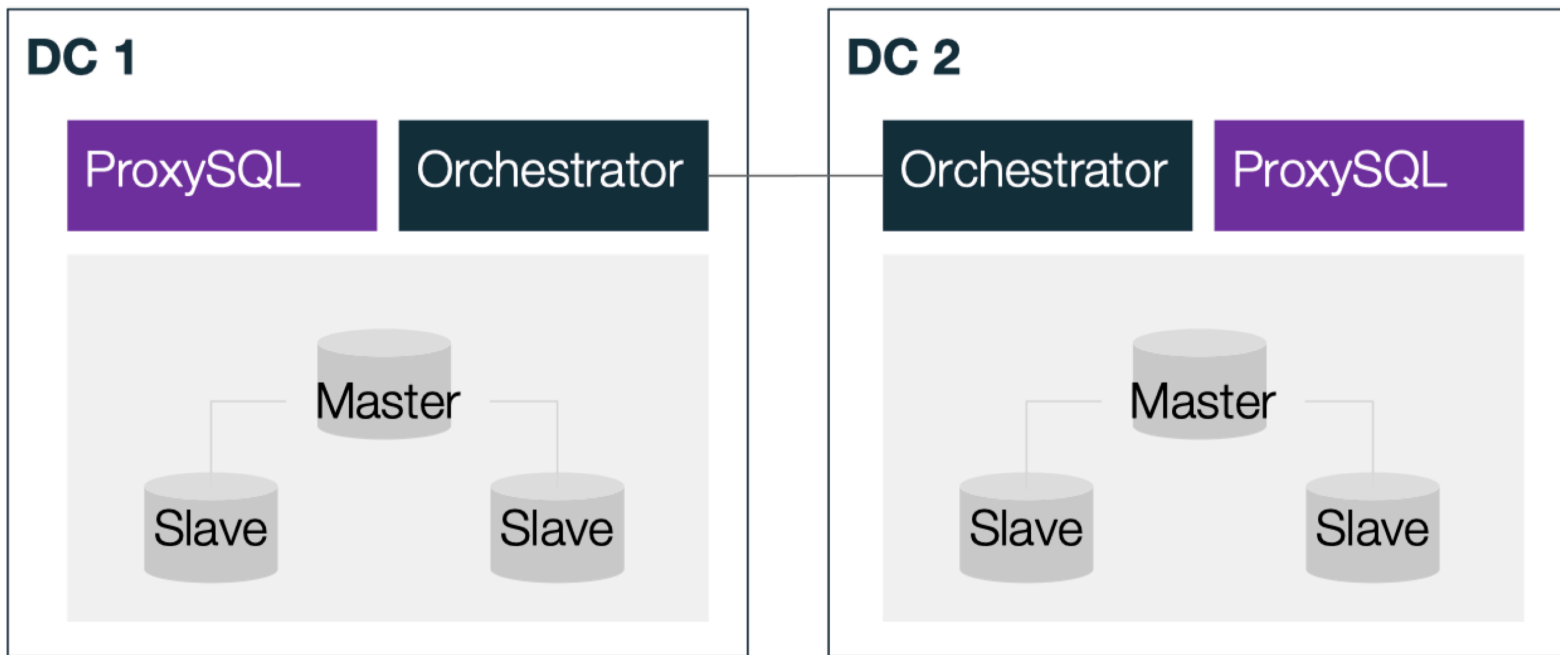


特点:

- 每个节点上都有一个独立的agent
- 利用raft构建集群，利用GTID做index选主
- Leader对外提供写服务
- Follow节点可以对外提供读服务
- 适合结合Docker工作



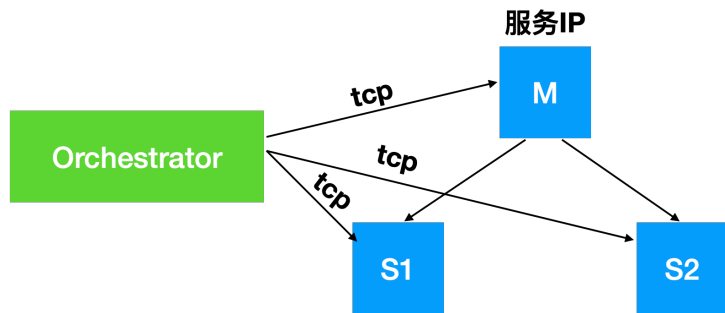
GTID+增强半同步及多IDC架构及使用架构



第三代增强半同步复制

•MySQL层面参数配置

- set global slave_net_timeout=4;
- change maser to master_connect_retry=1, master_retry_count=86400;



- 1.Orchestrator监控M，当为M挂掉
- 2.获取Slave上Show slave status输出，依据io_thread状态对Master进一步判断
- 3.因为基于GTID+增强半同步，master_auto_position=1的特性，两个s间，非常好处理数据一致性。



第三代增强半同步复制

•特点

- 利用MySQL自特性，复制并行度高，dump_thread读取binlog可并行, 半同步ack线程独立，实现binlog group commit, writeset等
- 基本不用担心复制延迟，主库能并行的，从库也能并行
- 强依赖于增强半同步（依赖多个slave），利用半同步保证RPO，RTO依赖于复制延迟

•缺点

- 增强半同步中存在幽灵事务(local commit)
- 增强半同步运维上可能出现影响

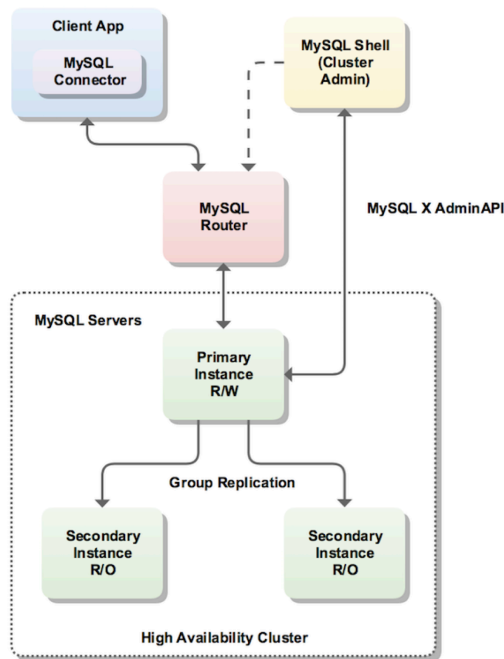


第三代增强半同步复制

•这代产品的特点

- 简单，方便运维
- 充分利用MySQL自身的特性，并行复制，增强半同步
- 扩展灵活，都可以支持调用外部脚本

第四代MySQL原生高可用



- 利用MGR MySQL原生高可用特点
- MySQL Router默认包含在MySQL发行包中，MySQL Router可以部署多个
- MySQL Router提供基于端口号的读写分离
- mysql shell 支持快速构建MGR
- 开源社区可以利用ProxySQL替代MySQL Router



第四代MySQL原生高可用

优点:

原生高可用，官方推荐Single master ，但实际建议使用multi master模式，单点写入

推荐ProxySQL+MGR结合， 性能为王环境：mysql router

缺点:

使用上有一些限制

运维方面有难度，特别对于multi master模式，需要控制写入及并发(更新丢失)



MySQL Replication Auto Failover

使用要求:

- 必须使用GTID复制&master_auto_position=1
- Change master to中启用: **SOURCE_CONNECTION_AUTO_FAILOVER**
- 所有节点必须存在一样的复制用户及密码
- 复制用的用户名和密码必须写到change master to 语句中

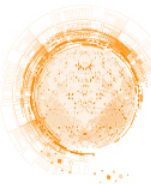
实现原理:

- 从库以change master to中master_connect_retry(2-4s)定义的时间间隔总共重试master_retry_count(3-5次)定义次数, 后认为原始主库挂掉;
- 从asynchronous_connection_failover_add_source定义的可以用服务器列表中, 选择权可用权重高的节点自动连接过去



小结

- 从第二代高可用开始后，整个高可用技术中出现： 配置中心 (consul)， DNS， Proxy或LVS类技术等综合应用
- 从原来的单集群运维扩展到： 数据库托管服务， RDS平台
- 从实现上看，更偏重于自我定制高可用平台



高可用选择建议&趋势

	MySQL版本	可用软件	存在问题
第一代	MySQL 5.5	MHA	传统复制，主库存在没把日志传输到从库上的风险，数据补偿，不能处理MySQL本身的幽灵事务
第二代	MySQL 5.6	Facebook开源的mysql, Maxscale	利用binlog server和主库做半同步，保证数据安全，binlog server没太好的开源解决方案
第三代	MySQL 5.7	xenon, mysql replication manager Github orchestrator	增强半同步，基本完美
第四代	MySQL 8.0 MySQL 5.7	mysql-router、proxysql mgr	原生高可用解决方案

现在选择高可用推荐： 第三代， 第四代架构中选择



高可用选择建议&趋势

•互联网行业

- 集成化高，低成本
- K8s，基于主从高可用，云原生高可用k8s+vitess+orchestrator

•金融行业

- 强一致性，更多的副本，多IDC
- K8s+MGR

•关注技术

- K8s, Vitess, MGR, 复制，服务注册（第五代MySQL高可用方案）
- MySQL生态解决方案

QA环节



• 吴炳锡

- 知数堂联合创始人， 3306 π 社区联合创始人，腾讯TVP成员， 15年MySQL从业老兵。
- ChinaUnix.net MySQL版版主； 担任多年DTCC，SACC， Oracle技术嘉年华讲师及顾问。
- 擅长多年MySQL系统架构设计及培训教学经验 & MySQL大规模运维管理优化、高可用方案、多IDC架构设计，企业级应用数据库设计等

THANKS

