

数据库高可用容灾方案设计和实现

丁顺

UCloud 存储研发工程师



目录

01

高可用数据库概述

02

业界典型高可用数据库架构

03

高可用数据库自动化运维

04

总结



01

高可用数据库概述



什么是高可用数据库

- **一系列的数据库构成的总体系统**

- 任何时刻，至少有一个节点接受客户端请求，提供数据库服务
- 大多数系统有一个主节点，处理主要请求
- 若干个备用节点

- **系统可以做容灾切换**

当主节点无法提供服务，让一个备用节点成为主节点

- **相比单个数据库服务，减少宕机的影响时间**



高可用数据库带来的便利

- 系统可用性提高
- 读写分离
- 变更不停服
- 备份不影响服务性能



高可用数据库关键问题

- **各个数据库数据如何做同步？**
 - 要保证切换后的数据库也有最新的数据
 - 同步对主库和备库有什么影响？
- **容灾切换如何进行？**
 - 切换之后，要确保数据的一致性
 - 架构不同，容灾切换的复杂度有差别
- **如何提高运维效率？**



02

业界典型高可用数据库架构



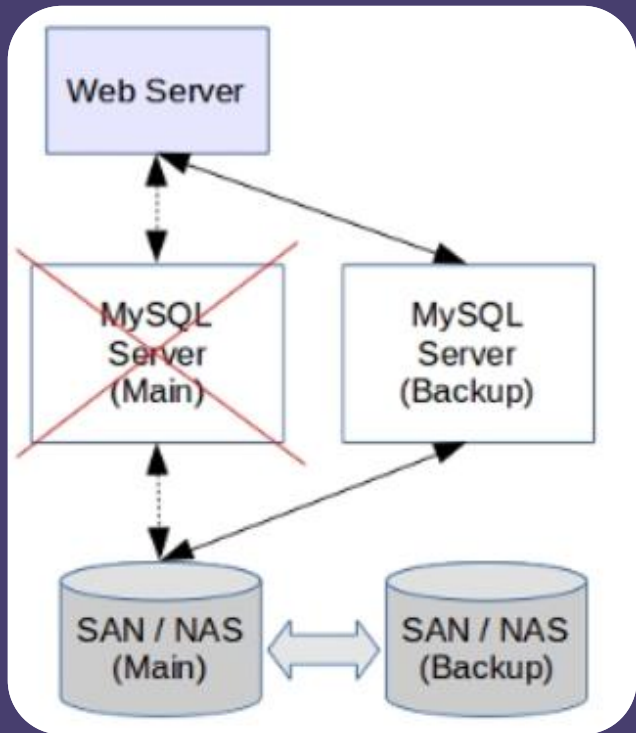
高可用架构概览

- 按照数据同步方式划分
 - 共享存储
 - 操作系统实时数据块复制
 - 数据库主从复制
 - 数据库高可用集群
- 每种数据同步方式可以衍生出不同的架构



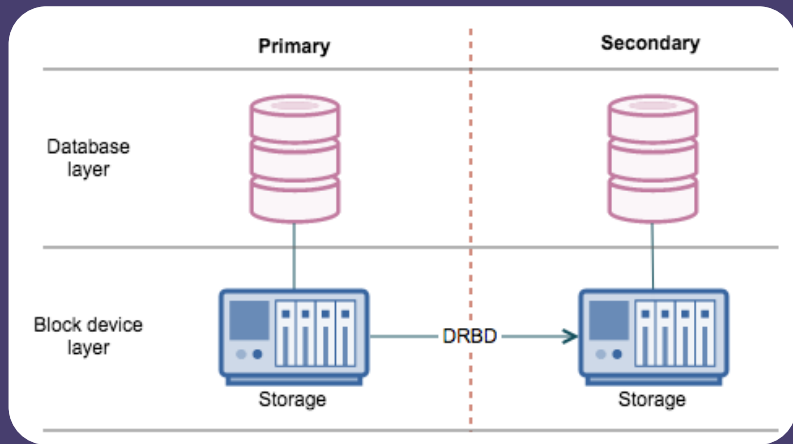
共享存储

- 若干DB服务使用同一份存储
- 容灾切换
 - 启动备用DB，成为新的主DB
 - 导流到新的主DB
- 对共享存储可用性要求高
“把鸡蛋放在了一个篮子里”
- 对网络性能要求高
- 但它是未来的发展趋势
计算存储分离，和DB深度结合



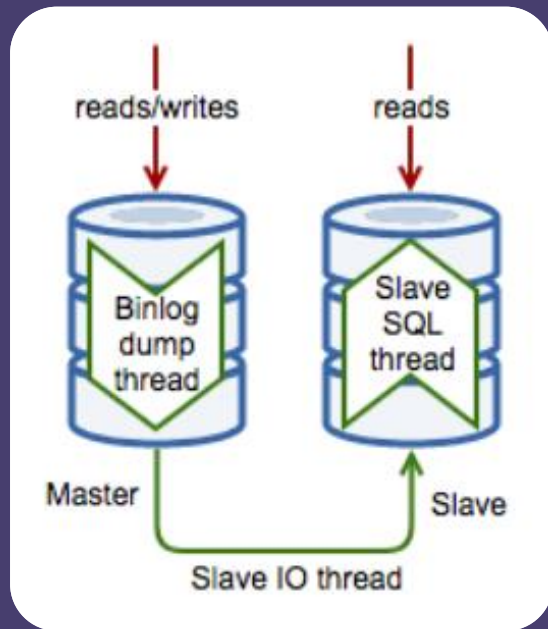
操作系统实时数据块复制

- **典型：DRBD**
远程RAID1
- **容灾切换**
 - 激活备用节点的相关存储设备
 - 启动数据库服务
- **只能有1个数据副本提供服务**
- **恢复时间较长**



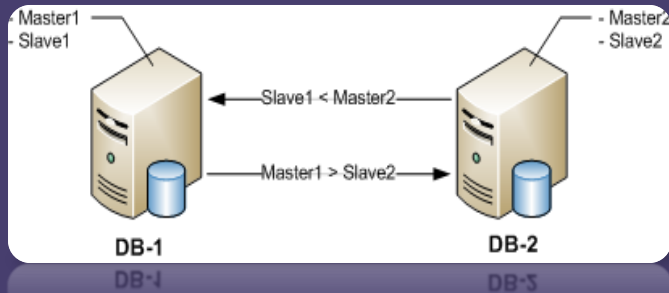
数据库主从复制

- 一个主库，多个从库
 - 主库同步数据库日志到各个从库
 - 从库各自回放日志
- 容灾切换
 - 从库追完数据，提升为主库
 - 请求导流到新主库
- 最经典的数据同步模式
- 衍生出诸多架构改进



主从复制（续）：双主架构

- **经典主从**：容灾后，原来的主需要设置为从
那可不可以一开始就设置好呢？可以！
两个DB互相向对方做数据复制
- **无论往哪一个DB写数据，另一个都自动同步**
确保容灾后数据依旧同步
- **容灾切换步骤简化**
只要切换请求流量



主从复制（续）：改进技术

- 日志自动寻址技术

切换后，自动定位新主库的日志同步点

MySQL：GTID（Global Transaction ID）

- 异步复制改进

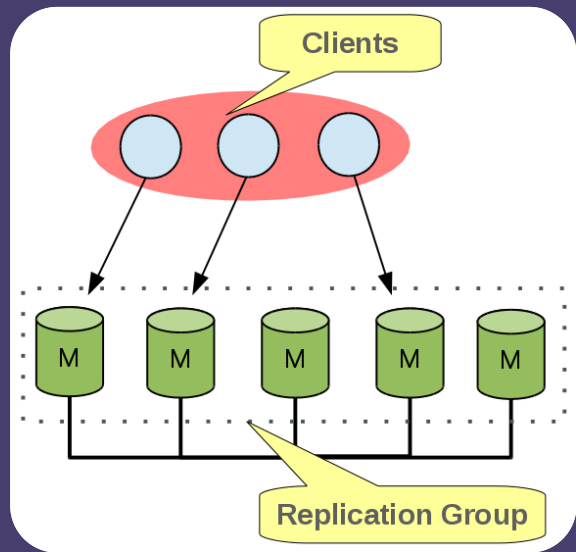
半同步复制

保证更好的数据一致性



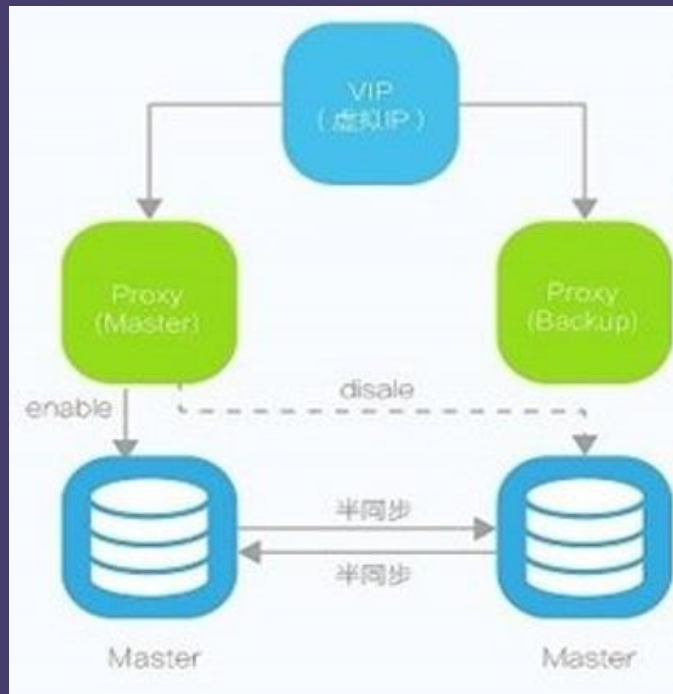
数据库高可用集群

- 数据库提供多节点一致性同步机制
利用该机制构建多节点同步集群
- 典型案例
 - MGR (MySQL Group Replication)
 - Galera
- 几乎同步的复制
- 理论上每个节点都可以读写
- 容灾切换
只需要切换请求的流量

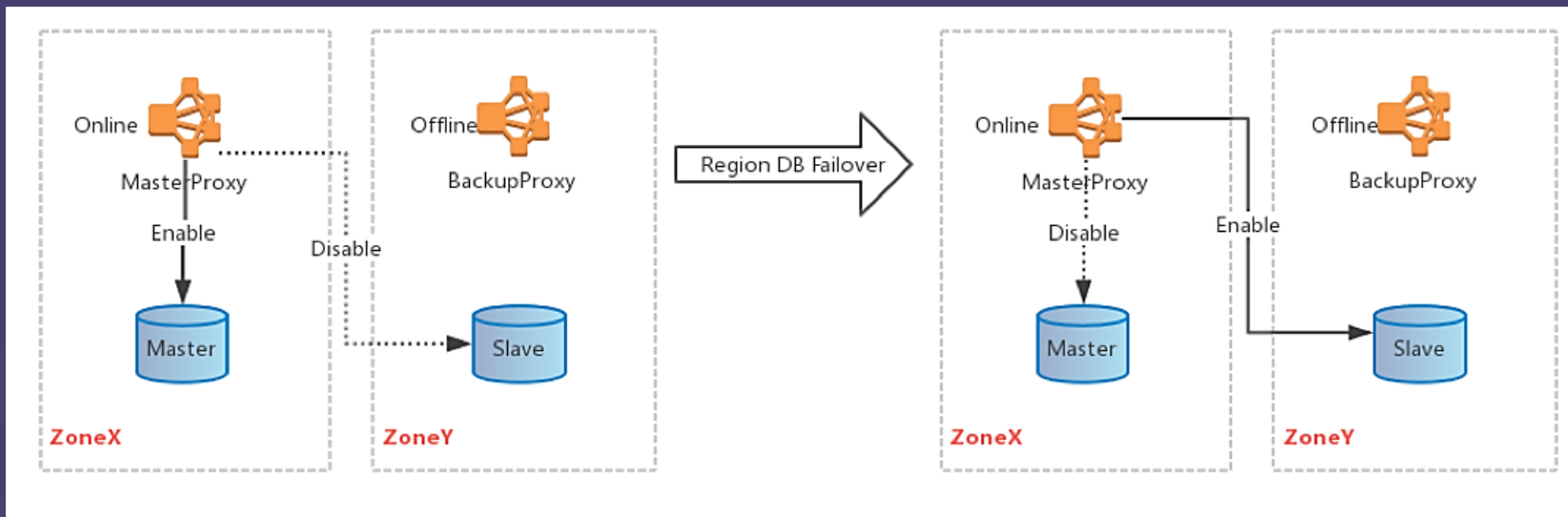


案例：UDB MySQL高可用实例架构

- UDB：UCloud云数据库产品
 一键创建高可用数据库架构
- 设计思路
 - 考虑原生MySQL兼容
 - 架构尽可能涵盖不同版本
 - 架构尽可能涵盖不同使用场景
- 基于数据库主从复制
 - ✓ 使用双主架构
 - ✓ 使用半同步复制
 - ✓ 使用GTID



UDB容灾切换：示意图



03

高可用数据库自动化运维



高可用数据库自动化运维：概述

- 自动化运维是高可用数据库的难点

如何同时管理大量高可用数据库？

- 自动化运维重点方向

- 容灾切换自动化
- 高可用数据库运行状况监控
- 健康状况自动检查和问题修复



容灾切换自动化

- 准确判断需要容灾
- 容灾不是你想切，想切就能切
- MySQL的业内常用方案

讨论：网络波动

备库数据要尽量和主库一致

- MHA
- Orchestrator



健康状况自动检查

- 同步情况自动监控搭配告警

复制有没有正常进行

主从延时

可以用Prometheus等工具定期采集

- 异常情况自适应调整

延时过大，帮助从库加快追日志速度

极端情况下，重做从库



UDB：海量高可用数据库自动化运维

- 高可用容灾集中式自动化管理

 - 自研自动容灾逻辑

 - 大规模，高并发，自动化

- 自动化问题探测，问题修复

 - 自动拉起DB，恢复服务

 - 自动恢复数据同步

 - 自适应流量控制

- 高效运维工具和巡检工具



UDB：高可用运维经验

- 日常需要做例行巡检，保证高可用数据库的健康
主从延时是导致高可用容灾无法切换的重要原因之一
- 定期容灾演练很有必要
尤其要演练切换后数据一致性
- 高可用切换记录日志，切换失败需要马上告警
切换记录日志可以做复盘分析
高可用不能容灾时第一时间介入，缩短影响时间



04

总结



总结

- 高可用架构是数据库运行稳定必不可少一部分
- 高可用架构设计时需要考虑诸多问题
 - 高可用数据同步
 - 高可用自动切换
 - 高可用自动化运维
- 在云环境下，推荐使用云数据库一键完成上述配置



UCloud云数据库UDB



关注“UCloud技术公告牌”，更多分享与交流

