

课程内容

- 主从架构详解
- 高可用架构分析
- 分库分表场景分析
- 表拆分方式

主从架构

为什么要主从架构？

- 1.如果主服务器出现问题，可以快速切换到从服务器提供的服务
- 2.可以在从服务器上执行查询操作，降低主服务器的访问压力
- 3.可以在从服务器上执行备份，以避免备份期间影响主服务器的服务

主从方案

(课前思考)

- 1、M-S
- 2、M-S-S-S
- 3、M-M-M-S
- 4、MM
- 5、SSSS

数据同步方式：1、GTID（底层也是基于bin-log），2、bin-log

主从复制：延时问题？

有哪些主从复制方式？

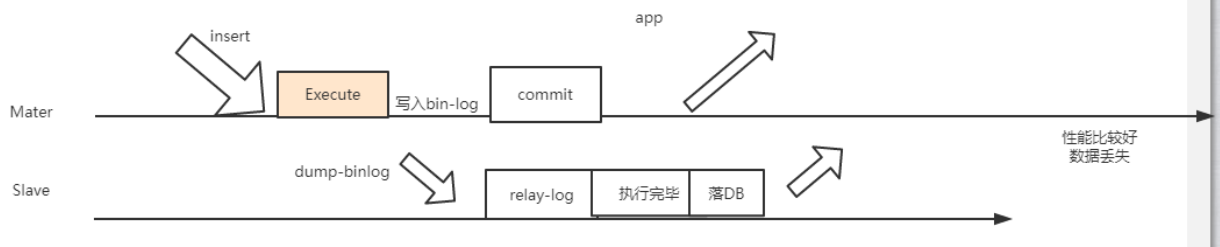
- 1、同步复制(Fully synchronized)

Mysql-Cluster



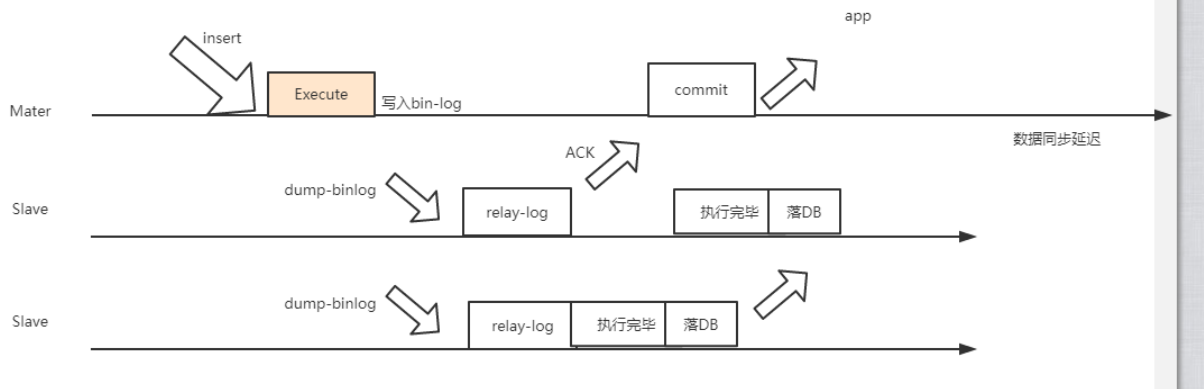
- 2、异步复制 (Asyn) mysql默认的复制方式

异步

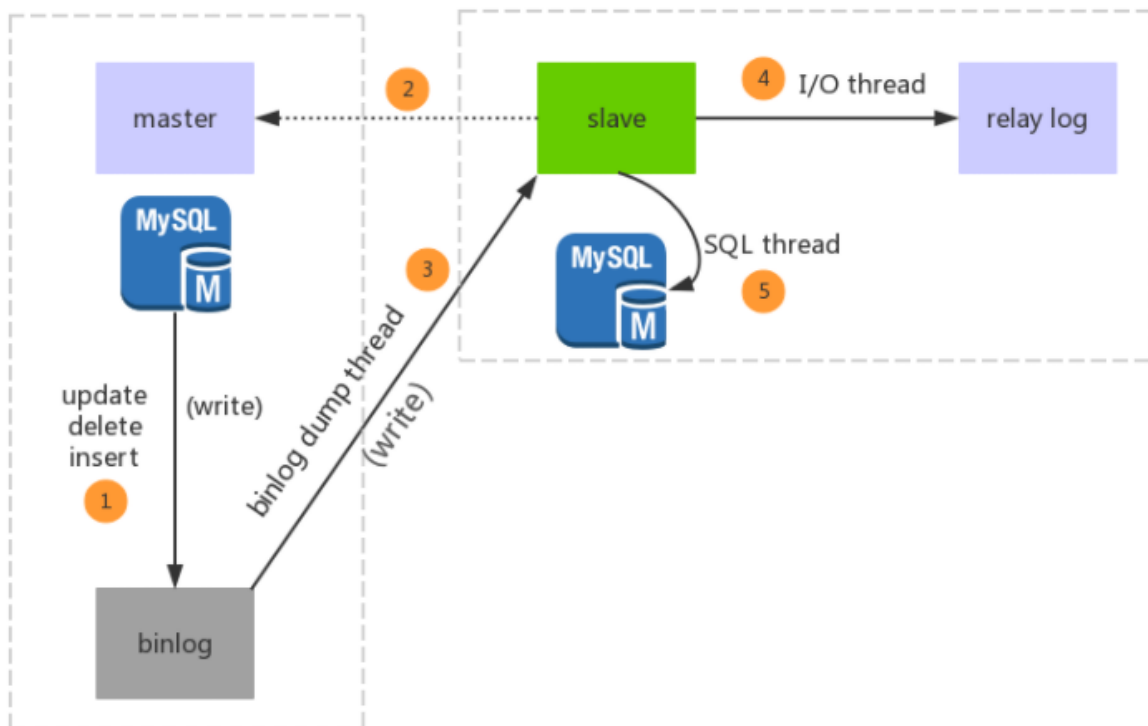


3、半同步复制

半同步



主从复制原理



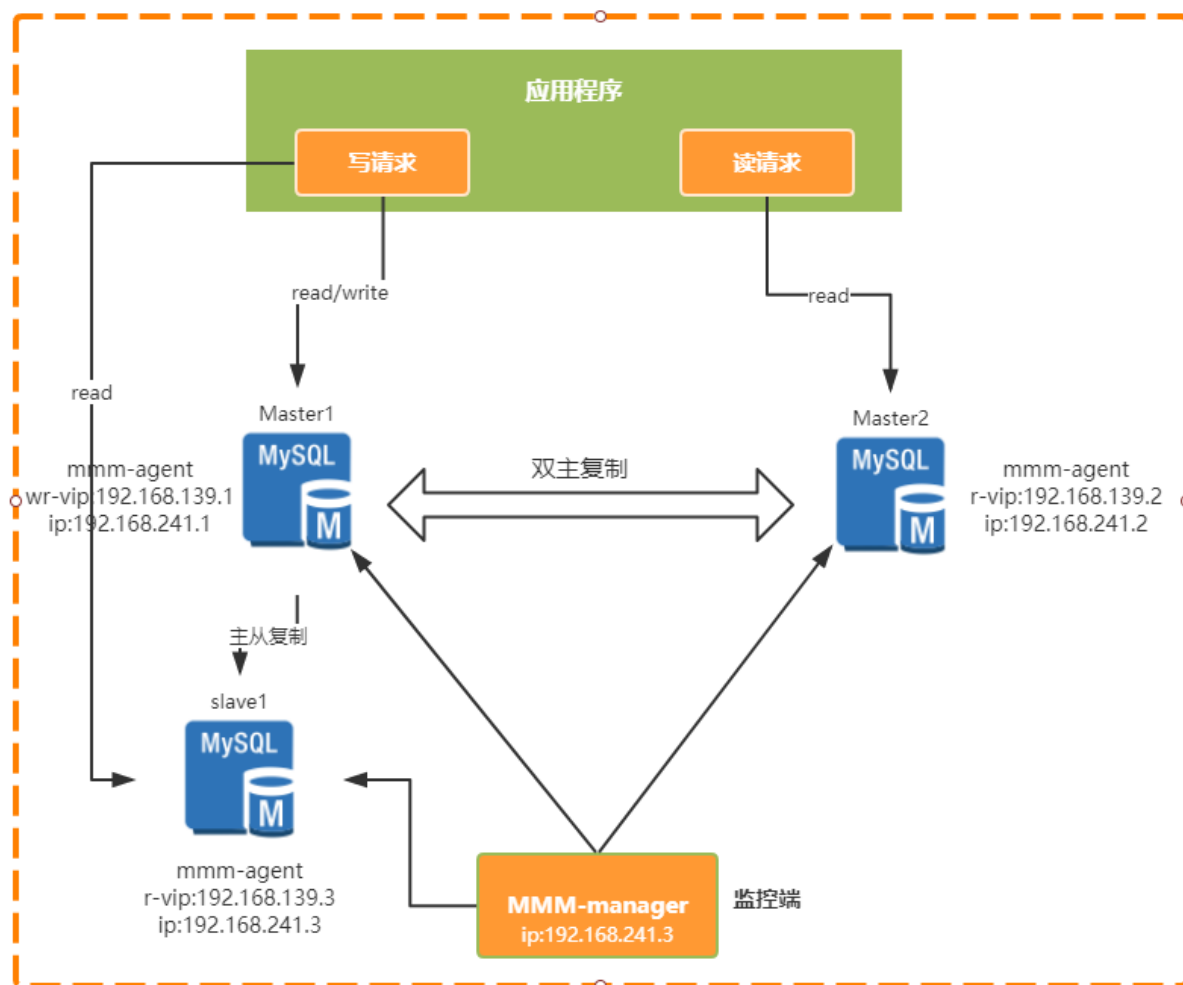
高可用架构方案总结

一、MMM高可用方案

MMM(Master-Master replication managerfor Mysql, Mysql主主复制管理器)是一套灵活的脚本程序，基于perl实现，用来对mysql replication进行监控和故障迁移，并能管

理mysql Master-Master复制的配置(同一时间只有一个节点是可写的)

MMM高可用方案



1.1 优点

- (1) 高可用性，扩展性好，出现故障自动转移，对于主主同步，在同一时间只提供一台数据库写操作，保证数据的一致性。
- (2) 配置简单，容易操作。

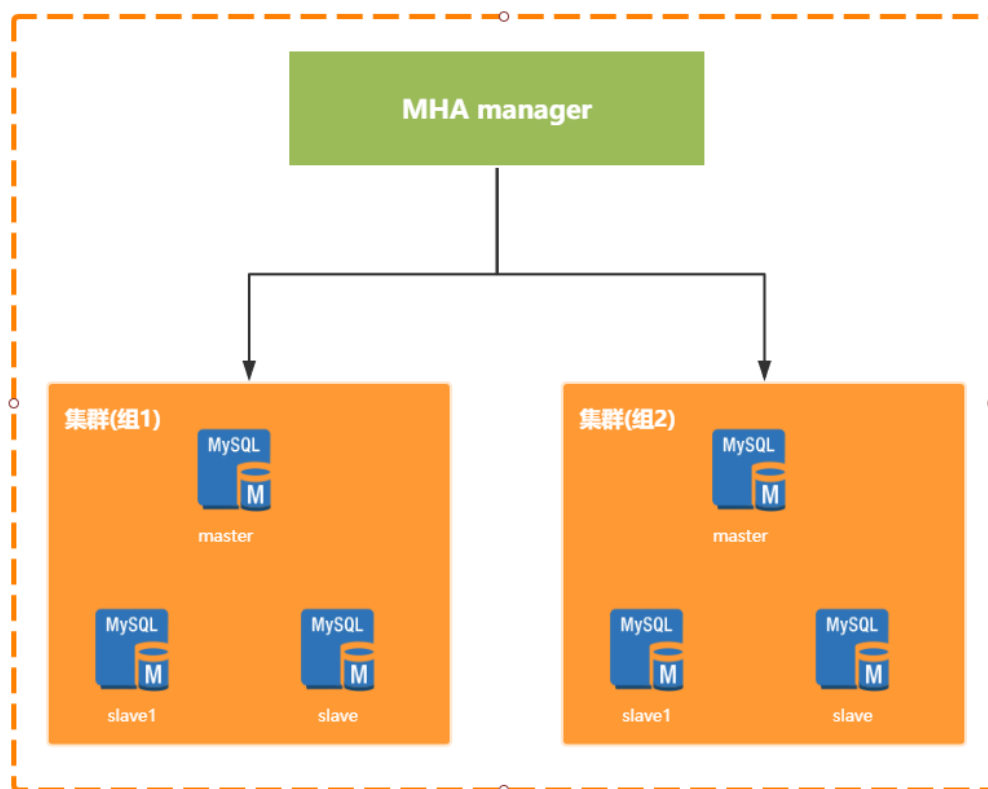
1.2 缺点

- (1) 需要一台备份服务器，浪费资源
- (2) 需要多个虚拟IP
- (3) agent可能意外终止，引起裂脑。

二、MHA方案

MHA服务，有两种角色，MHA Manager(管理节点)和MHA Node(数据节点)。在MySQL故障切换过程中，MHA能做到在0~30秒之内自动完成数据库的故障切换操作，目

前MHA主要支持一主多从的架构，要搭建MHA,要求一个复制集群中必须最少有三台数据库服务器。



2.1 优点

- (1) 不需要备份服务器
- (2) 不改变现有环境
- (3) 操作非常简单
- (4) 可以进行日志的差异修复
- (5) 可以将任意slave提升为master

2.2 缺点

- (1) 需要全部节点做ssh密钥
- (2) MHA出现故障后配置文件会被修改，如果再次故障转移需要重新修改配置文件。
- (3) 自带的脚本还需要进一步补充完善，且用perl开发，二次开发困难。

分库分表

什么是分库分表？

将一个表拆分多张表(库内分表与分库分表)

为什么需要分库分表？

(课前思考)

- 1、微服务：基于业务，需要分拆需要分表
- 2、读写分离
- 3、建索引，优化查询
- 4、换数据库oracle, sybase, db2

提升性能，任然没有办法解决

能不分尽量不分？

不要过度拆分

有哪些拆分方式？

横向拆分

数据量庞大的表：3000万条（B+树）- 树高过高，IO次数增加

拆成3张表结构一模一样的表：1000万条

表可以分区：可以解决数据量大的问题！?高并发场景-

磁盘读写能力：300M/S

t_order->t_order_0,t_order_1

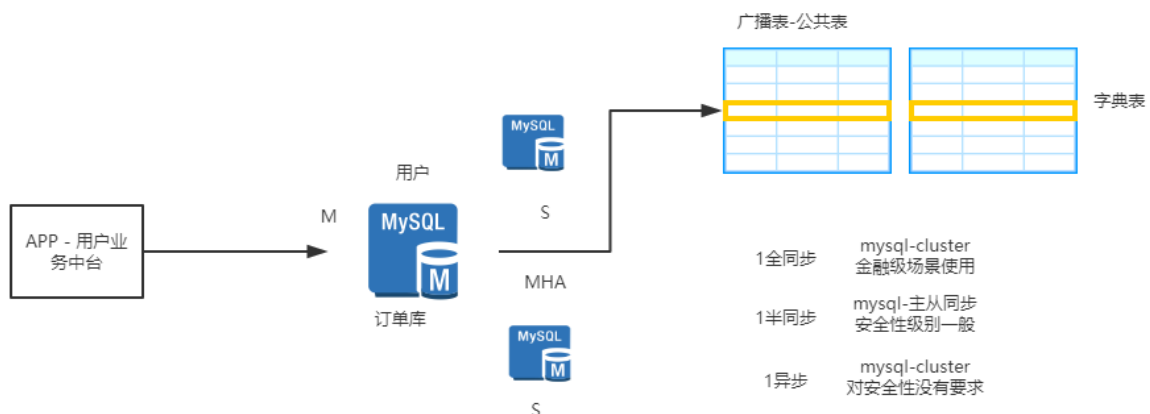
表主键怎么定义？

表怎么路由？

纵向拆分

拆库

根据业务拆！



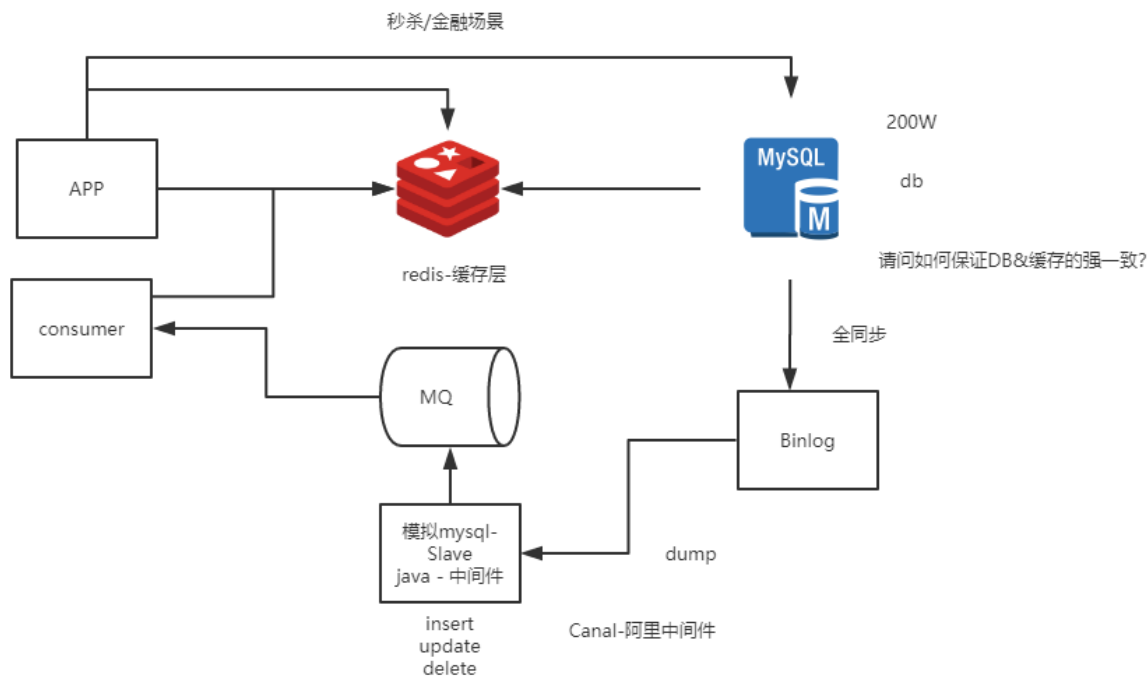
拆分后产生的问题

- 1、扩库jion
- 2、主键重复问题
- 3、分页查询
- 4、关联查询

.....

- shardingsphere (京东数科) 在apache孵化
- Mycat (阿里巴巴-基于cobar) 不是阿里的
- Tddl
- Atlas (奇虎360)
- 还有哪些你熟知的?

请问如何保证DB&缓存的最终强一致性?



在实际开发中，我们经常会遇到一些需求，在数据库操作成功后，需要进行一些其他操作，如：发送一条消息到MQ中、更新缓存或者更新搜索引擎中的索引等。

遇到这种看似无解的问题，最好的办法是换一种思路去解决它：不要同时去更新数据库和其他组件，只是简单的更新数据库即可。

如果数据库操作成功，必然会产生binlog。之后，我们通过一个组件，来模拟的mysql的slave，拉取并解析binlog中的信息。通过解析binlog的信息，去异步的更新缓存、索引或者发送MQ消息，保证数据库与其他组件中数据的最终一致。