

架构实战营 - 模块4

第1课：数据库存储架构

一手微信study322 价格更优惠
有正版课找我 高价回收帮回血

李运华

前阿里资深技术专家(P9)

教学目标

1. 掌握数据库读写分离架构
2. 掌握数据库分库分表架构
3. 掌握数据库分布式事务的算法

一手微信study322 价格更优惠
有正版课找我 高价回收帮回血



站在巨人的肩膀上，看的更远！

目录

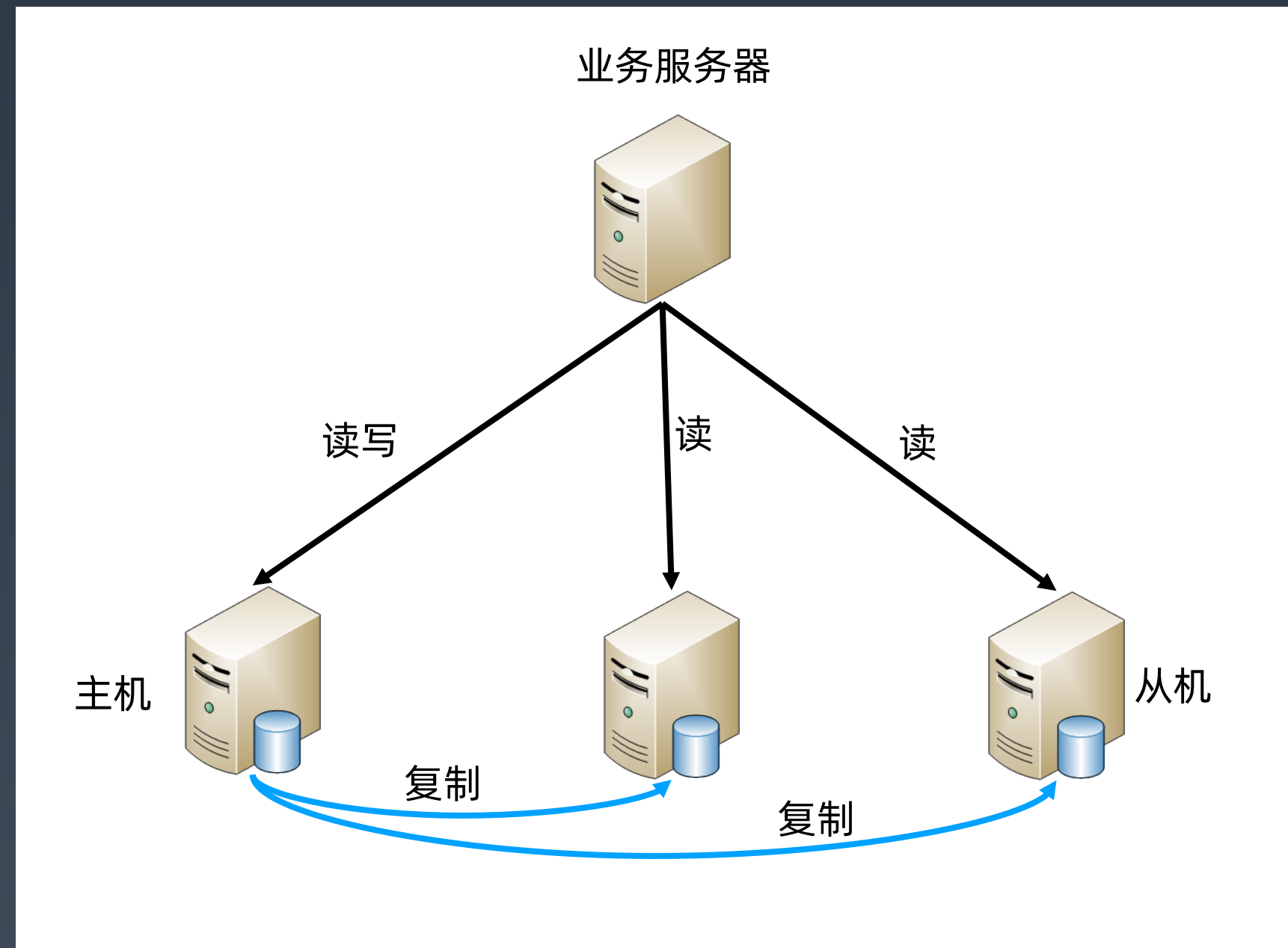
1. 数据库读写分离
2. 数据库分库分表
3. 数据库分布式事务算法

一手微信study322 价格更优惠
有正版课找我 高价回收帮回血

数据库读写分离

手微信study222 价格更优惠
有正版课找我 高价回收帮回血

数据库读写分离



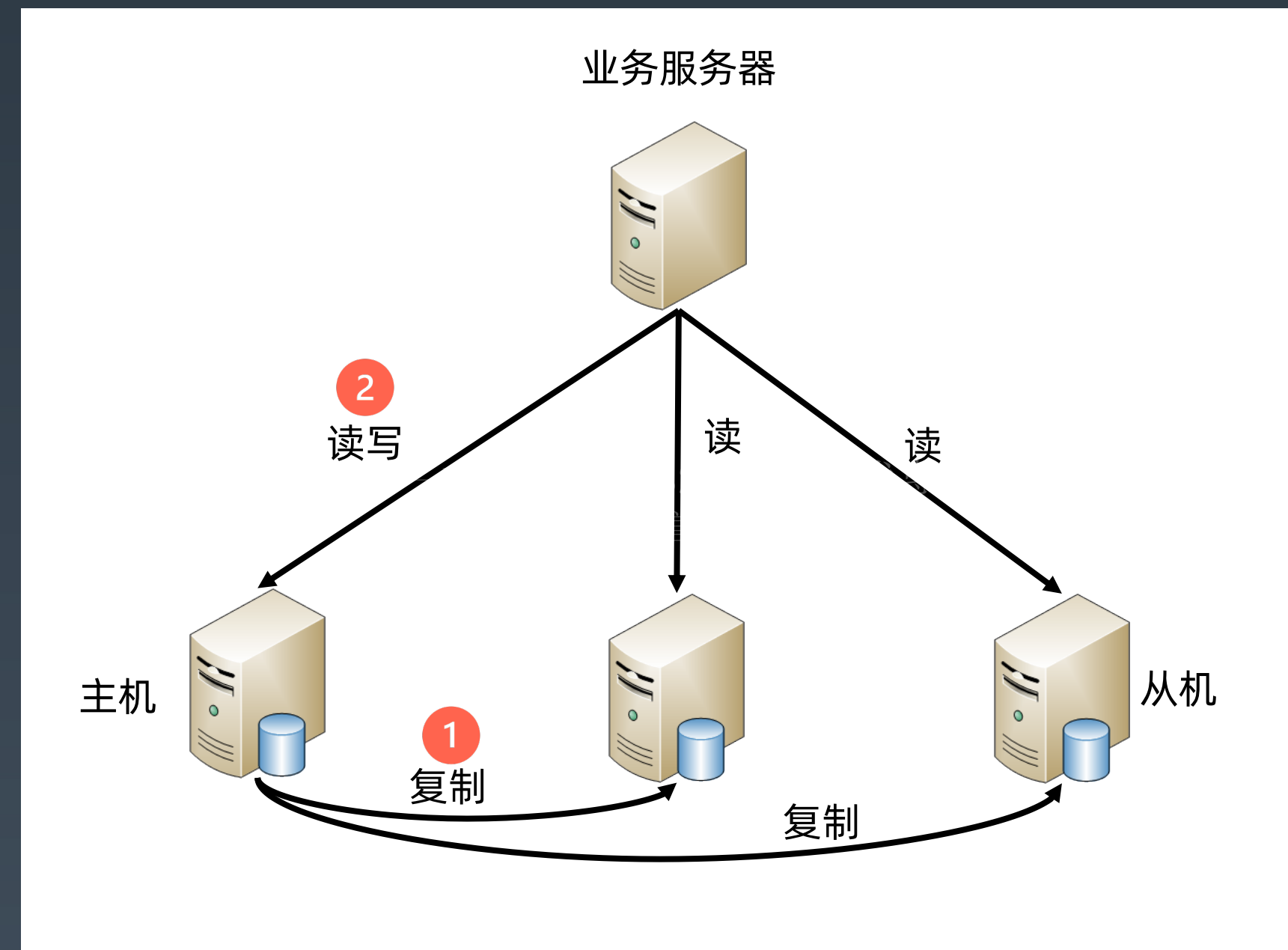
【实现原理】

1. 数据库服务器搭建主从集群，一主一从、一主多从都可以。
2. 数据库主机负责读写操作，从机只负责读操作。
3. 数据库主机通过复制将数据同步到从机，每台数据库服务器都存储了所有的业务数据。
3. 业务服务器将写操作发给数据库主机，将读操作发给数据库从机。

【如何判断要读写分离？】

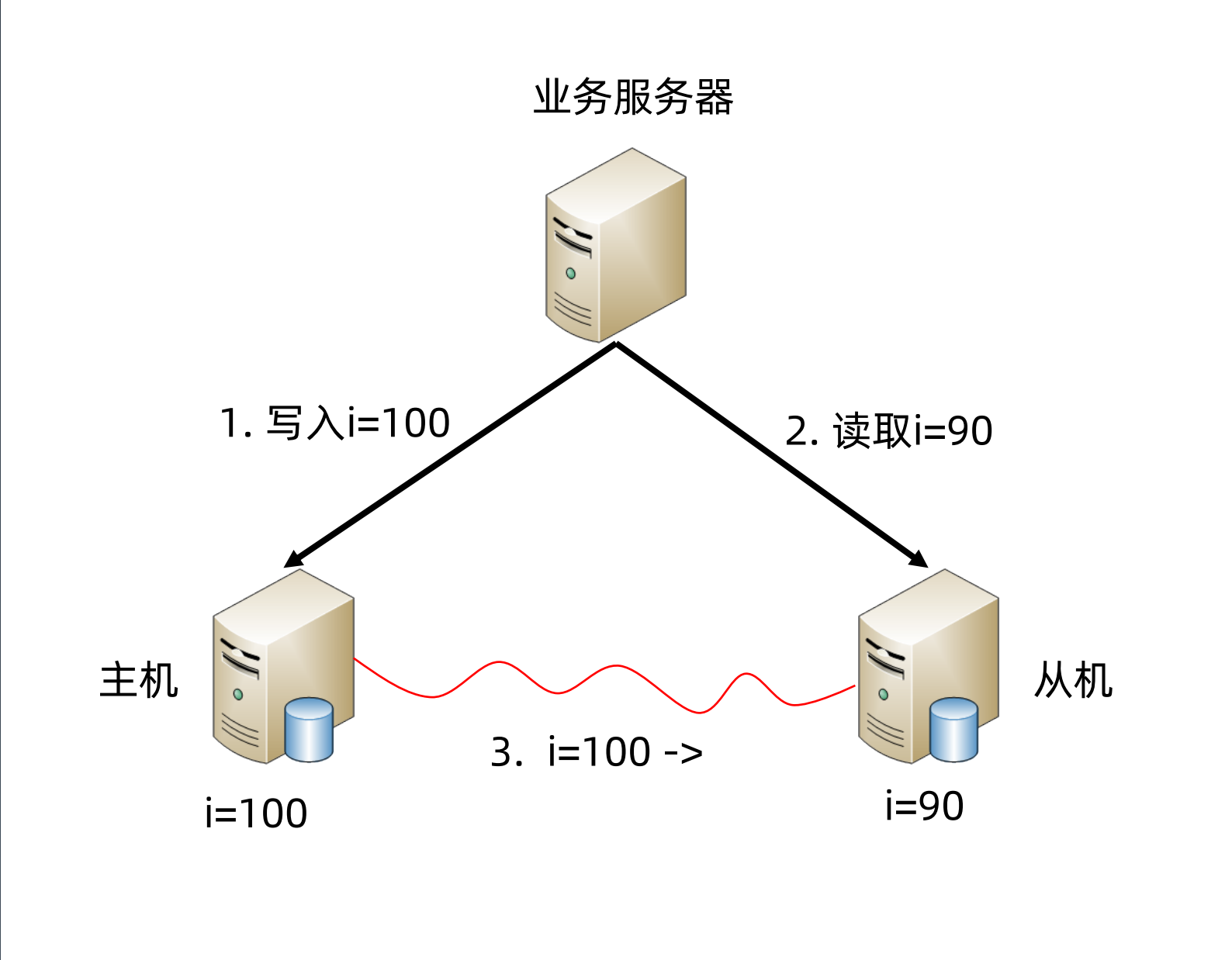
1. 业务量持续增长
2. 先优化（优化索引，加入缓存）再重构

数据库读写分离复杂度分析



1. 复制延迟; 2. 任务分解;

数据库读写分离复制延迟



【应对方法1 - 读写绑定】

写操作后的读操作指定发给数据库主服务器。

【缺点】

业务侵入很大，容易留坑

【应对方法2 - 二次读取】

读从机失败后再读一次主机

【缺点】

如果有很多二次读取，将大大增加主机的读操作压力

【应对方法3 - 业务分级】

关键业务读写操作全部指向主机，非关键业务采用读写分离

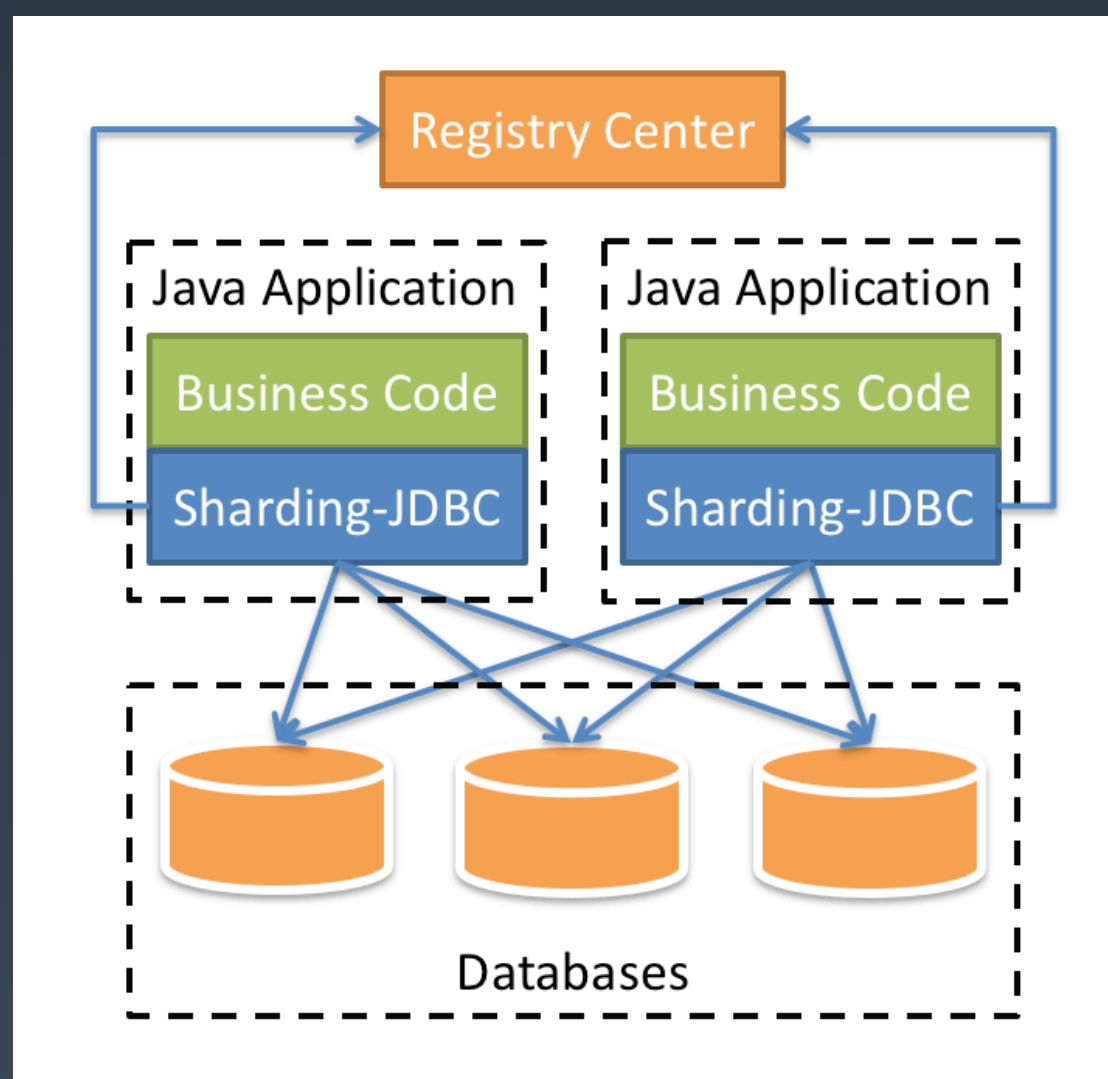
【缺点】

编码人员容易全部采用读写主机



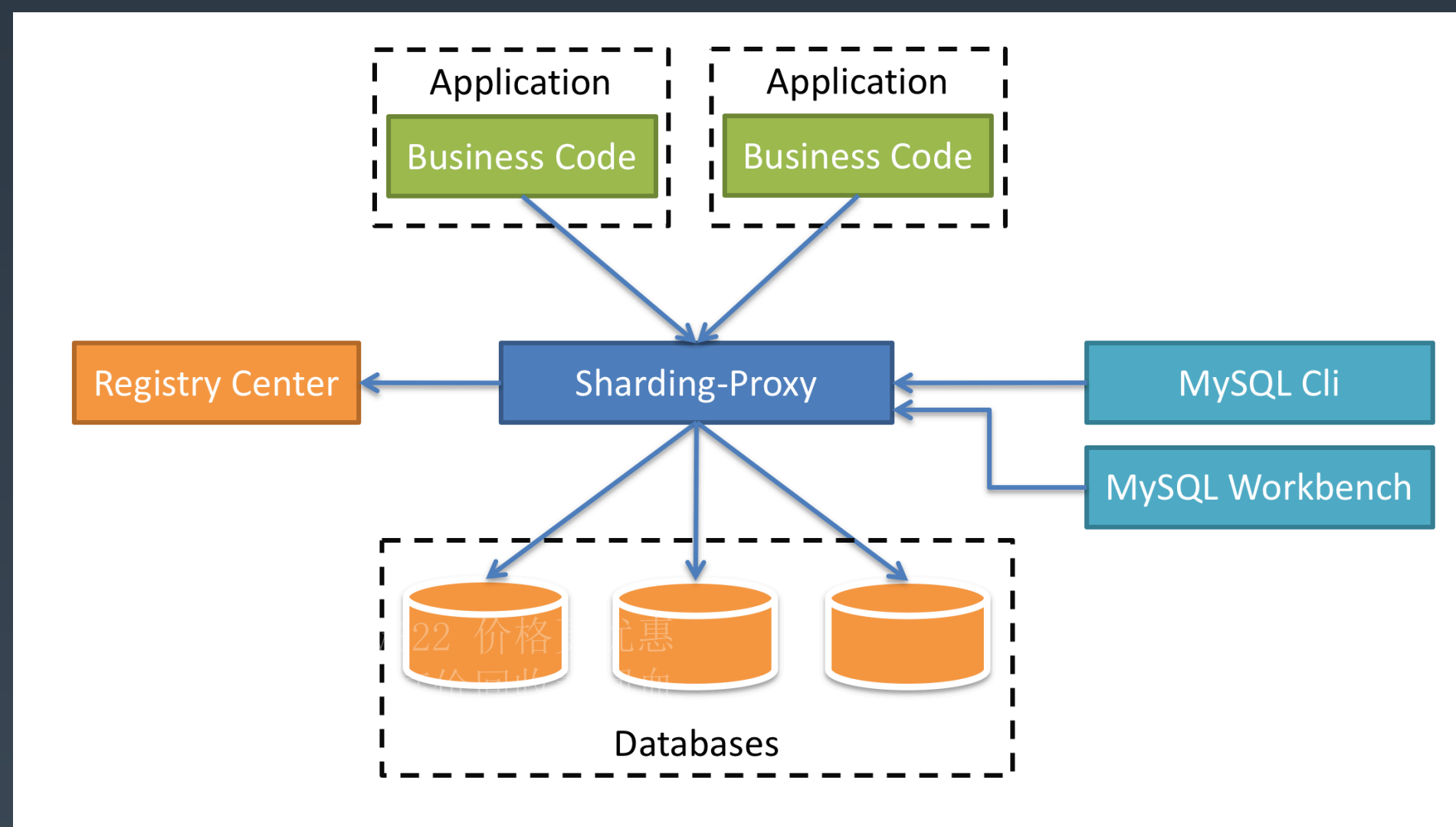
你觉得应该优选哪种应对方法？

数据库读写分离任务分解



程序代码封装模式

1. 实现简单，基于 JDBC 封装
2. 维护简单，无部署，**无需考虑高性能高可用**
3. 每个语言都要实现一遍



中间件封装模式

1. 实现复杂，独立服务器，需要实现连接管理
2. 维护复杂，需要独立部署，并且要考虑集群部署支持高性能高可用
3. 跨语言

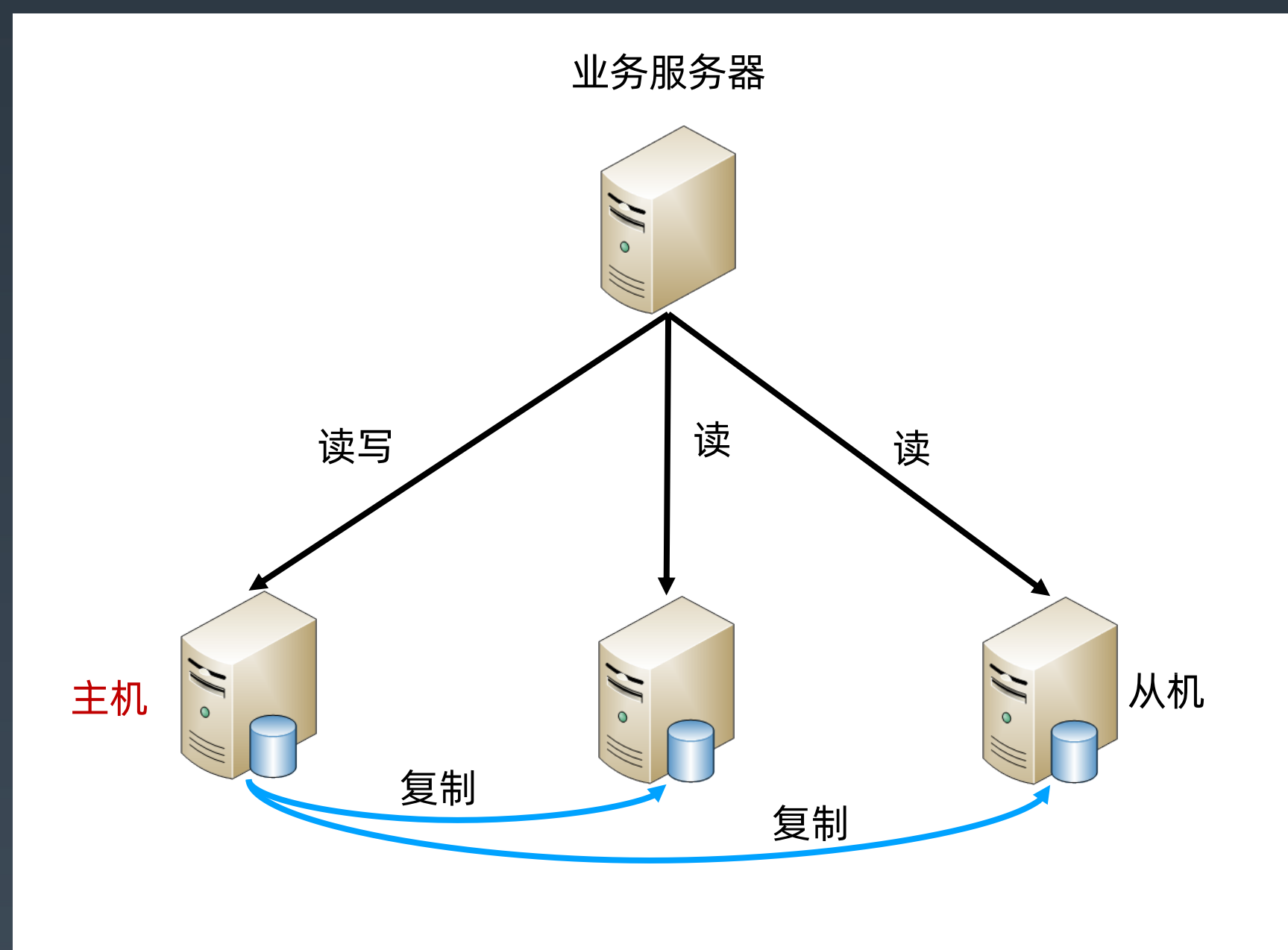


为何程序代码封装无需考虑高性能高可用部署？

数据库分库分表

手微信study322 价格更优惠
有正版课找我 高价回收帮回血

数据库分库分表



主机节点写入性能瓶颈



study322 价格更优惠
找我 高价回收帮回血

数据库分库分表

数据分库

数据分表

加机器（叠加法则）

数据分库



数据库集群



用户数据



商品数据



订单数据

【Join问题】

原本在同一个数据库中的表分散到不同数据库中，导致无法使用 SQL 的 Join 查询

【解法】

1. 小表冗余

将一些小表在每个库中冗余一份，例如字典表

2. 代码 Join

在代码里面实现 Join 功能

3. 字段冗余

例如订单表直接记录商品类型

【事务问题】

表分散到不同的数据库中，无法通过事务统一修改

【解法】

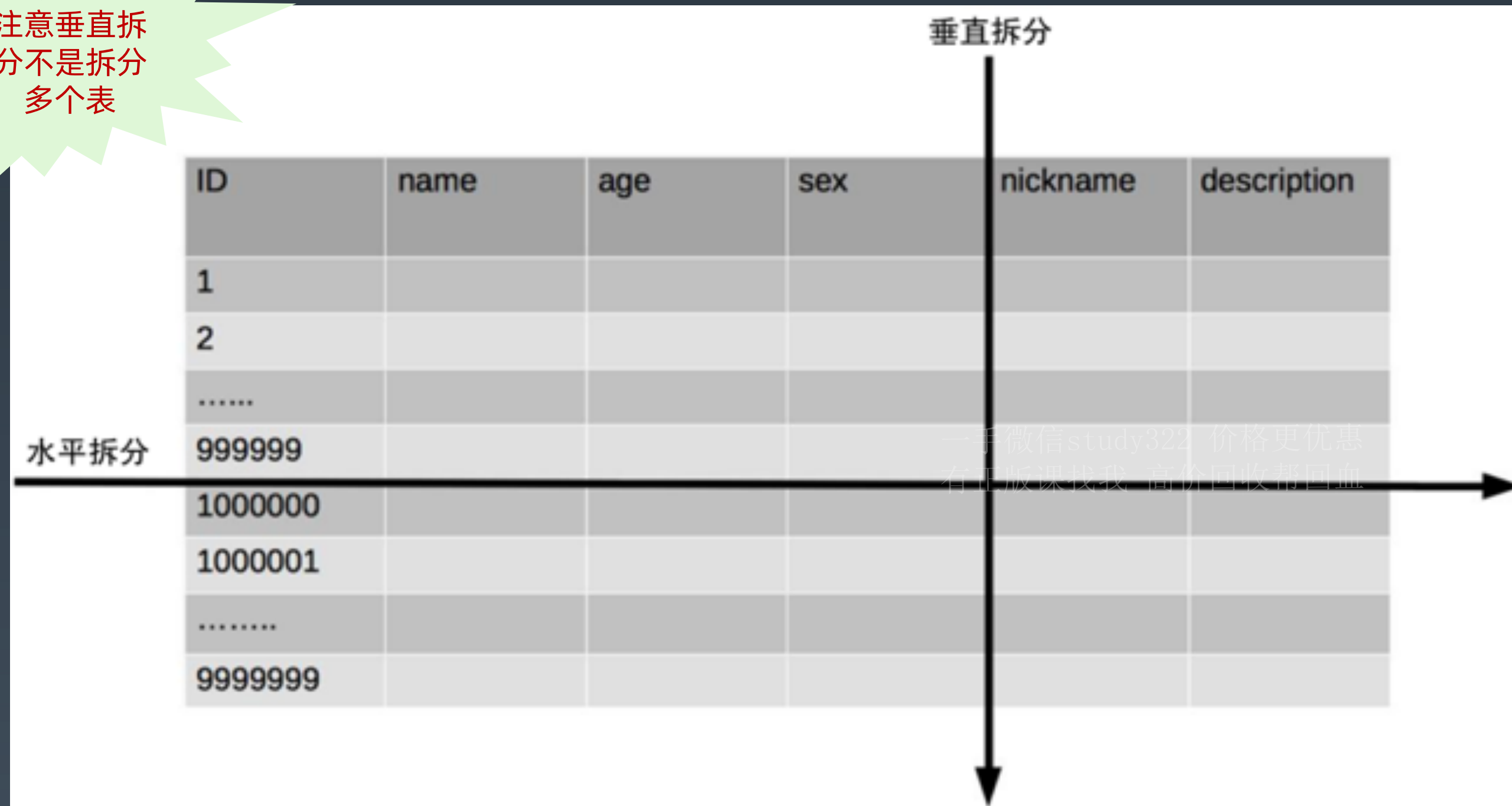
分布式事务

【案例】

先扣商品库存，扣成功后生成订单，如果因为订单数据库异常导致生成订单失败，业务程序又需要将商品库存加上；而如果因为业务程序自己异常导致生成订单失败，则商品库存就无法恢复了，需要人工通过日志等方式来手工修复库存异常

数据分表

注意垂直拆分不是拆分多个表

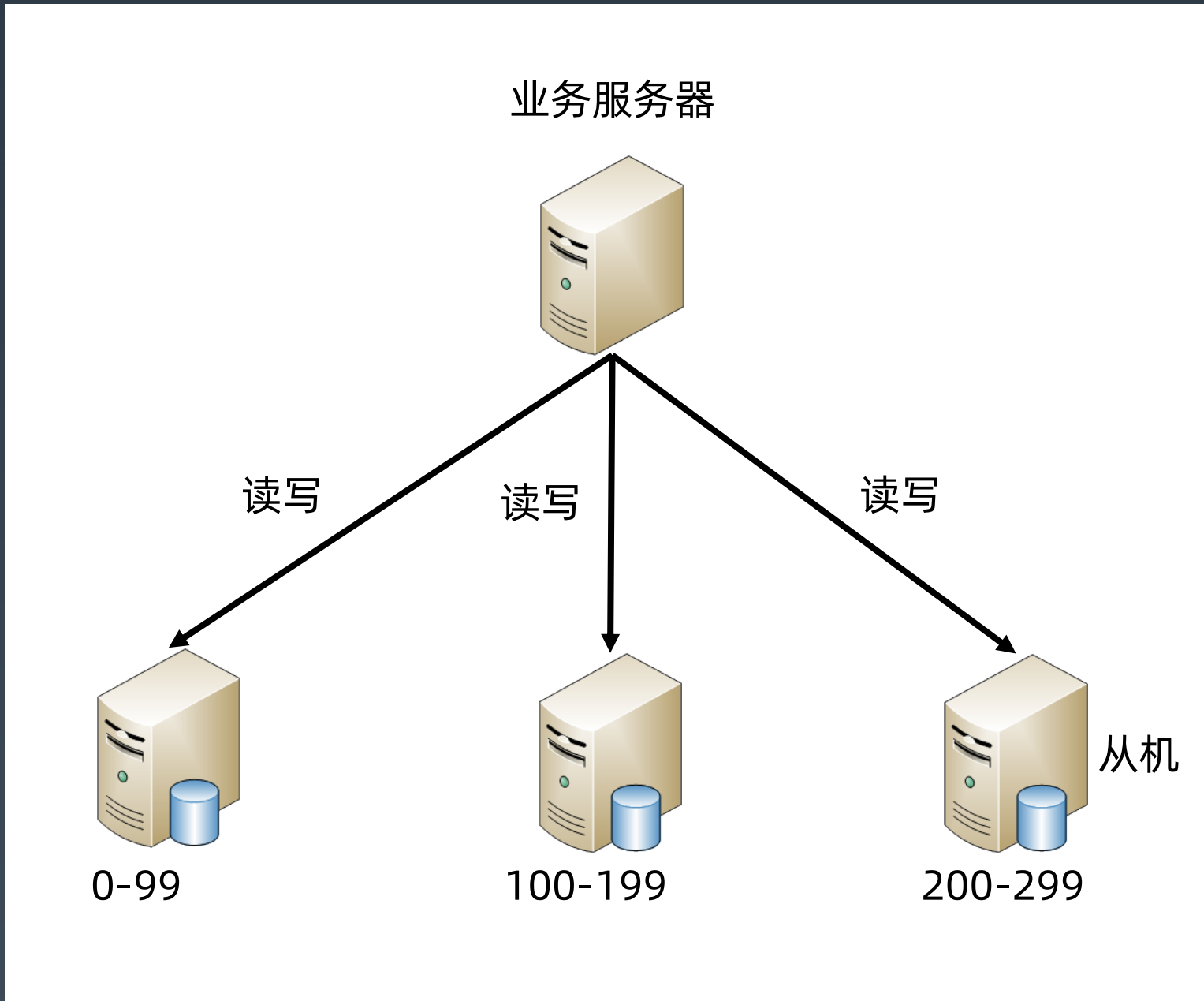


垂直拆分：按列拆分，优化单机处理性能；
水平拆分：按行拆分，提升系统处理性能

多大的表需要拆分？

1. B+tree的层数：3层大约是2000万条
2. Innodb buffer pool：2000万数据，每条数据100字节，单表就2G了
3. 数据量持续增长的表

水平分表复杂度和应对方法

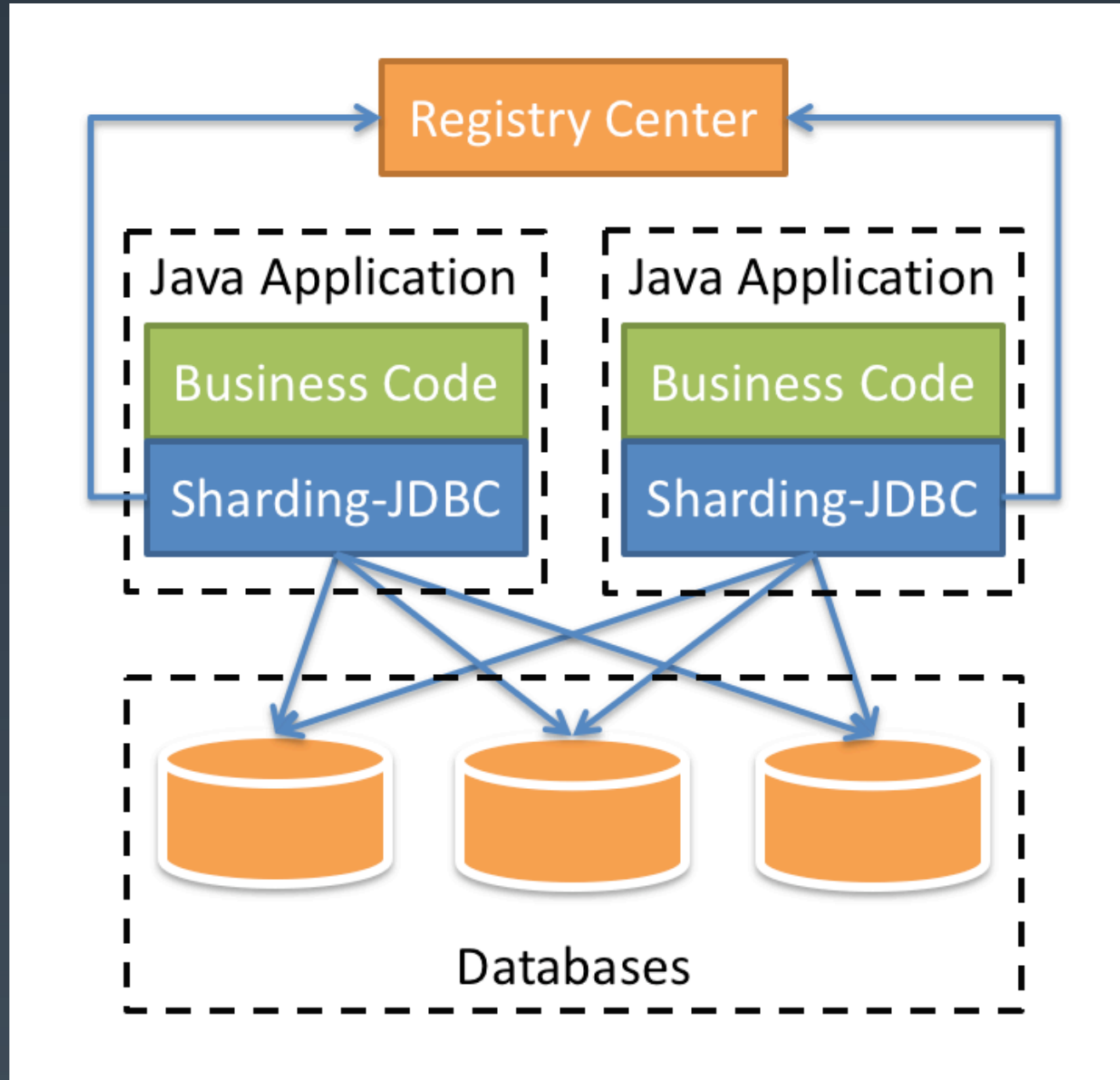


【路由问题】
路由算法进行计算数据归属。

【count 操作】
如何计算 count

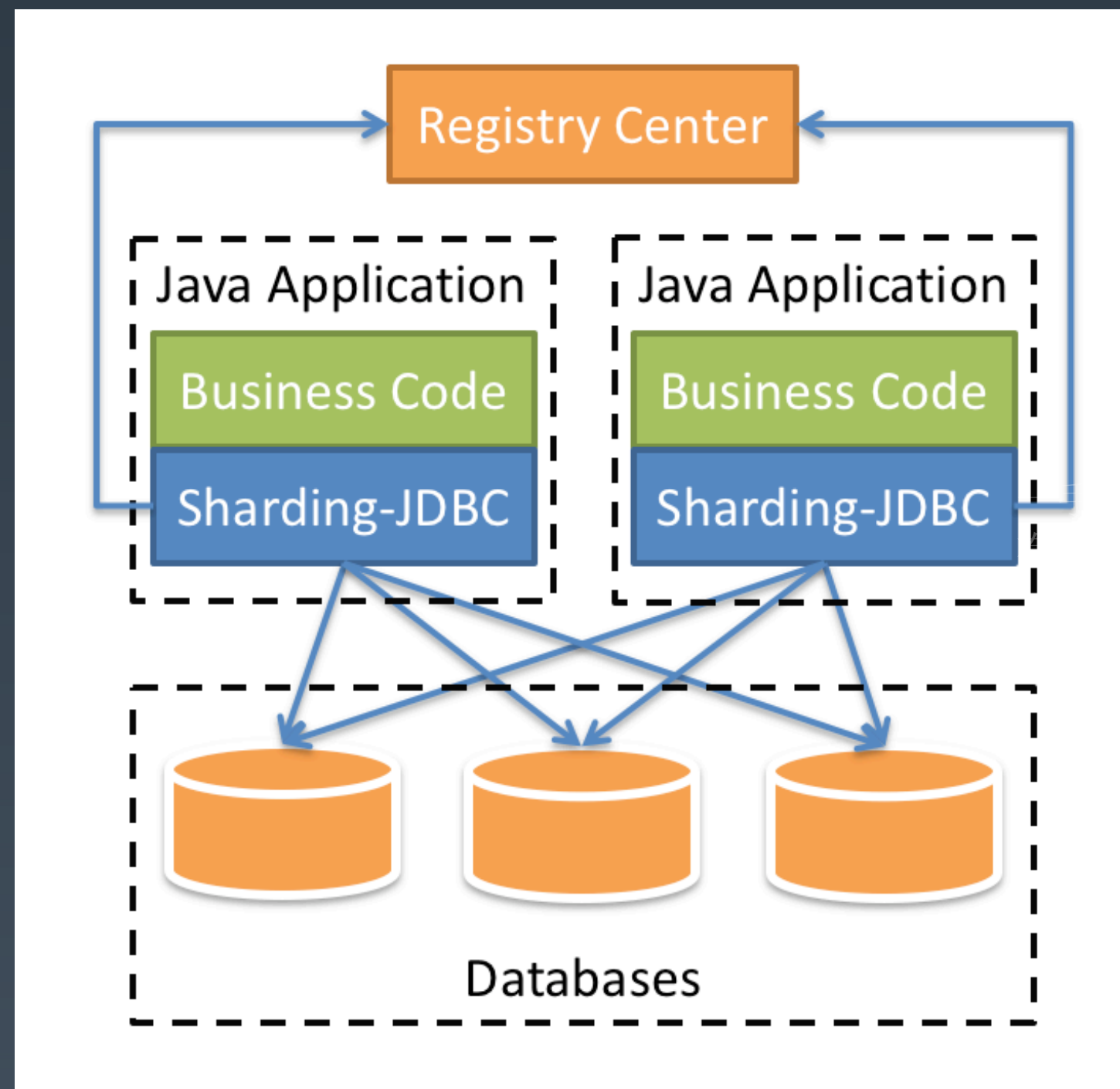
【join 操作】
如何执行与其它表 join

【order by】
如何执行 order by



水平分表能够通过加服务器来不断提升性能么？

水平分表伸缩瓶颈



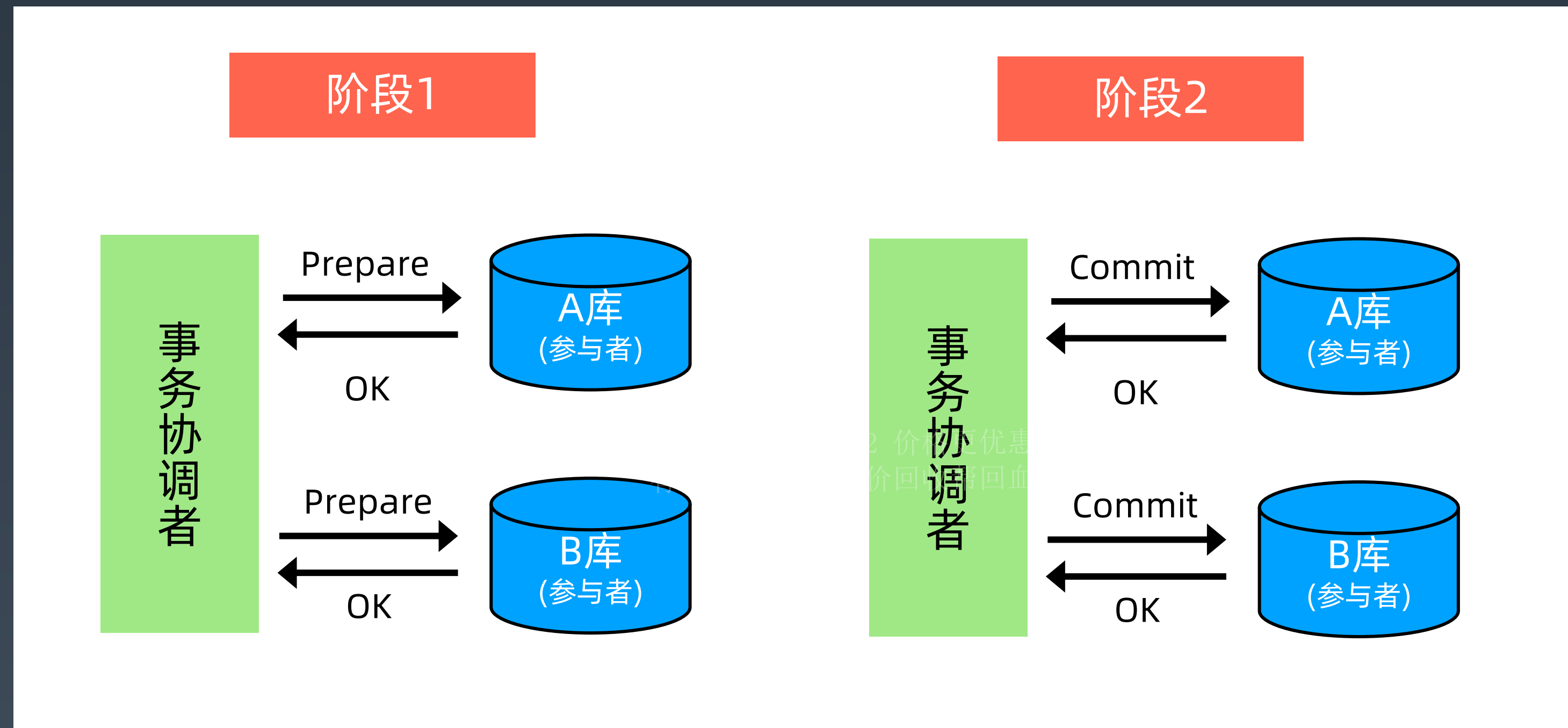
每个应用都需要连接所有的分片，当应用数量增多后，数据库连接会逐渐成为瓶颈。

以 MySQL 为例，默认100连接，实测50~100连接性能最高，超过200后会显著下降

数据库分布式事务

一手微信study322 价格更优惠
有正版课找我 高价回收帮回血

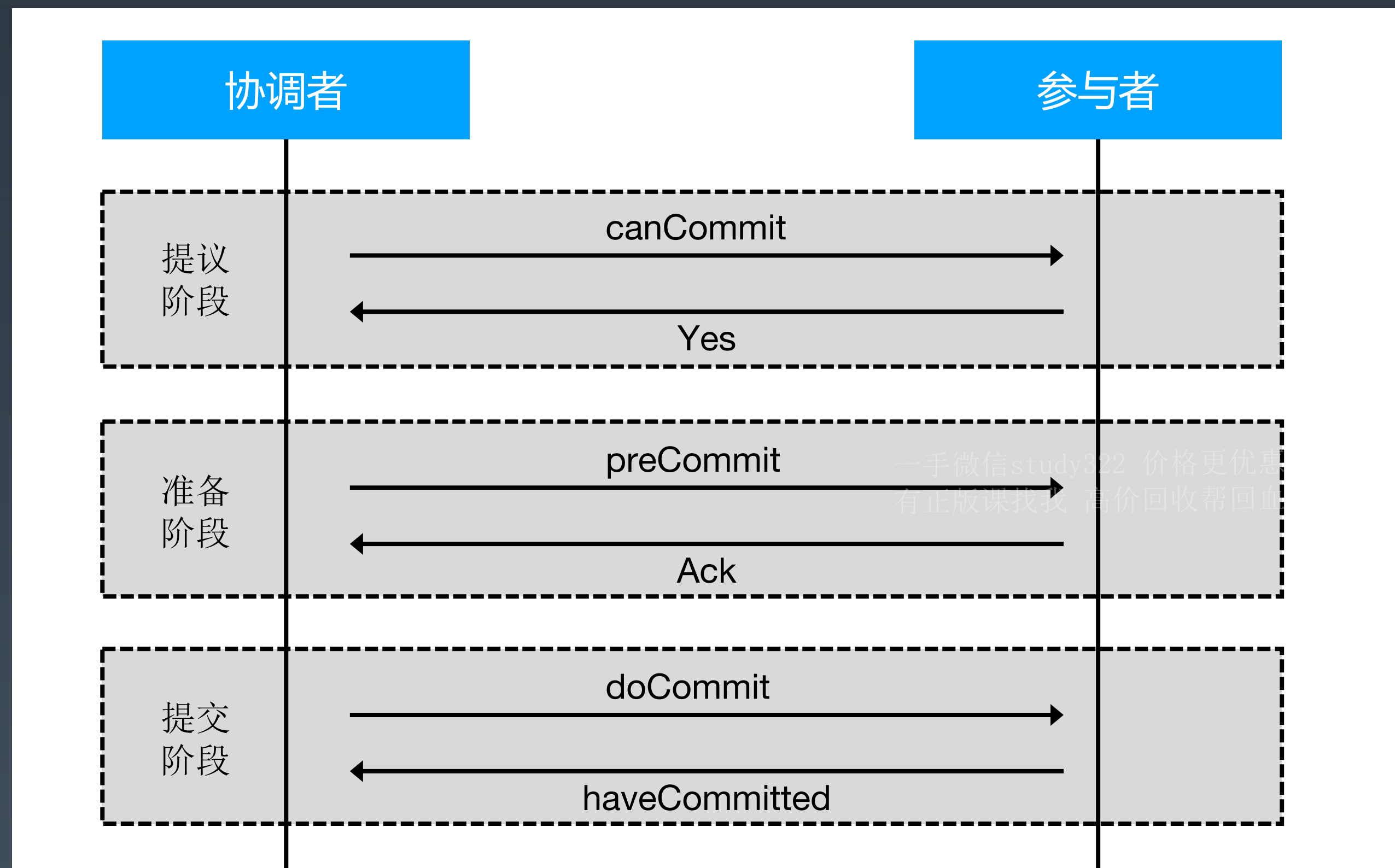
分布式事务算法 - 2PC



阶段1：单个参与者故障会导致整体事务失败

阶段2：事务协调者故障会导致整体状态不一致，参与者一直等待事务协调者指令，可能需要人工修复

分布式事务算法 - 3PC



1. Yes 丢了会导致协调者取消事务
2. 部分 preCommit 丢了会导致部分参与者超时取消事务，其它参与者收到了 preCommit 消息，然后超时提交事务，因此造成脑裂
3. Ack 丢了会导致参与者超时提交事务
4. doCommit 丢失会导致参与者超时提交事务



2PC和3PC哪个更好，应用更广？

MySQL XA

```
<?PHP
$dbtest1 = new mysqli("172.20.101.17","public","public","dbtest1")or die("dbtest1 连接失败");
$dbtest2    = new mysqli("172.20.101.18","public","public","dbtest2")or die("dbtest2 连接失败");

//为XA事务指定一个id, xid 必须是一个唯一值。
$xid = uniqid("");

//两个库指定同一个事务id, 表明这两个库的操作处于同一事务中
$dbtest1->query("XA START '$xid'");//准备事务1
$dbtest2->query("XA START '$xid'");//准备事务2

try {
    // $dbtest1
    $return = $dbtest1->query("UPDATE member SET name='唐大麦' WHERE id=1") ;
    if($return == false) {
        throw new Exception("库dbtest1@172.20.101.17执行update member操作失败! ");
    }

    // $dbtest2
    $return = $dbtest2->query("UPDATE memberpoints SET point=point+10 WHERE memberid=1") ;
    if($return == false) {
        throw new Exception("库dbtest1@172.20.101.18执行update memberpoints操作失败! ");
    }

    //阶段1: $dbtest1提交准备就绪
    $dbtest1->query("XA END '$xid'");
    $dbtest1->query("XA PREPARE '$xid'");
    //阶段1: $dbtest2提交准备就绪
    $dbtest2->query("XA END '$xid'");
    $dbtest2->query("XA PREPARE '$xid'");

    //阶段2: 提交两个库
    $dbtest1->query("XA COMMIT '$xid'");
    $dbtest2->query("XA COMMIT '$xid'");
}
catch (Exception $e) {
    //阶段2: 回滚
    $dbtest1->query("XA ROLLBACK '$xid'");
    $dbtest2->query("XA ROLLBACK '$xid'");
    die($e->getMessage());
}

$dbtest1->close();
$dbtest2->close();

?>
```

【外部XA】

跨多MySQL实例的分布式事务，由应用程序代码作为2PC的事务协调者

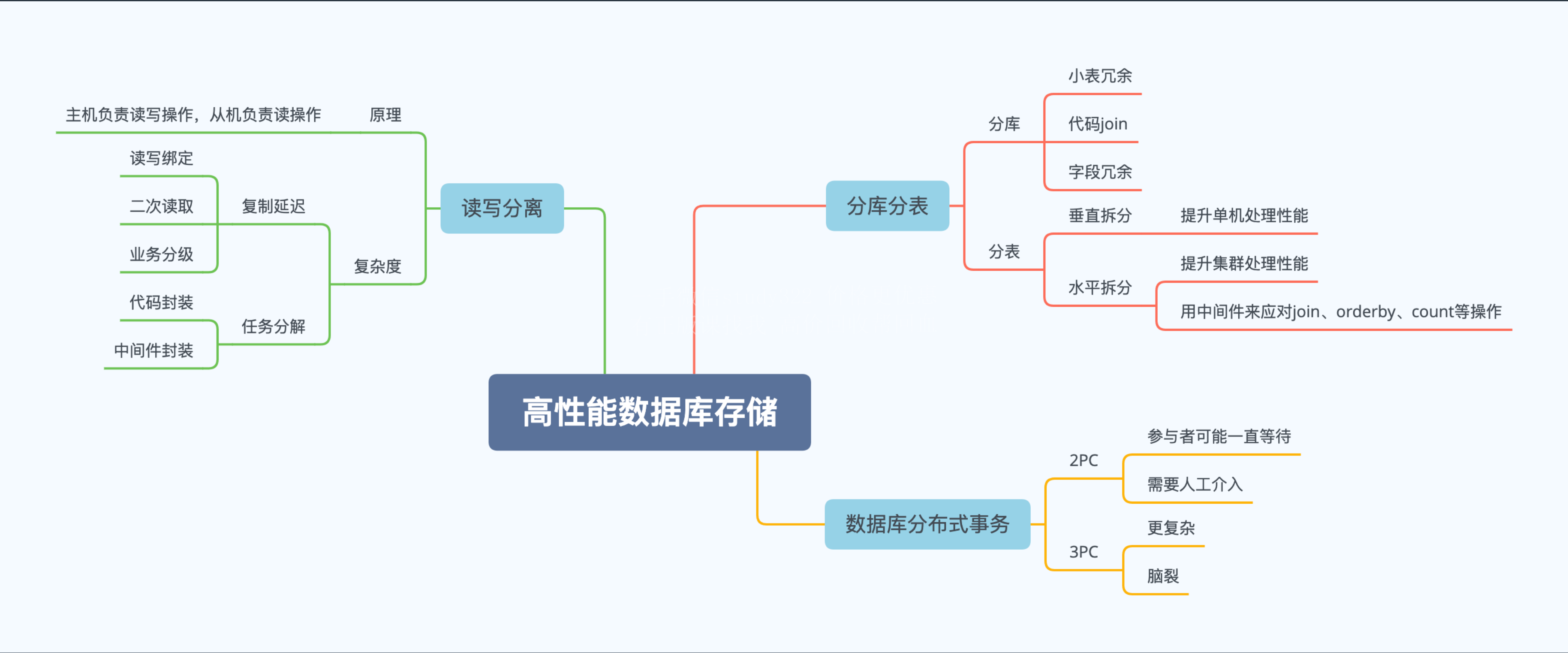
【内部XA】

跨同一实例多存储引擎的事务，由Binlog作为2PC的事务协调者

详细参考：

<https://blog.csdn.net/soonfly/article/details/70677138>

本节思维导图



随堂测验

【判断题】

1. 读写分离主要是提升系统读性能，对写性能没有帮助
2. 数据分库和数据分表都可以提升系统写性能
3. 水平分表后，尽量用中间件来实现各种 SQL 操作
4. 3PC 比 2PC 功能更强大，应该优先采取 3PC
5. 2PC 中的事务协调者可以是代码，也可以是独立的系统

加微信 study322 价格更优惠
有正版课找我 高价回收帮回血

【思考题】

为什么数据库系统自己不实现分库分表的功能，而 Redis、MongoDB 等却都提供 sharding 功能？

Q&A



茶歇时间



八卦，趣闻，内幕.....

THANKS

一手微信study322 价格更优惠
有正版课找我 高价回收帮回血