

# 架构实战营 - 模块4

## 第2课：存储架构模式 - 复制架构

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. 熟悉高可用的关键指标
2. 掌握常见的存储复制架构和优缺点

一手微信study322 价格更优惠  
有正版课找我 高价回收帮回血



万变不离其宗！

# 目录

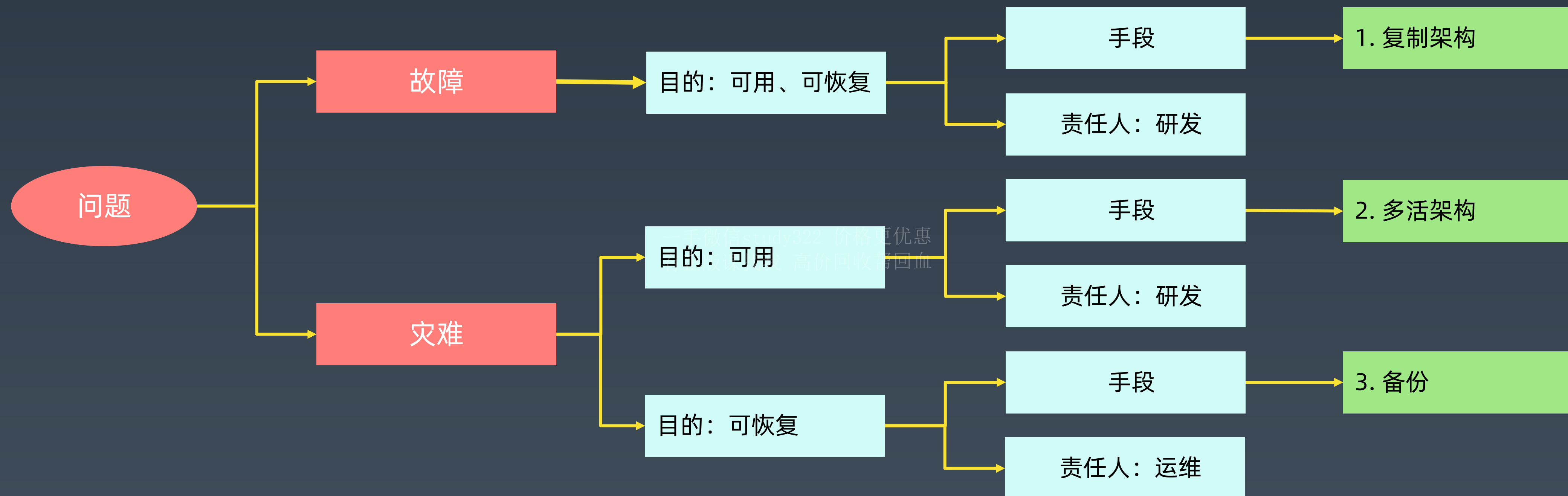
- 1. 高可用关键指标
- 2. 主备&主从架构
- 3. 双机切换架构
- 4. 集群选举架构

一手微信study322 价格更优惠  
盗版课找我 高价回收帮回血

# 高可用关键指标

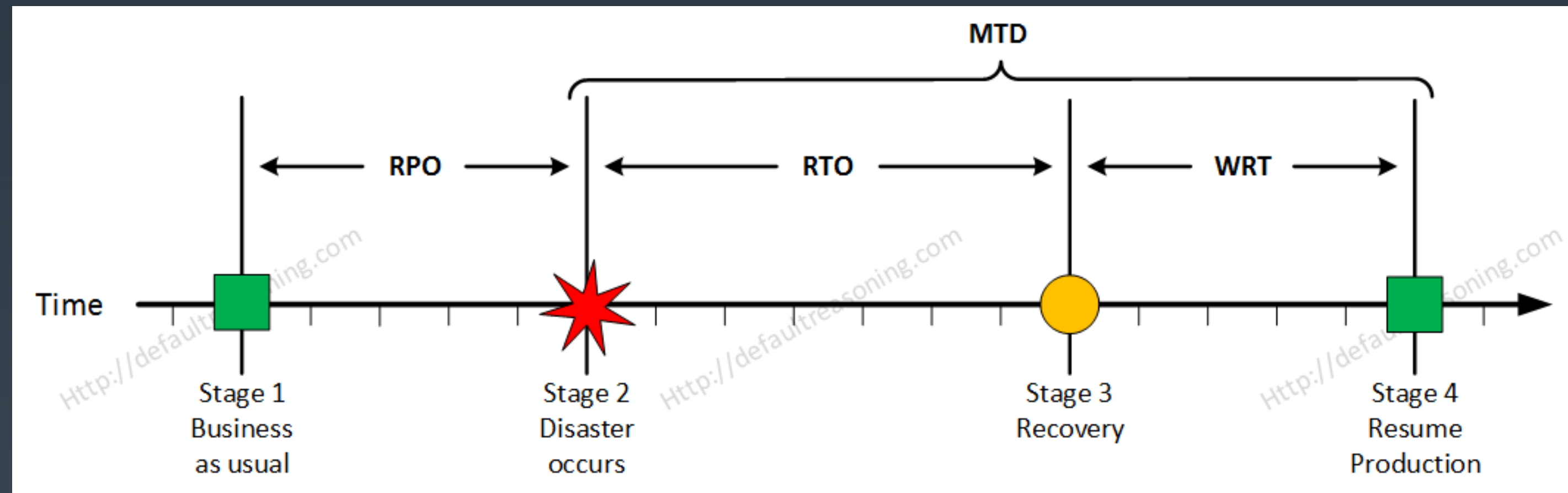
一手微信study322 价格更优惠  
有正版课找我 高价回收帮回血

# 存储类问题处理框架图



直接上多活架构就好了，还要复制架构干啥呢？

# 高可用存储几个核心指标



高可用计算  
不涉及哪个  
指标？

## 【RPO】

Recovery Point Objective, 恢复点目标, 指“最大可接受的数据损失”, 因为数据备份和复制都是有时间限制的, 不可能做到绝对实时。

## 【RTO】

Recovery Time Objective, 恢复时间目标, 指“最大可接受的系统恢复所需时间”, 因为定位、处理、恢复需要时间。

## 【WRT】

Work Recovery Time, 工作恢复时间, 指“系统恢复正常后, 恢复业务所需的时间”, 因为要进行各种业务检查、校验、修复。

## 【MTD】

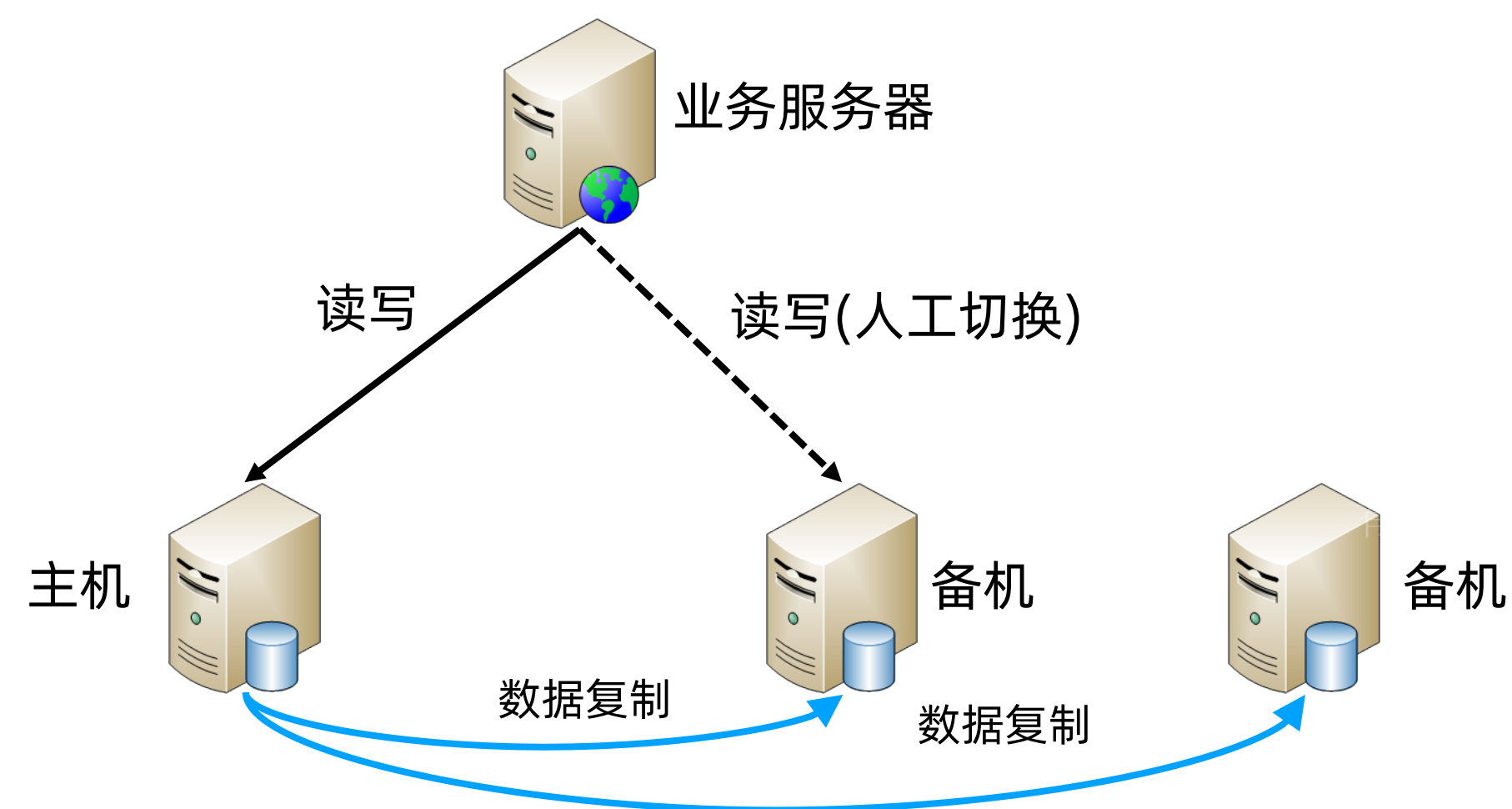
Maximum Tolerable Downtime, 最大可容忍宕机时间, 等于  $RTO + WRT$

# 主备&主从架构

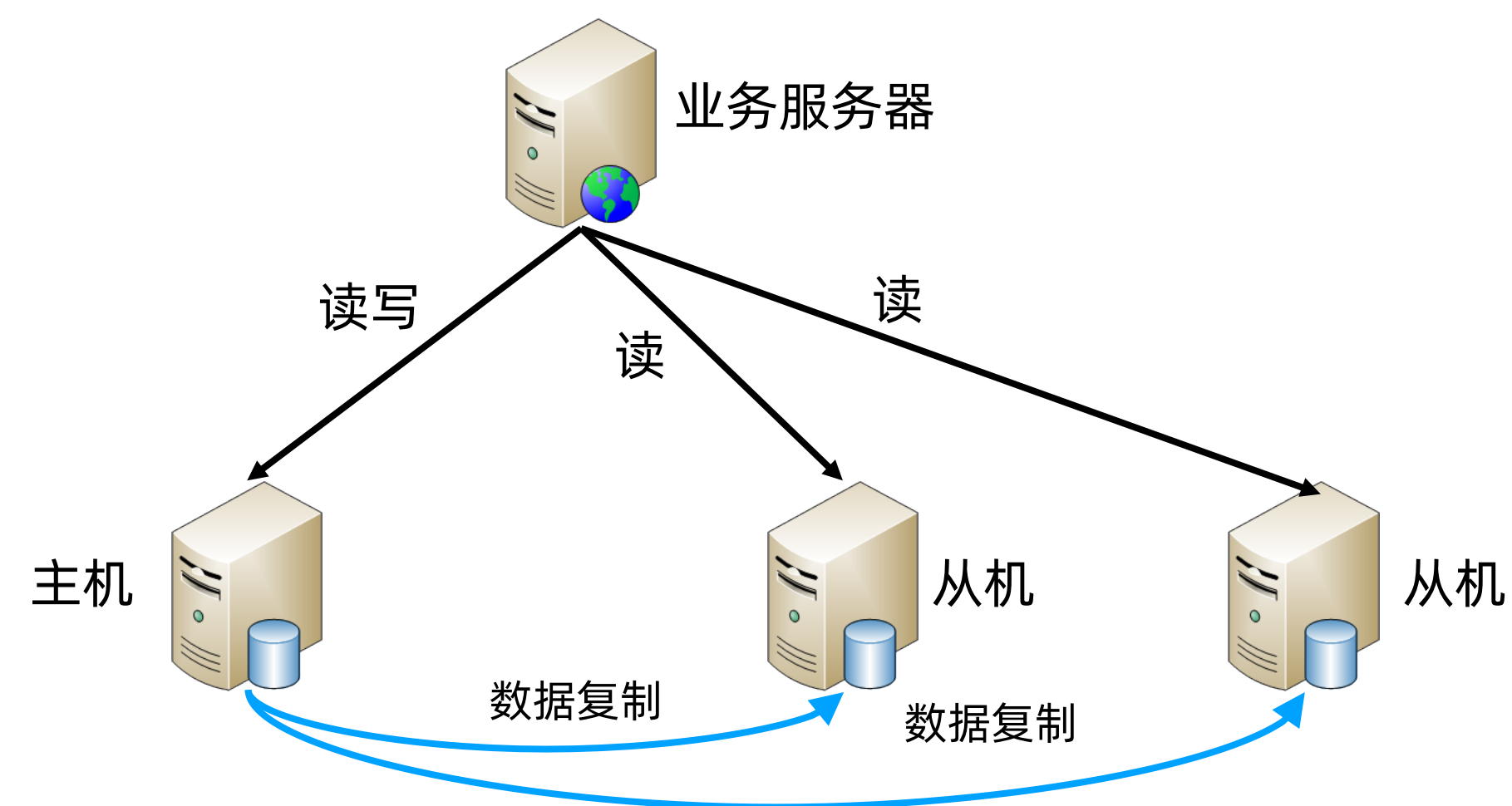
一手微信study322 价格更优惠  
有正版课找我 高价回收帮回血

# 主备复制 & 主从复制

主备复制(数据可恢复)



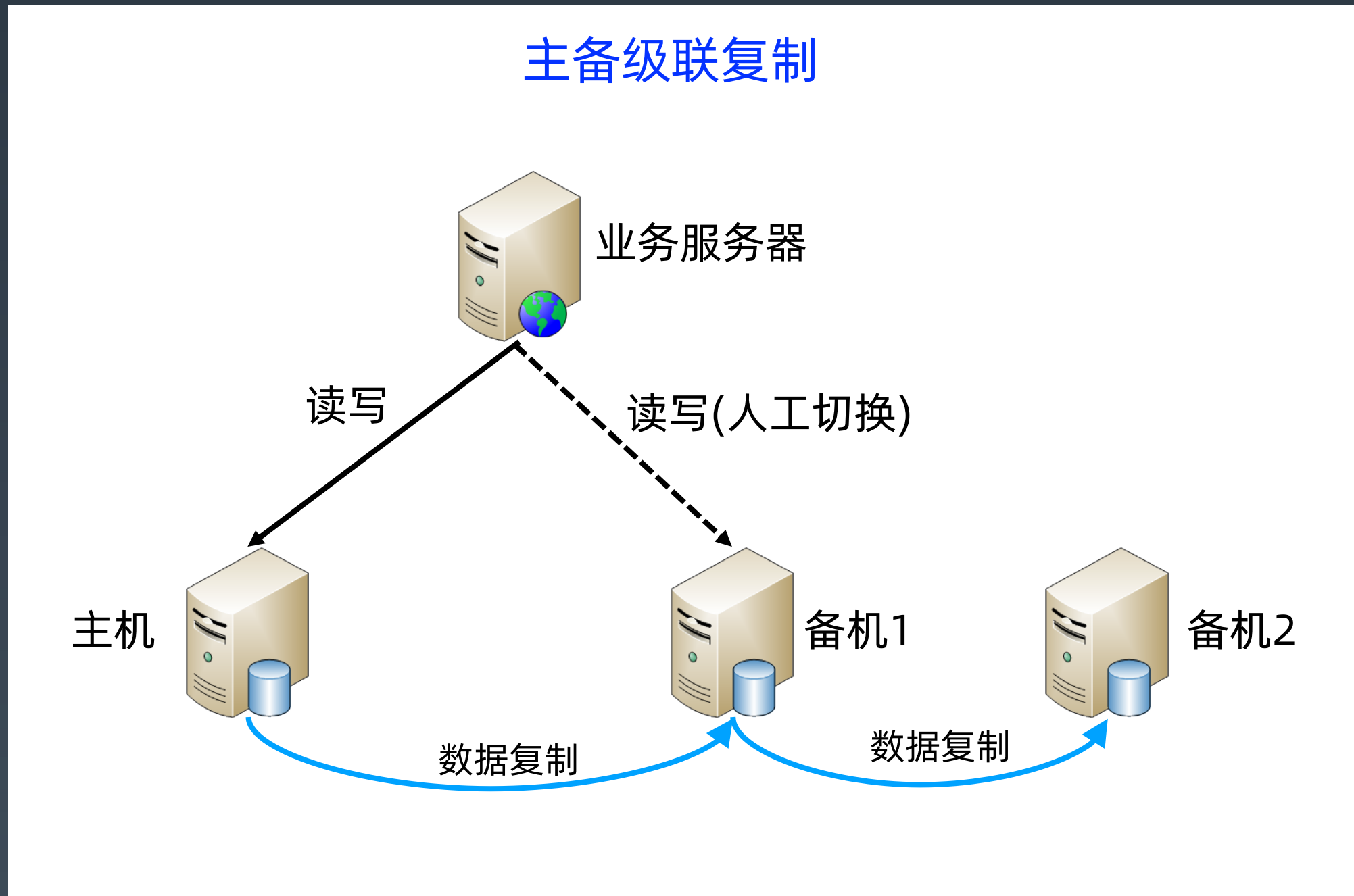
主从复制(数据可恢复 + 高性能读)



**本质：**通过冗余来提升可用性，通过叠加来提升读性能  
**优点：**实现简单，只需要数据复制，无状态检测和角色切换  
**缺点：**需要人工干预，RTO比较大  
**变化：**备机是否提供复制源功能，备机部署地点，主从主备混合部署



# 主备级联复制



**变化：**  
备机作为复制源，例如图中备机1是备机2的复制源

**优点：**  
主机故障后，切换备机1为主机，方便快捷，直接修改配置即可（图中虚线部分），无需修改备机2的配置，无需判断备机1和备机2的数据覆盖问题

**缺点：**  
备机1对备份非常关键，备机1宕机会导致两台备份机都备份失败

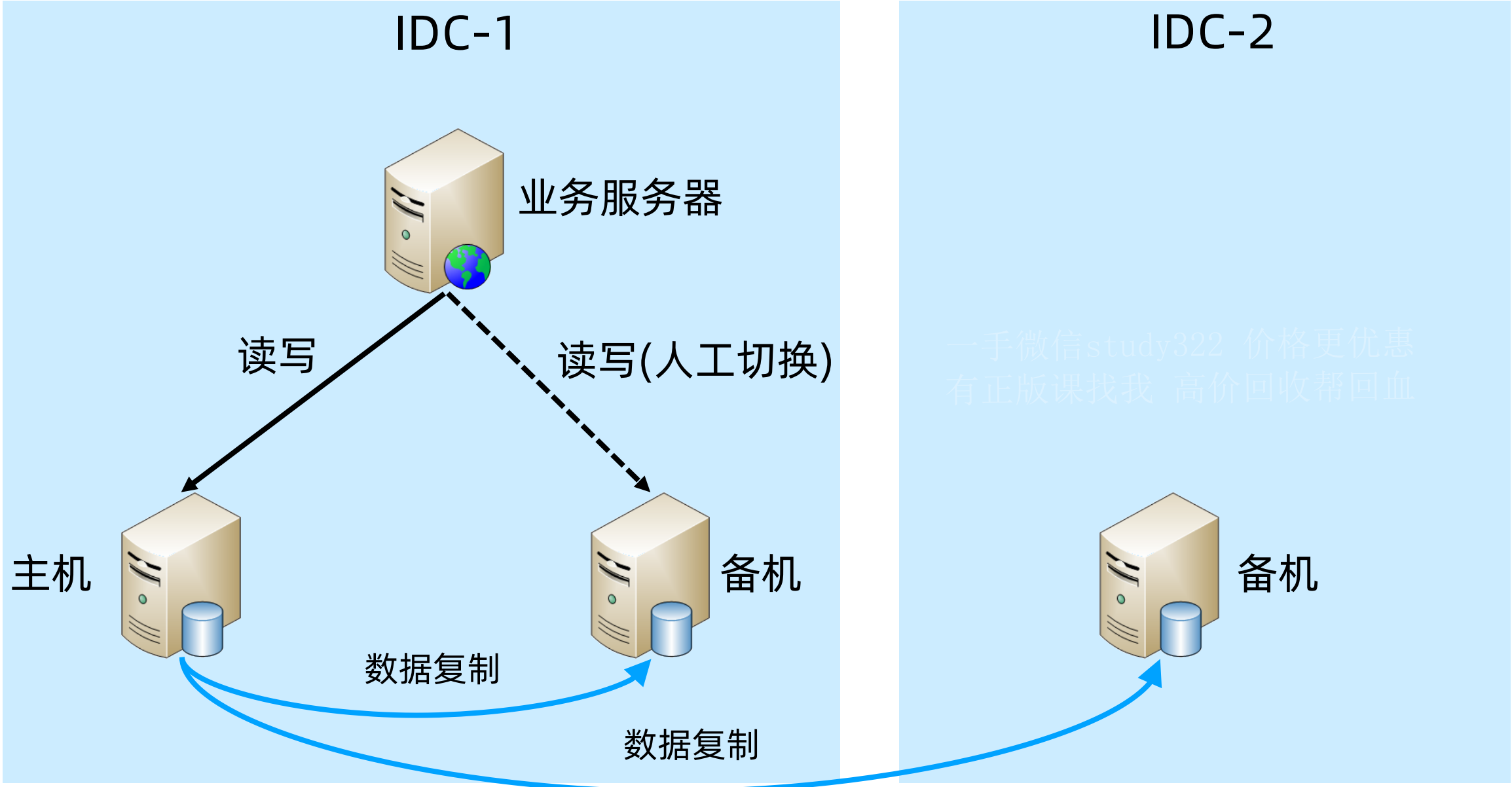
**应用：**  
MySQL、Redis 支持这种模式



主从架构可以这么做么？

# 主备架构的灾备部署

主备跨机房复制



**【场景1】**

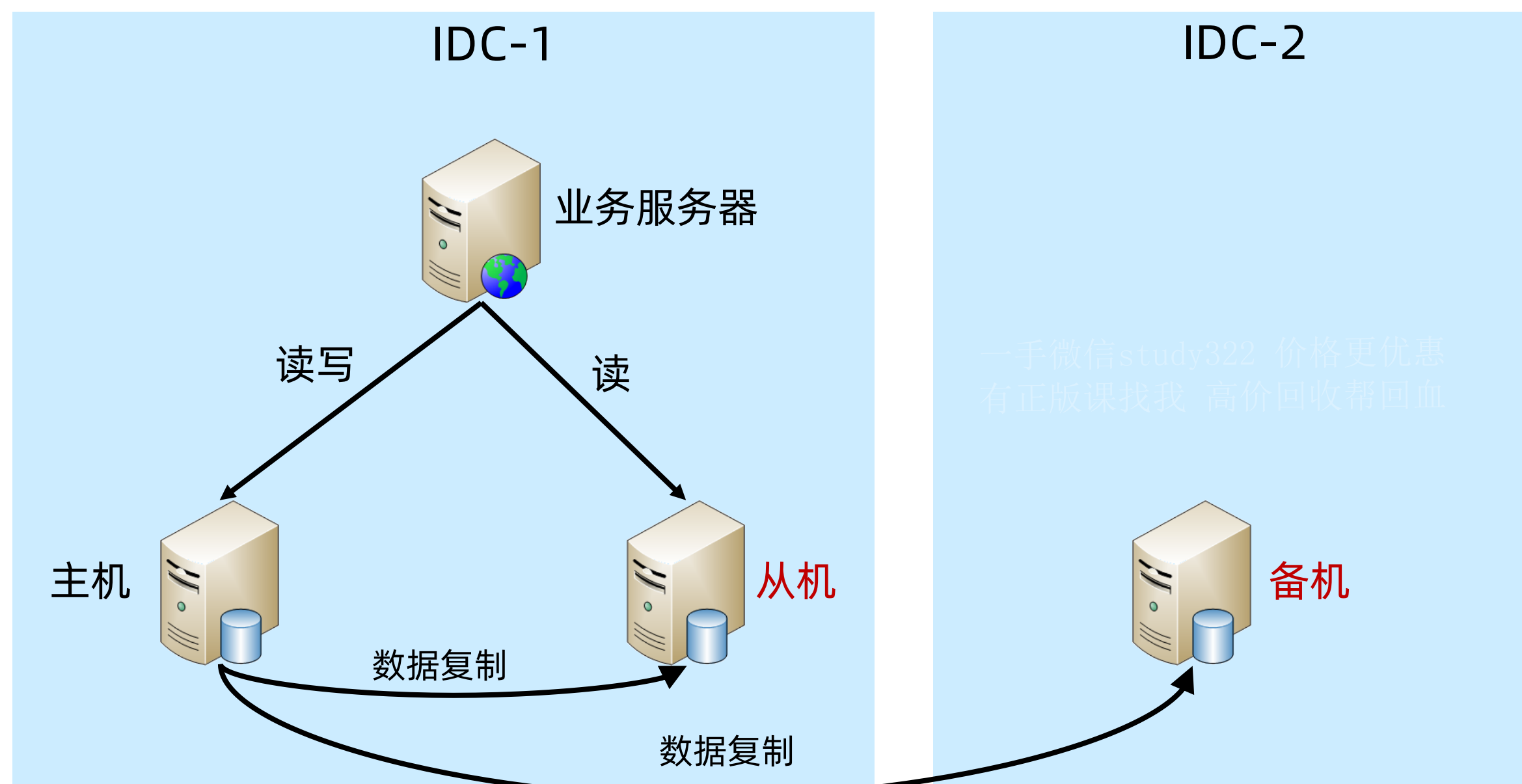
IDC-1和IDC-2在同一个城市，可以应对机房级别的灾难

**【场景2】**

IDC-1和IDC-2不在同一个城市，可以应对城市级别的灾难

# 主从架构的灾备部署

主从跨机房复制



## 【场景1】

IDC-1 和 IDC-2 在同一个城市，可以应对机房级别的灾难

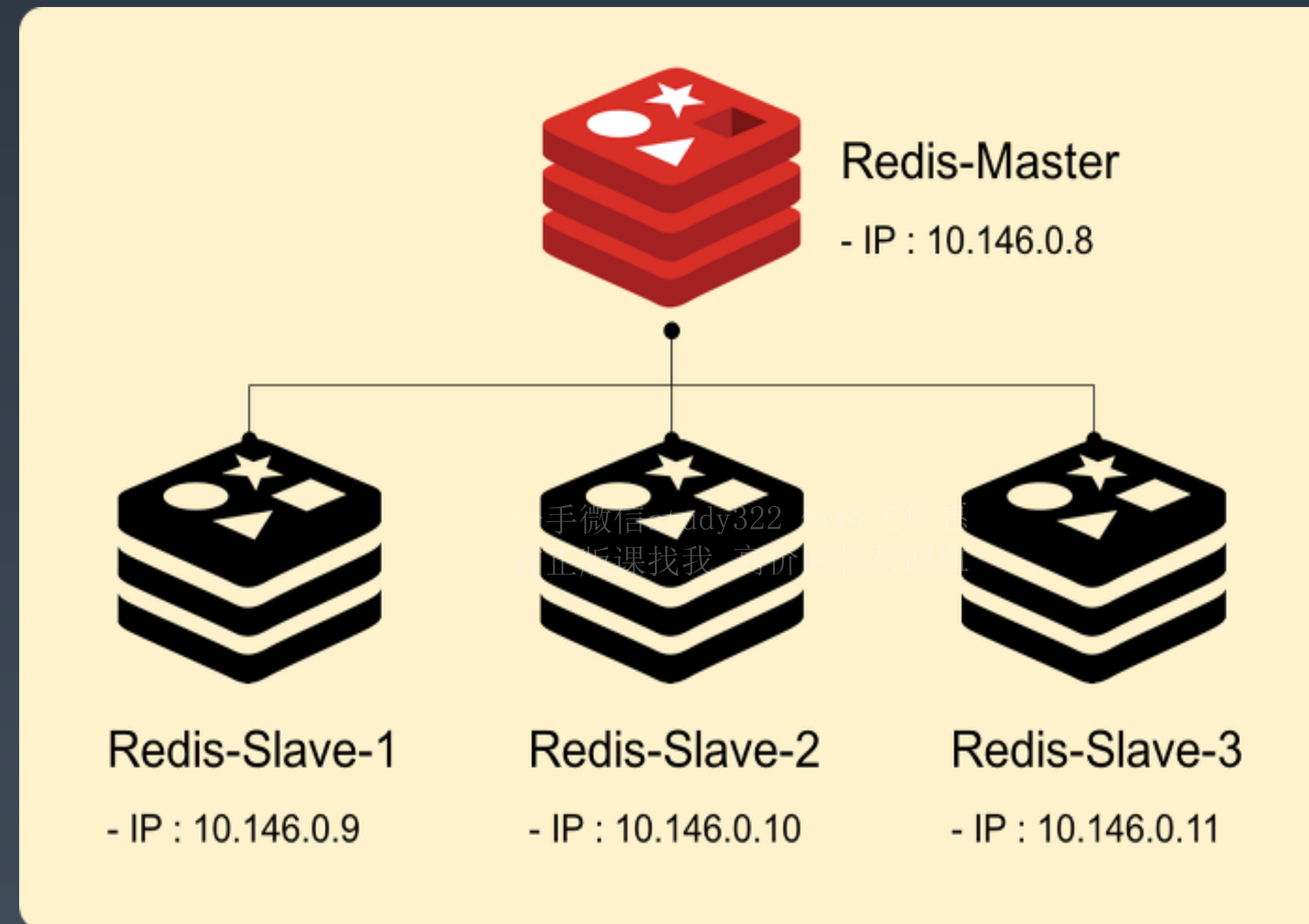
## 【场景2】

IDC-1 和 IDC-2 不在同一个城市，可以应对城市级别的灾难



为什么不把 IDC-2 的备机改为从机提供读取服务？

# 案例学习 - Redis



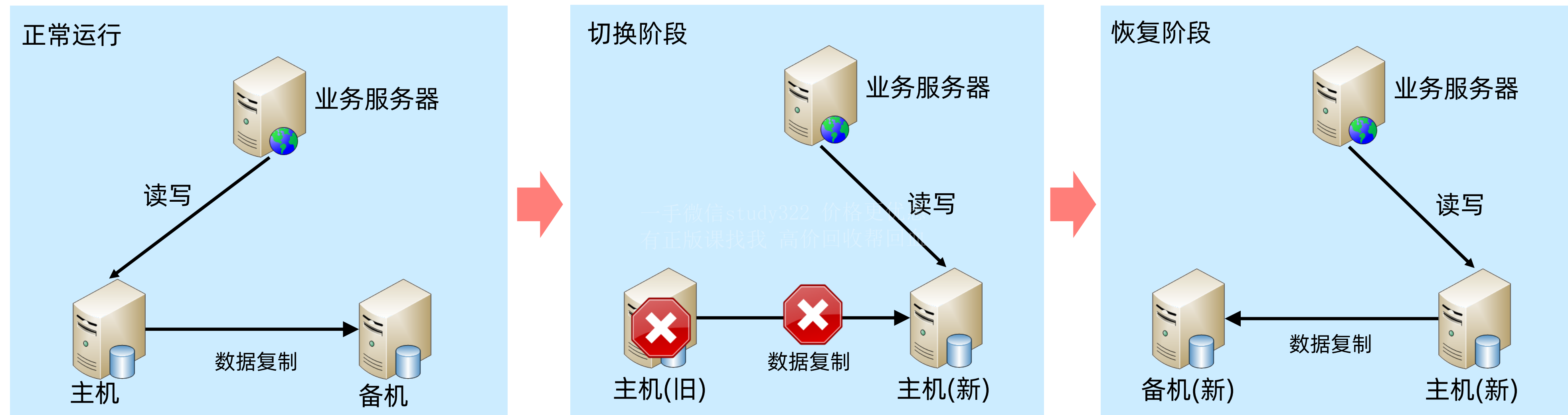
Redis 的 Master-Slave 方案属于哪种架构模式？

# 双机切换架构

一手微信study822 价格更优惠  
有正版课找我 高价回收帮回血

# 双机切换 - 主备切换

## 主备切换



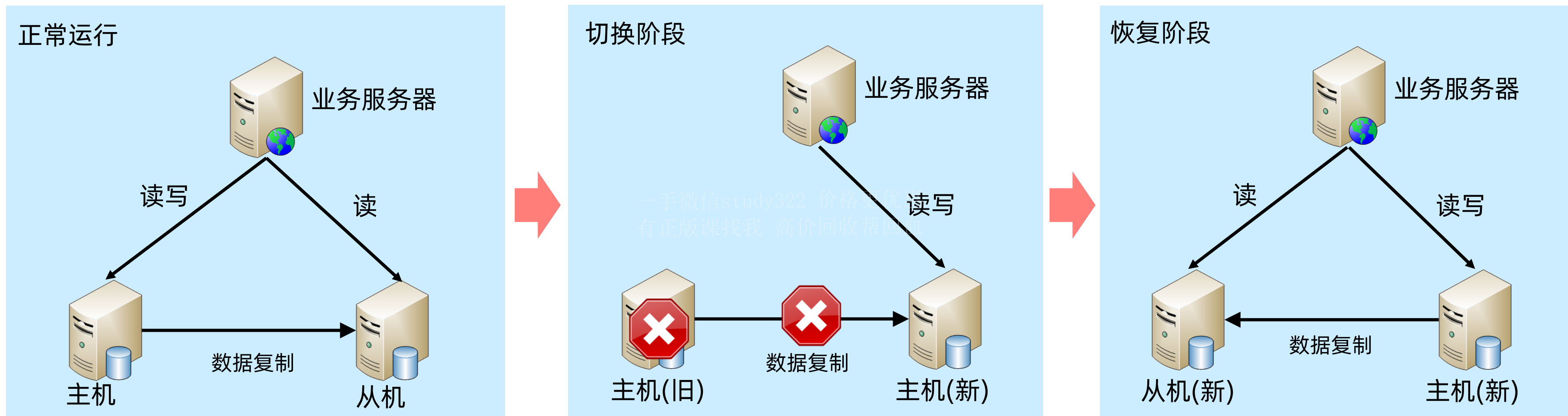
**优点:** 可以自动实现故障恢复, RTO短

**缺点:** 实现复杂, 需要实现数据复制、状态检测、故障切换 (参考模块2第3课)、数据冲突处理

**应用:** 内部系统、管理系统

# 双机切换 - 主从切换

## 主从切换



整体和主备切换类似，差异点在于“切换阶段”，只有主机提供读写服务，主机性能有风险

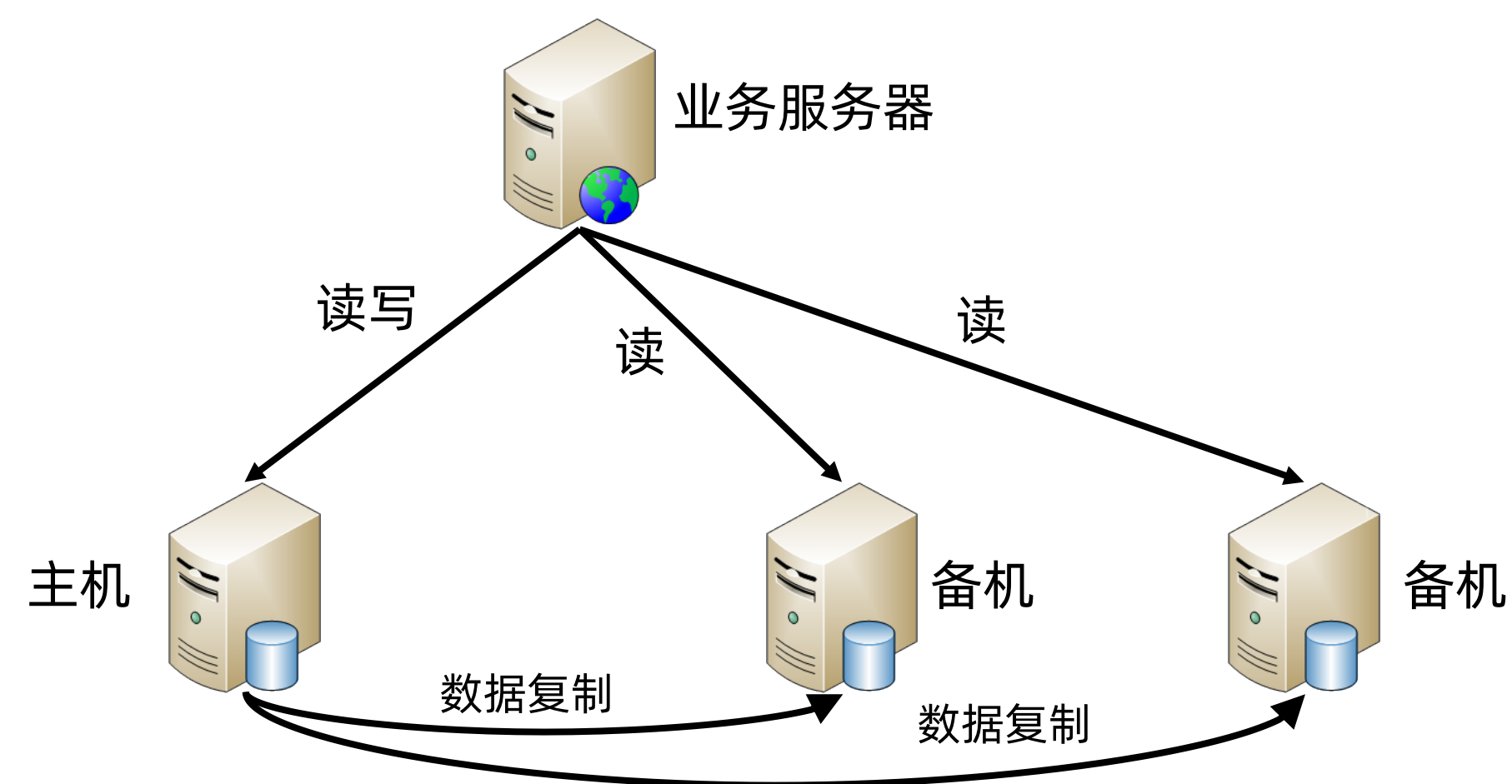
# 集群选举架构

· 微信study522 价格更优惠  
有正版课找我 高价回收帮回血

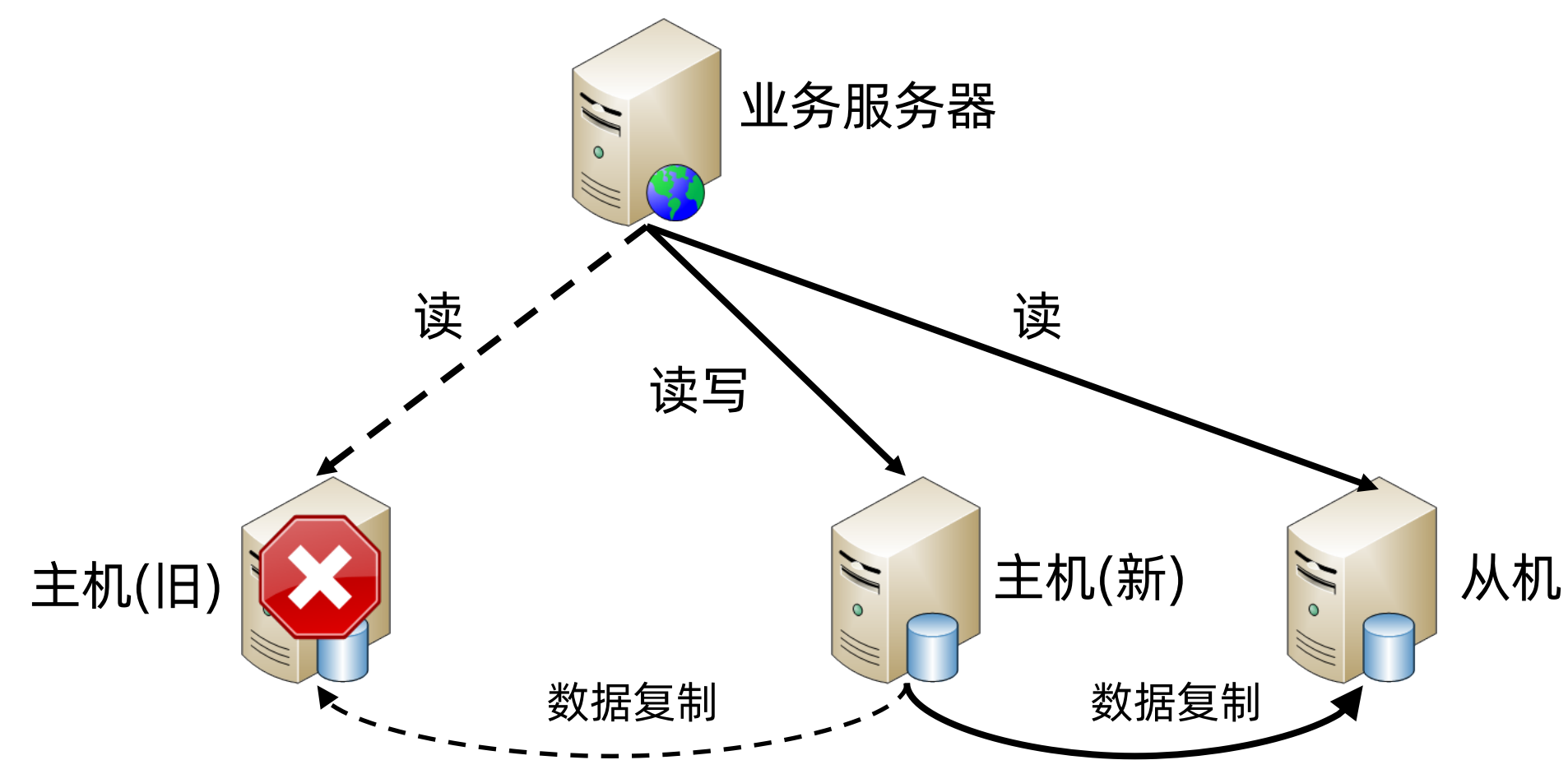


# 集群选举

正常运行



选举

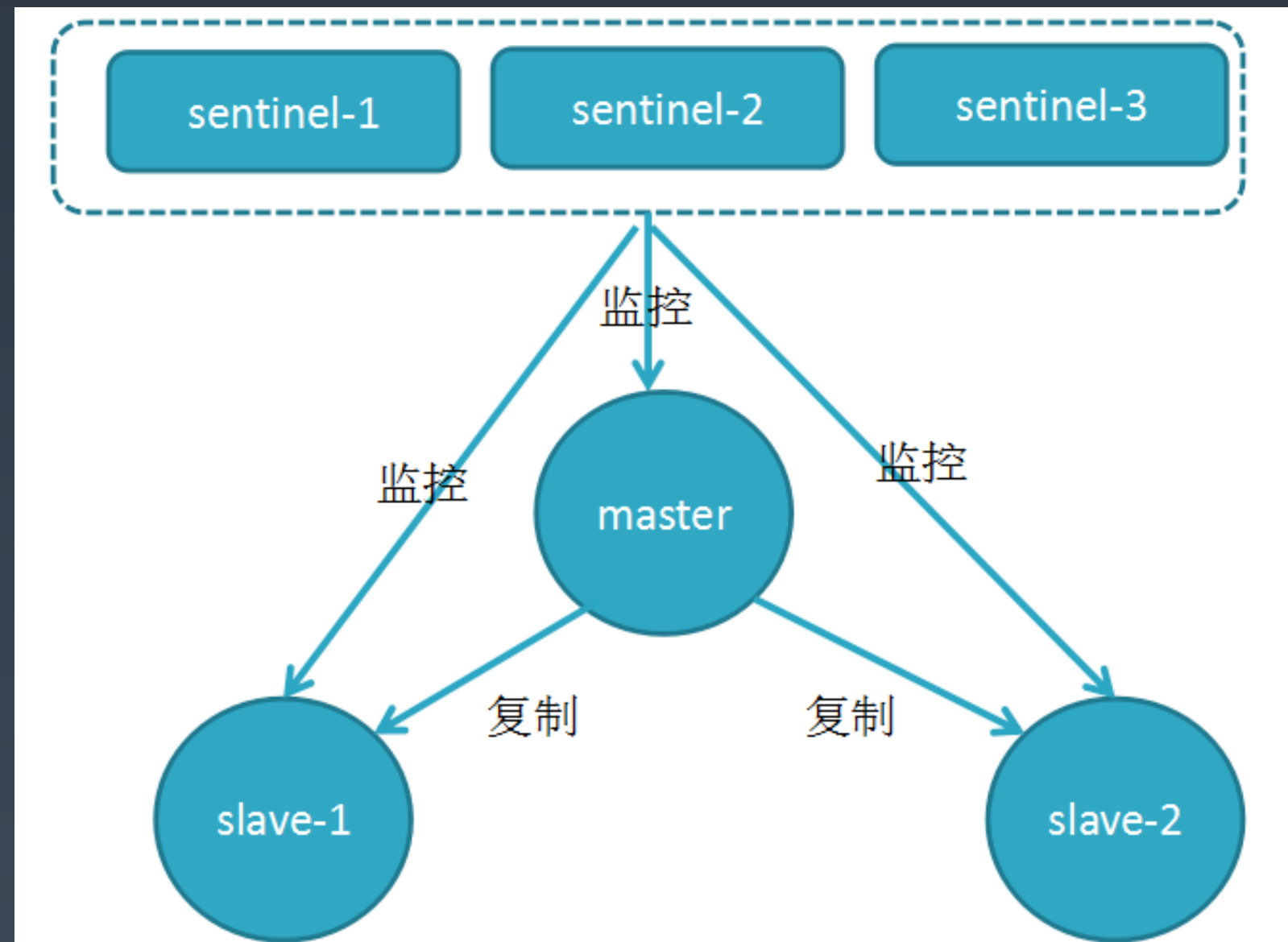


**优点：**可以自动实现故障恢复，RTO短，可用性更高

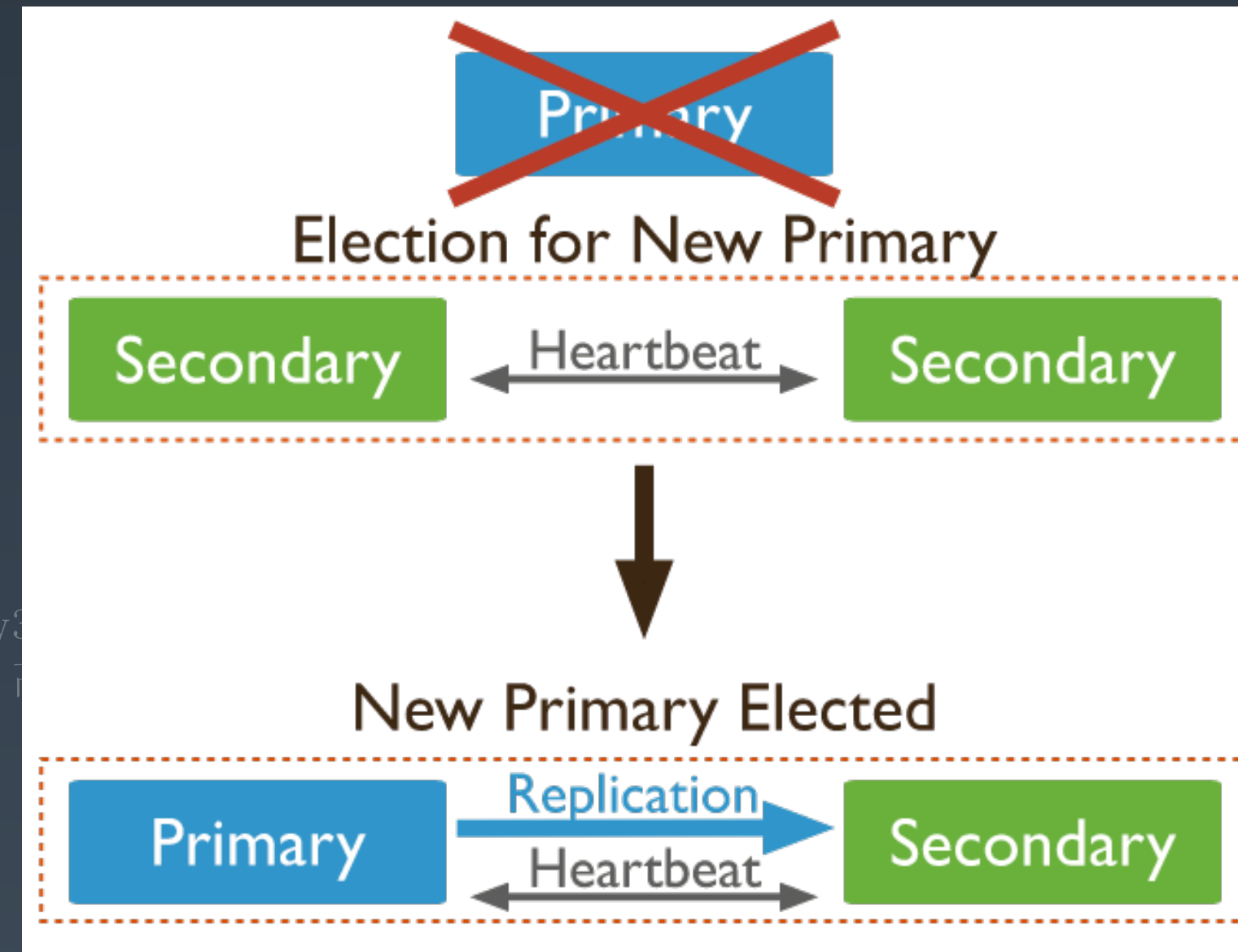
**缺点：**实现复杂，需要实现数据复制、状态检测、**选举算法**、故障切换（参考模块2第3课）、数据冲突处理

**应用：**通用，例如Redis、MongoDB等

# 集群选举案例 - Redis vs MongoDB



**Redis:** 使用 sentinel 集群来解决“决策者”单点问题，sentinel 又是通过 Raft 算法进行选举的。



**MongoDB:** 3.2之前是 Bully 选举算法（ES 也是用这个），3.2之后改为 Raft 算法



Redis Sentinel 架构简单还是 MongoDB replica set 架构简单？

# Bully vs Raft vs ZAB vs Paxos

Bully

谁的 ID 最大或最小谁来当老大（一般选择 ID 大的），Elasticsearch 用到

Raft

分布式一致性协议，etcd、MongoDB 用到

ZAB

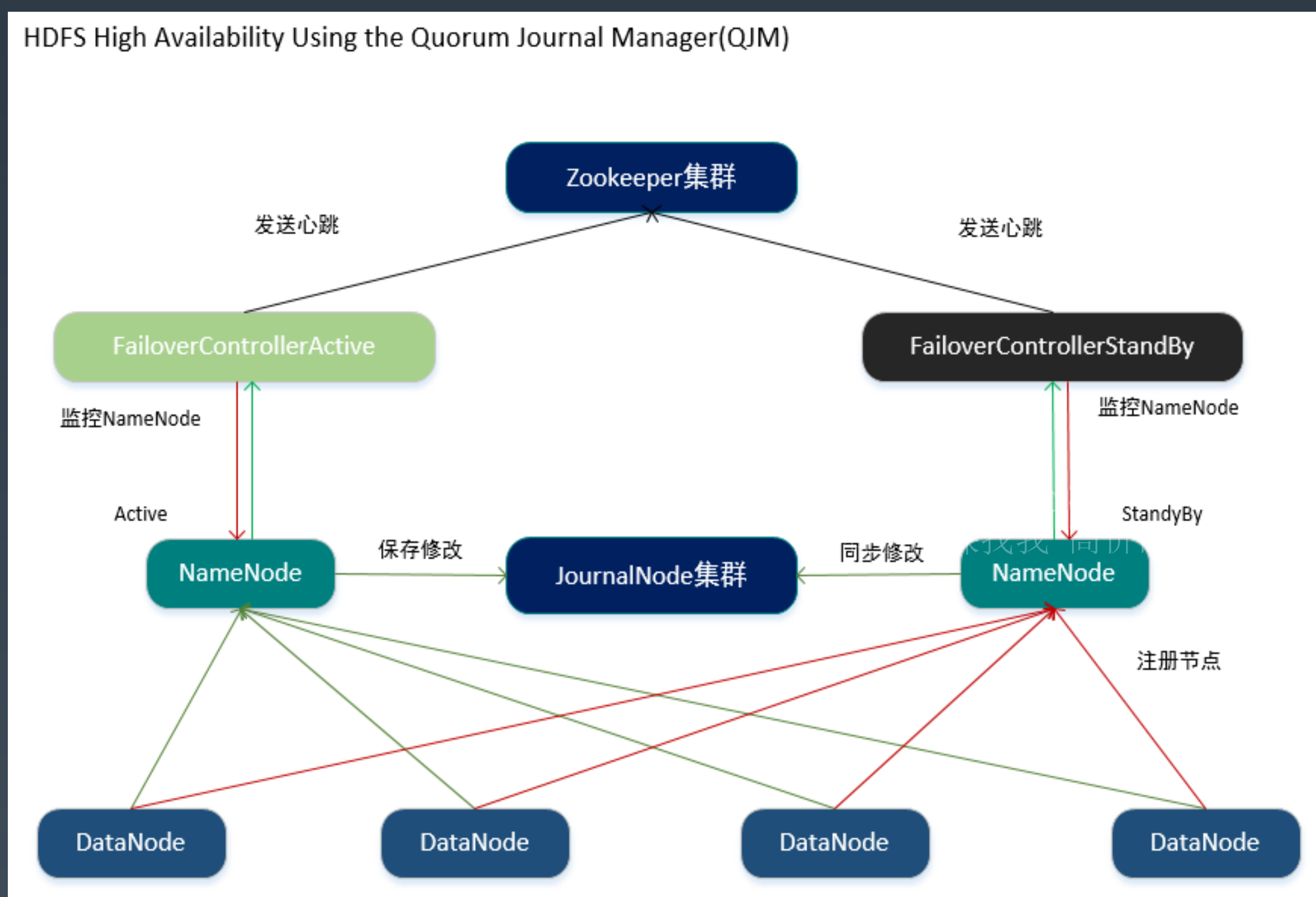
分布式一致性协议，ZooKeeper 用到

Paxos

分布式一致性协议，OceanBase、Chubby 用到

“There is only one consensus protocol, and that's Paxos”-all other approaches are just broken versions of Paxos” -- Google Chubby Mike Burrows

# 最佳实践 - 基于 ZooKeeper 实现



基于 ZooKeeper 来实现双机切换或者集群选举，能够大大降低复杂度，优势有几点：

1. ZooKeeper 已经保证了自我的高可用
2. 基于 ZooKeeper，切换或者选举过程实现比较简单
3. ZooKeeper 可以有多用途



基于 ZooKeeper 的状态决策是民主式、独裁式、协商式中的哪一种？

# 本节思维导图



# 随堂测验

## 【判断题】

1. 对于高可用架构来说，RTO 和 RPO 指标越高越好
2. 主备和主从架构可以结合起来形成高可用架构方案
3. 主备架构和主从架构需要人工处理故障，而双机架构可以自动切换故障，因此肯定选双机架构
4. 集群选举复杂度要高于双机切换
5. Bully 算法太简单了，不适合高可用集群选举

一手微信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 MongoDB 要改掉 Bully 而 Elasticsearch 可以用 Bully ?



# 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