

# 阿里游戏异地多活设计之道

李运华@阿里游戏（资深技术专家）

# 目录

---



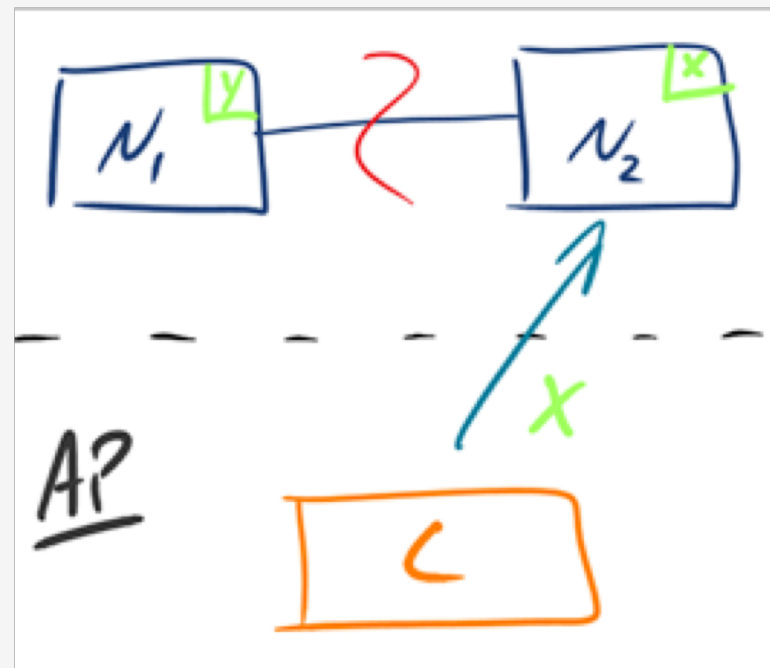
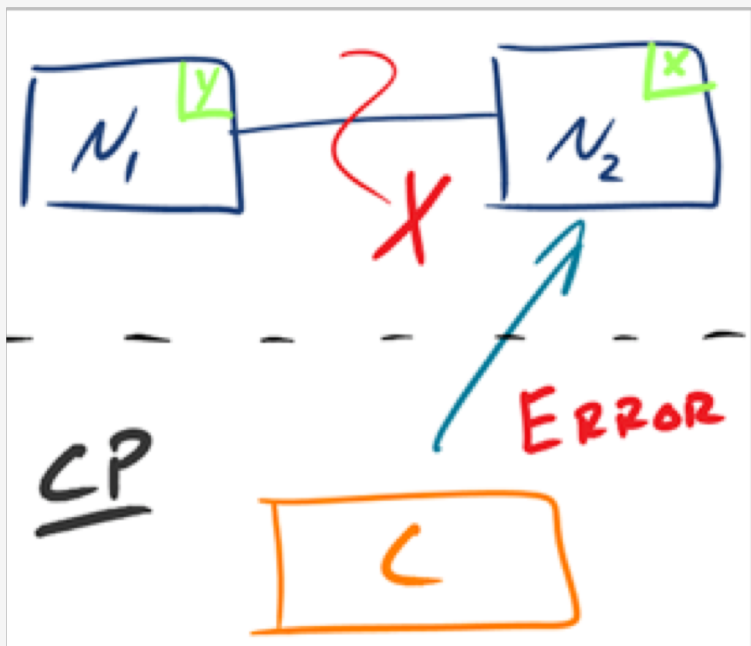
**1 异地多活 – 1个原理**

**2 异地多活 – 3大原则**

**3 异地多活 – 4大步骤**

**4 阿里游戏实践**

# 大道至简- CAP理论



异地多活本质上是一个AP方案

# 大道至深 – CAP细节

---

- CAP关注的粒度是数据，而不是系统
- CAP是忽略网络延迟的
- 正常运行情况下，可以同时满足CA
- 放弃 != 无为，需要为分区恢复后做准备。

# 目录

---

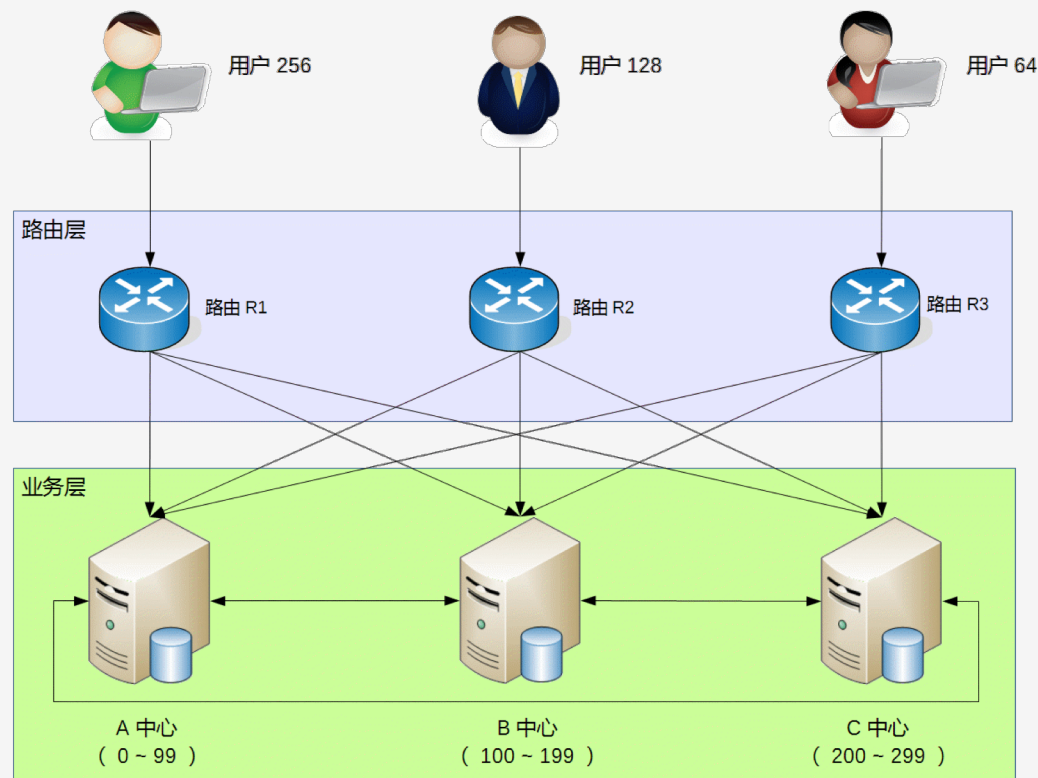
**1** 异地多活 – 1个原理

 **2** 异地多活 – 3大原则

**3** 异地多活 – 4大步骤

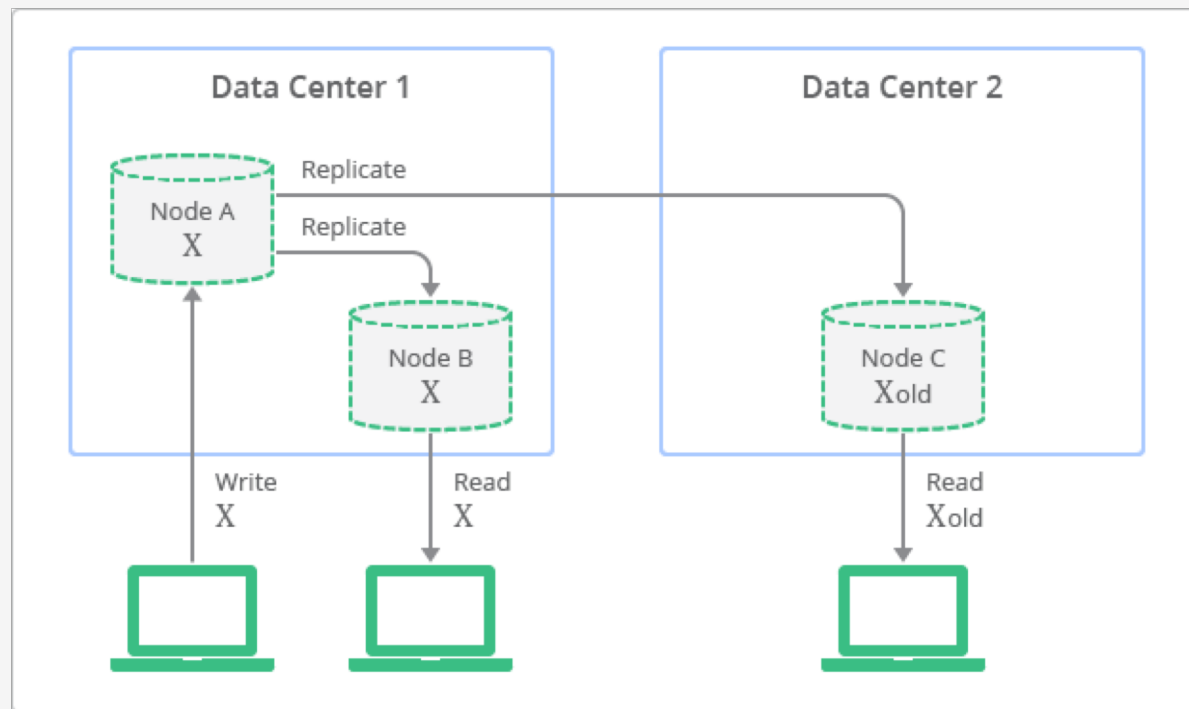
**4** 阿里游戏实践

# 原则1 – 只保证核心业务



丢車保帥：只保证登录的异地多活

# 原则2 – 只能做到最终一致性



复制肯定有时间窗，抛弃实时一致性的幻想

# 原则3 – 只保证绝大部分用户

---



不要为了0.01%的用户，而影响了99.9%的用户



# 目录

---

**1 | 异地多活 – 1个原理**

**2 | 异地多活 – 3大原则**



**3 | 异地多活 – 4大步骤**

**4 | 阿里游戏实践**

# 异地多活设计步骤

---

1. 业务分级
2. 数据分类
3. 数据同步
4. 异常处理

# 步骤1 – 业务分级

---

## ➤访问量

登录 > 注册 > 修改密码

## ➤核心场景

聊天>朋友圈>摇一摇

## ➤收入来源

订单>搜索>编辑

优先保证核心业务

# 步骤2 – 数据分类

---

## ➤数据量

修改数据量

## ➤一致性

余额、库存：强一致性

用户信息：最终一致性

## ➤唯一性

账号：全局唯一；

昵称：全局不唯一

## ➤可丢失性

账户余额变更：不可丢失；

微博：可丢失；

Session：可丢失；

## ➤可恢复性

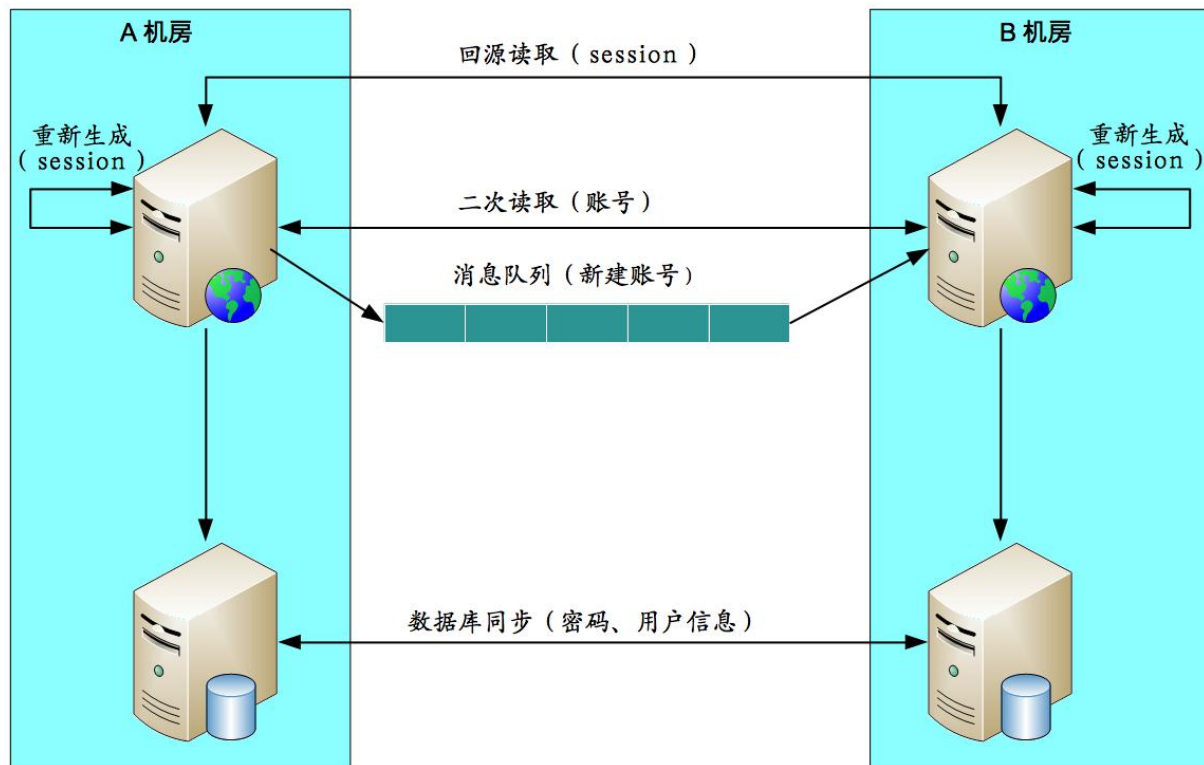
微博：重发即可（体验？）

用户密码：“忘记密码”找回

新闻：可以让运营人员重发（工作量？）

不同数据不同处理方式

# 步骤3 – 数据同步



多管齐下，“不择手段”

# 步骤4 - 异常处理

---

## ➤ 业务逻辑兼容

数据短时间不一致：业务有损，旧数据

数据无法获取：转账申请，支付核对中

## ➤ 事后补偿

礼包、红包；

礼物、物品（暴雪炉石回档补偿）；

## ➤ 人工修正

人工订正数据，达到最终一致性；

重要的事情说三遍：日志、日志、日志

“客户第一”，要亏自己亏

# 目录

---

**1 异地多活 – 1个原理**

**2 异地多活 – 3大原则**

**3 异地多活 – 4大步骤**

**➔ 4 阿里游戏实践**

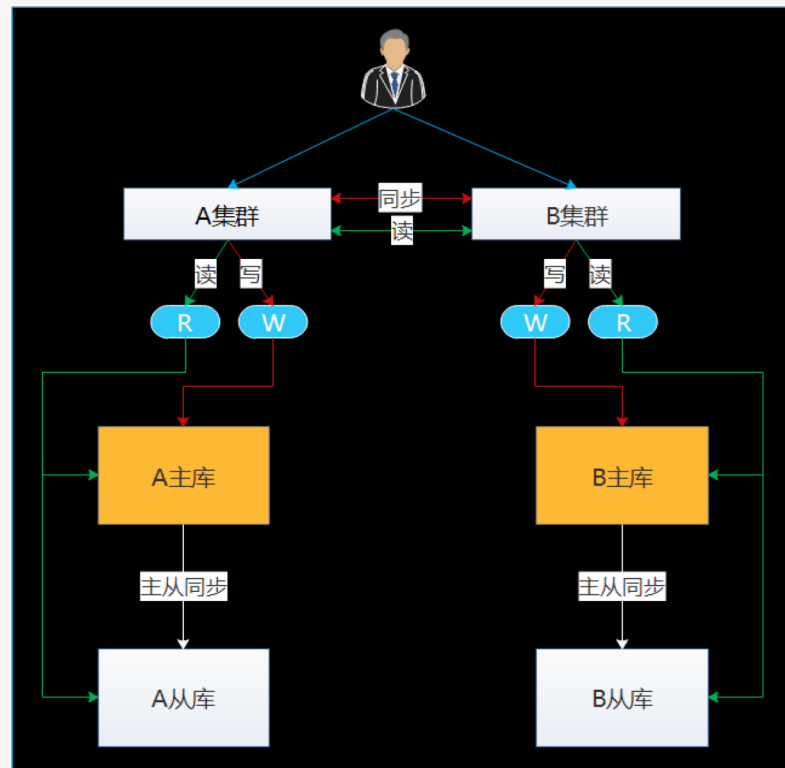
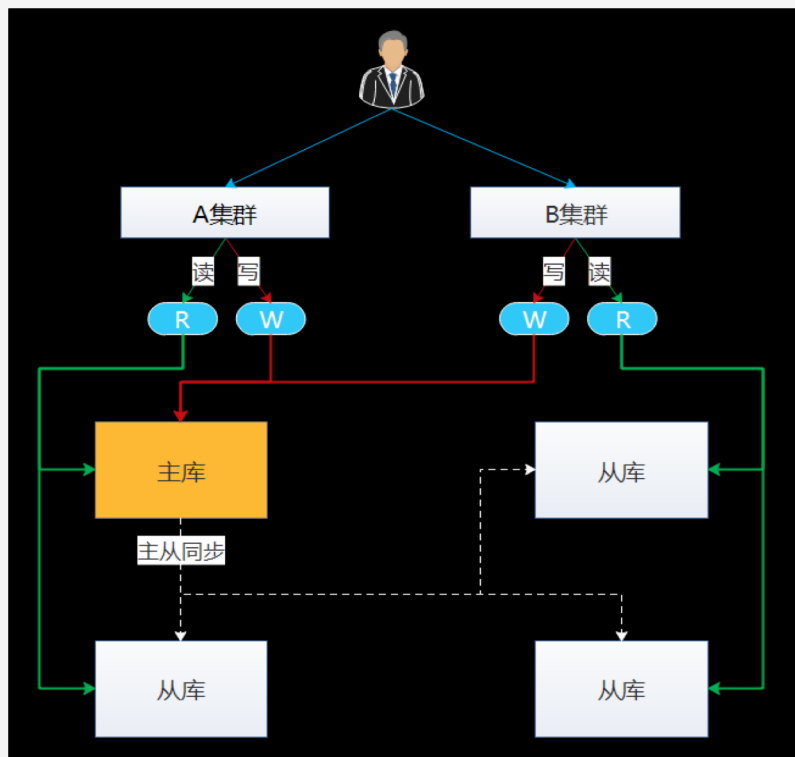
# 阿里游戏接入系统异地多活

---

1. 业务分级  
登录、支付
2. 数据分类  
session、角色id、账号密码
3. 数据同步  
session: 不同步, 重复生成  
角色id: 全局唯一, 消息队列同步 + 算法重复生成  
账号密码&支付: 第三方管理, 无需处理
4. 异常补救  
人工修复  
礼包、代金券补偿



# 阿里游戏接入系统异地多活



机房掉电、DDOS攻击。。。这都不算个事