

58数据库架构最佳实践

58沈剑

shenjian@58.com

关于-我

- 百度-高级工程师
- 58同城-高级架构师
- 58同城-技术委员会主席
- 58同城-技术学院优秀讲师
- 58同城-业界技术分享嘉宾
- 58到家-技术总监
- 58到家-技术委员会负责人
- **本质：程序员一枚！**

微博



微信



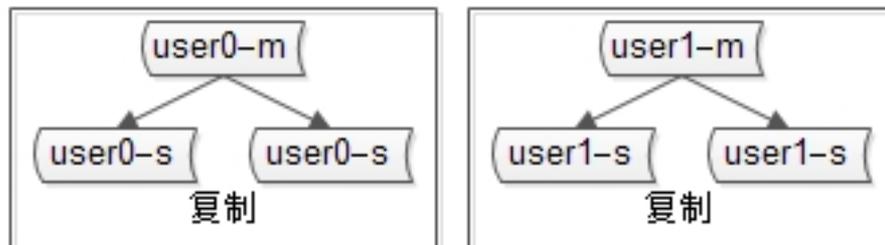
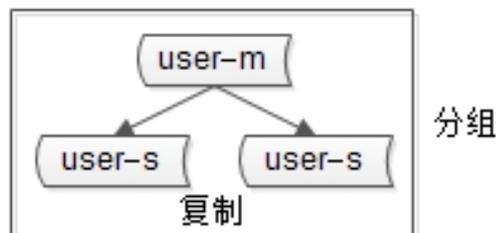
目录

- 基本概念
- 58最佳实践
- 总结

一、基本概念

基本概念

- **单库** : single
- **分片** : sharding
- **路由** : routing
- **复制** : replication
- **分组** : group

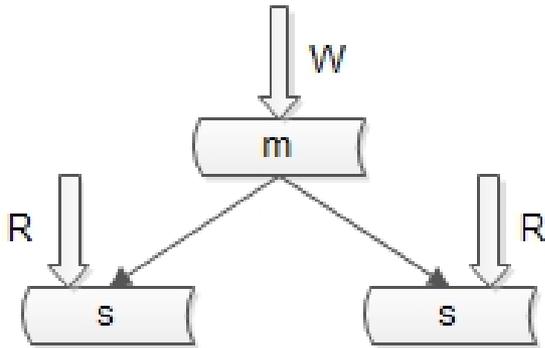


实际应用：分组+分片

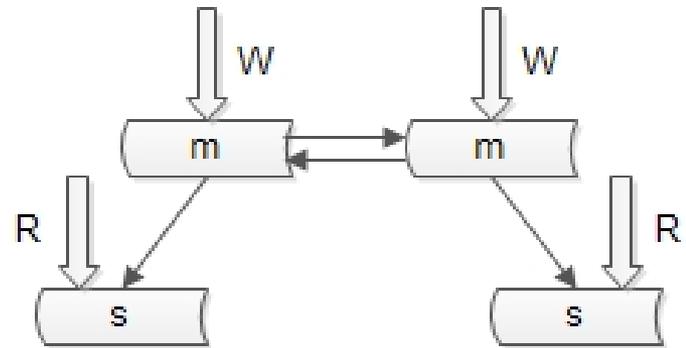
二、58最佳实践

(一) 可用性实践-传统玩法

- 如何保证数据库“读”高可用？
- 如何保证数据库“写”高可用？
- 存在什么问题？
- 存在什么问题？



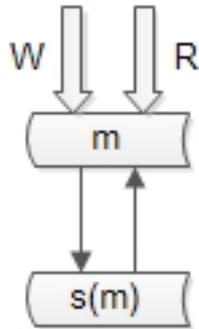
“读” 高可用



“写” 高可用

(一) 可用性实践-58的玩法

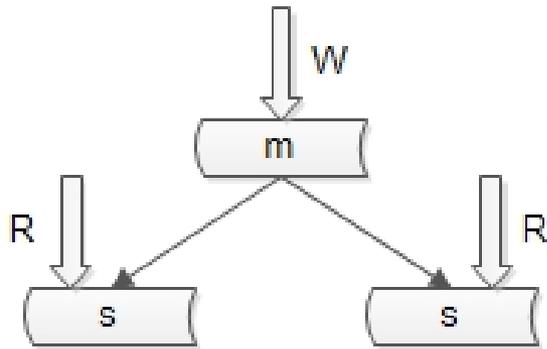
- 58怎么玩数据可用性：“双主”当“主从”用
- 如何解决读写一致性问题？
- 如何解决读写可用性问题？
- 存在什么问题？



“读写” 高可用

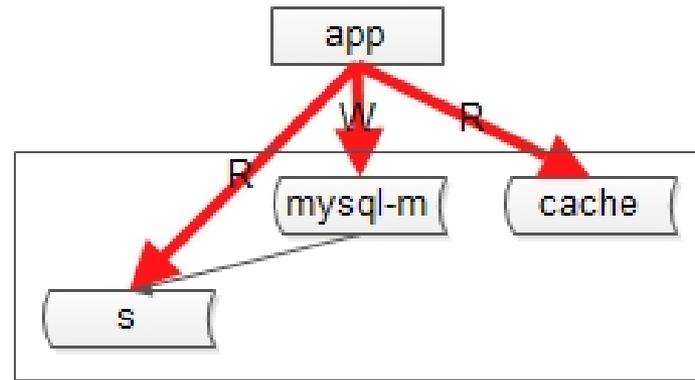
(二) 读性能实践-传统的玩法

- 增加读库
- 存在什么问题？



增加读库

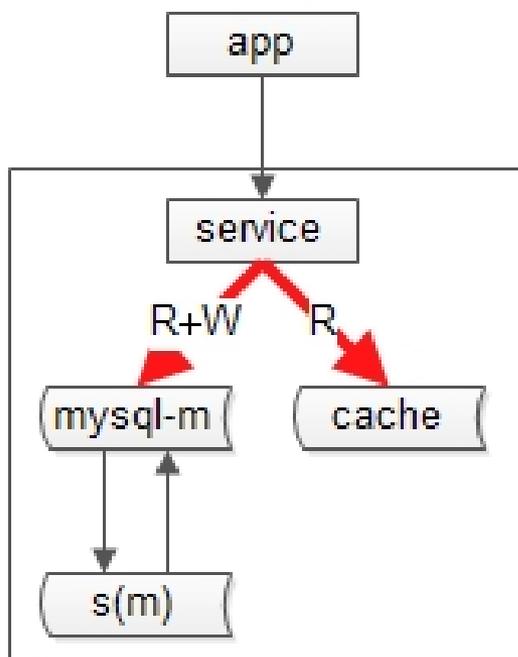
- 增加缓存
- 存在什么问题？



常见玩法：缓存+数据

(二) 读性能实践-58的玩法

- 服务+数据+缓存 一套



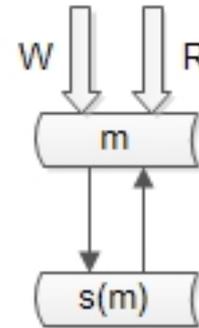
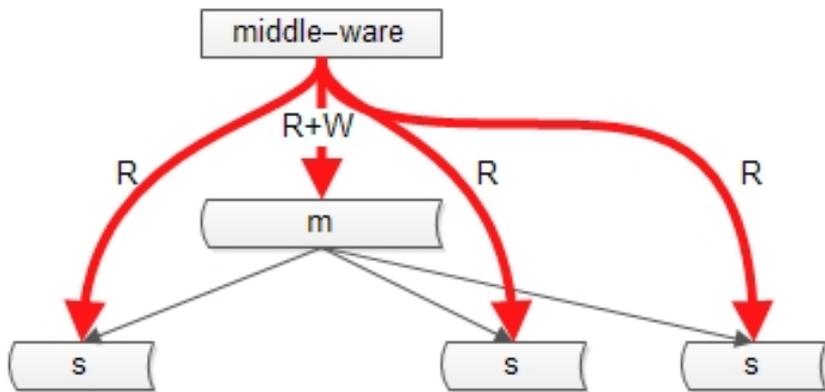
58玩法：服务+缓存+数据

(三) 一致性实践-主从不一致

- 为什么会不一致？
- 主从不一致如何优化？

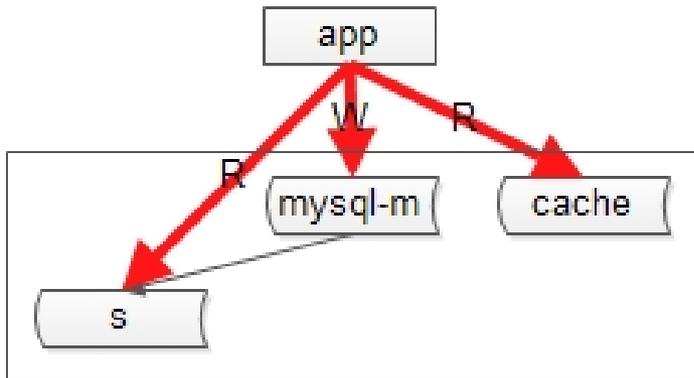
(1) 引入中间件

(2) 强制读主



(三) 一致性实践-缓存不一致

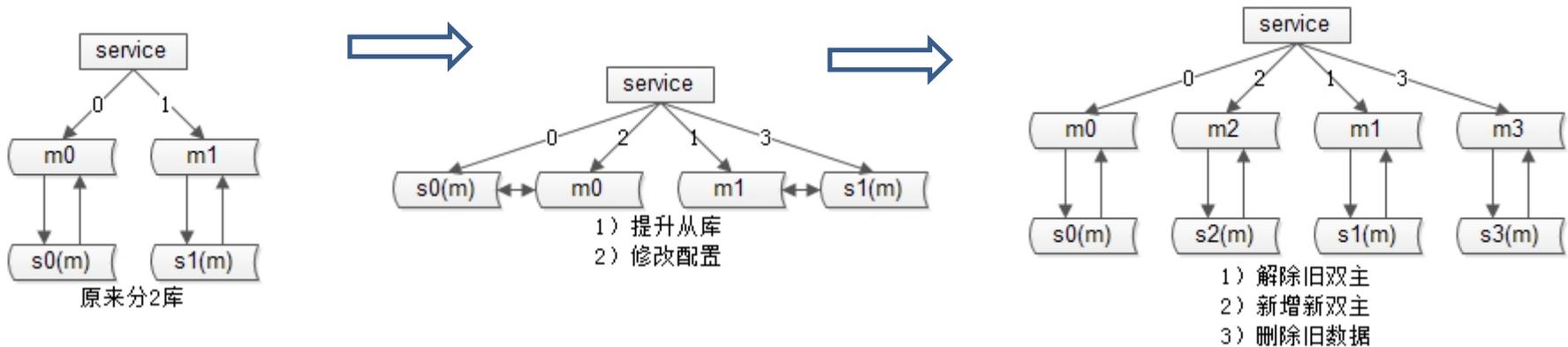
- 为什么会不一致？
- 缓存不一致如何优化？
 - (1) 缓存双淘汰
 - (2) 设定过期时间



常见玩法：缓存+数据

(四) 扩展性实践-秒级扩容

- 原来水平切分成n个库，要扩容成2n个库，如何实现？
- 存在什么问题？

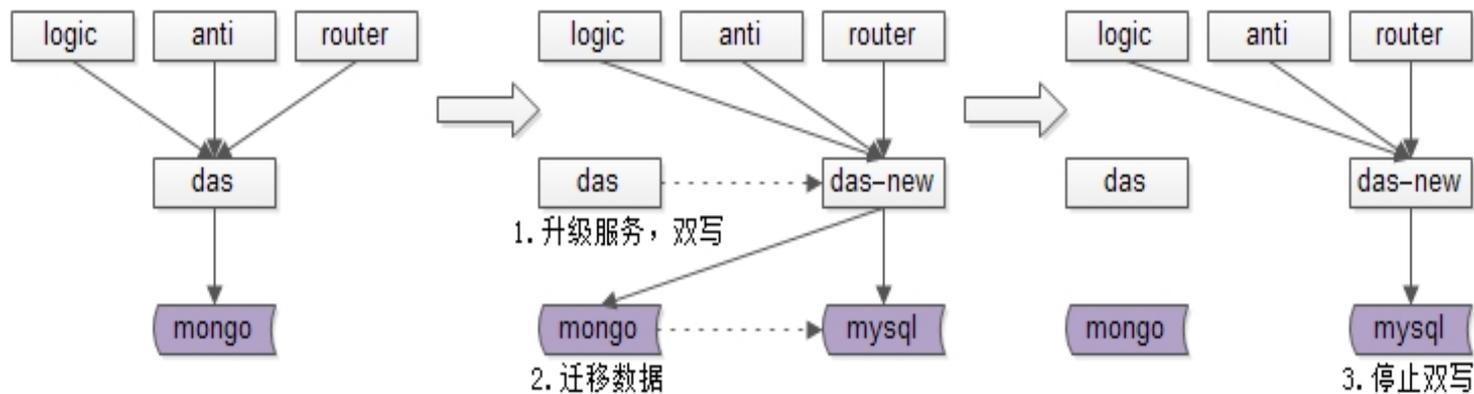


(四) 扩展性实践-平滑倒库

- 解决什么问题？

(1) 2库扩3库 (2) 增加字段 (3) 底层存储介质变化

- 目标：平滑迁移，不停服务
- 扩展方法：双写法倒库



三、总结

总结

- 基本概念：分片，路由，复制，分组
- 可用性实践：双主当主从用
- 读性能实践：服务+数据+缓存一套
- 一致性实践：
 - (a) 主从一致性：强制读主
 - (b) 缓存一致性：缓存双淘汰
- 扩展性实践：
 - (a) 秒级扩容
 - (b) 平滑双写倒库

Q&A

“架构师之路” 公众号

谢谢！

