

直播应用中的高并发场景实战

3-2 QPS 从0到10w+ 的用户
中台的架构是如何演变的?

Produced By Idea讲师

用户中台业务背景

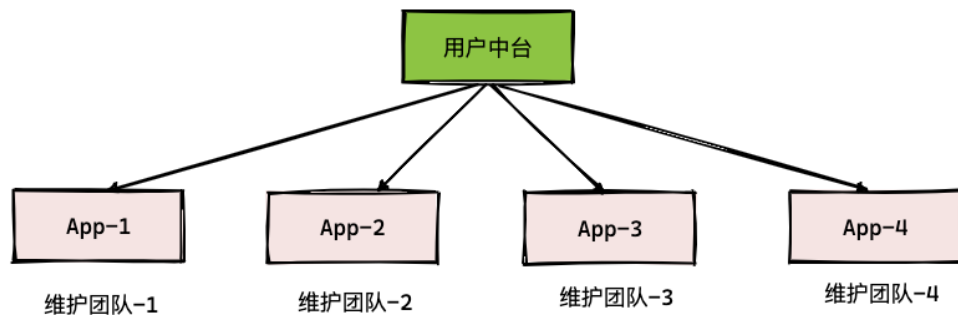


用户中台业务背景

- ◆ 用户数据的统一管理
- ◆ 方便不同业务线的接入
- ◆ 性能，可扩展性，可维护性更高



用户中台业务背景



多个接入方，共同使用一个用户中台体系



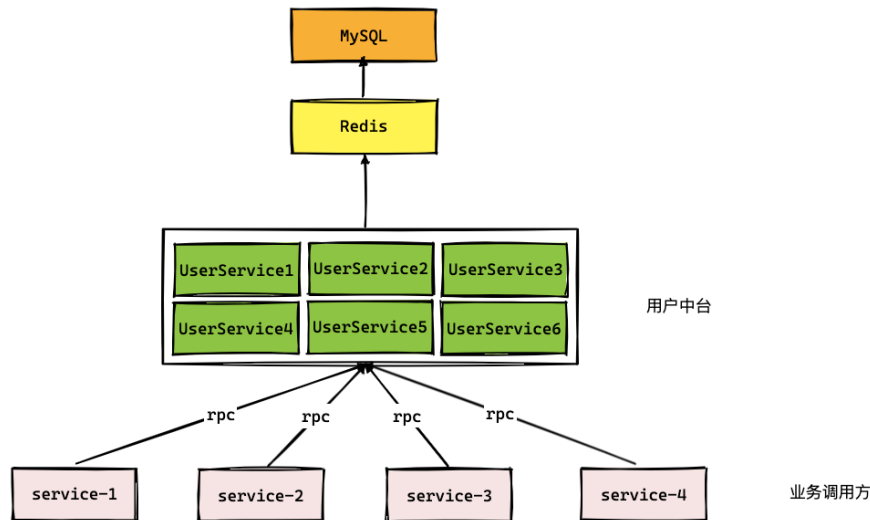
用户中台架构分析



用户中台架构设计

◆ 性能瓶颈: MySQL

◆ 单点风险

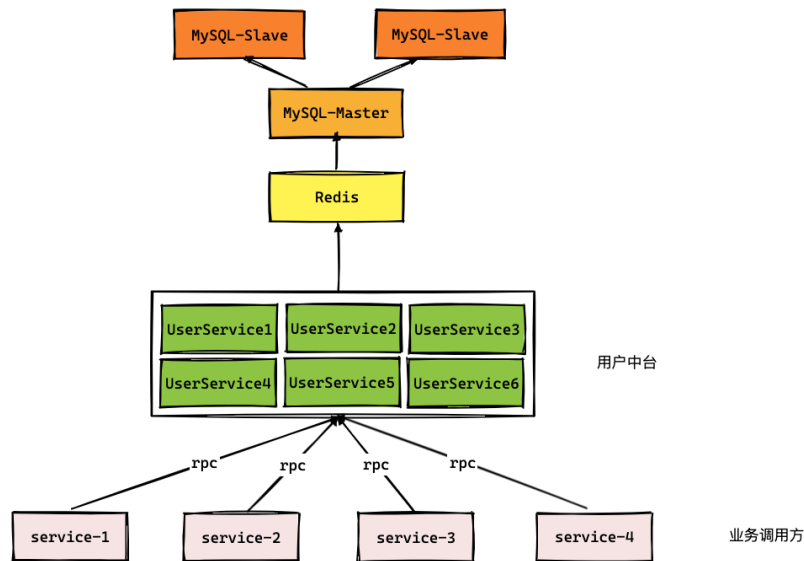


第一版: MySQL大单机 + Redis大单机



用户中台架构设计

- ◆ 容量瓶颈：Redis空间不足
- ◆ Redis单点风险

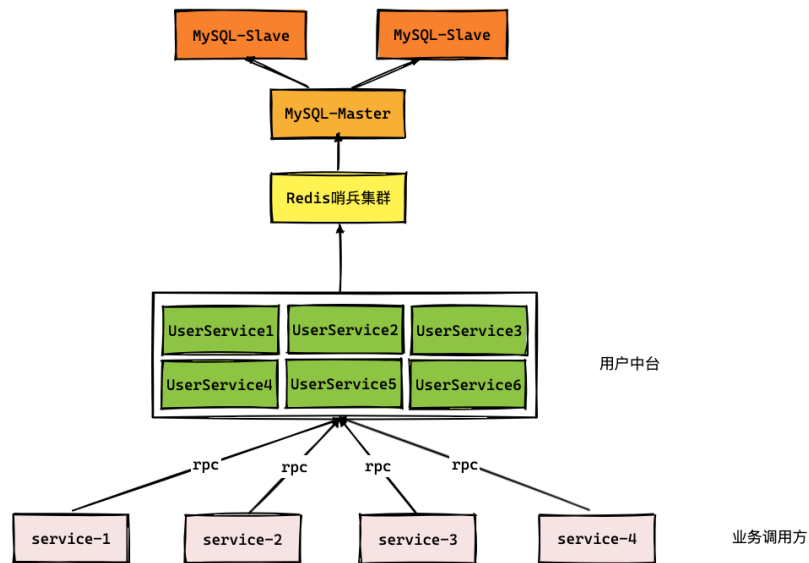


第二版：MySQL读写分离 + Redis大单机



用户中台架构设计

- ◆ 容量瓶颈：Redis空间不足
- ◆ Redis单点风险

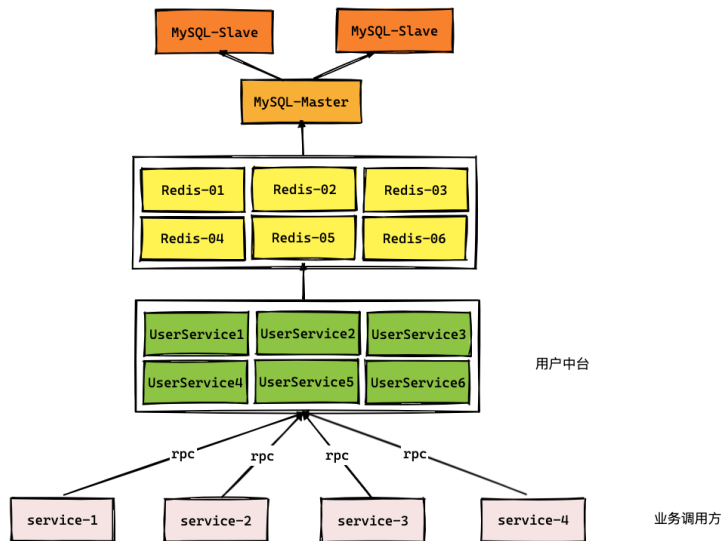


第三版：MySQL读写分离 + Redis哨兵



用户中台架构设计

◆ 机器成本高



第四版：MySQL读写分离 + Redis哨兵



用户中台架构设计

Redis机器的容量预估工具

http://www.redis.cn/redis_memory/



用户中台架构设计

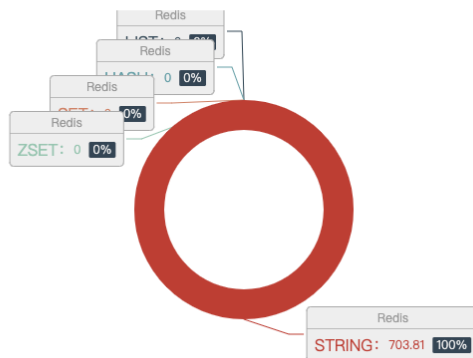
Redis容量预估

Key类型	Key个数	Key长度 单位 (字节)	元素/字段数	元素/字段长度 单位 (字节)	Value长度 单位 (字节)
string	1000000	50			500
list	Key数量	Key长度	元素/字段数	元素/字段长	
hash	Key数量	Key长度	元素/字段数	元素/字段长	Value长度
set	Key数量	Key长度	元素/字段数	元素/字段长	
zset	Key数量	Key长度	元素/字段数	元素/字段长	Value长度

开始计算

所需总内存: 703.81M

■ STRING
■ LIST
■ HASH
■ SET
■ ZSET



用户中台的思考



用户中台架构的思考

MySQL读压力如何解决?

- ◆ 读压力分担给从节点，从库做横向扩容

MySQL主从架构下有什么问题?

- ◆ 主从延迟，读写分离



用户中台架构的思考

Redis存储空间不足

Redis分片集群
下有什么问题？

- ◆ 传统的主从，哨兵两种架构都是单点存储，如果存储数据过多，可以考虑使用分片集群架构
- ◆ 片键重定向，部分指令失效



流量增加的处理手段

面对流量的增加，核心的处理手段就是横向扩容，所以在进行架构设计的时候需要考虑到这一点





撸起袖子，干~

