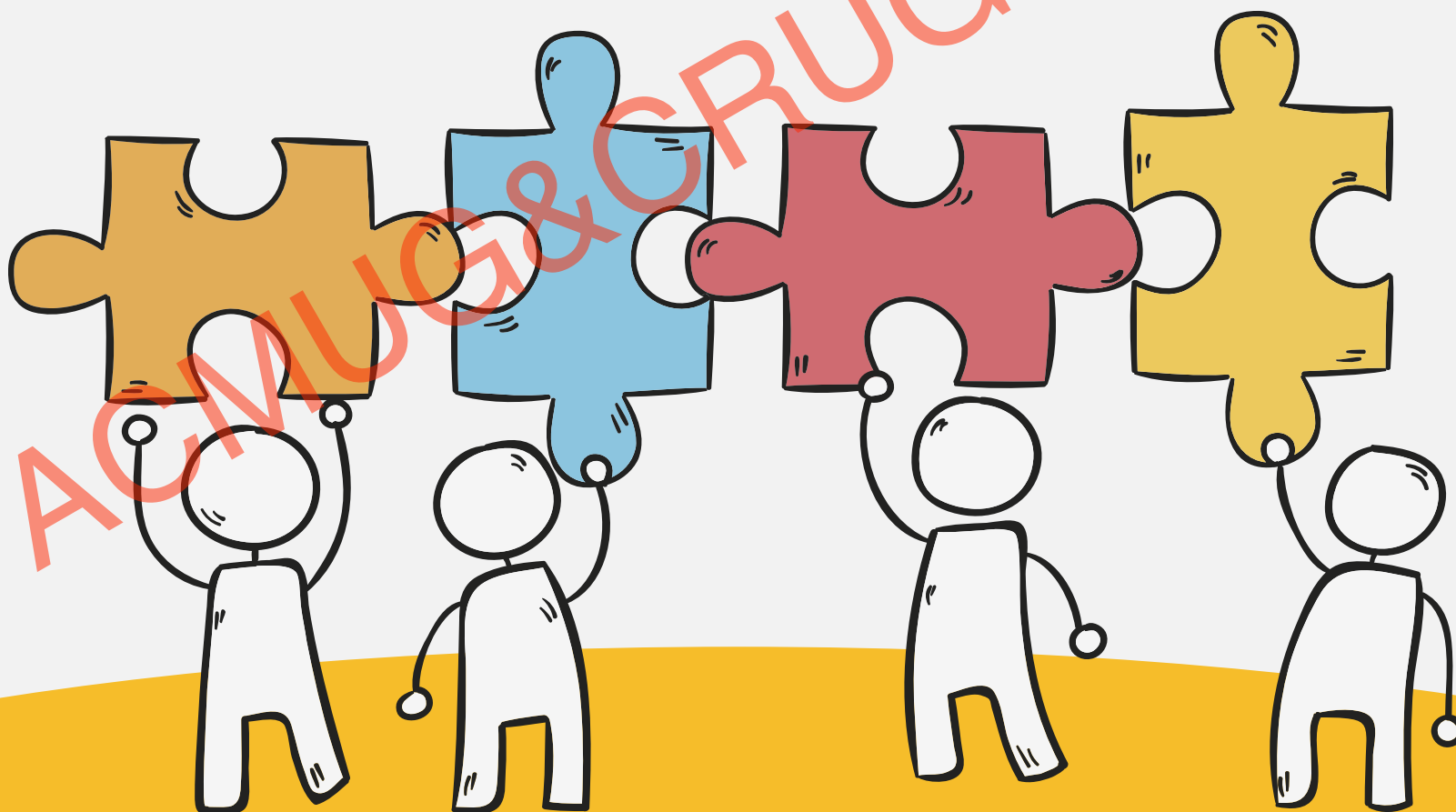


ACMUG & CRUG - 2018 - SZ

- Redis 高可用技术方案之华山论剑



CONTENTS

1

个人介绍

2

Redis简介

3

Redis高可用

4

Redis多活

01

自我介绍

介绍自己的工作经历和社会经验



自我介绍

极数云舟数据库架构师
极数学院联合创始人

Redis中国用户组发起人兼主席

前新浪微博高级DBA

阿里云MVP

ACMUG & CRUG 2018

02

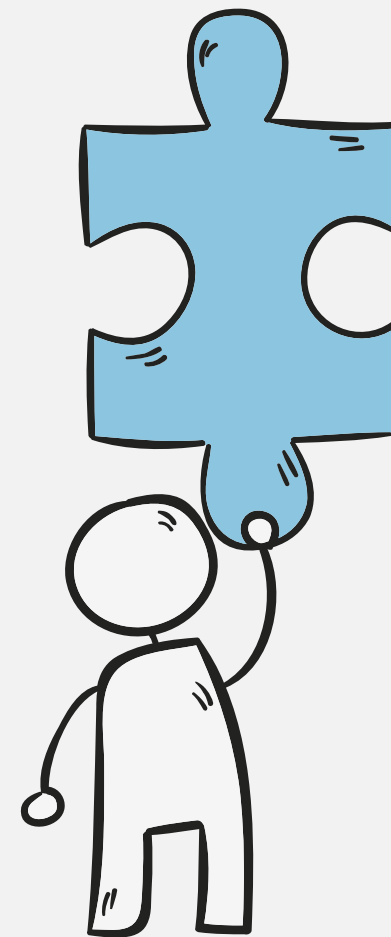
Redis简介

介绍Redis的特性

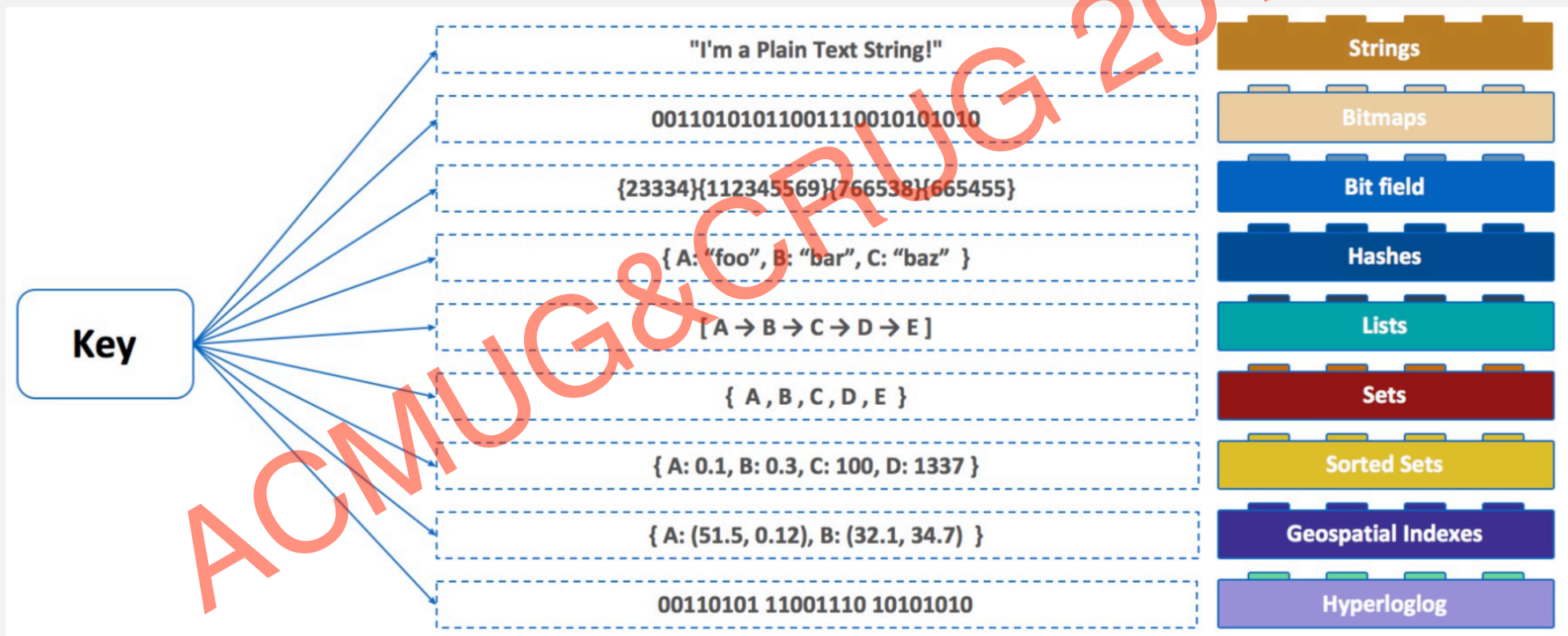
介绍Redis的数据结构

介绍Redis的内存管理策略

介绍Redis 4.0的新特性



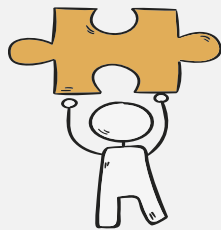
Redis数据结构



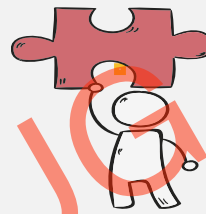
SKIPLIST vs ZIPLIST =? 高性能：高压缩率

Redis特性

单进程



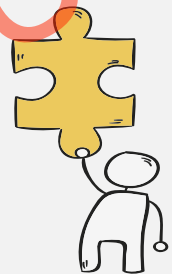
丰富的数据结构



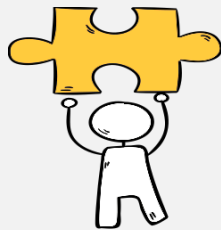
高可用



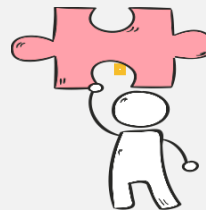
高性能



支持持久化



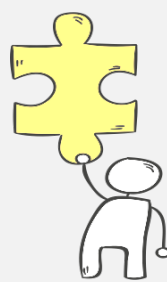
低延迟



分布式



自定义功能



Redis内存管理策略

1

no- eviction

禁止驱逐数据

2

volatile-lru

从已设置过期时间的数据集中挑选最近最少使用的数据淘汰

3

volatile-ttl

从已设置过期时间的数据集中挑选将要过期的数据淘汰

4

volatile-random

从已设置过期时间的数据集中任意选择数据淘汰

5

allkeys-lru

从数据集中挑选最近最少使用的数据淘汰

6

allkeys-random

从数据集中任意选择数据淘汰

7

another?

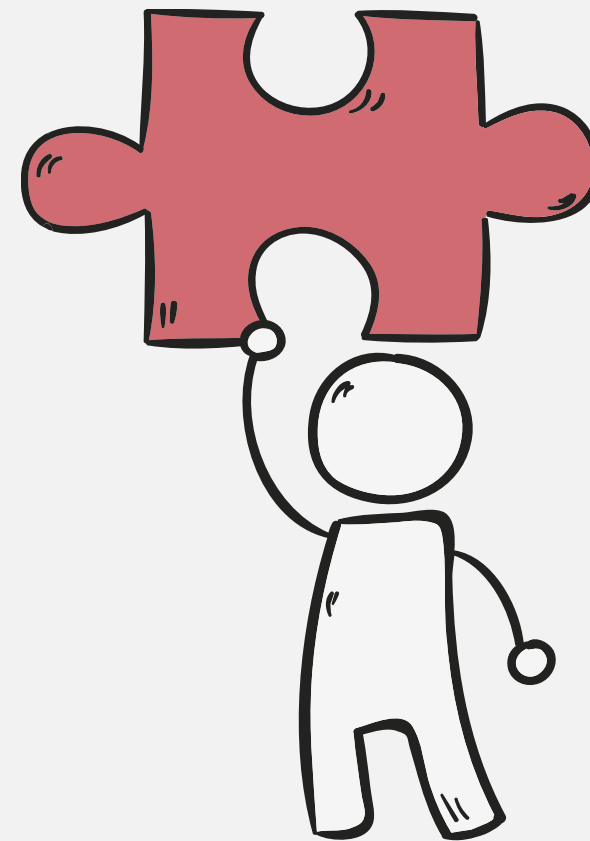
Redis 4.0新特性



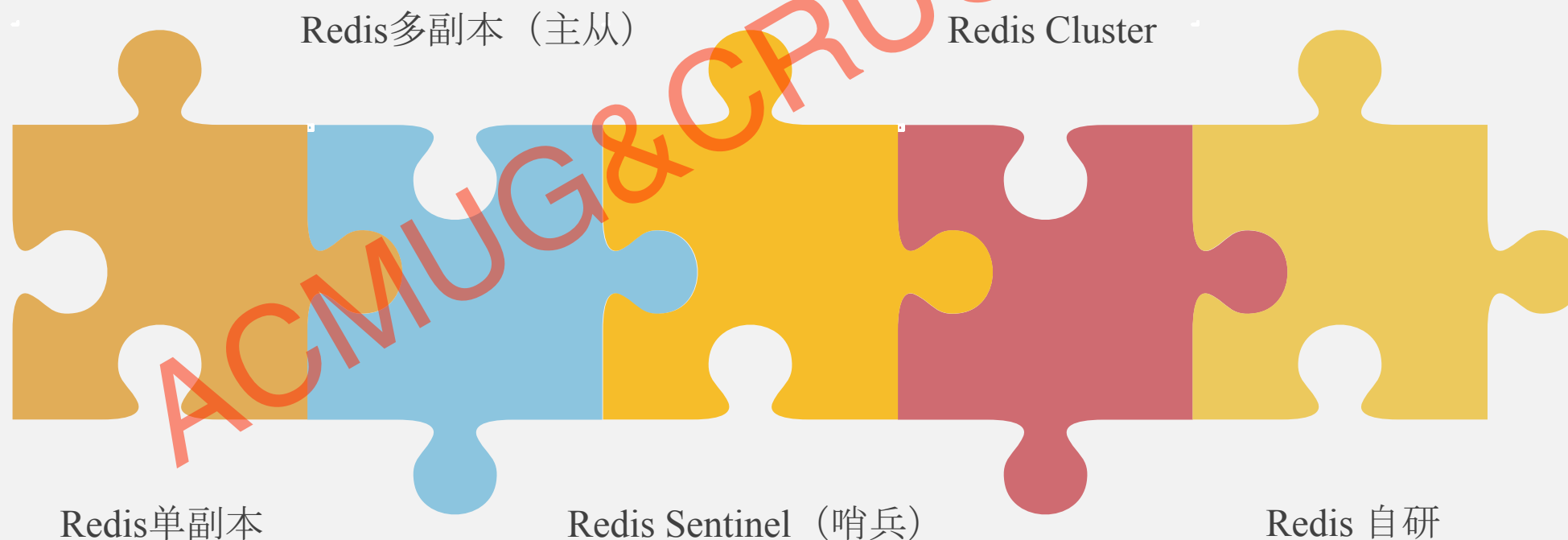
- ✓ 支持Module, 方便扩展自定义新功能
- ✓ PSYNC 2.0, 优化主从切换全量复制问题
- ✓ 提供LFU (Last Frequently Used) 算法, 优化缓存剔除
- ✓ 非阻塞DEL、FLUSHALL、FLUSHDB, 避免阻塞
- ✓ 提供AOF-RDB混合持久化策略, 数据更可靠
- ✓ 提供memory命令, 内存监控统计更全面
- ✓ 交换数据库功能swapdb, 减少select操作命令
- ✓ Redis Cluster兼容NAT和Docker
- ✓ 引入Jemalloc库, 优化内存访问

03 Redis高可用

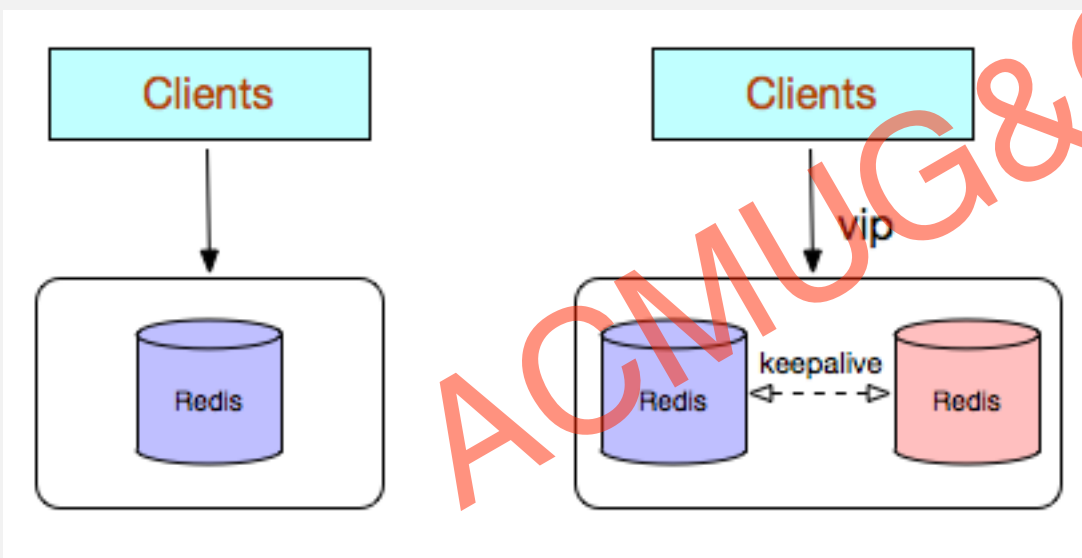
介绍Redis各种高可用的架构和优缺点



Redis高可用架构常见的5种形式



Redis单副本



- ◆ 优点
 - ✓ 架构简单、部署方便
 - ✓ 高性价比，当缓存使用时无需备用节点（单实例可用性可以用supervisor或crontab保证），当然为了满足业务的高可用性，也可以牺牲一个备用节点，但同时时刻只有一个实例对外提供服务。
 - ✓ 高性能
- ◆ 缺点
 - 不保证数据的可靠性
 - 当缓存使用，进程重启后，数据丢失，即使有备用的节点解决高可用性，但是仍然不能解决缓存预热问题，因此不适用于数据可靠性要求高的业务。
 - 高性能受限于单核CPU的处理能力（Redis是单线程机制），CPU为主要瓶颈，所以适合操作命令简单，排序、计算较少的场景。也可以考虑用memcached替代。

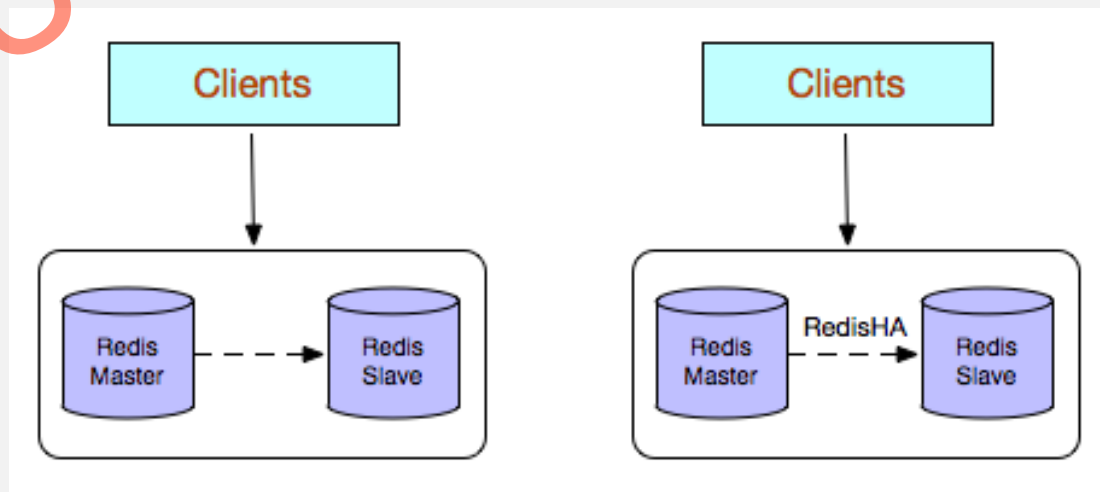
Redis多副本（主从）

◆ 优点

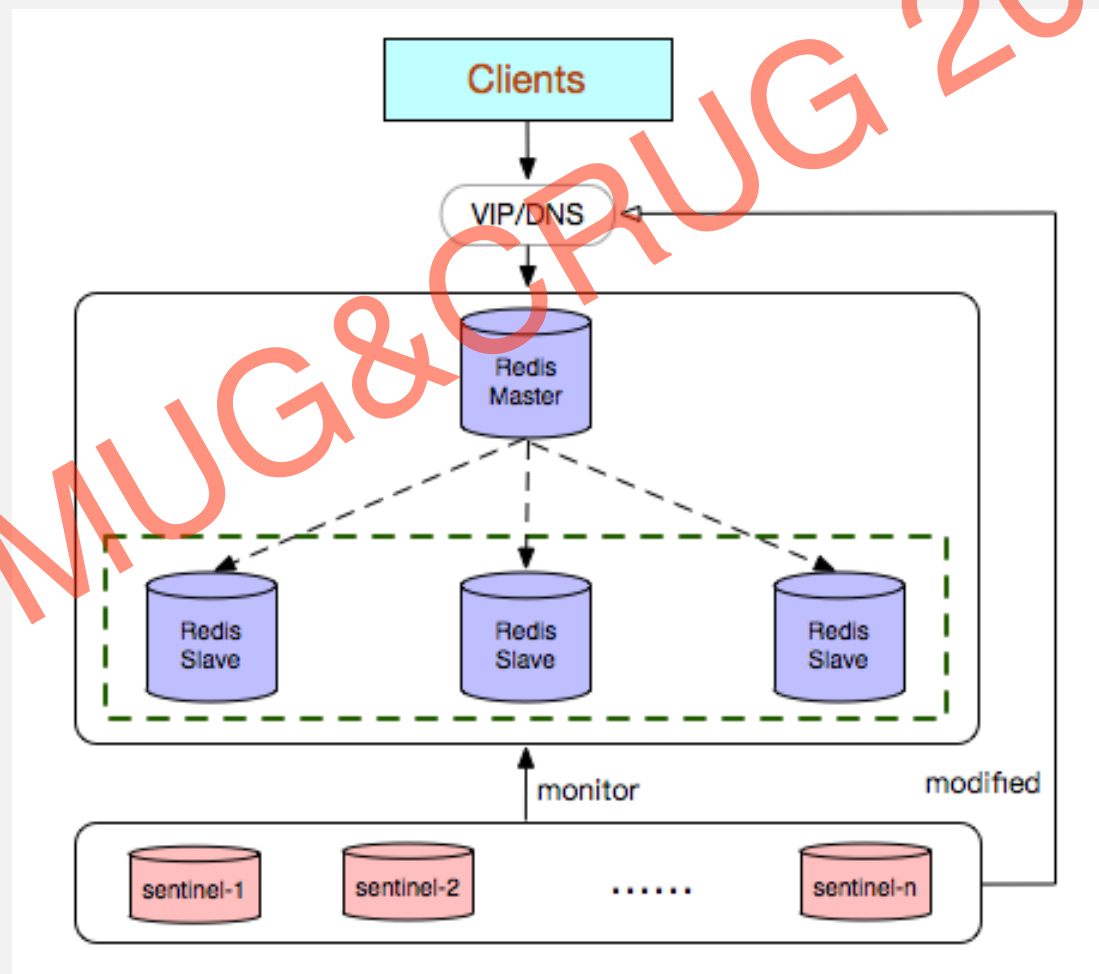
- ✓ 高可靠性，一方面，采用双机主备架构，能够在主库出现故障时自动进行主备切换，从库提升为主库提供服务，保证服务平稳运行。另一方面，开启数据持久化功能和配置合理的备份策略，能有效的解决数据误操作和数据异常丢失的问题。
- ✓ 读写分离策略，从节点可以扩展主库节点的读能力，有效应对大并发量的读操作。

◆ 缺点

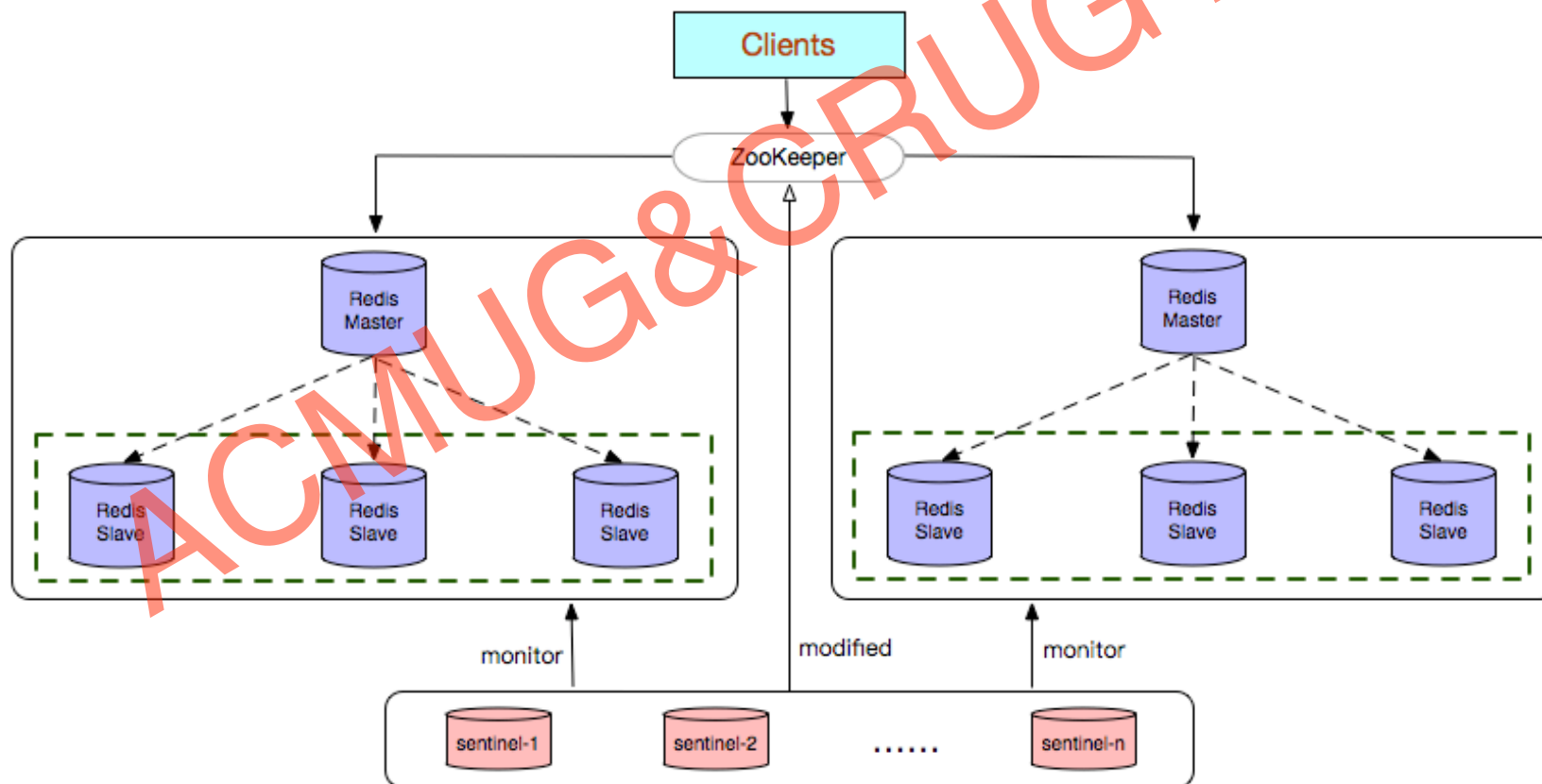
- 故障恢复复杂，如果没有RedisHA系统（需要开发），当主库节点出现故障时，需要手动将一个从节点晋升为主节点，同时需要通知业务方变更配置，并且需要让其他从库节点去复制新主库节点，整个过程需要人为干预，比较繁琐。
- 主库的写能力受到单机的限制，可以考虑分片
- 主库的存储能力受到单机的限制，可以考虑Pika
- 原生复制的弊端在早期的版本也会比较突出



Redis Sentinel (哨兵)



Redis Sentinel (哨兵)



Redis Sentinel（哨兵）

◆ 优点

- ✓ Redis Sentinel集群部署简单
- ✓ 能够解决Redis主从模式下的高可用切换问题
- ✓ 很方便实现Redis数据节点的线形扩展，轻松突破Redis自身单线程瓶颈，可极大满足对Redis大容量或高性能的业务需求。
- ✓ 可以实现一套Sentinel监控一组Redis数据节点或多组数据节点

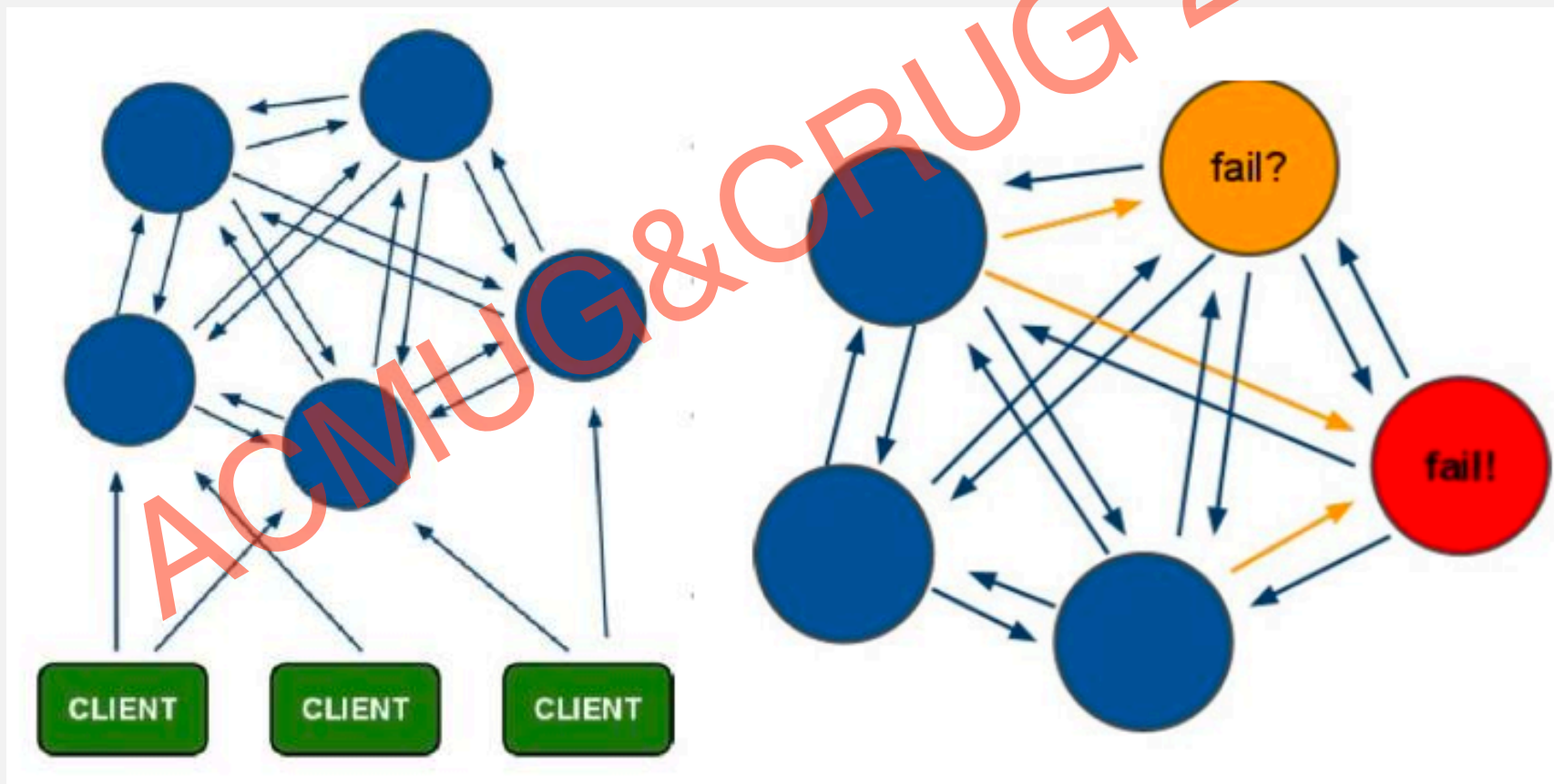
◆ 缺点

- 部署相对Redis主从模式要复杂一些，原理理解更繁琐
- 资源浪费，Redis数据节点中slave节点作为备份节点不提供服务
- Redis Sentinel主要是针对Redis数据节点中的主节点的高可用切换，对Redis的数据节点做失败判定分为主观下线和客观下线两种，对于Redis的从节点有对节点做主观下线操作，并不执行故障转移。
- 不能解决读写分离问题，实现起来相对复杂

建议：

1. 如果监控同一业务，可以选择一套Sentinel集群监控多组Redis数据节点的方案，反之选择一套Sentinel监控一组Redis数据节点的方案
2. 合理设置参数，防止误切，控制切换灵敏度控制
 - a) quorum
 - b) down-after-milliseconds 30000
 - c) failover-timeout 180000
 - d) maxclient 20000
 - e) timeout 3600
3. 部署的各个节点服务器时间尽量要同步，否则日志的时序性会混乱
4. Redis建议使用pipeline和multi-keys操作，减少RTT次数，提高请求效率
5. 自行搞定配置中心（zookeeper），方便客户端对实例的连接访问

Redis Cluster



Redis Cluster

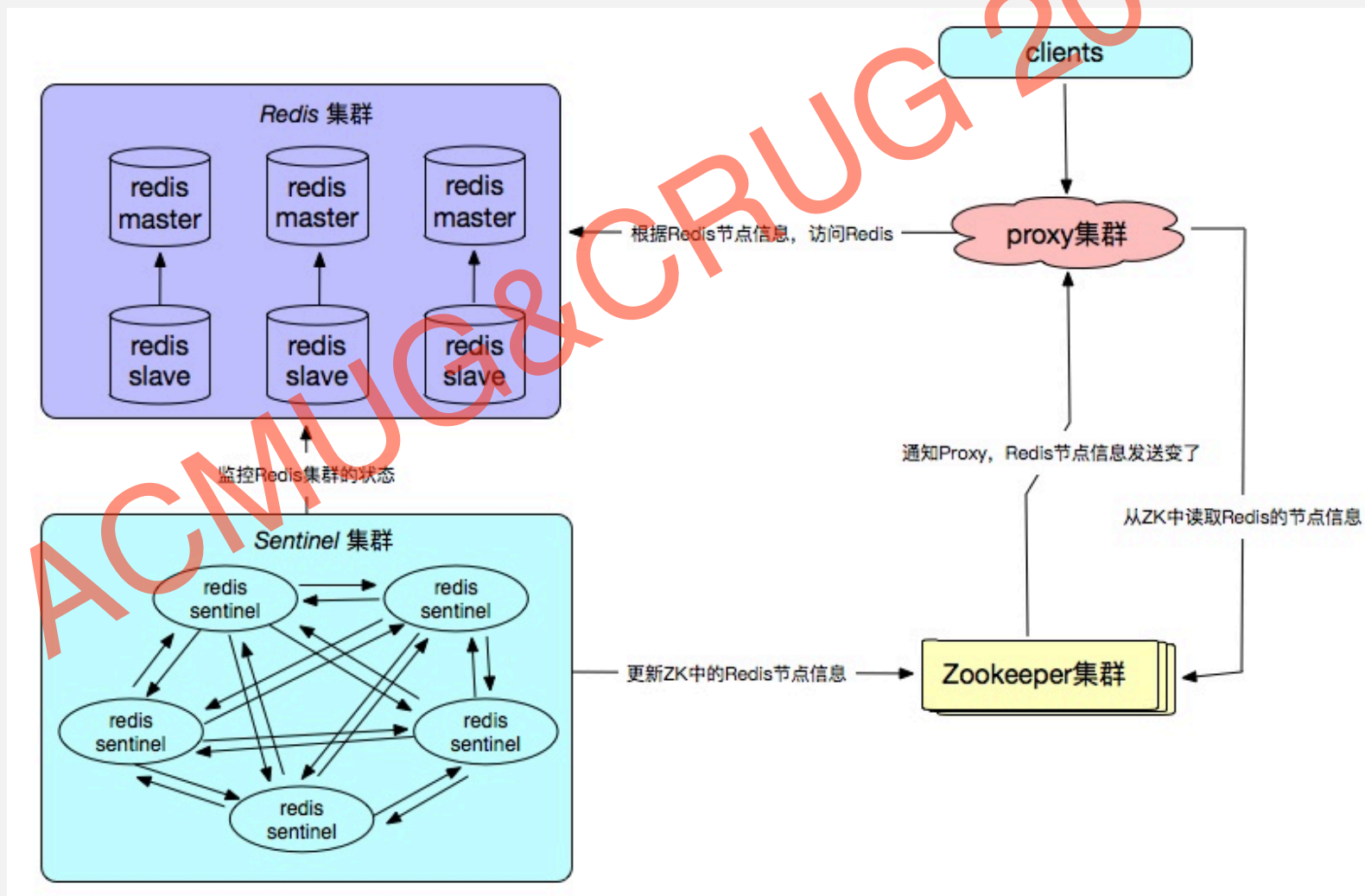
-
-
- ◆ 优点
 - ✓ 无中心架构
 - ✓ 数据按照slot存储分布在多个节点，节点间数据共享，可动态调整数据分布。
 - ✓ 可扩展性，可线性扩展到1000多个节点，节点可动态添加或删除。
 - ✓ 高可用性，部分节点不可用时，集群仍可用。通过增加Slave做standby数据副本，能够实现故障自动failover，节点之间通过gossip协议交换状态信息，用投票机制完成Slave到Master的角色提升。
 - ✓ 降低运维成本，提高系统的扩展性和可用性。

Redis Cluster

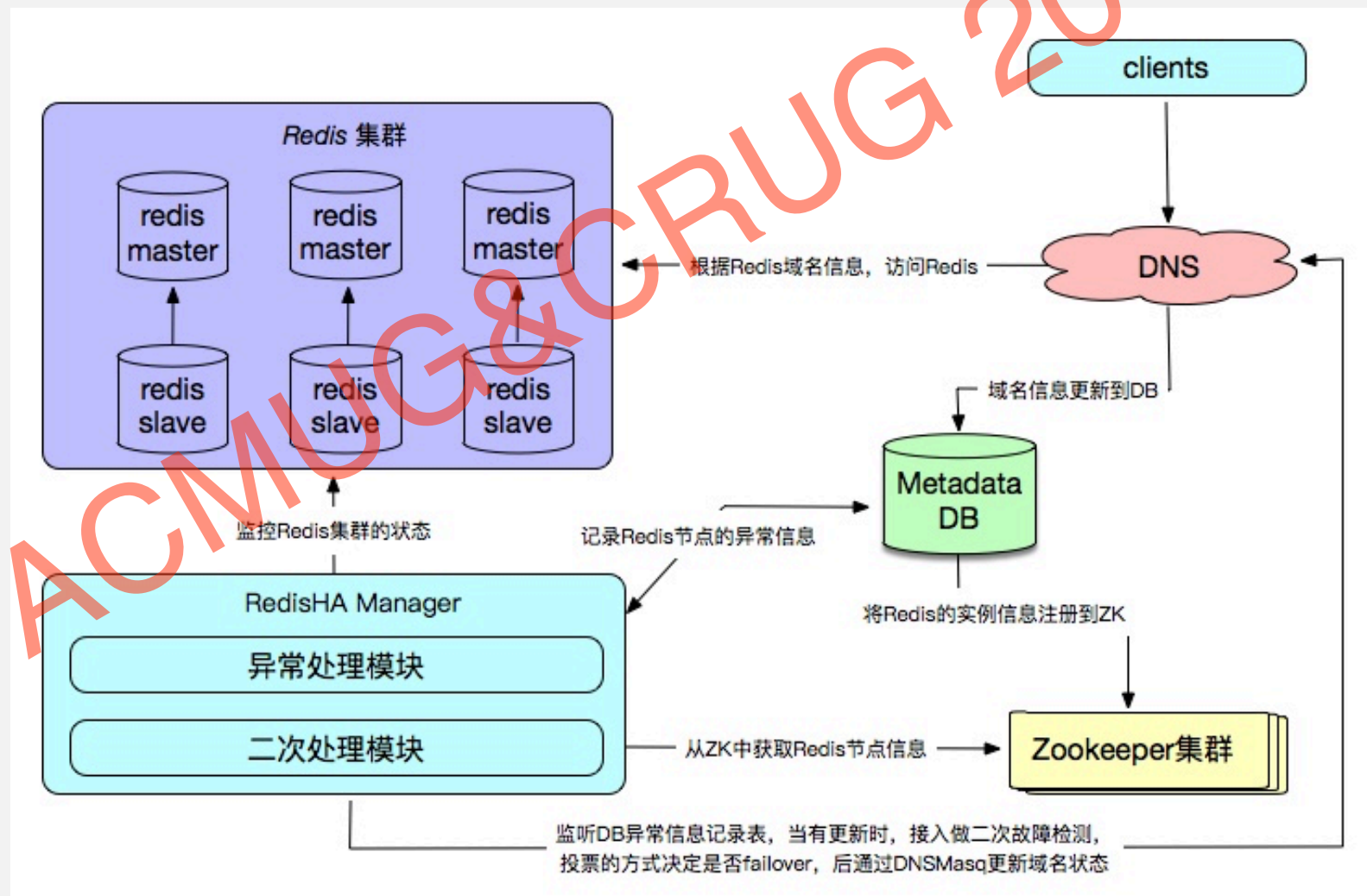
◆ 缺点

- Client实现复杂，驱动要求实现Smart Client，缓存slots mapping信息并及时更新，提高了开发难度，客户端的不成熟影响业务的稳定性。目前仅JedisCluster相对成熟，异常处理部分还不完善，比如常见的“max redirect exception”。
- 节点会因为某些原因发生阻塞（阻塞时间大于cluster-node-timeout），被判断下线，这种failover是没有必要的。
- 数据通过异步复制，不保证数据的强一致性。
- 多个业务使用同一套集群时，无法根据统计区分冷热数据，资源隔离性较差，容易出现相互影响的情况。
- Slave在集群中充当“冷备”，不能缓解读压力，当然可以通过SDK的合理设计来提高Slave资源的利用率。
- key批量操作限制，如使用mset、mget目前只支持具有相同slot值的key执行批量操作。对于映射为不同slot值的key由于执行mset、mget等操作可能存在于多个节点上因此被支持。
- key事务操作支持有限，只支持多key在同一节点上的事务操作，当多个key分布于不同的节点上时无法使用事务功能。
- key作为数据分区的最小粒度，因此不能将一个很大的键值对象如hash、list等映射到不同的节点。
- 不支持多数据库空间，单机下的redis可以支持到16个数据库，集群模式下只能使用1个数据库空间，即db 0。
- 复制结构只支持一层，从节点只能复制主节点，不支持嵌套树状复制结构。
- 避免产生hot-key，导致节点成为系统的短板。
- 避免产生big-key，导致网卡打爆、慢查询。
- 重试时间应该大于cluster-node-time时间
- Redis Cluster不建议使用pipeline和multi-keys操作，减少max redirect产生的场景。

Redis 自研



Redis 自研



Redis 自研

- ◆ 优点：

- ✓ 高可靠性、高可用性
- ✓ 自主可控性高
- ✓ 贴切业务实际需求，可缩性好，兼容性好

- ◆ 缺点：

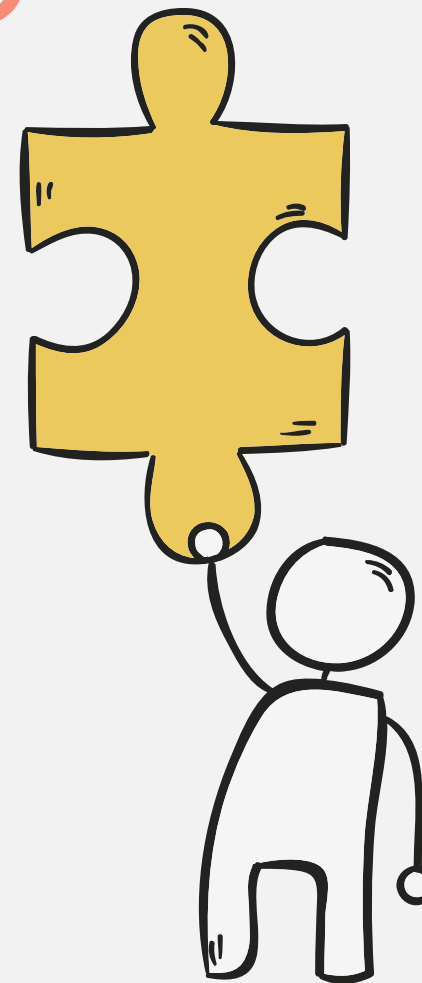
- 实现复杂，开发成本高
- 需要建立配套的周边设施，如监控，域名服务，存储元数据信息的数据库等。
- 维护成本高

04

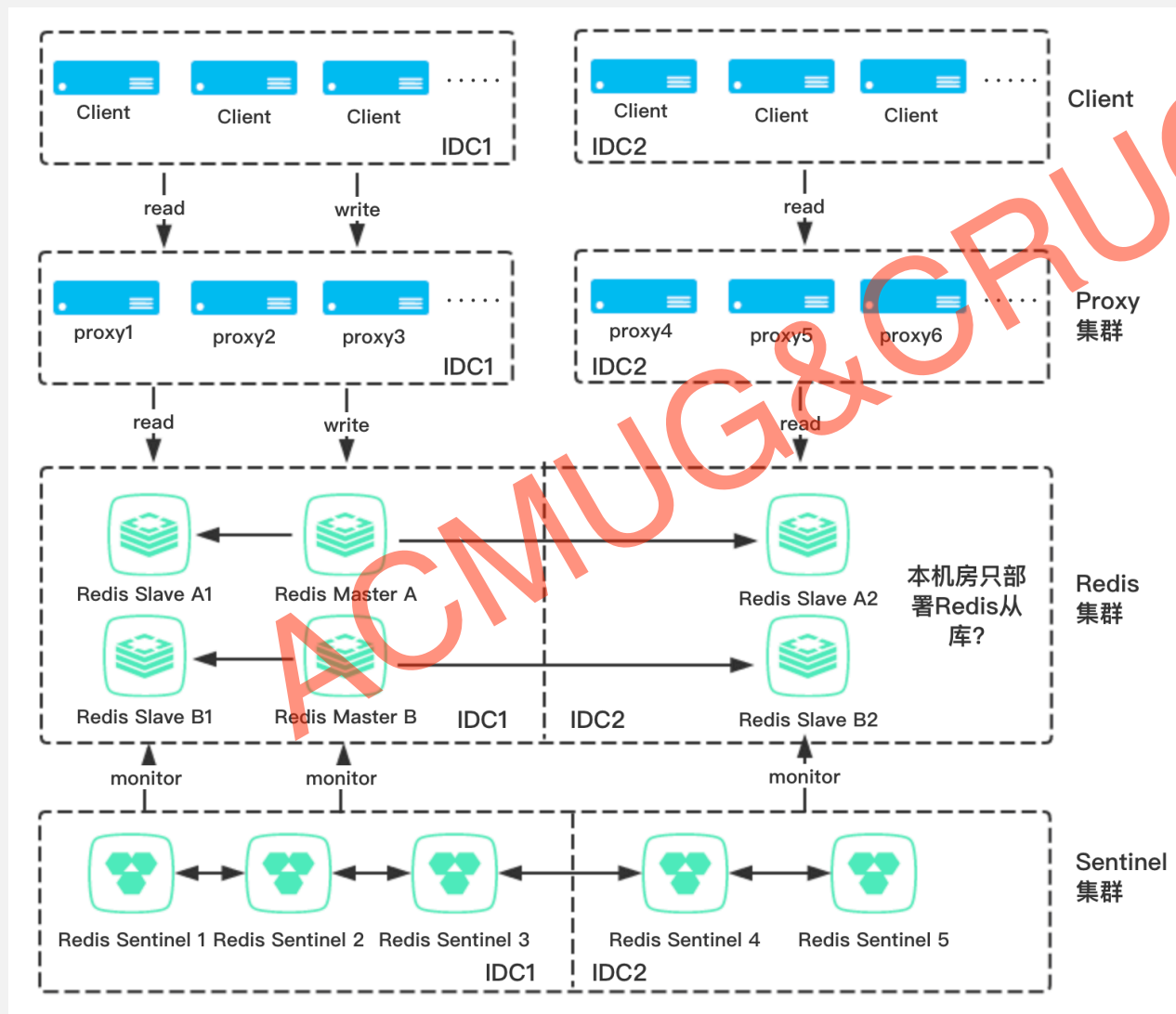
Redis 多活

Redis 多活的一些参考解决方案

抛砖引玉、集思广益、促进繁荣、建设生态



Redis多活架构 – 主从复制模式

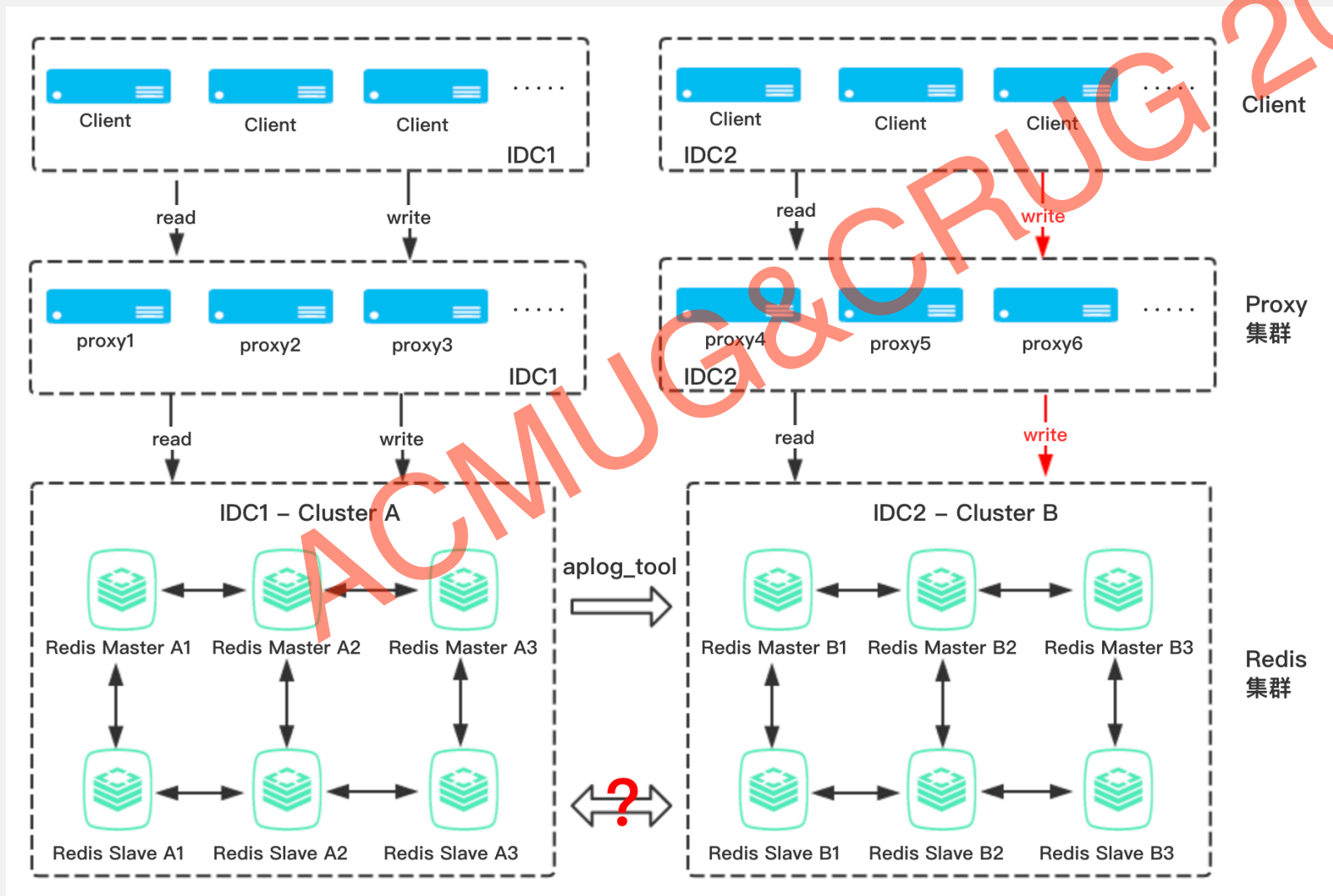


集群间如何实现数据的双向同步？

双写？

双向的主从同步？

Redis 多活 – Redis Cluster 集群模式



集群间如何实现数据的双向同步？

双写？

脑裂？

ACMUG & CRUG - 2018 - SZ

THANK YOU

