

Redis 高级讲义

峰云就她了

xiaorui.cc

github.com/rfyiamcool

key的规范

- * 加入业务的前缀
- * 长度控制在30个字符以内
- * 一级key不要超过千万
- * 同样ziplist类型的hash比strings省内存

Value的规范

- * 选择合适的数据结构
- * 长字符压缩存取 (snappy, msgpack, more...)
- * 避免big key (删除和迁移时阻塞)
- * 避免hot key (单点性能)

优化慢请求

- * 避免使用 $O(n)$ 的指令 (keys *, hgetall, smembers, sunion ...)
- * 使用scan, hscan, sscan, zscan
- * 业务层规避这类设计

提高吞吐

- * 使用pipeline批量传输, 减少网络RTT
- * 使用多值指令 (mset, hmset)
- * 使用script lua
- * 干掉aof ?

(big key) or (hot key)

- * big key

- * scan / small range get

- * del > unlink (redis 4.0 async del)

- * hash shard

- * hot key

- * hash shard

不推荐使用命令

- * pub sub

- * redis transction

- * more ...

redis lua

- * 减少RTT消耗
- * 保证多指令原子性
- * 自定义指令

lua 场景

- * zset的zpop

- * semaphore分布式锁

- * 自增id生成器

redis module

- * 注册新指令
- * 性能比redis lua更强劲
- * redis 4.0 以上

经历过的性能指标

- * 1w 的稳定长连接
- * 9w TPS
- * 队列千万级别
- * 百万数量key

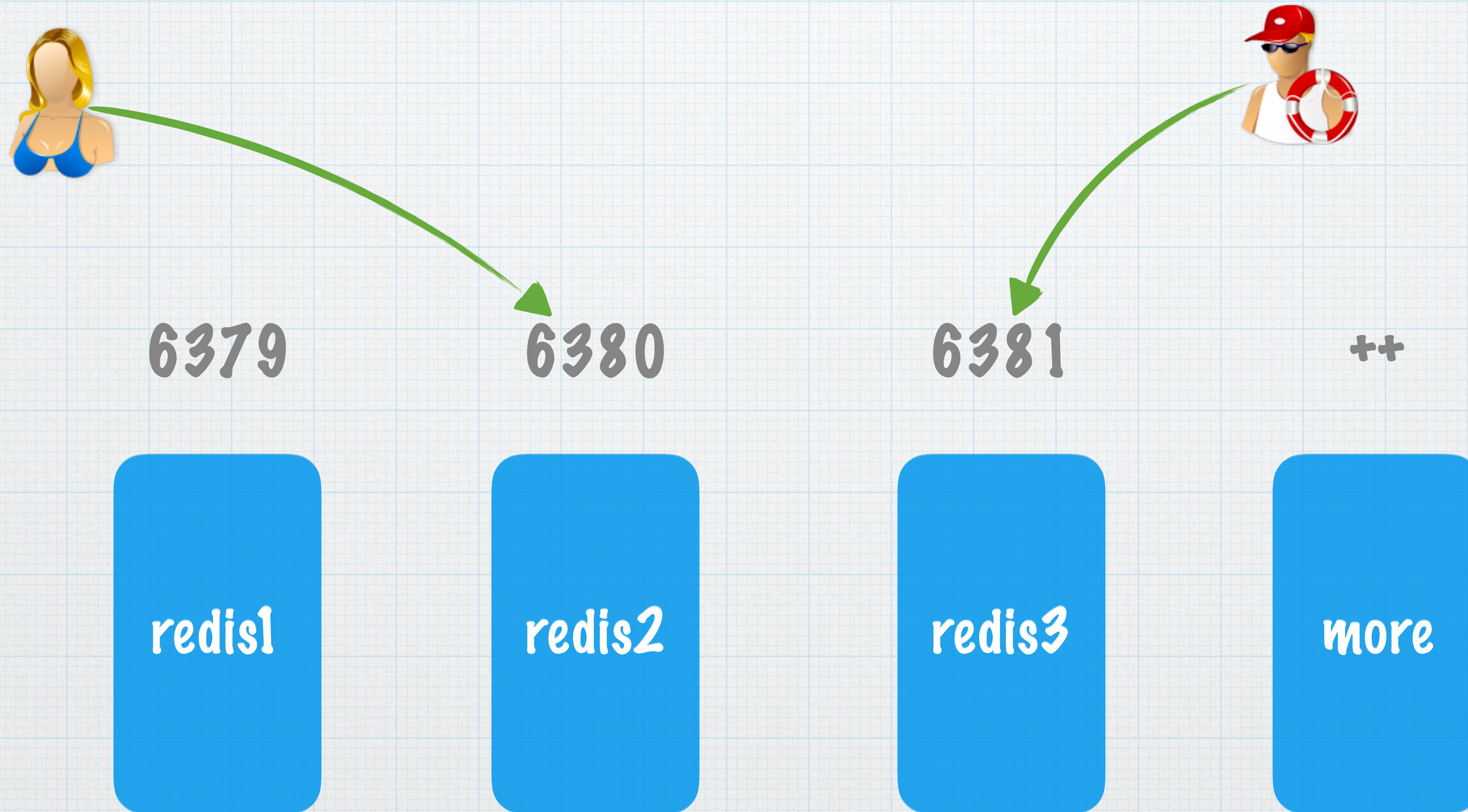
单节点

单机进化到多实例

- * 什么是多实例
- * 为什么要多实例化
- * 多实例化需要注意什么？

单机多实例

what 多实例



why 多实例

- * 最大程度的使用内存
- * 避免单实例RDB时
 - * 被kernel oom
 - * 使用swap造成阻塞.
- * 单实例启动太慢
- * 扩展, 迁移, 内存随便整理

- * copy on write will block

- * 绕开redis单工作线程的问题

 - * 阻塞指令

 - * busy event

 - * hashcrc

 - * more ...

How 多实例

- * 128G 总内存.

- * 11G 为一个实例, 启动个10实例.

- * 空出18G做缓冲.

- * 后台脚本来触发bgsave.

- * 启动时也是一个一个的启动

简约集群

* vip 多线程版 twemproxy

* codis

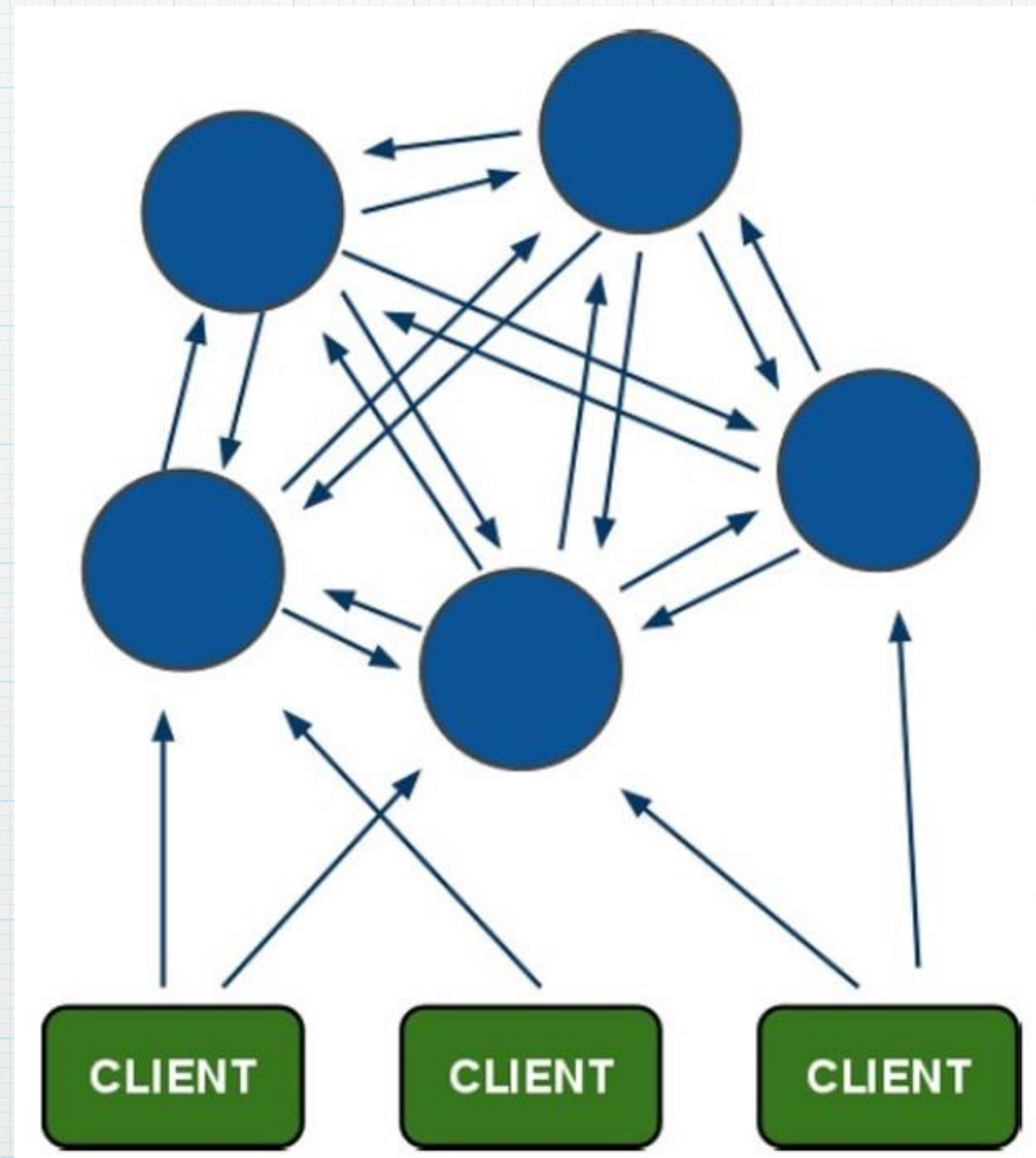
* redis cluster

集群

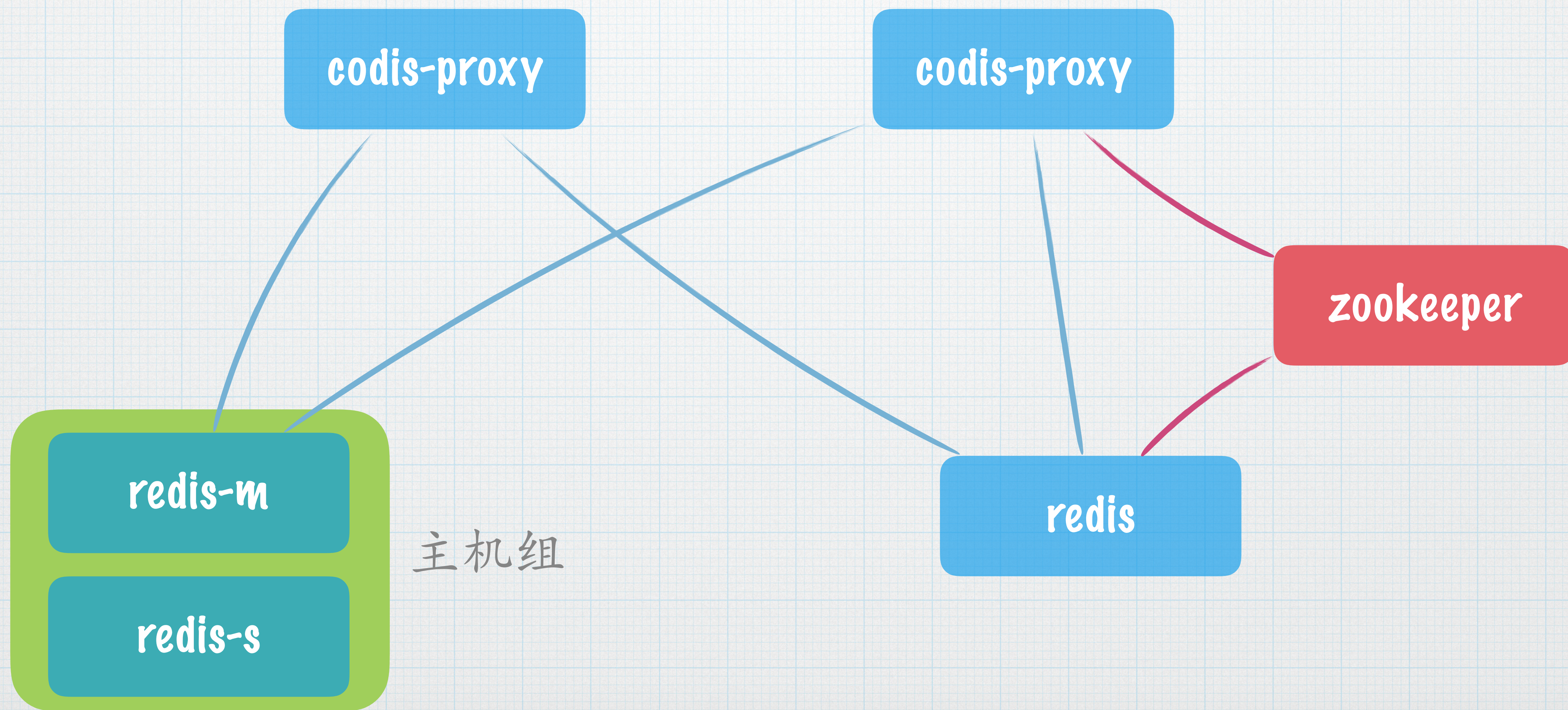
codis vs redis cluster

	cluster	codis
hash_tag	y	y
design	中心化	去中心化
pipeline	client move order	支持
slot	y	y
多租户	y	y
性能	high	this < cluster
code	复杂	简单
范围	广	也有不少大厂

redis cluster



codis



redis 使用的高级场景

- * 定时器

- * 去重优先级**fifo**队列

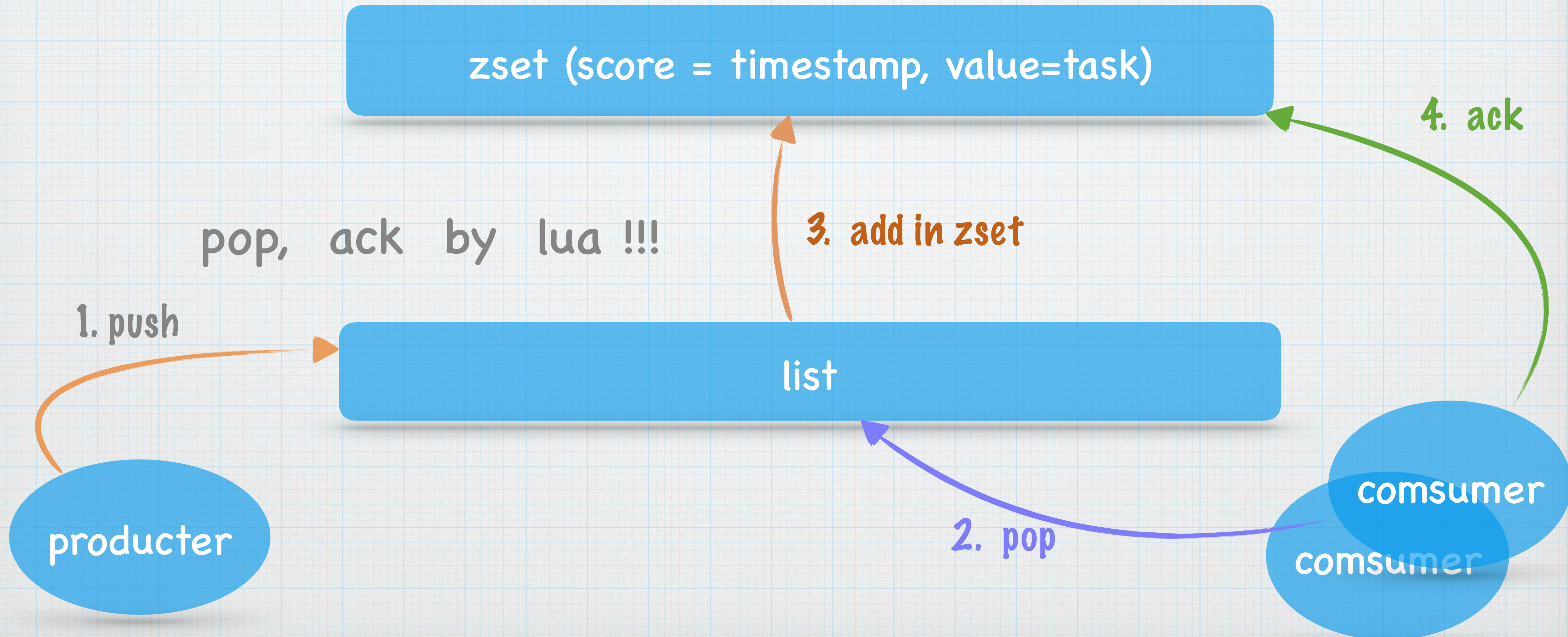
- * 分布式锁

- * More ...

定时器



伪ack队列



去重优先级的FIFO队列

set (作为去重特性)

queue_1

queue_2

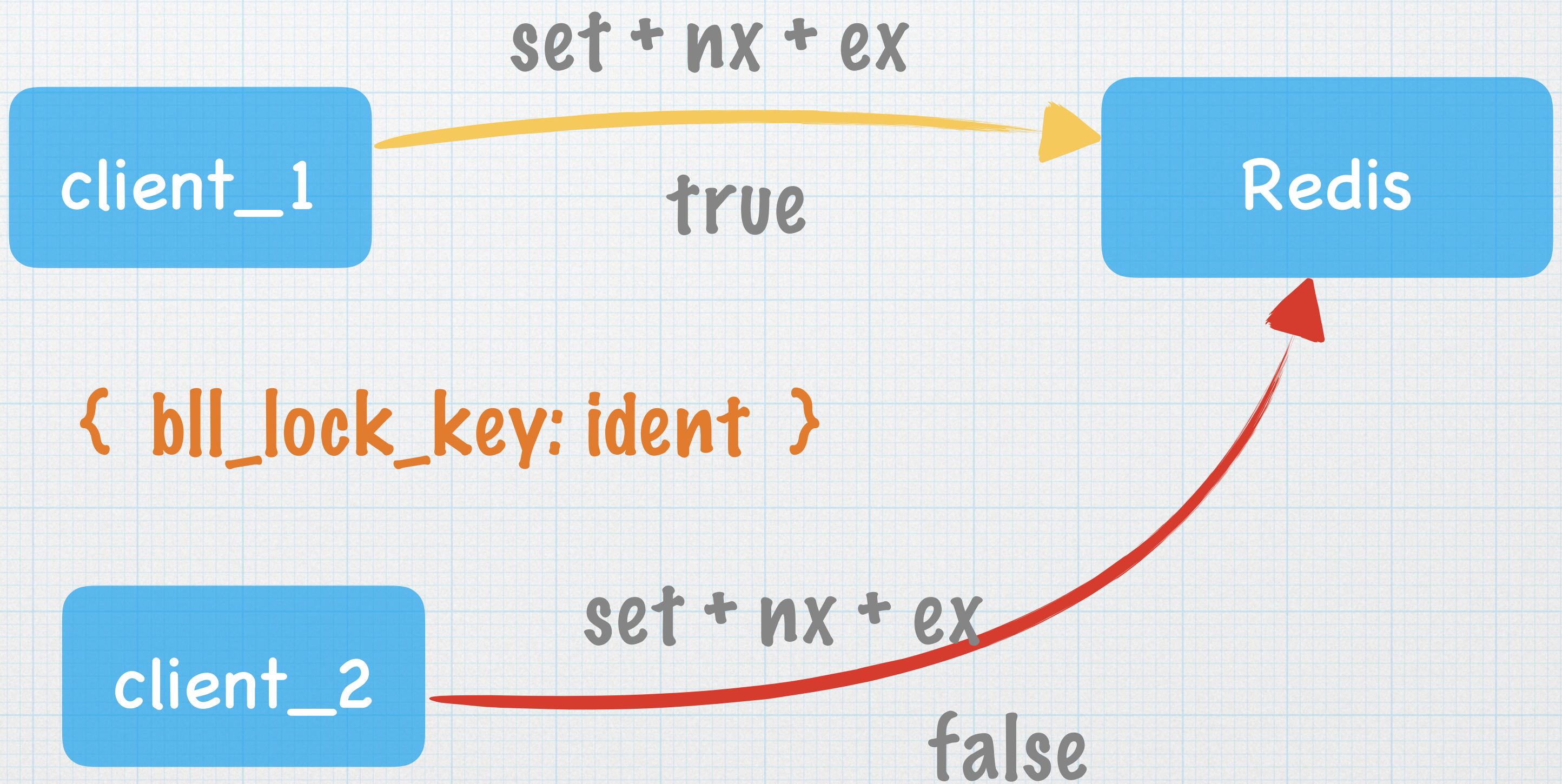
queue_3

* redis lua 封装增删改查

分布式锁

- * 安全可靠
 - * say no
- * 可重入锁
 - * say yes
- * 公平调度
 - * say hard

lua make (compare and set) !!!



排查问题

- * 外部

 - * redis-cli monitor

- * 内部

 - * keyspace

 - * slow log

* 内存碎片

* -- bigkeys

* string, bytes 空间

* set, list, zset, hash, 元素个数

* rdb tool

* 具体占用空间

监控

- * info -> instantaneous_ops_per_sec
- * info -> used_memory_human
- * /proc/{pid}/smaps
- * connected_clients

项目分享

- * https://github.com/rfyiamcool/go_redis_semaphore
- * https://github.com/rfyiamcool/zset_zpop
- * https://github.com/rfyiamcool/redis_unique_queue
- * https://github.com/rfyiamcool/redis_modules_ackqueue
- * https://github.com/rfyiamcool/go_redis_lock
- * <https://github.com/rfyiamcool/kvdis>

“别说话！”

一峰云就她了