

Redis

redis 安装

- [Redis 安装](#)
- [Redis 安装 key](#)
- [Redis 安装](#)

Redis ????



缓存穿透解决方案

1. 接口层增加校验：如用户鉴权校验，id做基础校验， $id \leq 0$ 的直接拦截；
2. 对查询接口为空的对象也进行缓存，将null也存入缓存中。要保证好时效性，设置失效时间。
3. 布隆过滤器：将所有可能存在的数据哈希到一个足够大的bitmap中，一个一定不存在的数据会被这个bitmap拦截掉，从而避免了对底层存储系统的查询压力。

?????key



如何找到大key

- **string类型通过命令查找**

```
redis-cli -h 127.0.0.1 -p6379 -a "password" --bigkeys
```

- **RdbTools工具**

```
rdb dump.rdb -c memory --bytes 10240 -f redis.csv
```



如何找到大key

- **直接删除大key会造成阻塞，因为redis是单线程执行，阻塞期间，其他所有请求可能都会超时。超时越来越多，会造成redis连接会耗尽，产生各种异常**
- **低峰期删除：凌晨，观察qps，选择低的时候，无法彻底解决阻塞**
- **分批次删除：对于hash，使用hscan扫描法，对于集合采用srandmember每次随机取数据进行删除。对于有序集合可以使用zremrangebyrank直接删除，对于列表直接pop即可。**
- **异步删除法：用unlink代替del来删除，这样redis会将这个key放入到一个异步线程中，进行删除，这样不会阻塞主线程。**

????????????

将库存名额加载到了Redis，必须要保证精确计数。如果redis的库存数量错误，就会导致超卖的问题产生。

在秒杀的时候分为两步。

第一步：判断库存名额是否充足。

第二步：减少库存名额，扣减成功就是抢到。

我们必须要保证两种操作的原子性。所以我们要使用lua脚本，redis的特性，专门支持原子。

还有一种情况，我们解决了超卖是防止超过我们的库存量，那么会不会有一种情况是我扣减了库存，但是消息没发出，订单生成失败，导致我没有卖成功呢。

我们扣减库存后，发送消息队列，这里要有重试策略，如果发送消息失败进行重试，超过重试次数后，则要持久化磁盘，由补偿服务来进行扫描，进行后续业务处理。类似于mysql的日志持久化。