

# Redis大厂面试汇总

---

## 缓存雪崩、缓存穿透、缓存击穿

---

答题思路：

- 三者的概念，发生场景、解决方案
- 三者的区别和影响

涉及知识点：Redis第七章：缓存问题—缓存穿透、缓存雪崩、缓存击穿

穿透：不存在的key

雪崩：大量的key失效

击穿：一个key或一些key 热点key

## 大Key，热点Key的处理

---

### Hot Key

答题思路：

- hot key的概念，场景，问题
- hot key的发现
- hot key的处理

涉及知识点：Redis第七章：缓存问题—hot key

### Big Key

答题思路：

- big key的概念、场景，影响  
String > 10k list大于5000个
- big key的发现
- big key的处理

涉及知识点：Redis第七章：缓存问题—big key

## 数据库一致，缓存失效，数据并发竞争

---

### 数据库一致

答题思路：

- Catch Aside Pattern
- 数据源不一致
- 场景的适用性（互联网）
- 保证最终一致，一致的时间处理

涉及知识点：

Redis第一章：Cache Aside Pattern

Redis第七章：数据不一致

## 缓存失效

答题思路：

缓存失效带来的问题：缓存穿透、缓存雪崩、缓存击穿（高并发）

会让数据库压力过大而宕机

redis的缓存过期策略：LRU

Redis设置的expiretime TTL

缓存失效的处理：

Redis做DB时，不能失效 保证数据的完整性，数据一致问题，定时任务，在DB变化后，更新缓存

可以失效但不穿DB，失效后读取本地缓存或服务熔断

异步更新DB，数据时时同步

涉及知识点：

Redis第七章：缓存穿透、缓存雪崩、缓存击穿

Redis第二章：缓存过期和淘汰策略

Redis第七章：缓存更新策略

## 数据并发竞争

答题思路：

- 数据并发竞争的概念、场景
- 数据并发竞争的影响
- 解决方案：
  - 将并发串行化：分布式锁+时间戳、利用队列
  - 使用CAS：秒杀

涉及知识点：

Redis第七章：数据并发竞争

Redis第七章：乐观锁

## 热点数据和冷数据是什么

---

答题思路：

热数据：hot key 位于Redis中 命中率尽量高

冷数据：不经常访问的数据 位于DB中

冷热的交换：maxmemory+allkeys LRU

交换比例：热20万、冷200万

Redis作为DB时，冷数据不能驱逐，保证数据的完整性

涉及知识点：

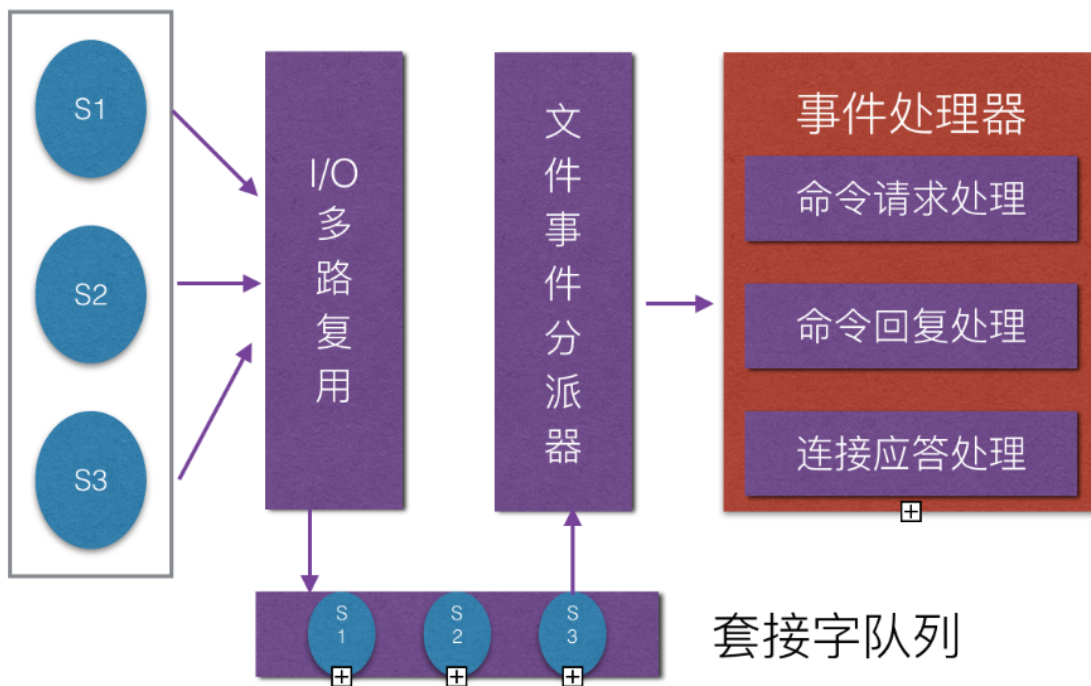
Redis第七章：hot key

Redis第二章：缓存过期和淘汰策略

## 单线程的redis为什么这么快

答题思路：

- redis在内存中操作，持久化只是数据的备份，正常情况下内存和硬盘不会频繁swap
- 多机主从，集群数据扩展
- maxmemory的设置+淘汰策略
- 数据结构简单，有压缩处理，是专门设计的
- 单线程没有锁，没有多线程的切换和调度，不会死锁，没有性能消耗
- 使用I/O多路复用模型，非阻塞IO；
- 构建了多种通信模式，进一步提升性能
- 进行持久化的时候会以子进程的方式执行，主进程不阻塞



涉及知识点：

Redis第二章：底层数据结构

Redis第二章：缓存过期和淘汰策略

Redis第三章：通信协议

Redis第三章：事件处理机制

Redis第四章：RDB和AOF

Redis第六章：主从、集群

## redis的过期策略以及内存淘汰机制

答题思路：

- 为什么要过期
- 什么情况下不能过期
- 如何设置过期
- expires 原理
- 如何选择缓存淘汰策略

涉及知识点：

Redis第二章：缓存过期和淘汰策略

## Redis 为什么是单线程的，优点

---

答题思路：

- Redis采用单线程多进程集群方案
- Redis是基于内存的操作，CPU不是Redis的瓶颈
- 瓶颈最有可能是机器内存的大小或者网络带宽
- 单线程的设计是最简单的
- 但是对多核CPU利用率不够，所以Redis6采用多线程。

单线程优点：

- 代码更清晰，处理逻辑更简单 不用去考虑各种锁的问题，
- 不存在加锁释放锁操作，没有因为可能出现死锁而导致的性能消耗
- 不存在多进程或者多线程导致的切换而消耗CPU

涉及知识点：

Redis第二章：缓存过期和淘汰策略

Redis第三章：通信协议

Redis第三章：事件处理机制

## 如何解决redis的并发竞争key问题

---

Rediskey的设计，尽量不竞争

必须竞争：秒杀、分布式锁

同数据并发竞争

## Redis分布式锁问题

---

答题思路：

- 分布式锁的概念，应用场景
- Redis的实现方式
- 分布式锁的本质分析
- redis、zookeeper、etcd三者的对比和应用场景
- redisson的使用

涉及知识点：Redis第七章：分布式锁

## 有没有尝试进行多机redis 的部署？如何保证数据一致的？

---

答题思路：

- redis多机部署方案：Redis主从+哨兵、codis集群、RedisCluster
- 多机：高可用、高扩展、高性能
- 三者的区别，适用场景
- 数据一致性指的是主从的数据一致性
- Redis是AP模型，主从同步有时延。所以不能保证主从数据的时时一致性，只能保证数据最终一致性

- 保证数据一致性方案：

#### 1、忽略

如果业务能够允许短时间不同步就忽略，比如：搜索，消息，帖子，职位

#### 2、强制读主库，从库只做备份使用

- 使用一个高可用主库提供数据库服务
- 读和写都落到主库上
- 采用缓存来提升系统读性能

#### 3、选择性读主

写主库时将哪个库，哪个表，哪个主键三个信息拼装一个key设置到cache里

读时先在cache中查找：

cache里有这个key，说明1s内刚发生过写请求，数据库主从同步可能还没有完成，此时就应该去主库查询

cache里没有这个key，说明最近没有发生过写请求，此时就可以去从库查询

涉及知识点：

Redis第六章：主从复制、哨兵模式、proxy端分区、官方cluster分区