

# Java 最常见的 208 道面试题：第十八模块答案

Java团长 5天前

## 十八、Redis

### 179. redis 是什么？都有哪些使用场景？

Redis是一个开源的使用ANSI C语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value数据库，并提供多种语言的API。

Redis 使用场景：

- 数据高并发的读写
- 海量数据的读写
- 对扩展性要求高的数据

### 180. redis 有哪些功能？

- 数据缓存功能
- 分布式锁的功能
- 支持数据持久化
- 支持事务
- 支持消息队列

### 181. redis 和 memecache 有什么区别？

- memcached所有的值均是简单的字符串，redis作为其替代者，支持更为丰富的数据类型
- redis的速度比memcached快很多
- redis可以持久化其数据

### 182. redis 为什么是单线程的？

因为 cpu 不是 Redis 的瓶颈，Redis 的瓶颈最有可能是机器内存或者网络带宽。既然单线程容易实现，而且 cpu 又不会成为瓶颈，那就顺理成章地采用单线程的方案了。

关于 Redis 的性能，官方网站也有，普通笔记本轻松处理每秒几十万的请求。

而且单线程并不代表就慢 nginx 和 nodejs 也都是高性能单线程的代表。

### 183. 什么是缓存穿透？怎么解决？

缓存穿透：指查询一个一定不存在的数据，由于缓存是不命中时需要从数据库查询，查不到数据则不写入缓存，这将导致这个不存在的数据每次请求都要到数据库去查询，造成缓存穿透。

解决方案：最简单粗暴的方法如果一个查询返回的数据为空（不管是数据不存在，还是系统故障），我们就把这个空结果进行缓存，但它的过期时间会很短，最长不超过五分钟。

## 184. redis 支持的数据类型有哪些？

string、list、hash、set、zset。

## 185. redis 支持的 java 客户端都有哪些？

Redisson、Jedis、lettuce等等，官方推荐使用Redisson。

## 186. jedis 和 redisson 有哪些区别？

Jedis是Redis的Java实现的客户端，其API提供了比较全面的Redis命令的支持。

Redisson实现了分布式和可扩展的Java数据结构，和Jedis相比，功能较为简单，不支持字符串操作，不支持排序、事务、管道、分区等Redis特性。Redisson的宗旨是促进使用者对Redis的关注分离，从而让使用者能够将精力更集中地放在处理业务逻辑上。

## 187. 怎么保证缓存和数据库数据的一致性？

- 合理设置缓存的过期时间。
- 新增、更改、删除数据库操作时同步更新 Redis，可以使用事物机制来保证数据的一致性。

## 188. redis 持久化有几种方式？

Redis 的持久化有两种方式，或者说有两种策略：

- RDB (Redis Database)：指定的时间间隔能对你的数据进行快照存储。
- AOF (Append Only File)：每一个收到的写命令都通过write函数追加到文件中。

## 189. redis 怎么实现分布式锁？

Redis 分布式锁其实就是在系统里面占一个“坑”，其他程序也要占“坑”的时候，占用成功了就可以继续执行，失败了就只能放弃或稍后重试。

占坑一般使用 setnx(set if not exists)指令，只允许被一个程序占有，使用完调用 del 释放锁。

## 190. redis 分布式锁有什么缺陷？

Redis 分布式锁不能解决超时的问题，分布式锁有一个超时时间，程序的执行如果超出了锁的超时时间就会出现問題。

## 191. redis 如何做内存优化？

尽可能使用散列表 (hashes)，散列表 (是说散列表里面存储的数少) 使用的内存非常小，所以你应该尽可能的将你的数据模型抽象到一个散列表里面。

比如你的web系统中有一个用户对象，不要为这个用户的名称，姓氏，邮箱，密码设置单独的key,而是应该把这个用户的所有信息存储到一张散列表里面。

## 192. redis 淘汰策略有哪些？

- volatile-lru：从已设置过期时间的数据集 (server. db[i]. expires) 中挑选最近最少使用的数据淘汰。
- volatile-ttl：从已设置过期时间的数据集 (server. db[i]. expires) 中挑选将要过期的数据淘汰。
- volatile-random：从已设置过期时间的数据集 (server. db[i]. expires) 中任意选择数据淘汰。
- allkeys-lru：从数据集 (server. db[i]. dict) 中挑选最近最少使用的数据淘汰。
- allkeys-random：从数据集 (server. db[i]. dict) 中任意选择数据淘汰。
- no-eviction (驱逐)：禁止驱逐数据。

## 193. redis 常见的性能问题有哪些？该如何解决？

- 主服务器写内存快照，会阻塞主线程的工作，当快照比较大时对性能影响是非常大的，会间断性暂停服务，所以主服务器最好不要写内存快照。
- Redis 主从复制的性能问题，为了主从复制的速度和连接的稳定性，主从库最好在同一个局域网内。

(完)

## Java团长

专注于Java干货分享



扫描上方二维码获取更多Java干货