



1. Redis 是什么？都有哪些使用场景？

Redis 是一个使用 C 语言开发的高速缓存数据库。

Redis 使用场景：

- 记录帖子点赞数、点击数、评论数；
- 缓存近期热帖；
- 缓存文章详情信息；
- 记录用户会话信息。

2. 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

3. Redis 和 memcache 有什么区别？

- 存储方式不同：memcache 把数据全部存在内存之中，断电后会挂掉，数据不能超过内存大小；Redis 有部份存在硬盘上，这样能保证数据的持久性。
- 数据支持类型：memcache 对数据类型支持相对简单；Redis 有复杂的数据类型。

关注公众号：Java 编程专栏，获取最新面试题，架构师资料



- 使用底层模型不同：它们之间底层实现方式，以及与客户端之间通信的应用协议不一样，Redis 自己构建了 vm 机制，因为一般的系统调用系统函数的话，会浪费一定的时间去移动和请求。
- value 值大小不同：Redis 最大可以达到 512mb；memcache 只有 1mb。

4. 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。

5. 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。

6. Redis 支持的数据类型有哪些？

Redis 支持的数据类型：string（字符串）、list（列表）、hash（字典）、set（集合）、zset（有序集合）。

7. Redis 支持的 Java 客户端都有哪些？

支持的 Java 客户端有 Redisson、jedis、lettuce 等。

8. jedis 和 Redisson 有哪些区别？

- jedis：提供了比较全面的 Redis 命令的支持。

关注公众号：Java 编程专栏，获取最新面试题，架构师资料



- Redisson：实现了分布式和可扩展的 Java 数据结构，与 jedis 相比 Redisson 的功能相对简单，不支持排序、事务、管道、分区等 Redis 特性。

9. 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。

10. 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。

11. 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。

12. 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问题。

关注公众号：Java 编程专栏，获取最新面试题，架构师资料



13. Redis 如何做内存优化？

尽量使用 Redis 的散列表，把相关的信息放到散列表里面存储，而不是把每个字段单独存储，这样可以有效的减少内存使用。比如将 Web 系统的用户对象，应该放到散列表里面再

整体存储到 Redis，而不是把用户的姓名、年龄、密码、邮箱等字段分别设置 key 进行存储。

14. 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。

15. 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。

关注公众号：Java 编程专栏，获取最新面试题，架构师资料