

# 写在前面

本系列文章，收集了市面上各大互联网公司面试题。笔者小牛（小牛的公众号名片二维码放文末啦）也开发了个人网站,也有不少热心的网友贡献了面试经历。题库还处于更新中，欢迎大家访问。

<http://interviewtop.top>

InterviewTop汇总互联网知识点面试题

目前支持的题库：

公司

字节跳动  
美团  
快手  
腾讯  
百度  
网易  
京东  
华为

岗位

题库

2022届秋招日程汇总表

进入interviewtop

目前题库覆盖公司有：各类银行科技岗，百度，阿里，字节，腾讯，美团，快手，网易，华为，shopee，京东。

腾讯

全部  
银行科技岗  
阿里巴巴  
字节跳动  
美团  
快手  
腾讯

后端

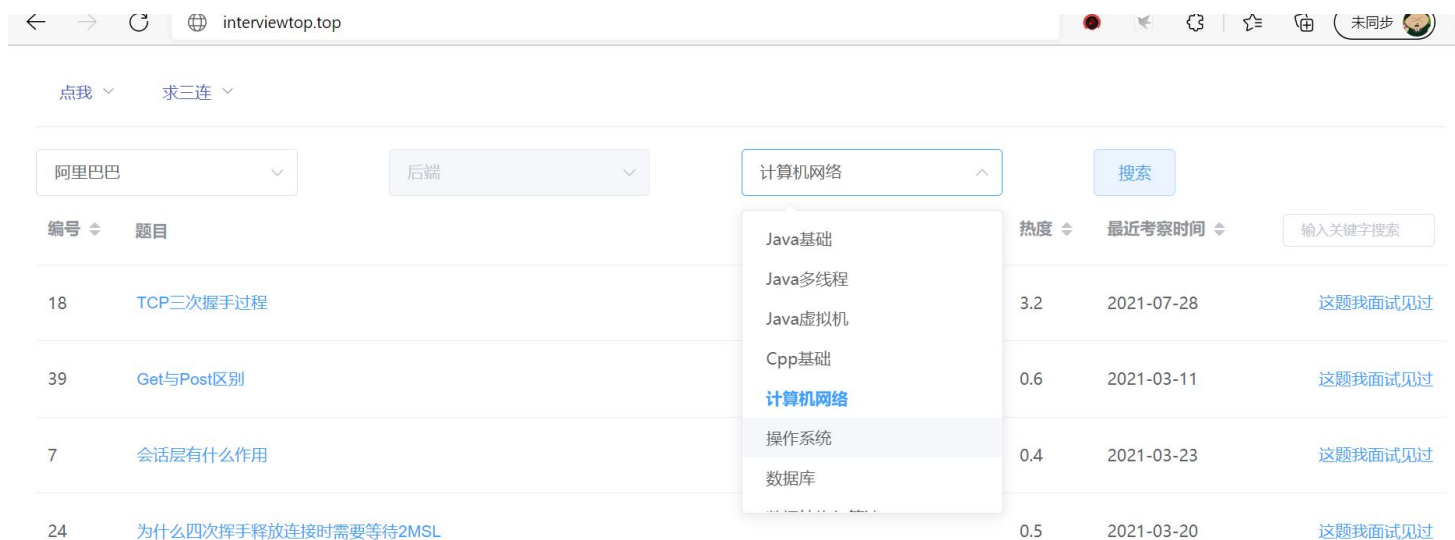
Cpp基础

搜索

热度 最近考察时间 输入关键词搜索

0.5	2021-03-12	<a href="#">这题我面试见过</a>
0.3	2021-03-21	<a href="#">这题我面试见过</a>
0.5	2021-03-15	<a href="#">这题我面试见过</a>
0.6	2021-03-10	<a href="#">这题我面试见过</a>
0.6	2021-03-16	<a href="#">这题我面试见过</a>

题库包括：计算网络，操作系统，数据库，Redis，Java基础，Java多线程，Java虚拟机，设计模式，数据结构与算法。



大家可以进入网站搜索大厂对应题库的高频面试题。

## Redis单线程原理

首先必须明确，Redis单线程指的是网络请求模块使用了一个线程（，其他模块仍用了多个线程。并不是一个线程完成了所有功能。

原理上，其采用了利用epoll的多路复用特性，因此可以采用单线程处理其网络请求。

## Redis数据类型

String：字符串类型，最简单的类型

Hash：类似于Map的一种结构。

List：有序列表。

Set：无序集合。

ZSet：带权值的无序集合，即每个ZSet元素还另有一个数字代表权值，集合通过权值进行排序。

## 什么情况下使用redis

1. 针对热点数据进行缓存
2. 对于特定限时数据的存放
3. 针对带热点权值数据的排序list
4. 分布式锁

## redis与memcache的区别

1. redis处理网络请求采用单线程模型，而memcache采用多线程异步IO的方式
2. redis支持数据持久化，memcache不支持
3. redis支持的数据格式比memcache更多

## 简述缓存穿透

缓存穿透指缓存和数据库均没有需要查询的数据，攻击者不断发送这种请求，使数据库压力过大。

## 简述缓存穿透的解决方法

1. 在数据库操作访问前进行校验，对不合法请求直接返回。
2. 对于经常被访问的，并且数据库没有的键，缓存层记录键=null。

## 简述缓存击穿

缓存击穿指缓存中没有数据，但数据库中有该数据。一般这种情况指特定数据的缓存时间到期，但由于并发用户访问该数据特别多，因此去数据库去取数据，引起数据库访问压力过大

## 简述缓存穿透的解决方法

1. 设置热点数据永远不过期。
2. 对并发读数据设置并发锁，降低并发性

## 简述缓存雪崩

缓存雪崩指缓存中一大批数据到过期时间，而从缓存中删除。但该批数据查询数据量巨大，查询全部走数据库，造成数据库压力过大。

## 简述缓存雪崩的解决方法

1. 缓存数据设置随机过期时间，防止同一时间大量数据过期。
2. 设置热点数据永远不过期。
3. 对于集群部署的情况，将热点数据均与分布在不同缓存中。

## Redis有哪些集群部署方式

1. 主从复制
2. 哨兵模式
3. Cluster集群模式

## 简述主从复制模式

在主从复制中，有主库（Master）节点和从库（Slave）节点两个角色。

从节点服务启动会连接主库，并向主库发送SYNC命令。

主节点收到同步命令，启动持久化工作，工作执行完成后，主节点将传送整个数据库文件到从库，从节点接收到数据库文件数据之后将数据进行加载。此后，主节点继续将所有已经收集到的修改命令，和新的修改命令依次传送给从节点，从节点依次执行，从而达到最终的数据同步。

通过这种方式，可以使写操作作用于主库，而读操作作用于从库，从而达到读写分离。

## 简述哨兵模式

哨兵模式监控redis集群中Master的工作的状态。在Master主服务器宕机时，从slave中选择新机器当作master，保证系统高可用。

每个哨兵每10秒向主服务器，slave和其他哨兵发送ping。

客户端通过哨兵，由哨兵提供可供服务的redis master节点。

哨兵只需要配master节点，会自动寻找其对应的slave节点。

监控同一master节点的哨兵会自动互联，组成哨兵网络，当任一哨兵发现master连接不上，即开会投票，投票半数以上决定Master下线，并从slave节点中选取master节点。

## cluster集群

cluster提出了虚拟槽的概念。

1. redis cluster默认有16384个槽，在集群搭建的时候，需要给节点分配哈希槽尽可能相同数量虚拟槽。
2. 如果目前redis执行set操作，redis先对这个key经过CRC16 hash运算，并把结果对16384取余，得到槽编号。
3. 根据槽编号，寻找到其对应的redis节点，在节点上执行hash命令。
4. 如果此时执行get操作，节点先验证该key对应的槽编号是不是归本节点管，如果是则保存数据。如果不是，则发送正确节点编号给客户端。

# 简述Redis的RDB

RDB即将当前数据生成快照，并保存于硬盘中。可以通过手动命令，也可以设置自动触发。

## 简述Redis的save命令

save命令是redis手动触发RDB过程的命令。使用该命令后，服务器阻塞，直到RDB过程完成后终止。该过程占用内存较多。

## 简述Redis的bgsave命令

bgsave命令不阻塞主进程（严格意义上也不是完全不阻塞，详看下面过程），该命令fork一个子进程用于执行RDB过程。其具体过程为：

1. 判断此时有没有子进程用于RDB，有的话直接返回。
2. redis进行fork子进程过程，此时父进程处于阻塞状态。
3. 子进程创建RDB文件，完成后返回给父进程

## 简述Redis自动触发RDB机制

1. 通过配置文件，设置一定时间后自动执行RDB
2. 如采用主从复制过程，会自动执行RDB
3. Redis执行shutdown时，在未开启AOF后会执行RDB

## 简述Redis的AOF

AOF通过日志，对数据的写入修改操作进行记录。这种持久化方式实时性更好。通过配置文件打开AOF。

## 简述AOF的持久化策略

1. always。每执行一次数据修改命令就将其命令写入到磁盘日志文件上。
2. everysec。每秒将命令写入到磁盘日志文件上。
3. no。不主动设置，由操作系统决定什么时候写入到磁盘日志文件上。

## 简述AOF的重写

随着客户端不断进行操作，AOF对应的文件也越来越大。redis提供了bgrewriteaof函数，针对目前数据库中数据，在不读取原有AOF文件的基础上，重写了一个新的AOF文件，减少文件大小。

## RDB与AOF优缺点比较

AOF占用的文件体积比RDB大。一般来说利用AOF备份对系统的消耗比RDB低。对于备份时出现系统故障，RDB数据可能会全丢，但AOF只会损失一部分。

RDB恢复速度比AOF低。

## 简述Redis淘汰机制

1. noeviction：默认禁止驱逐数据。内存不够使用时，对申请内存的命令报错。
2. volatile-lru：从设置了过期时间的数据集中淘汰最近没使用的数据。
3. volatile-ttl：从设置了过期时间的数据集中淘汰即将要过期的数据。
4. volatile-random：从设置了过期时间的数据中随机淘汰数据。
5. allkeys-lru：淘汰最近没使用的数据。
6. allkeys-random：随机淘汰数据。

## MySQL与Redis区别

mysql是关系型数据库，并且其将数据存储在硬盘中，读取速度较慢。

redis是非关系型数据库，并且其将数据存储在内存中，读取速度较快。

## 简述Redis过期策略

1. 定期删除，redis默认是每100ms就随机抽取一些设置了过期时间的key，并检查其是否过期，如果过期就删除。因此该删除策略并不会删除所有的过期key。
2. 惰性删除，在客户端需要获取某个key时，redis将首先进行检查，若该key设置了过期时间并已经过期就会删除。

实际上redis结合上述两种手段结合起来，保证删除过期的key。

## Redis基本数据类型实现原理

字符串：采用类似数组的形式存储

list：采用双向链表进行具体实现

hash:采用hashtable或者ziplist进行具体实现

集合：采用intset或hashtable存储

有序集合：采用ziplist或skiplist+hashtable实现

## Redis快的原因

1. redis是基于内存的数据库，内存数据读取存储效率远高于硬盘型
2. redis采用多路复用技术通过采用epoll的非阻塞IO，提升了效率



我是小牛，微软程序员一名，各大互联网公司面试经验等等，欢迎大家扫描左侧二维码关注后端技术小牛说！

关注公众号：后端技术小牛说，回复interviewtop，获得最新大厂面试八股文