

一面

1、mysql 索引结构？

2、redis 持久化策略？

(1)RDB: 快照形式是直接把内存中的数据保存到一个 dump 的文件中，定时保存，保存策略。当 Redis 需要做持久化时，Redis 会 fork 一个子进程，子进程将数据写到磁盘上一个临时 RDB 文件中。当子进程完成写临时文件后，将原来的 RDB 替换掉。

(2)AOF: 把所有的对 Redis 的服务器进行修改的命令都存到一个文件里，命令的集合。使用 AOF 做持久化，每一个写命令都通过 write 函数追加到 appendonly.aof 中。aof 的默认策略是每秒钟 fsync 一次，在这种配置下，就算发生故障停机，也最多丢失一秒钟的数据。缺点是对于相同的数据集来说，AOF 的文件体积通常要大于 RDB 文件的体积。根据所使用的 fsync 策略，AOF 的速度可能会慢于 RDB。Redis 默认是快照 RDB 的持久化方式。

3、zookeeper 节点类型说一下；

4、zookeeper 选举机制？

5、zookeeper 主节点故障，如何重新选举？

6、syn 机制？

7、线程池的核心参数；

8、threadlocal 的实现，原理，业务用来做什么？

9、spring di 的原理；

10、四次挥手；

11、gcroot 选择；

12、标记清除算法的过程，标记清楚算法如何给对象分配内存空间？

13、cms 算法的缺点；

二面

1、correnthashmap？

2、threadlocal 原理？

3、hashmap；

4、Java 数据类型，同步机制；

5、讲讲贪心算法；

6、如果线上用户出现 502 错误你怎么排查？

7、并发量很大，服务器宕机。你会怎么做？

三面

1、syn 和 lock 的区别，哪个更好？怎么选择？

2、hashmap 源码，为什么 8 个节点变成红黑树 又为什么到了 6 个节点才恢复为链表？

3、缓存穿透，怎么解决？

4、负载均衡算法，实现；

5、轮询和随机的缺点；

6、分布式服务治理；

7、dns 迭代和递归的区别；

8、算法题：最长回文串；这道牛客题霸上有原题，大家可以去看看：NC17 最长回文串

9、为什么连接的时候是三次握手，关闭的时候却是四次握手？

因为当 Server 端收到 Client 端的 SYN 连接请求报文后，可以直接发送 SYN+ACK 报文。

其中 **ACK** 报文是用来应答的，**SYN** 报文是用来同步的。但是关闭连接时，当 **Server** 端收到 **FIN** 报文时，很可能并不会立即关闭 **SOCKET**，所以只能先回复一个 **ACK** 报文，告诉 **Client** 端，"你发的 **FIN** 报文我收到了"。只有等到我 **Server** 端所有的报文都发送完了，我才能发送 **FIN** 报文，因此不能一起发送。故需要四次握手。