

# 小霸王

大家好,简单自我介绍一下: 4年经验,某马培训毕业,普通本科,坐标北京,这次跳槽公司保密一下希望大家谅解。  
在这里跟大家分享一下本次面试的经验和平时学习的过程

在不知道儒猿技术窝(前身狸猫技术窝)之前自己都是瞎看书,比如周志明jvm,并发编程的艺术,redis设计与实现等等书籍,书太晦涩完全是死记硬背过几天就忘记了,后来通过1群echo\_\_陈大佬了解到技术窝,用半年时间把儒猿的专栏看了两到三遍,在这里感谢一下大佬算是入门的师傅;然后跟着大佬一起报了石杉老师架构班,今年2月审核通过但是武汉疫情4月才来北京开始学习,目前学架构班大概7个月左右。

架构班目前我学了1-13和大数据03,04内容:<http://note.youdao.com/s/dxE3UxWN>;

架构班学习上建议跟着老师的大纲一步一步学习,因为跳着看后人的心会浮躁起来,比如前面电商项目一和微服务源码很多视频都几个小时,跳看后回头在补时效率就很低了。

本次跳槽主要是想验证一下自己这半年多的学习成功,平时学习时间安排 早上5-9点左右看老师视频,在公司上午复习下之前学的技术栈,主要通过自己的笔记和图来回顾,下午忙完领导安排活后实战和整理笔记,周末全天;知识点上是否吃透可以通过找会的人让他考考你或者跟不会的人能讲明白这个技术点或老师的代码都能自己手撸出来;

面试前半个月找石杉老师指导一下简历上技术栈和项目(这个很重要不然简历根本没有吸引力),按照简历技术栈来复习自己的知识体系(主要看自己平时自己做的笔记同时也沉淀自我笔记),项目上找架构班大佬虐两到三次项目细节;

一: 面试了哪些公司?

百度 美团 京东 滴滴 小米 爱奇艺 58集团 新浪 好未来 boss 猿辅导

拿到了哪些公司的offer: 百度 美团 京东 滴滴 小米 爱奇艺 58集团 好未来 boss直聘

jvm,mysql,jdk集合和JUC并发源码,网络,微服务框架,分布式事务,分布式锁源码,redis

以上知识点对应老师的课程:

0.jvm和mysql来自救火队长专栏

1.jdk 石杉老师的架构班:08\_JDK源码剖析系列内容

- 2.网络 石杉老师的架构班:09\_大白话网络,10\_Java NIO技术精讲以及原理剖析,11\_支撑1亿图片的分布式存储系统项目实战,12\_Netty核心功能精讲以及核心源码剖析,13\_基于Netty的高并发基础架构项目实战,03\_支撑每日10亿数据量的Kafka架构实战和04\_探索世界上最复杂的消息系统: Kafka内核源码分析
  - 3.微服务体系 石杉老师的架构班:springcloud netflix五个组件源码,分布式事务atomikos和byteTcc源码,分布式锁curator和redisson源码
  - 4.redis 石杉老师亿级流量和redis设计与实现
- 

三: 第一部分的内容jdk的相关的在老师架构班有两百多讲实战结合eureka源码写注册中心核心代码和Hadoop源码写editLog,老师也从中抽了一部分放在面试突击第三季这里

问题: 说一下HashMap的工作原理(简单描述下回答思路)

1. 底层数据结构
  2. 初始化核心参数
  3. hash算法
  - 4.寻址算法
  5. hash冲突
  6. 扩容机制
  7. put方法,get方法和remove方法执行流程
  - 8.hashMap是否是线程安全
- 

问题: 说一下线程安全ConcurrentHashMap的工作原理(简单描述下回答思路)

- 1.底层数据结构
  2. hash算法
  - 3.寻址算法
  4. put方法流程,get方法,remove方法,size方法
  - 5.jdk 1.7和1.8底层实现区别,可能会被问到1.7如何去锁segment
- 

问题: volatile硬件底层原理,cas和synchronized原理?

- 1.学习资料:石杉老师面试突击第三季volatile硬件级别讲解(强烈建议看一下)

自己整理的笔记: <http://note.youdao.com/s/dxE3UxWN>

- 2.重点说一下回答思路:

- 1.为了解决处理器与主存之间的速度鸿沟,引入高速缓存,却又导致了缓存一致性问题
- 2.为了解决缓存一致性问题,引入了如MESI等技术,又导致了处理器等待问题
- 3.为了解决处理器等待问题,引入了写缓冲区和无效队列,又导致了重排序和可见性问题
- 4.为了解决重排序和可见性问题,引入了内存屏障

- 5.每个技术点跟面试官举工作中的场景或者源码中场景知道为什么要学它,学了它后有什么用,比如

volatile(aqs中state concurrentHashMap中Node节点),线程池的ctl(atomic 高三位:线程池状态 低位:线程池worker线程数量),concurrentHashMap(cas+synchronized),工作中优化停机,心跳时间等

---

问题: 线程池的工作原理和线程池的应用场景

- 1.学习资料:石杉老师面试突击第三季线程池工作原理
- 2.自己整理的笔记:<http://note.youdao.com/s/G4t8hOyT>

---

问题: 说一下ThreadLocal工作原理和使用场景

- 1.学习资料:石杉老师面试突击第三季threadLocal工作原理
- 2.使用场景:石杉老师架构班中比如zuul源码中RequestContext和分布式文件系统写editLog的txId
- 3.自我整理的笔记:<http://note.youdao.com/s/341vlwEp>

---

四: 第二部分的内容jvm相关的知识,之前试看周志明jvm相关书籍都是死记硬背理论缺乏实战效果不理想,之后是看救火队长的jvm专栏写的通俗易懂同时很多案例分析,建议先看专栏在看书,专栏中都有答案这里把jvm面试中问题说一下

问题: jvm的内存分配和执行原理以及类加载过程

问题:常用垃圾收集器有哪些,你们线上用的什么收集器,CMS优化,G1工作原理

问题:线上机器cpu负载过高如何去排查,如何去优化JVM参数

---

问题:是否有遇到过线上OOM情况,如何去排查的(这个拿老师案例说一下)

1.触发OOM的地方metaspace,线程虚拟机栈,堆内存,堆外内存专栏对应的场景都有案例我面试说了一个堆外内存溢出的情况

2.业务背景:公司IM系统使用netty来进行消息推送,有人在pipeline中自己创建byteBuf未释放

3.分析问题:

1.先通过OOM后自动生成的堆快照信息通过mat分析可能出现的地方和线程的调用栈定位到代码行

2.当并发量较大时堆内存持有堆外内存引用的对象进入老年代一直不能被回收,当通过jdk

ByteBuffer.allocateDirect(xx)时java.nio.Bits#reserveMemory尝试的去分配堆外内存,如果堆外内存空间不够时会执行一次System.gc()通知jvm来进行一次full gc

3.但是由于线上jvm参数-XX:DisableExplicitGC禁止system.gc()导致堆外内存溢出

---

五: 第三部分的内容mysql相关的知识,我之前看过mysql45讲,曾经以为自己mysql不用在学一遍了,后来儒猿群朋友推荐看队长的mysql专栏才知道细节上的东西,客观评价一下mysql 45讲讲的两个点就是死锁和join了其他我觉得一般,太多时间要看评论才能弄明白问题,mysql知识点队长专栏都有详细讲解,这里我就把面试问到问题列一下

---

问题:mysql buffer pool的结构组成,建议从查询和更新操作来讲

笔记:<https://www.processon.com/view/link/5f27c4e17d9c0835d3a35995>

问题:mysql通过什么机制来保证事物数据不丢失,MVCC机制和read-view视图

问题:mysql的索引结构,为什么不使用B树使用B+树,三层索引大概能支持多少数据,sql如何优化,覆盖索引,回表查询,索引下推等概念

问题:mysql的锁有哪些,如何分析死锁问题,死锁如何解决

问题:现在有一个场景需要保证数据库和redis的双写一致性,你会如何去做

问题:mysql的水平分库分表,什么情况下水平分表,什么情况下分库,分库分表的中间件你如何进行技术的选型,分片键有多个如何去做,单表迁移到分库分表方案,分库分表如何扩容

---

六: 第四部分的内容redis相关知识点石杉老师的亿级流量和redis设计与实现

问题:redis数据结构类型有哪些,平时在哪些场景下用过

1.redis数据结构类型有哪些:<http://note.youdao.com/s/Omt2UtMp>

2.应用场景:<http://note.youdao.com/s/8NPHq37A>

问题:redis过期淘汰策略,内存淘汰策略,持久化方式

1.内存淘汰策略:<http://note.youdao.com/s/O0iHxERv>

2.rdb持久化:<https://www.processon.com/view/link/5f39d0cce0b34d07117334ab>

3.aof持久化:<https://www.processon.com/view/link/5f39fa95e401fd0be031aee5>

问题:redis主从集群+哨兵,redisCluster工作原理,一致性hash的好处

---

问题:redis的master和slave的集群可能出现主从之间网络实例抖动,导致重连失败一直需要进行全量复制你觉得是什么原因?

1.复制原理:<https://www.processon.com/view/link/5f3b3d18079129531b60ee86>

2.导致的原因:从节点拿offset来主节点复制积压缓冲区找时找不到导致全量复制

---

问题:缓存雪崩,缓存穿透,缓存击穿如何解决,平时生产redis需要注意什么

1. 学习资料面试突击第一季:

<https://gitee.com/shishan100/Java-Interview-Advanced#分布式缓存>

2.生产问题

1. fork耗时导致高并发请求下延时(最好控制在10g中)

- fork子进程需要拷贝父进程的空间内页表

2. 避免复制风暴

- 别一个master挂多个slave导致都全量复制的时候,占用带宽
- 可以slave上在去挂slave,即星状变成树

3.匹配key

- keys指令导致的高cpu问题(禁止)
- 如果必须通过scan扩容不会重复(高位累加)&扩容的时候会重复

#### 4.危险命令

- flushall:清空所有数据。
- flushdb:清空本库数据。 请确认在没有请求的情况下再执行以上操作。(深夜更新时)另外, redis 提供了异步清空命令
- flushall async 和flushdb async 异步清空数据, 不会阻塞当前线程

---

七: 第五部分的内容分布式锁和微服务的相关内容,石杉老师架构班有详细的spring cloud netfix 5个组件和zk curator redisson的源码分析,工作原理上面试突击第二季有讲解

问题:分布式锁使用过哪些,使用场景

- 方案:<http://note.youdao.com/s/9I3mdlyC>
- Zk curator源码:<http://note.youdao.com/s/S7rvozW2>
- Redisson:<http://note.youdao.com/s/GgheHWp6>

---

问题: 能说一下springcloud netfix中feign和hystrix工作原理

- 1.feign请求核心流程: <http://note.youdao.com/s/BKApT2bP>
- 2.feign整合hystrix核心流程: <http://note.youdao.com/s/CNXJKhqi>
- 3.hystrix熔断器原理:<http://note.youdao.com/s/7U4ixMXk7>
- 4.hystrix存在问题,熔断打开后过了滑动窗口的时间后只根据一次请求的结果来确定是否继续熔断(二次开发统计一定比率)

---

八: 第六部分分布式事务内容:

问题:分布式事务有使用过吗?在哪些场景下?说一下byteTcc的方案?seata有了解过?saga的原理有了解过吗?

- 1.学习资料:面试突击第二季或者石杉老师架构班的分布式事务原理和实战
- 2.笔记:<http://note.youdao.com/s/l2SJpVam>
- 3.byteTcc源码:<http://note.youdao.com/s/ToAMxuGZ>
- 4.seata官网:<http://seata.io/zh-cn/docs/dev/mode/tcc-mode.html>

---

九: 第七部分网络相关内容

学习资料:面试突击第三季或石杉老师架构班大白话网络,nio源码,netty源码,kafka源码,自研分布式文件系统和IM系统

问题: 说一下http和https的请求流程

- 1.http请求流程:<https://www.processon.com/view/link/5f113c8ae401fd06f3df1bed>
- 2.https请求流程:<https://www.processon.com/view/link/5f11521de401fd06f3df7e75>

---

问题: bio和nio说一下,netty对比原生的nio好在哪里

1.netty好处:<http://note.youdao.com/s/Dt9XWGgc>

netty的源码:

1.核心流程:<https://www.processon.com/view/link/5f3e3f1f5653bb06f2dc1b27>

2.服务端:<https://www.processon.com/view/link/5f448c1de0b34d638e054d4f>

3.netty的实战:说rocketmq中代码或者石杉老师架构班自研IM系统

4.rocketmq源码:

1.学习资料:儒猿技术窝原子弹:从0开始带你成为消息中间件实战高手

源码:

producer端:<https://www.processon.com/view/link/5fa91d8d7d9c081baf205c23>

broker端:<https://www.processon.com/view/link/5fade419f346fb2d03b275db>

---

十: 第八部分设计类型的题目:

1.统计一个ip访问次数

- 在不加锁的情况下ConcurrentHashMap+putIfAbsent+Atomic

2.自研的网关的功能

- 这里说的是基础架构自研的,自己了解一点基于netty实现,参考zuul的前,中,后拦截器思路实现,自定义组件:appld认证,加密解密,加签验签,监控等

3.设计模型的了解和应用场景

- type=1调用类a执行,type=2类b执行,type=3类c执行通过策略模式来进行优化
- 工作场景说了下库存更新:模板方法+命令模式+工厂模式,权限删除访问者模式,在介绍了一些开源中间件的源码中设计模式

---

4.比如搜索系统会调用百度,Google,必应,有道,调用过程中其中一个方法调用执行完则返回结果

- 线程池提交多个异步任务,任务执行完后放入到一个阻塞队列中
- 主线程中阻塞队列take元素获取到返回结果

5.让你设计一个秒杀系统你怎么设计

- 架构班石杉老师秒杀的图:<http://note.youdao.com/s/KWjR5hxo>

6.如何设计一个消息中间件,消息中间件事务如何去设计,平时使用消息中间件需要考虑什么问题

---

学习资料:石杉老师架构班kafka讲解

producer:<https://www.processon.com/view/link/5f87744c07912906db271efd>

broker端:<https://www.processon.com/view/link/5f93d5c71e085307a0a1826a>

consumer端:<https://www.processon.com/view/link/5f9bae6107912907289bfdff>

事务设计:参考rocketmq事务消息说的方案:<http://note.youdao.com/s/VZz4RZN5>

使用注意事项面试突击:<https://gitee.com/shishan100/Java-Interview-Advanced>

---

十一: 第九部分开放题:

## 0.每次换工作的原因

### 1.你想换一个什么样岗位

### 2.工作什么样的事情给你带来成就感

3.现在有两个工作你怎么选择,工作A做的事情是高并发的场景他的业务上可能没有起色在走下坡路;另外一个场景并发量没有那么高技术上有数据量分库分表,分布式事务等场景,业务上给公司带来的价值更高

### 4.平时工作中技术和业务怎么权衡

### 5.一个需求需求需要做你们平时开发流程是什么样

### 5.一个需求需求需要做你们平时开发流程是什么样

### 6.你目前薪资多少,期望薪资多少,多久能到岗

### 7.你有什么要问我的吗

### 8.对我们团队的工作有了解吗

## 十二: 第十部分算法:

### 1.算法一般在一面和二面可能有

2.我平时没怎么刷属于放弃,个人认为时间成本和回报不成正比,算法其实也挺邪门的,我整个三四十次面试下来就出算法题5-6个,但是也有同学面试出很多算法

3.如果不喜欢刷算法的同学,建议算一种类型比如链表,同时面试的时候适当卖惨,比如说自己只会链表类型算法,其次算法很菜,大部分面试官听到后会直接会给你出链表类型题目

4.如果算法没做出来还让你继续面的话一定要通过技术来征服面试官,最后可以问一下面试官对算法的态度

5.如果要刷算法建议每天控制在1个小时左右把自己会的题刷个5-10遍

## 十三: 第十一部分项目:

技术上的问题大概就是上面列举出来的,一面聊基础会问的很深和很细,二面和三面主要看你项目,我本次面试的重点说的项目是自研分布式文件系统(石杉老师架构班项目), 订单服务,消息推送服务开发等,要想进比如京东 百度 美团 阿里这样的大厂, 技术深度和广度都需要很扎实,同时项目需要有含金量跟面试官有的聊,平时自己准备时:

### 自研或二次开发中间件项目:

1.自研的东西对比其他开源的内容有什么好处(可以从技术架构上,公司数据存储上,业务需求等方面考虑)

2.自研的带来的收益有哪些(最后表现出来的结果)

3.做这件事技术上最重要的点是哪些(用到的技术会被深问)

4.哪个地方做的比较完美,哪些地方后续还需要扩展(明白自己系统的优势和劣势)

### 业务项目

#### 1.项目整理技术架构

2.你负责的项目的核心业务流程,需要有亮点比如熔断限流降级,分布式事务,分布式锁,分库分表,消息中间件异步化等

共性的内容:



- 1.项目每天的访问量
- 2.日活跃用户
- 3.平时和高峰期的qps
- 4.日交易量
- 5.服务部署多少台每台机器的配置
- 6.生产出现过的问题,如何去排查问题的

---

小结:

我本次面试独角兽+大厂大概面了十几家左右,面试一般是4轮,前3轮技术面试,第四轮hrbp简单聊工作履历和离职原因 新的岗位是否了解等,最后聊一下谈薪资和股票事情,聊薪资注意一下:不要把自己所有offer都告诉hr,这样hr会按照总包的价格来压低你的基础工资,在就是不管去不去先拿offer再说不要不好意思,这样你之后面试才有底气谈工资

---

学习方法

1. 第一遍主要是看资料学习(比如老师讲源码看源码分析的笔记,比如老师写代码过一遍老师的代码在脑海中把流程给弄明白),看的过程可以记录一下重点的知识点(不建议跟着敲效率为0)
2. 第二遍自己写代码(卡主了在回头看一遍老师代码继续撸)或者写demo分析源码画流程图
3. 第三遍总结自己笔记(不建议用老师的笔记不是说老师笔记不好作为参考)
4. 第四遍第二天早上可以默写出昨天学习知识点核心的流程
5. 第五遍反复实战项目应用(比如比较重要项目建议敲两遍以上,重要技术点视频学两遍以上)
6. 第六遍跟看过的人交流或没看过的人,让你自己描述的东西能让别人听懂并能回答他问的问题

最后感谢一路陪伴自己儒猿群的小伙伴们,同时也感谢石杉老师这么好的课程,希望以后大家都能在这把技术点知识弄明白坚持学下去,祝愿大家以后也能跳槽一个更好的公司赚更多的钱

---

最后分享几个公司的面试题:

京东

京东二面:

- 1.存储平台主要做哪一块
- 2.画存储平台的架构图
- 3.容量不足如何去扩容
- 4.数据备份
- 5.namenode zk的选举不太了解说raft的选举
- 6.节点数据的主备,冷备,底层进行数据的传输
- 7.大文件hdfs断点续传
- 8.redis主要用在哪里,redisCluster,布隆过滤器的实现,redis的分布式锁,zk的curator的分布式锁
- 9.熔断如何来做的,hystrix的实现 滑动窗口的时间
- 10.qps多少,部署多少台机器,机器的配置



- 11.kafka consumer端的原理
  - 12.sql优化如何去做的
  - 13.你對自己后面规划
  - 14.为啥要离开
  - 15.期望薪资
  - 16.jvm调优
- 

京东三面:

- 1.每次换工作的原因
  - 2.为什么要做这个,有开源的内容有做过对比
  - 3.你们分布式文件系统的优势在哪里,意义在哪里
  - 4.具体有哪些成就感
  - 5.小文件定义的为多大
  - 6.做这件事技术上最重要的点
  - 7.如何保证数据的不丢失
  - 8.如果dataNode节点都宕机了怎么办
  - 9.哪个地方做的比较完美
  - 10.有做过业务相关的吗
  - 11.用微服务的架构为了什么
- 

爱奇艺

【爱奇艺一面:】

- 1.自我介绍
  - 2.说一下你目前负责的项目和业务逻辑
  - 3.并发请求如何处理,分布式锁用的什么
  - 4.订单数据如何存储的,数据库主键用的什么,订单id如何生成的
  - 5.订单系统高峰qps有多少,部署多少台机器,分配的jvm内存多大,每台能扛的并发量,订单系统的瓶颈点在哪里
  - 6.说一下这两年在公司中有学到什么东西吗?比如说技术的提升或工作上的提升?
  - 7.在学校学了些什么东西?
  - 8.linux用的多吗?tcp的连接状态大概有那几个能说一下吗?
  - 9.挑一个能体现个人能力的经验或者分享吗?源码的话那个项目看的比较多能分享一下吗?
  - 10.你还有什么想说的或想问的?
- 

【爱奇艺二面】:

- 1.自我介绍一下

- 2.图片存储系统用的什么?自研的基于什么技术来做的?
- 3.说一下kafka nio的网络模型?轮序的过程是否有问题,轮序的效率是否高吗?有了解过epoll()的底层原理吗?
- 4.说一下kafka发送消息,接收消息存储消息响应数据的过程
- 5.你们redis是怎么用的,redis有了解底层的数据实现吗?redis是单线程还是多线程?
- 6.平常编程用java比较多一点?说一下synchronized关键字锁的升级的过程?偏向锁?自旋锁?
- 7.如果我启一个java的线程的话会经过哪些环节?
- 8.分布式锁的使用?如果用redis实现锁具体有什么要注意的?redLock有听说过吗?如何解决redLock的问题?如果不用redis锁还有其他分布式锁吗?
- 9.上万的高峰qps如何坐高性能,高可用的?hystrix如何做的熔断?hystrix的滑动窗口有什么问题?
- 10.系统平常用什么方式部署?底层部署使用的什么?
- 11.你有什么想问的吗?

---

【爱奇艺三面】：

- 1.简单介绍一下
- 2.为什么要离职?新的工作有什么样的期望?
- 3.薪资方面?
- 4.主要负责的项目?文件系统架构?高可用?nameNode的一主多备的一致性?高性能?高并发?纠删码的存储方式节省空间?
- 5.分布式共识算法?paxos和ZAB有了解过吗?raft的执行流程?
- 6.Hadoop的源码看过哪些?
- 7.kafka的topic底层数据存储结构?
- 8.延时消息怎么去实现?rocketmq的自定义延时消息怎么实现?
- 9.eureka的主要来做什么?hystrix的熔断和限流的如何做的?hystrix如何去二次开发优化滑动窗口的问题?
- 10.redisCluster的高可用是如何实现?
- 11.canal和es(使用的版本和写入数据多久后才能读到),es的优化有了解过吗?
- 12.hbase的源码有了解过吗?hbase存储数据的问题?hbase如何分片的?列式存储和行式存储区别在哪里?
- 13.redis的雪崩 击穿 穿透,redis的master和slave的集群可能出现主从之间网络实例抖动,导致重连失败一直需要进行全量复制你觉得是什么原因?
- 14.mysql的隔离事务级别有几级?innodb底层事务的实现?mysql的binlog是什么时候写的?
- 15.jvm的垃圾回收器有哪些?CMS如何去优化?
- 16.对我们团队的工作有了解吗?

---

【爱奇艺四面】：

- 0.自我介绍
- 1.分布式文件系统中元数据怎么存储的,元数据都放内存吗?量太大的话怎么去优化?
- 2.上传文件时如何去保证文件写dataNode一定成功了,怎么保证文件数据不丢失?

- 3.dataNode节点磁盘是怎么去做分区存储的?说了一个啥名称不太懂的那种
  - 4.当初为啥要自研分布式文件系统,技术架构如何去选型的?
  - 5.client读写文件每次都需要和nameNode通信如何减少网络开销?
  - 6.是否考虑过缓存的方式来进行读写数据?
  - 7.nameNode如何去保证高可用的?
  - 6.redis分布式锁和zk分布式锁
  - 7.volatile的硬件原理
  - 8.synchronized和lock的区别?线程被中断后释放会释放锁?
  - 9.分布式事务有使用过吗?在哪些场景下?说一下byteTcc的方案?seata有了解过?saga的原理有了解过吗?
- 

美团

#### 【美团一面】

- 1.看机会处于什么样的考虑,未来的规划?
  - 2.有在时间工作中使用过jstat,jmap,mat工具,能给一个实际的例子说明一下吗
  - 3.synchronized和lock方法有啥差别,底层的实现方式有什么不一样
  - 4.reentrantLock中state的变量为啥通过volatile来修饰,从硬件级别说一下
  - 5.synchronized的锁,锁消除-->锁粗化-->偏向锁-->轻量级锁-->重量级锁-->自旋锁,普通方法和静态方法锁区别
  - 6.threadLocal的工作原理
  - 7.线程池的工作原理,如何去设计线程池的线程的数量,哪些是cpu密集型,哪些是io密集型的,如何动态的修改线程池的线程数量
  - 8.说一下nio,nio的reactor模型一般怎么用
  - 9.innodb buffer的作用背景和一条sql的读写数据的流程
  - 10.mysql的索引的结构,比如 1,1 2,2 3,3在b+树如何去存储的
  - 11.分布式事务,2pc,3pc,tcc,本地消息表,可靠消息服务,最大努力通知,saga
- 

#### 【美团二面】

- 1.目前负责哪方面的内容,说一下图片上传系统
- 2.是否调过业内的小文件存储,七牛云等
- 3.整体的分布式文件系统的架构
- 4.相同的文件是否进行校验,磁盘上存储的dataNode文件有做加密
- 5.相同的文件名是否导致冲突,文件丢失如何去解决的
- 6.副本多少个,副本节点如何去分配
- 7.如果dataNode写数据失败后如何去保证数据不丢失
- 8.如何去保证nameNode的高可用,一主多备是热备还是冷备
- 9.nameNode的元数据写在哪儿,nameNode如果出现卡死的情况如何去保证数据不丢失
- 10.kafka的网络模型能介绍一下吗

- 11.fastdfs如何来传文件的,对于当前客户端系统来说来访问文件的话需要什么参数
  - 12.当读请求访问量较大的时候如何去优化
  - 13.目前文件系统能支持多少文件的存储
  - 14.当前系统存储的问题
  - 15.消息系统,目前是轮序的方式每隔10s的方式
  - 16.如果每次用户在线量变大了后改成推模式或者推拉的模式
  - 17.客户端和用户的tcp连接如何去建立
  - 18.hystrix主要在哪里的场景下应用
  - 19.如何去设计hystrix中熔断开关打开的比率
  - 20.如何更好的设计消息系统的高可用
- 

美团三面没整理

儒猿技术学