

# 西木子同学面经

---

- 1、自我介绍
- 2、与儒猿的故事
- 3、专栏学习之路
- 4、架构班学习之路
- 5、报架构班前的愿望
- 6、我面试了哪些公司
- 7、面试准备内容
- 8、简历准备
- 9、面试内容
- 10、面试
- 11、对儒猿群的评价
- 12、总结

附录：面试题

儒猿-石杉架构课

<https://sourl.cn/bFTskZ>

## 1、自我介绍

大家好，我今年29岁，渣渣专，半路出家的程序员。工作四年多，换了三家公司，个个都是外包...我算绝对平庸的那种人，理解能力一般，记忆能力更差，说实话，如果天分满分是10，我可能到不了5。

这不是谦虚，是事实。所以每次看到大神的经历，内心总有一种无力感，感觉自己根本不可能做到，除了羡慕，卧槽，好牛批，好像别的也没啥能做的。继续看下一个传奇，反正我这个学历和履历，还能有什么指望呢？

---

## 2、与儒猿的故事

不过还好，如果非要说我有什么优点，可能只有一个，毅力（虽然也是去年刚特训出来的）。我一直相信一句话：人一能之，己百之；人十能之，己千之。果能此道矣，虽愚必明，虽柔必强。

从疫情开始在家无事可做，我看了很多本书，心灵鸡汤喝了不少，有没有用不好说，但是当时确实斗志又来了，当时就发誓这次一定要坚持学习！（好吧，不知道是多少次了，反正之前每次都半途而废，三天激情一过，一切回归原点），但是有些时候可能也就是命运，机缘巧合之下，我进了儒猿群。

---

看到了那群早晨四五点起床学习的大神，真的是那种直冲心灵的震撼。我没想到有人会那么努力，高中也没这么拼啊。。。观望几天之后，我决定加入，之前每次坚持都失败，很大部分原因都是因为无人监督，放弃成本太低。在几百人面前，坚持几天放弃了，这种事情我觉得还是挺羞耻的。

## 儒猿-石杉架构课

从此也开始了我的早起之旅...前期极其痛苦，早起其实不难，难得是早睡。在保证身体的前提下，在十点左右就要必须休息了，玩手机的都知道，那种躺在床上玩手机的感觉多爽，早睡比登天还难。

晕晕乎乎几天后，我果断的删除了手机所有的娱乐软件，晚上9点把手机锁住（有个番茄闹钟），不睡也玩不成手机了。慢慢的坚持一两周，终于适应了。

---

### 3、专栏学习之路

因为正好我们项目组经常出现full gc的问题，所以我第一个专栏选择了jvm，刚看几章就已经兴奋的发抖的那种，为什么？

因为我理解能力真的不行，我买过jvm的书，也看过很多公众号，可还是没学会调优，一度怀疑自己真的不适合做这行。但是这个专栏，毫不客气的说，只要不是智障，都能轻松看懂，并且只要

一步一步跟着练习，绝对能学会的那种。

我跟大佬们不一样，这个专栏我学了将近四个星期，而且是每天将近3-5个小时的那种，老师的例子我跟着一个一个的敲，结合自己项目一个一个实验，进度很慢，但是能明显感觉到自己的提升。

---

不出意外，学完后我很轻松的对项目进行了调优，效果也很明显，fgc频率大为降低，而且生产碰到的jvm问题都能很快定位并解决。

尝到甜头后，我又依次学了mysql，rocketmq，亿级流量缓存，面试突击系列等专栏，不得不说，儒猿的小伙伴学习氛围是真的好，大家一起讨论问题，兴致勃勃，压根不觉得累。

### 儒猿-石杉架构课

十一假期的时候，出去旅游，早晨5点爬起来在酒店学习，晚上回来洗完澡继续奋战。那时候除了学习，真的什么都想不到了。那几个月真的充实无比，专栏也慢慢刷完了，接着就报了架构班。

---

## 4、架构班学习之路

其实之前我早就知道架构班了，但是太贵了，我当时真的不敢赌，但是几个月的学习后，我算彻底服了，我从来也不相信别人的推销，我只信自己。

毕竟知识不会骗人，在此之前，慕课网，奈学，msb，拉钩，网易，图灵等架构班视频我几乎都看了一点，有的看得多，有的看的少，可以说我算比较有发言权的了，他们的课程也不能说不好，只能说不适合我。

我个人是不太喜欢别人给我念ppt的，本身我记忆力就不行，念完我也学不会...不过也有人觉得挺好，这个东西就是仁者见仁了，毕竟我也确实从其他课上也学到过东西。

作为一个不太聪明的我来说，我更喜欢老师那种白话文，手撸代码，从零画图，实战代码的讲解，简单，粗暴，有效！

---

虽然没什么钱，我也咬牙用信用卡买了（不推荐，各位还是量力而行，我是因为之前债务太多导致的没钱），反正我不担心。我已经练出了足够的毅力和耐心，剩下的就交给老师了。

老师说过的一句话我一直记得：可能你天分不如别人，智商不如别人，吸收也没人家好。人家吸收个8成9成，你只能吸收5成甚至更少，但是记住，就算你成不了仙，掉下来，那也是半仙

这句话给我鼓舞极大，是啊，大家学的都是一样，他们年薪50万70万，我笨点，那就30万，他们花1年，我花三年呗！总比我年薪一二十万浑浑噩噩好吧？拼个半年一年，博一个璀璨未来，这个生意，我没有理由不做！

---

## 儒猿-石杉架构课

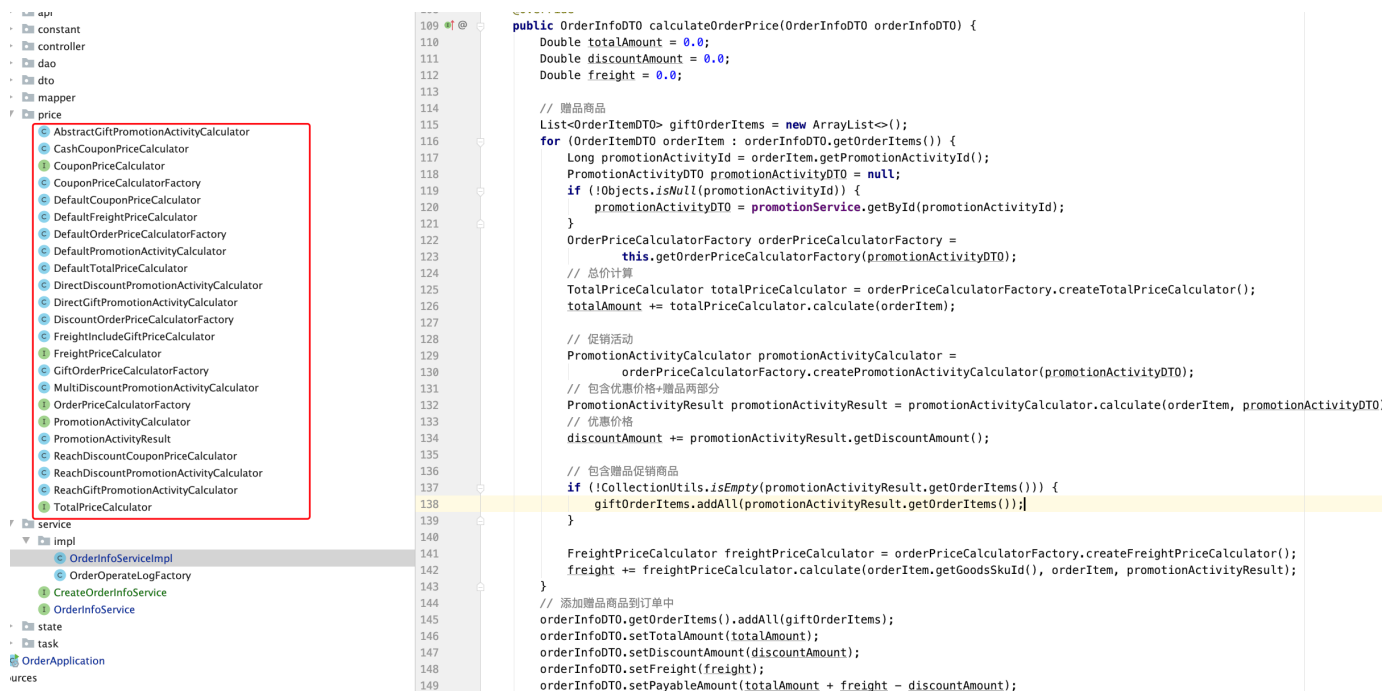
---

我入坑一个月后，有几个一起学习的小伙伴，收到我的反馈后也纷纷入坑，大家一起按照老师的指导慢慢学习。

### (1) 阶段1的地狱式折磨

让我见识到了优秀的代码和规范的开发流程，这一阶段超级痛苦，但是回报超值，对未来成长有着巨大作用。

从老师的代码里就能明显看出确实很认真的在做项目级的开发，一个价格计算的组件，各种拆分，设计模式，这些东西对平时开发来说用处极大。值得每个同学反复观看...



## (2) 阶段2就到了springcloud源码系列

这是我第一个源码学习，在此之前我对源码内心一直有种恐惧，但是一个月后，在老师带领下，cloud源码完美结束，成果也很显著

<https://soursl.cn/bFTskZ>

基本用这个框架碰到的问题，都能迅速通过源码定位问题，并且对它也做了一些线上调优。更主要的，我觉得就是学到了源码的学习方法，从此对源码不再恐惧了！

接下来就是不停的学习，分布式事务源码和实战，分布式锁源码，jdk源码及注册中心实战，网络，nio及分布式存储项目，秒杀项目，kafka源码等等。随着学的东西越来越多，我的笔记也越来越厚。。。

## (3) 自研分布式文件系统

我个人觉得架构班最有代表性的就是自研中间件这块，毕竟这个东西一般人别说写，想恐怕都很少有人想，而老师则是带着直接手撸了一遍。

可以说你学完再看其他中间件真的压力大减，毕竟解决方案无非就是那么几种，基本都很类似。这些中间件写完，我觉得架构能力会有很大提升。

比如并发这块，很多组件都能用到。网络也是一块重点，敲完会有很深入的理解。

NetworkResponse

NetworkResponseQueues

NioClient

NioProcessor

NioServer

ReplicateManager

StorageInfo

StorageManager

resources

test

target

pom.xml

dfs-namenode

.settings

src

main

java

com.zhss.dfs.namenode.server

Command

DataNodeInfo

DataNodeManager

DoubleBuffer

EditLog

EditLogFactory

FSDirectory

FSEditlog

FSImageUploadServer

FSNamesystem

NameNode

NameNodeRpcServer

NameNodeServiceImpl

RemoveReplicaTask

ReplicateTask

test

target

pom.xml

dfs-rpc

.settings

src

target

pom.xml

```
7 /**
8  * 负责管理editLog日志的核心组件
9  * @author zhonghuashishan
10  */
11
12 public class FSEditlog {
13
14     /**
15      * editLog日志文件清理的时间间隔
16      */
17     private static final Long EDIT_LOG_CLEAN_INTERVAL = 30 * 1000L;
18
19     /**
20      * 元数据管理组件
21      */
22     private FSNamesystem namesystem;
23
24     /**
25      * 当前递增到的txid的序号
26      */
27     private long txidSeq = 0L;
28
29     /**
30      * 内存双缓冲池
31      */
32     private DoubleBuffer doubleBuffer = new DoubleBuffer();
33
34     /**
35      * 当前是否在将内存缓冲刷入磁盘中
36      */
37     private volatile Boolean isSyncRunning = false;
38
39     /**
40      * 在同步到磁盘中的最大的一个txid
41      */
42     private volatile Long syncTxid = 0L;
43
44     /**
45      * 是否正在调度一次刷盘的操作
46      */
47     private volatile Boolean isSchedulingSync = false;
48
49     /**
50      * 每个线程自己本地的txid副本
51      */
52     private ThreadLocal<Long> localTxid = new ThreadLocal<>();
53
54     public FSEditlog(FSNamesystem namesystem) {
```

### 5、报架构班前的愿望

附两张图，之前我的愿望...

6



石杉  
手机在线

SVIP7



你与石杉的小幸运 🍀 即将消失，今天记着打卡哦。

11-03 上午5:52

你与石杉已超过1天未互发消息，小幸运 🍀 已消失。

昨天 下午10:10

老师您好，我是大专非计算机专业的，28岁，四年经验，一直在外包，目前工资是18K，想在明年9月份或者十月份左右的时候找一份甲方互联网的工作，月薪大致在22K以上就行，但是架构班的课程实在太多了，这十个月我也不知道应该把重点放在哪些地方，很多计划都不好做（因为每个课我都想学。。。），而且我从来没有面试过甲方，更别提互联网公司了，所以希望老师在有空的时候能稍微指点一下，我有耐心学习，但是向这种方向方面的问题，老师的几句指点肯定比我乱闯要好得多，谢谢



上午9:17

发送



杉架构课

l.cn/bFTskZ







## 6、我面试了哪些公司

极兔速递（3轮技术+1轮HR，通过）

掌门1对1（3轮技术+1轮HR，通过）

微盟（2轮技术+1轮HR，通过）

一智科技（3轮技术+1轮HR，通过）

剩下的就是几个上海的中小公司，基本也都过了

这里说一句群里很多跟我一样低学历的兄弟都比较关心的问题，学历不好不是一定没机会，但是机会确实比别人少点，可能人家有十次机会，成功一个就能进大厂，你只有一次机会，失误了就没了。

由于我的专升本还没下来，学历是真真实实的大专，面试大厂的机会确实少了点。所以我有两个小建议：

儒猿-石杉架构课

第一，想办法起码弄个自考本科，别说没用，再不济也比没有好。

第二，学历不行，就拼技术，死死抓住每一个机会，切忌怨天尤人，躺平放弃。人嘛，你应该考虑的是如何成功，而不是为你的失败找各种借口。

---

## 7、面试准备内容

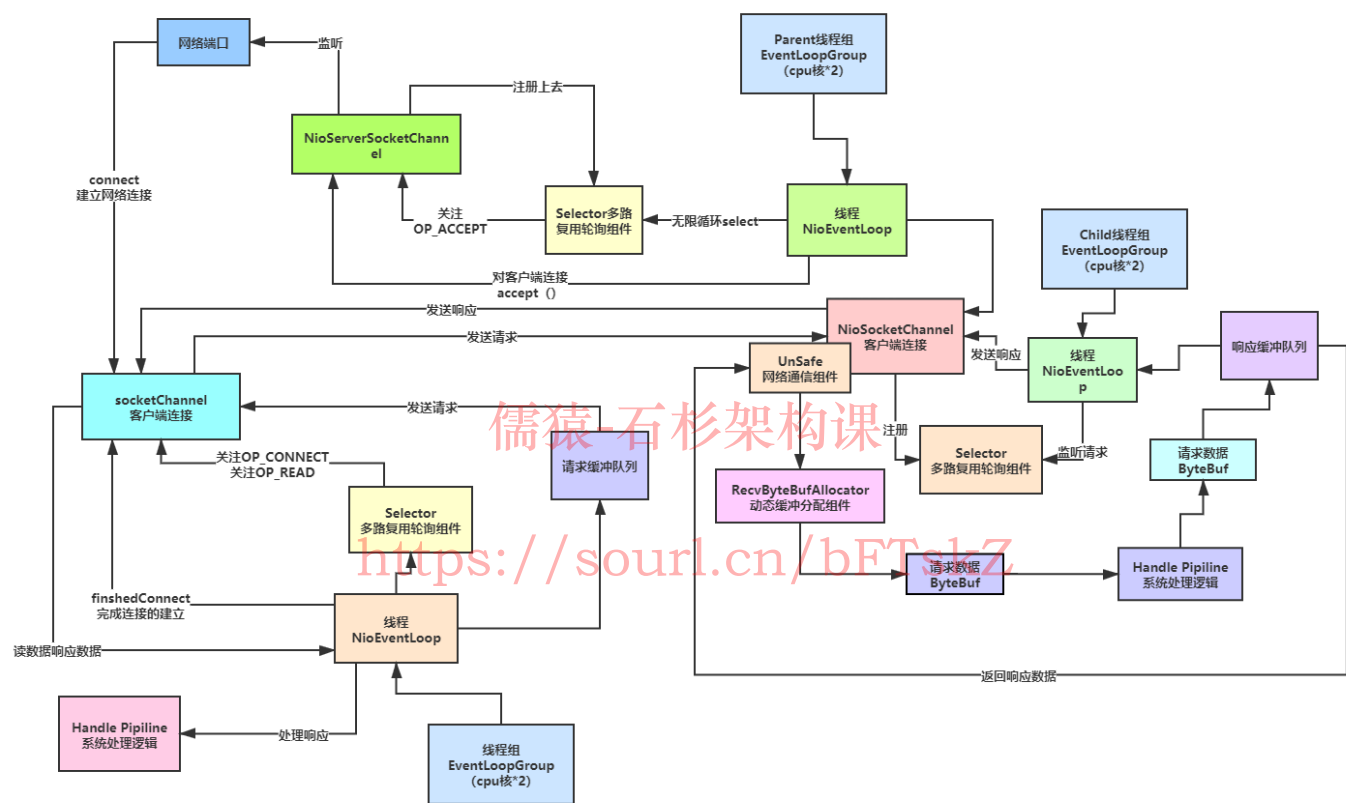
1、基础准备：jvm，mysql，rocketmq，redis，分布式事务，分布式锁，springcloud，jdk下并发包和集合包，线程池，kafka，es，dubbo，zk，spring，nio，网络等等...

基础知识我本来想贴一下自己笔记的，不过一来其实之前的面经都分享过了，二来我的笔记也确实没他们做的好，都是一个老师教的，笔记差别不大，我就偷个懒了，哈哈。

我就贴一下我做笔记的小技巧吧。如果是基础类的知识，最好的方法那肯定就是画脑图，比如 jvm，mysql，知识点多，不画图很难全部记住，就类似下面的

mysql图

如果是源码类的，流程图就不用说了，必须得画



2、项目准备：秒杀+分布式文件存储

复习的那一刻我算知道以前学习的时候坏毛病带来的弊端，你可以没有任何学习方法，但一定要记住一件事！记笔记！记笔记！记笔记！最好画图！

千万不要觉得你听懂了就不用记了，科学证明，你能记住你所学新知识的一两成就算很不错了。

不过还好，之前也只有一部分没记，那部分真的忘得死死的，复习的时候竟然跟新学的差不多...花了很大功夫又捡起来重新画图，慢慢的才把知识又串起来了。这里不得不再次吐槽我的智商，有些专栏我画了图，甚至还看了八九遍才勉强记住...

所以千万别再问为啥学过的东西很快忘了怎么办...咋地，你还有我忘得快嘛...多复习，多画图，多解答，真没啥好方法了...

---

接下来就是项目准备，其实很多人都卡死这一关了，毕竟基础知识只要花点功夫，几乎每个人都能学会，但是项目不一样，你平时做增删改查，现在突然让你编一个高大上的项目，其难度不亚于直接让你设计一个完整的架构，你不可能对你没做过的东西信手拈来。当你觉得编的还不错的时候，面试官随口一个细节问题就很快让你哑口无言，原形毕露。

<https://sourl.cn/bFTskZ>

我这里的建议就是你的项目要是真实做过的，如果流量不高，那就想办法夸张个100倍，然后看看你的项目哪些地方有瓶颈，再引入技术解决。

这里难的并不是如何引入技术，而是你对引入的技术是不是真的熟悉，因为任何中间件的引入都会导致复杂度升高，以及可用性问题，别人也一定会抓着你这个东西使劲问。

---

比如你学了rocketmq，想引入项目，那你就要结合你的项目谈谈

1、需不需要有序性、需不需要消息不丢失，如何解决幂等之类的

2、还会让你说下它的存储原理，架构，优势，缺点，和kafka的区别，当时基于什么考虑用的这个

3、结合业务聊聊为什么你为什么要保证消息不丢失，这样设计吞吐量下降的问题你准备怎么解决

4、你mq挂了怎么办，你的业务会受到怎么样的影响，你准备怎么降低这个影响，有没有降级方案等等等等

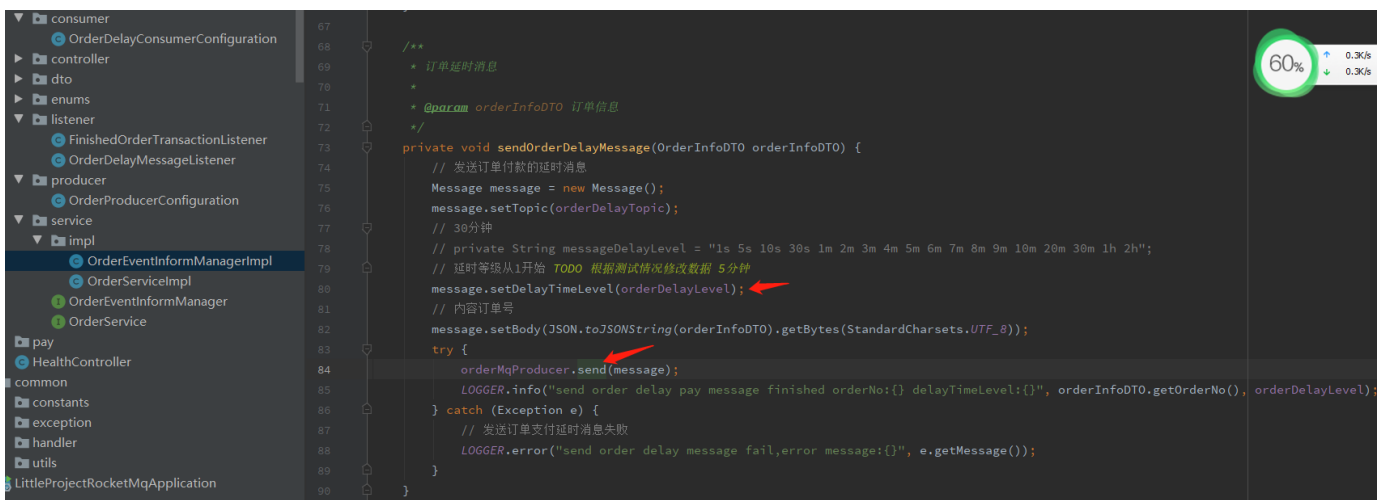
所以说难得从来不是你项目垃圾，而是你真的有认真考虑过这些问题吗？项目不是一天两天就能完善好的，最好找几个朋友，一起对各个细节从头到尾打磨一下，把学到的知识都融入到你的项目里。

这里推荐一波那个酒店小程序实战，我觉得那个专栏被严重低估了，我的项目细节很多都是在那个在专栏找到具体实现的，毕竟你理论说的再6，没实战过，总是有点发虚的

里面如何使用事务，如何保证有序性，如何实现延迟消息等都有具体代码实现和实战测试，基本学完不用担心细节方面的问题了。

<https://soul.cn/bFTskZ>

举个小例子，如果要在项目引入rocketmq，还要显得高大上，起码得用用事务，延迟消息，日保证保证有序这些东西吧，可是这些东西如果只是理论，你自己恐怕都不敢往上说，生怕人家来一句，你项目里怎么用的？没实际写过就是虚，心态差的当场就能被问懵。我是直接从那个专栏实战找的答案。



---

由于代码太多，我也不可能全部截图，但是就这几个方法，估计你看一遍就知道延时队列的用法和如何保证有序的实际开发用法，而且加上云平台的实战，可以实际看到效果。再融入你项目的时候你就压根不会虚，好家伙，代码你都写过，还运行了，看到效果了，你说这是虚假的经验？这就是真实的啊...

再加上mq这块你再研究研究原理，我还真不知道他怎么把你问倒，就算有些偏门题你不会，就这些实际经验，足可以拉开很多人了。所以每当有人问我怎么把技术融入项目，我就让他们看这个专栏。凭空想象的绝对不靠谱，三两下就被人问懵，但是自己跟着写一遍，实战一遍，那就是你真正的经验。

我是蛮期待他们继续推出其他专栏实战的，比如天天都有人说布隆过滤器，什么缓存雪崩，缓存穿透的解决方法，有几个人见过实际怎么开发的？比如es，我自己也会背几个原理，但是实际用es的时候，深度分页之类的具体怎么做等等。

希望以后如果项目要引入新技术，不用到处百度找可能有bug的demo，从实战专栏就能解决，这样实际开发中或者面试准备的时候，就真的爽到飞起了。

---

## 8、简历准备

简历就是你的门面，简历好不好直接取决于你有没有面试机会。我是直接通过简历指导，让老师帮忙修改的，简历上无用的东西不要写，不熟悉的尽量也不要写，除非你会的真的太少，需要点东西撑门面...

1. 深入理解JVM原理，并且对自己负责的线上系统负责调优。有解决线上OOM及线上频发的FullGc问题经验。
  2. 深入理解volatile硬件级别原理，CAS和synchronized的底层原理，及并发下AQS原理。
  3. 深入研究过JDK下的常见集合类，JUC下的原子类，线程池和常见的线程安全的集合类源码。
  4. 深入理解分布式锁实现原理。对第三方封装的框架如Curator，Redisson实现的分布式锁源码进行过研究。
  5. 了解常见的分布式事务的解决方案，并有实际应用经验。
  6. 深入研究Kafka的架构原理及源码。
  7. 深入理解Mysql的sql查询和更新底层原理，对索引，锁，事务等进行过研究，有sql优化经验。
  8. 深入理解RocketMq的架构原理，并在项目中实现延迟消息，事务等机制。
  9. 深入理解NIO网络通信模型，有基于Reactor线程模型实现分布式存储开发的经验。
  10. 深入了解redis的常见架构，主从复制，持久化机制，基于redis做过秒杀场景下的各种缓存。
  11. 深入理解springcloud，对eureka，ribbon，feign，hystrix等组件都进行过源码级别的研究，并做过线上调优。
- 

## 9、面试内容

基本都是绕着简历来问的，常规的自我介绍之后，大部分面试官就会从项目开始，依次深挖你的项目用到的组件，比如秒杀项目大致用到了redis，rocketmq，就会问：

### 儒猿-石杉架构课

#### (1) redis

1、Redis用来做什么的，数据类型用到了哪些？  
<https://soul.cn/bFTskZ>

2、不了解redis主从复制，集群模型，选举机制？

3、底层数据结构，持久化，为什么快，内存淘汰机制？

4、redis经典三连问-缓存雪崩，缓存穿透，缓存击穿的解决方案？

5、用redis做分布式锁的优缺点和解决方案？

6、如何解决大key，如何查看redis的负载情况等等？

---

7、场景题：比如如何找共同好友，如果需要记录某一操作有多少人执行，用什么数据结构，或者实现查看附近的人功能？

8、也有问布隆过滤器和位图的区别的？

9、也会引申出Redisson的封装，如何实现的分布式锁，以及redLock等源码实现？

反正他顺着你的项目为出发点，先问点基础知识，再绕着你的项目提出各种场景题。但总归绕不开redis的核心概念。

---

## (2) Rocketmq

1、先结合项目，聊聊什么场景用的，为什么选择它？

2、顺便让你画出rocketmq的架构图？

3、说说整个发送流程，以及存储机制，也会引出类似零拷贝的具体运用？

4、事务消息的实现原理？

5、再根据你的项目，聊聊需不需要保证有序性，数据不丢失，幂等性问题如何解决，rocketmq的网络通信架构等等。。。

---

## (3) 秒杀

1、如何解决超卖的，有什么优缺点，有没有考虑过其他方案，为什么不用？

2、有没有做限流，怎么做的，具体说说？

3、部署了多少台机器，qps和tps多少，每日订单量多少，用户量多少等细节问题？



4、画一下你的项目架构图，然后按照你的图每一步问，可用性和扩展性问题？

5、这里很多面试官都会随机的插入一些自己的疑问，比如如何做的调优，基于什么判断已经是最优，jvm怎么设置的参数，有没有考虑分布式事务的问题等等。

(tips：这些问题就不是随便说的了，要结合你的项目，前后都要串起来，建议多找几个朋友或者找群里大佬帮你过一下，前提是自己要有足够的思考

我是用了云平台的秒杀实战，基本都是真实数据了..再结合业务，把专栏的内容融入了进去)

---

#### (4) MySQL

一般问的最多的就是：讲讲你在线上做过哪些调优？这里要从发现问题开始讲，如何定位，如何判断，如何解决都讲一下。

### 儒猿-石杉架构课

也会顺便问一下索引结构以及常见的概念，比如回表，聚簇索引，mvcc，让你讲下更新一行数据的流程，索引失效的原因和解决方案。锁类型，以及为什么用b+树，然后也会问点分库分表之类的，比如分表维度，数据倾斜，不停机分库分表方案，如何跨页查询等等。。。

#### (5) JVM

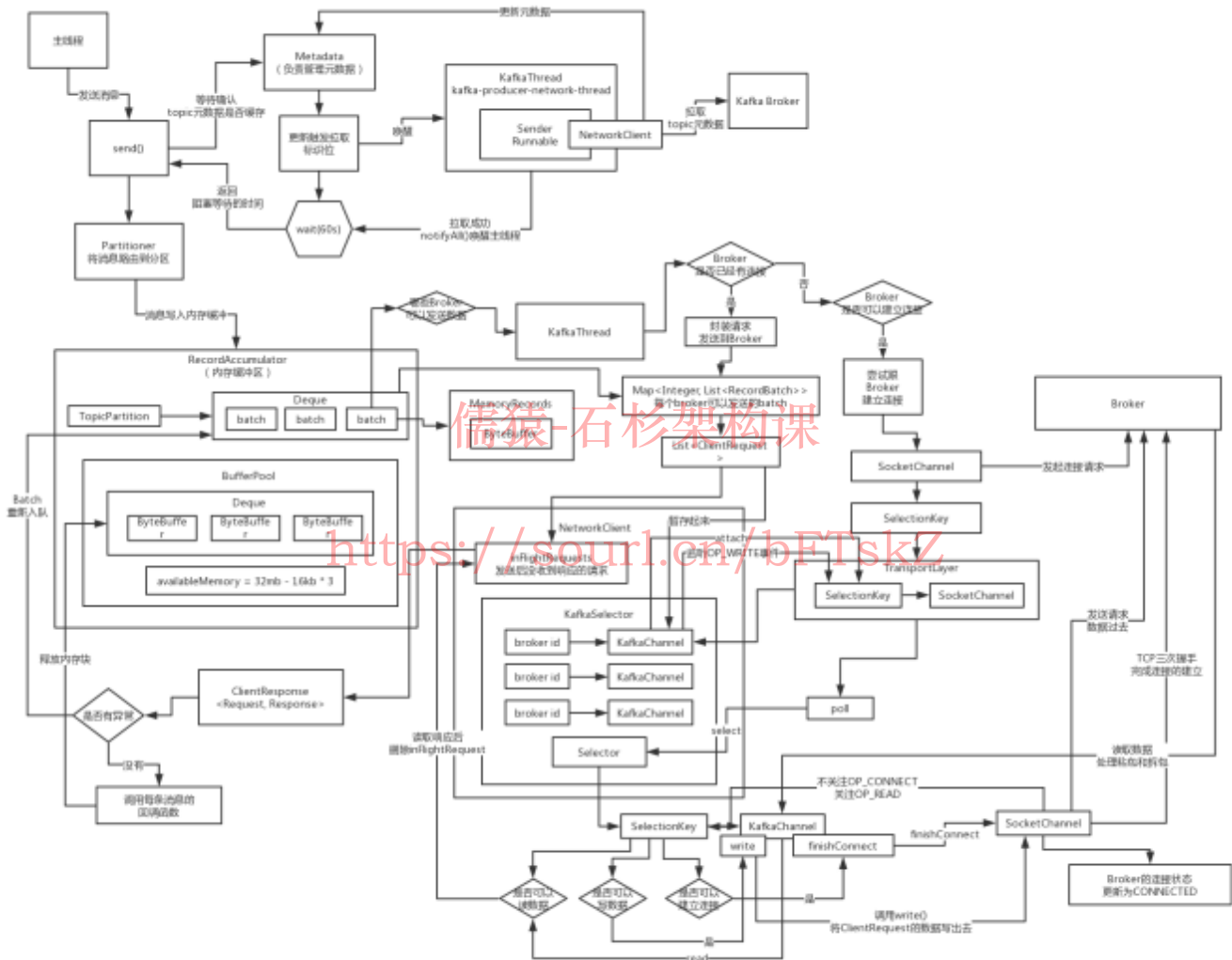
JVM的话先来说说你线上调优经验，如何解决fgc和oom问题的，主要是解决过程和想法，最好能说下从发现到定位，到排查到解决的完整步骤。然后就是让你画一下内存分配，说一下常见的垃圾回收器和回收算法，以及其中的区别。

这个就没得说了，专栏实例实在太多了，随便挑几个就行了，可以说自己项目碰到的常见的几个如内存分配不合理，或者一次加载大量对象，也能说自己了解的可能导致的原因，包括元数据空间导致的fgc之类的。

---

(6) Kafka

Kafka常问的就是先来一套架构原理图，再说说为什么比其他消息中间件吞吐量要高，如何解决频繁gc的问题的，ISR和HW机制说一下，reactor网络模型说一下，副本机制，写磁盘机制，消费者机制讲下。以及其他的一些细节。



(7) 其他 (这里就不一一列举了)

1、Springcloud一般就是先让你说下eureka的一些核心原理，如注册表，服务下线判断，自我保护机制，集群机制等等，如何做的调优，以及原理。也会问ribbon，feign，hystrix，zuul的一些原理架构图。

2、Jdk源码一般就是问一下读写锁原理，线程池原理，线程安全的集合有哪些并分别讲一下实现原理，lock锁的实现原理，用两个线程交替输出1-10等

3、分布式事务就是问有没有实战以及常见方案，每个方案的优缺点。

4、Nio一般就是问和bio区别，同步异步和阻塞非阻塞之间的区别，是哪里同步，哪里阻塞，多路复用是基于什么实现的以及他们的区别。你是如何实现reactor线程模型的，有什么缺点吗？

---

## 10、面试

1、面试技巧：无他，唯手熟尔。

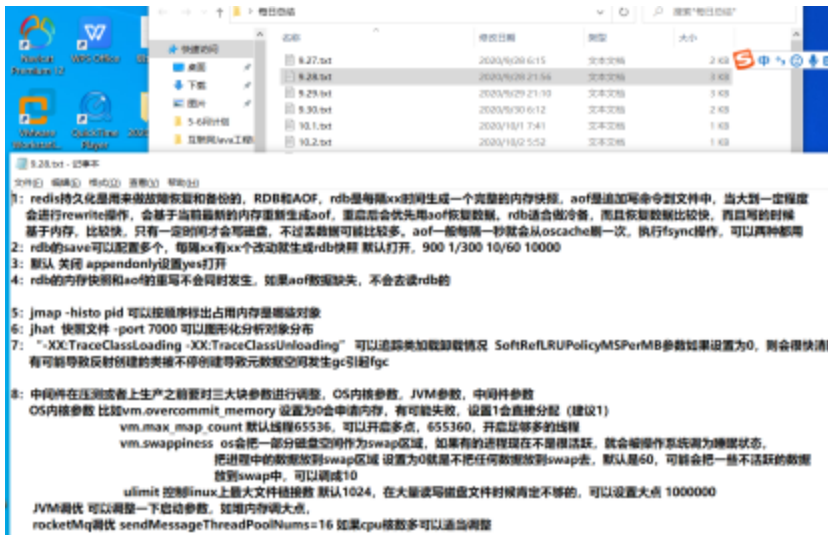
多面，多总结，先从小公司开始，最好先电面或者视频面，减少成本，等面得差不多了，就可以请假面试了。尽量不要裸辞，压力太大，很少有人承受得住，大神除外。

儒猿-石杉架构课  
<https://soul.cn/bF1skZ>

### 2、面试总结

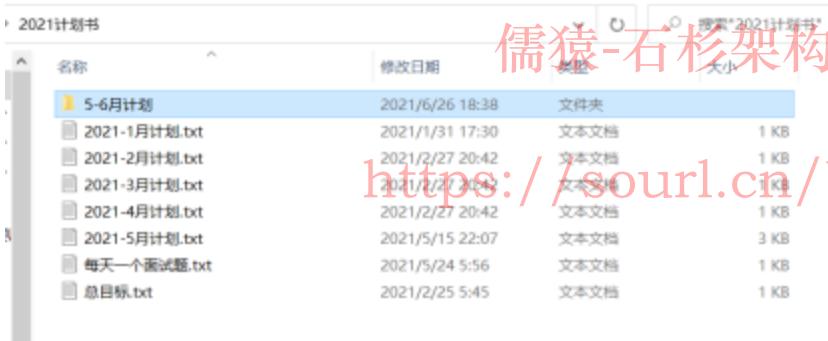
无论学什么，都是要努力才可以学到真东西。只有真正掌握技术的人，才有可能去享用它。如果你中途放弃了，之前所有的努力和付出都会变得没有价值。

给大家放几个图共勉，每日总结：

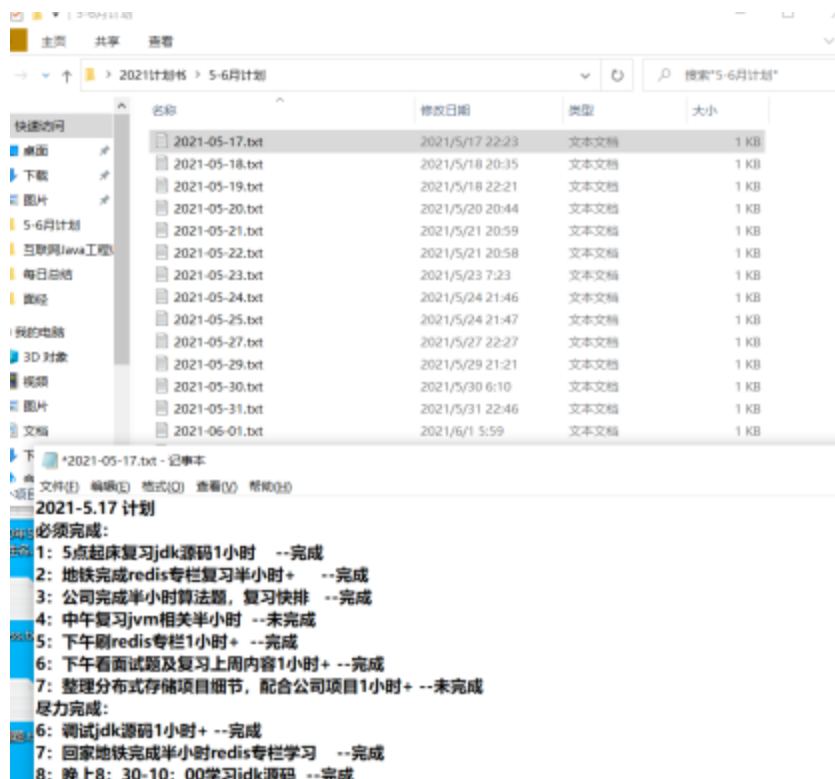


### 3、计划

#### 1、提前计划



#### 2、详细计划：（别问我哪来那么多时间...运气好，活不多）



## 儒猿-石杉架构课

### 11、对儒猿群的评价

<https://soursl.cn/bFTskZ>

我觉得儒猿最重要的就是那种独一无二的学习氛围，当然有人说太卷了，其实与我个人而言，说这些话都太空洞了，什么内卷，什么躺平，无非是自己选择罢了。如果用半年或者一年的时间，让你工资翻一倍，你愿不愿意义无反顾的拼一把？我反正是愿意的，我天资太差，除了努力，我找不到能改变命运的方法

而且从一开始我就明白一个事，一个县城的篮球教练和一个国家队的篮球教练，他们可能都让你忙碌训练一年，但是结果完全不一样，有时候大佬的一句话，真的可以让你少走两年弯路。

我算是石杉老师的粉丝了，我觉得那种被大佬指点的感觉挺神奇，他很轻松就能指点你应该从哪个方向努力，你无需思考太多，只管努力就行。

另外群里的大佬很多，我很菜，全靠他们带我，不厌其烦的指导我，真的超级感谢。我不是在这里打什么广告，真心感谢他们。（其他就不贴了，不是不重要，实在是太多了...）

儒猿-石杉架构课

<https://soursl.cn/bFTskZ>



西木子

20/9/15

我想进一个互联网公司，在那个公司积累，你觉得我是现在报架构班，还是先刷专栏，背面试题好呢🤔，主要没面试过互联网公司，不知道对技术深度追的深不深



wisdom

20/9/15

刷专栏和面试突击 可以去一个20k左右的公司 指的是吃透 也不是几个月能行的



wisdom

20/9/15

如果短期要跳槽的话以专栏和面试突击为主 如果是在一年之后跳槽的话可以先报班



wisdom

20/9/15

报班的事情是在一个公司稳定后一年左右不想跳槽能静下心来学架构班

<https://sourl.cn/bFTskZ>



西木子

20/9/15

嗯，我先把面试系列都学完再做笔记整理，争取明年进互联网公司🤔，有点担心我这大专学历，唉😭



西木子

20/9/24

不客气的说，感觉这个专栏拯救了我的自信，让我觉得我也能学会🤔



wisdom

20/9/24

都是这样过来的



西木子

20/9/24

听第一期面试突击的时候，我擦，我有一半知识没听过





西木子

20/9/24

后来画图画图的，我都能听懂😂



wisdom

20/9/24

嗯 多看几遍吃透



wisdom

20/9/24

真的是面试理论上随便玩

---

## 儒猿-石杉架构课

### 12、总结

<https://sourl.cn/bFTskZ>

其实对于百分之80以上的人来说，是没什么好的学习方法的。你可能花了很多年才明白一个学习方法超级有用，然后把心得告诉别人，他们先是惊喜，然后尝试，最后就会回归平庸。

这实在是太常见了，毕竟听懂不等于学会，不然满地都是大神了。百度搜一下学习方法，你能搜到不止一百条。所以当你发现你没什么好的学习方法，也尝试过别人的方法但收效甚微的，不用纠结了，唯有努力，别无他法。

学习方法可以自己慢慢摸索，摸得出来最好，摸不出来也无妨，只要坚持，结果一定不会太差。

---

题外话：这几天当初跟我一起学习架构班的大兄弟也狂砍了数个offer，货拉拉，微盟，韵达等等全部拿下！其他大厂还在面试，估计也能过，起步四十多万年薪了。

我俩一块入的这行，他三十多了，外包混了四五年，现在也算步入正轨了。我俩之前都属于菜的抠脚的那种，不然也不至于在外包那么多年...但是我们也坚持了下去，目前来看，结果不算太坏，更重要的是，有了希望。

最后就用一句我很喜欢的话收尾吧：“种一棵树最好的时间是十年前，其次是现在”。

---

## 附录：面试题

一面：

- 1: 讲一讲mvcc以及作用。
  - 2: mysql有几种锁，讲一讲。
  - 3: mysql的索引有几种，为什么不用hash。存储引擎有几种，区别是什么。
  - 4: 如果发现查询慢，有可能问题出在哪？是如何排查的？
  - 5: 如果没走索引，可能的问题和如何解决的？
  - 6: 说一说你的mysql调优经验吧。
  - 7: 讲一下rocketmq的整体架构。
  - 8: 你觉得nameSever的作用是什么？他是如何跟broker以及生产者消费者交互的？
  - 9: rocketmq的事务消息是如何实现的？
  - 10: 消息大致如何存储的？主要讲下那几个文件。
  - 11: rocketmq如何做到消息不丢失的，如何保证有序，如何保证不重复消费的？
- 

12: 讲一下对象从刚创建到清除的过程，对象是如何在内存中流转的。常见垃圾收集器以及各自特点。出现fgc的几种可能原因。

13: 讲讲你的jvm线上调优经验。比如线上出了频繁fgc或者oom，如何排查及解决。

14: 线程池的几种创建方式及主要参数，如果开了三个任务，执行完了，又进去两个任务，是会复用之前的线程还是新开线程？

15: 讲下volatile原理，aqs，reentrantLock和synchronized区别，读写锁的实现原理。常见并发包下的类有哪写？说说你熟悉的并发集合及原理。ConcurrentHashMap1.7和1.8区别。

16: 分布式锁有哪些常见实现, 源码你讲一讲。

17: 分布式事务聊一下。

---

二面:

1: 画一下你项目的架构图

2: 项目的redis你主要是用来做什么的, 有考虑过redis挂了怎么解决吗? 如何解决缓存雪崩, 穿透等之类的问题的? 如果没做, 是基于什么考虑的, 有没有备用方案? 有做过备份redis数据吗? 说下aof和rdb区别, 以及主从复制如何实现的。项目里有没有bigKey, 如果有你准备怎么解决? Redis分布式锁你是具体怎么用的, 它能防止超卖吗? 为什么不用redLock, 目前方案的缺陷在哪你知道吗? 假如redis当缓存, 失效的时候, 有大量的请求打到数据库了, 怎么解决? 如何解决缓存和数据库不一致的问题的?

### 儒猿-石杉架构课

3: 项目里用到rocketmq了? 先讲讲大概用在哪些地方了, 为什么要用? 结合你的项目聊聊为啥要保证有序, 事务等, 具体怎么做的, 目前的做法真的不会丢数据吗? 如何做到严格不丢, 带来的性能问题你怎么考虑的? 如果mq不可用了, 你有降级方案吗? 如果让你现在设计降级方案, 你怎么做?

---

4: 项目用到了springcloud? 先说说几个组件和作用, 画一下架构图。你是如何进行优化的? 你知道eureka的缺点吗? 如何避免服务雪崩, 你怎么处理降级和熔断的? 线程池和信号量有什么区别? 由于eureka的延时性, 假如分配你一台已经挂了的节点, 如何保证服务正常调用不会报错?

5: 项目有没有用设计模式, 聊聊应用场景, 你还知道哪几种? 你觉得设计模式的优缺点是什么?

6: kafka在项目中有用吗? 说一下大概发送流程, 他跟rocketmq的区别是什么? 零拷贝你了解吗? 在kafka有哪些具体应用? 时间轮聊一聊? 你觉得kafak源码给你最大的帮助是什么? kafka

如何实现的内存池循环利用，降低gc成本？

7：分库分表有用过吗？一般按什么维度拆分？如何解决数据倾斜问题？分库分表后如何解决深度分页查询问题？如何不停机迁移数据到分库分表上？有哪几种常见的中间件可以解决这些问题？

---

三面：

- 1：有了解过DDD吗？
  - 2：项目中解决过什么问题？哪些比较有挑战？
  - 3：如何做到快速锁定问题的？
  - 4：如何保证项目的稳定性？
  - 5：你对未来有什么规划吗？
- 

儒猿-石杉架构课

<https://soul.cn/bFTskZ>