

1、自我介绍

各位同学好，我是儿子娃娃，今年 25，二本非科班学历，一年多工作经验，之前就职于外包公司，平时工作主要语言是 python，工作形式是在家远程办公。

因为之前看过面试突击第一季，就被老师的课程给震撼了，俗话说得好，站在巨人的肩膀上，有老师这个巨人，跟老师学习可以大大缩减我到达目的地的时间，不断地突破自己在技术上面的天花板。。。。。。

2020 年 6 月报的班，学习架构班课程有 1 年时间左右，深刻的意识到单独靠自己默默的学下去，可能永远都达不到想要的结果，近期在石杉老师和助教的帮助之下，成功地面试上了蚂蚁金服。

2、面试了哪些公司

以下按照时间顺序排序

云上会展（2 轮技术+1 轮复试，通过）、别样（3 轮技术，通过）

字节（2 轮技术，挂了）

蚂蚁（3 轮技术+1 轮 hr，通过）

爱奇艺（3 轮技术，挂了）

达达（1 轮技术后约 2 面没下文）

携程（2 轮技术通过，推掉了第 3 轮交叉面试）

3、面试准备

（1）专栏

儒猿-石杉架构课

面试突击第一季、第二季、第三季：这里面涵盖了绝大部分常见的面试题，其中第一季的 mq 和 redis 是问的最多的，接着是第三季中关于并发和 mysql 的部分。

jvm 专栏：之前学习过一遍，面试前再复习了一遍，主要是 jvm 的几块内存区域、垃圾回收的触发时机和回收流程 (young gc、cms 和 g1) 还有一些异常情况 (比如 cms 的 concurrent mode failure)、如何优化 jvm (主要是合理分配堆内存，别让对象总是跑到老年代里头去导致频繁 full gc)

对于 jvm 这块的内容，到最后是在脑子里面形成了一个会动的场景图（几块内存区域在运行代码时候是怎么一起工作的、垃圾回收时候的几个步骤，每个步骤都在干嘛），面试的时候跟面试官讲的时候就会生动形象很多，而不是感觉像是在背书。

mysql 专栏：之前学习过一遍，但是后面忘得差不多了，面试前主要是看了下 buffer pool 这块内容，mysql 面试这块主要还是靠面试突击第三季的内容，当然有充足时间准备的同学，专栏一定要好好学习！

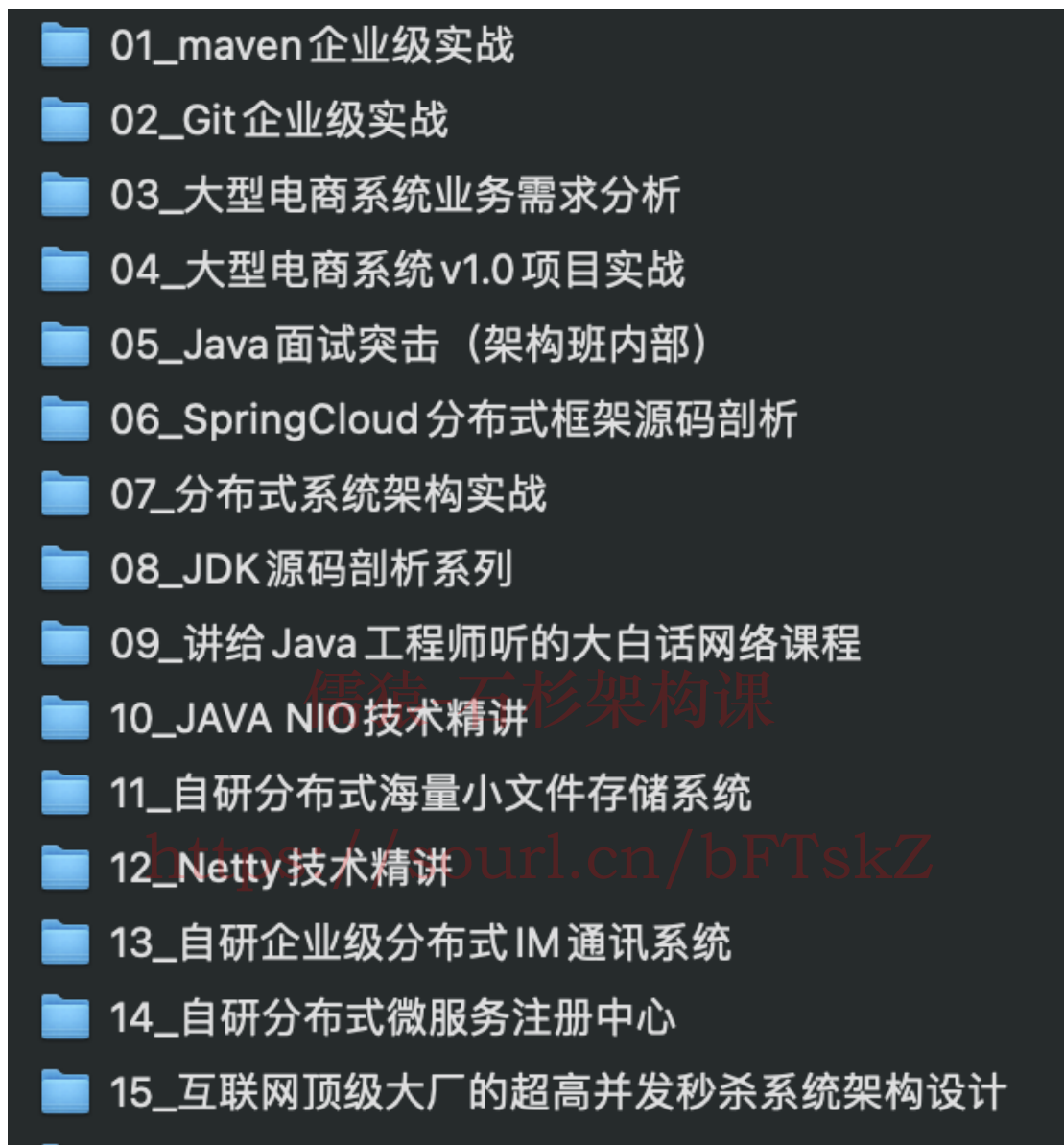
<https://sourn.cn/bFTskZ>

rocketmq 专栏：之前学习过一遍，后面面试主要还是面试突击第一季的 mq 连环炮，配合架构班的 kafka 核心原理和源码课去准备的，当然，有时间的同学，专栏一定不要错过，里面讲的都是都是非常有用的！

（2）架构班课程

架构班课程主要分为两个部分：Java 架构课程和大数据架构课程

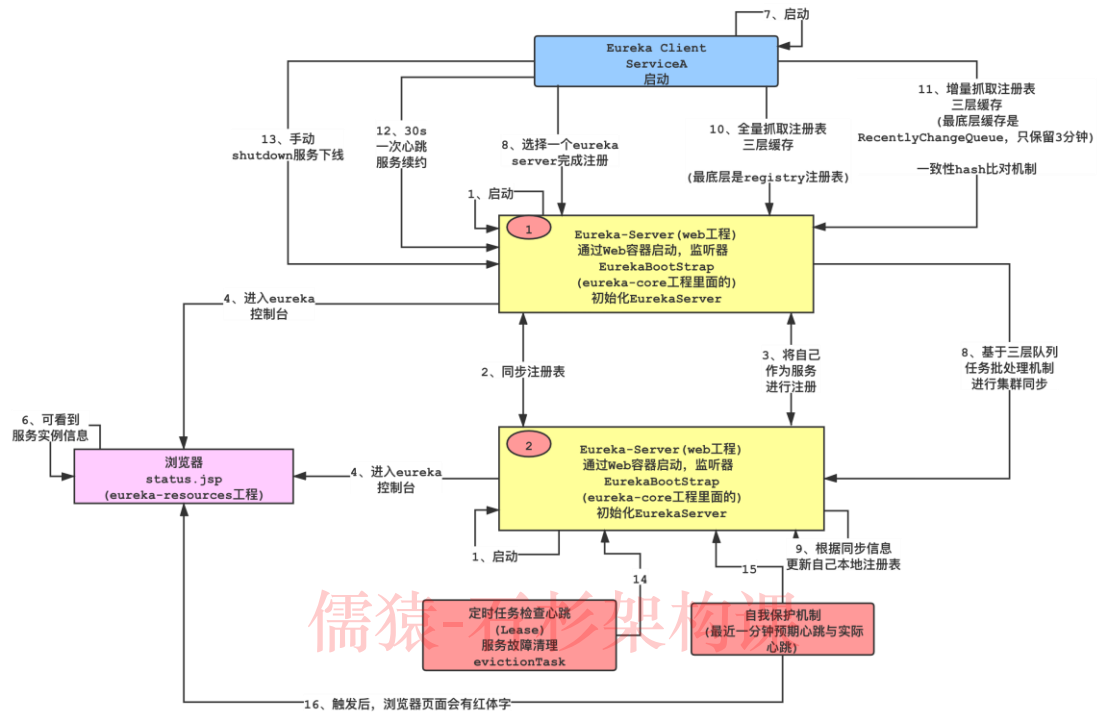
Java 架构课程学习的内容如下图：



大数据架构课程学习的内容有：

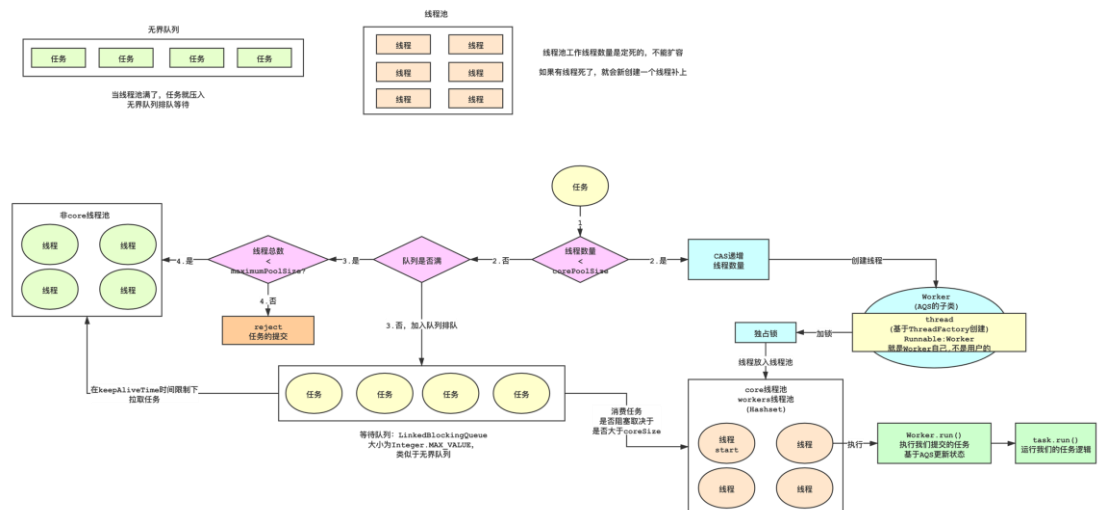
《hdfs 源码课和 hdfs 二次开发课程》、《kafka 原理课和源码课》、《zk 原理和源码课》

05_笔记图-Eureka 整体架构设计: <https://sourl.cn/JbwhHP>



<https://sourl.cn/bFTskZ>

06_笔记图-线程池原理: <https://sourl.cn/77uaNz>



07_笔记图-hystrix 源码: <https://sourl.cn/cd5bcU>

简历这块的准备主要走的是架构班内部系统的简历指导流程，先按照文档的指示，

一步步地写简历，主要就是个人简介、专业技能、还有项目，项目这块是重点。

比如项目简介、你在这个项目中的角色、项目的技术挑战以及如何解决的、项目线

上的一些部署架构、还有一些流量情况等等都得非常清楚。对于想要面试大厂的同

学，项目的负载情况一定要搞清楚，体现在简历上，大厂面试官都会问。

简历指导的流程一般是先填写好自己个人的基本情况，工作情况，以及跳槽的

诉求，还有每天能够学习的时间长短，打算准备多少时间，我的情况大概是这

样的：

同学状况：2020年6月入班

学习情况：学习进度一般

学习时间：工作日8小时学习，周末可以12小时

期望行业：互联网

期望岗位：Java开发工程师（高级）

学习周期：3-4个月左右

然后针对你的诉求，老师会为你量身定制一套复习计划，当时我的复习计划大概是长这个样子的：

二：综合建议

1: 专栏技术栈学习和复习计划(1个月)

- 1.从 0 开始带你成为JVM实战高手
- 2.从 0 开始带你成为消息中间件实战高手
- 3.从零开始带你成为MySQL实战优化高手

2: 亿级流量电商详情页系统实战(2周)

3: 架构班需要复习的内容(2-3个月)

- 1: Spring Cloud从0基础入门到精通核心组件源码
- 2: 05_多种开源技术实现的企业级分布式锁源码剖析
- 3: JDK源码
- 4: 01_精通基础开发技术之讲给Java工程师听的大白话网络课程
- 5: 01_支撑每日10亿数据量的Kafka架构实战
- 6: 02_探索世界上最复杂的消息系统：Kafka内核源码分析

然后就是按照复习计划，每天有条不紊地去复习推进了，当复习都完成了之

后，进入到简历指导下一步，编写简历

简历这块当初前前后后重写了差不多有四五次左右，简历这块个人认为最重要

的就是体现你的特长还有讲重点，特长部分可以在技能描述这块详细写出来，

就拿 jvm 举个例子，很多人只会写熟悉 jvm 基本原理，然后就完了，这样其实

并不太好

可以更细化一点，比如熟悉 jvm 原理、类加载机制、双亲委派机制、内存区域划分、垃圾回收算法、垃圾收集器 (ParNew, CMS, G1)。这样面试官要问问题的时候也更加方便，直接对着这些具体的技术点问就好了。

简历中最主要是介绍项目的背景、负载情况、项目描述、技术挑战、解决方案。对于每个部分的内容，想办法用最简单通俗的语言概括就好，不要长篇大论的写，面试官压根没时间一个一个字的去看。

就比如技术挑战的解决方案这块，刚开始我写的跟流水账一样，简历指导的时候被一顿痛批，后面按照修改建议，精简了流水账的内容，改成简洁通俗的内容。

具体的解决方案是留给面试过程中，和面试官沟通交流用的，以此为切入点，慢慢地聊一些技术方案，还有解决技术挑战过程中所涉及到的原理和源码。

关于如何准备项目，个人是结合之前工作所在的项目为基础进行准备的，本身就是微服务架构，用的也是 netflix 微服务技术栈，所以整体上来说还比较容易准备。

在这套项目基础上加上了老师之前讲的亿级流量缓存架构设计，然后就是之前工作时候遇到的一些技术挑战，整体上来说没有写的太华丽，没有把小文件和IM写进来，主要还是以业务为主线。

如果要写小文件和IM的同学，最好自己真的动手把老师留的一些课后作业给全部自己搞定了，然后真实开发部署上线测试，助教不是说老师9月就有小文件项目的云平台实战吗，可以跟着实操一遍。因为单单跟老师课堂这么敲，然后直接写上去，遇到一些厉害的面试官，很容易就被问露馅。

<https://soursl.cn/bFTskZ>

简历这块绝对是值得花大量时间去准备的，它就是你的一个敲门砖，学习系统上给的简历修改建议一定要虚心接受并调整，不然最后吃亏的一定是自己。

我的简历-技能描述部分

技能描述

1. 阅读过JDK并发包源码，了解核心关键字原理如cas、aqs、volatile、synchronize、java内存模型、并发数据结构、线程池等。能够熟练运用并发编程技术进行开发，深入理解底层原理。
2. 熟悉JVM原理、类加载机制、双亲委派机制、内存区域划分、垃圾回收算法、垃圾收集器(ParNew,CMS,G1)。
3. 熟练运用设计模式进行开发，常用构造器模式，工厂模式，策略模式，享元模式，模板方法等。
4. 熟悉Spring Cloud Netflix微服务技术栈，深入阅读过Eureka, Ribbon, Feign, Hystrix源码，处理过线上Hystrix线程池故障并进行过定制化优化。
5. 深入理解Kafka底层原理和架构设计，读过核心源码，如Producer端缓冲区(RecordAccumulator)设计，Batch打包发送机制，基于Nio与服务端的通信机制，粘包与拆包的解决，Broker端的Reactor高并发网络通信架构设计(Acceptor, Processor, Handler)，副本同步机制，消费端的消费逻辑、时间轮等机制。熟悉消息O丢失、防止重复消费、顺序消费技术方案。
6. 熟练使用Redis进行功能开发，了解Redis的基础数据结构、持久化、复制、主从、哨兵、Redis Cluster工作原理、熟悉缓存与数据库双写一致性、缓存雪崩、缓存穿透、缓存击穿技术方案。
7. 深入了解ZooKeeper底层原理和架构设计，如基于BIO实现的服务端集群通信，选举算法，zab协议，重选机制，海量客户端session管理等机制。
8. 熟悉分布式事务XA、2PC、3PC、TCC、最终一致性理论，可靠消息最终一致性方案。
9. 数据库：了解MySQL InnoDB引擎，B+树，聚簇索引，联合索引使用规则，事务隔离级别，mvcc。
10. 熟悉敏捷开发流程。

我的简历-项目部分

技术挑战

流量高峰期时调用下游酒店属性管理服务接口，频繁打满hystrix线程池触发RejectPolicy

默认情况下，hystrix会将同一个依赖服务下的所有接口调用均放到同一个线程池中，且大小为10，没有请求排队机制。

临时解决方案

- 开启请求排队机制，将未能分配到线程资源的请求，放入到请求队列中排队暂缓。

事后解决方案

- 为将酒店属性管理服务中个别流量较高的接口，独立分配hystrix线程池，保证请求有充分的线程资源进行处理。
- 开启hystrix线程池资源的扩容缩容策略，默认线程数量为1，最大线程数量为10，合理分配系统线程资源，降低cpu和内存负载。

酒店的属性支持多维度查询，且不同维度下的查询请求体冗长复杂，导致缓存key过大

- 一家酒店下拥有多个房间，以及不同的费率类型，每个房间以及不同的费率类型均有其对应的属性信息，属性信息支持从类目维度以及科目维度查询。
- 与下游服务的接口api接口调用采用的是GraphQL，每次请求构建出来的查询请求体信息较为复杂，直接将查询请求体内容作为缓存key放入到redis中会导致key过大，额外占用redis内存。

解决方案

- 使用hash算法计算出查询请求体的hash值，作为缓存key，其查询结果作为缓存value，存放对应酒店id管理的hash数据

<https://sourl.cn/bFTskZ>

结构下。

对于酒店属性的查询和更新请求同时执行，导致缓存与底层数据源不一致

对于同一家酒店，相同房号/费率条件下，同时接收到了其对应属性的查询请求和更新请求。

酒店属性更新请求先执行将缓存删除，但调用下游服务更新接口时执行较慢，导致后到的酒店属性查询请求先从下游服务查出未更新前的结果放入缓存，出现缓存与底层数据源数据不一致问题。

解决方案

- 采用缓存与数据库双写一致性方案，利用内存队列异步串行化同一家酒店，相同房号/费率条件下对应的属性查询和更新请求。
- 同时做了相同读请求去重复、读请求长时间阻塞、写入缓存中的数据被lru清理掉的相关应对方案。

6、面试题

如果是架构班的同学，一定要花时间完成简历指导完成过后的发放的模拟面试

题（大概有 100 道左右），这里面的这些题目都是面试常问的东西，尤其是

大厂。

这里面的模拟面试题，我就罗列部分项目相关、java 相关、专栏相关的内容。

（由于助教师兄说是架构班内部的资料，所以就不全部分享了）

关于个人和项目相关部分的面试题，是必须必须必须非常认真准备的，因为这里的题目，不论是去面小公司还是大厂，都是必问的。

因为一开始上来就是让你自我介绍一把，然后再介绍介绍之前自己做过的项

目，负责的模块，担任的职责，项目有哪些技术挑战，怎么解决这些技术挑战的，具体的流程是什么，自己在项目里面得到了哪些成长。

关于面试题的回答，尽量通俗易懂，能够让人马上抓到重点，这里我简单介绍一下

第一部分：项目相关

- 1、简单介绍介绍一下你在上家公司做过的项目、项目的主要流程、你所负责的模块、你在项目担任的职责
- 2、你在项目里面得到了学习到了什么，得到了哪些技术提升

- 3、有没有 code review 习惯，平时 code review 时候都会去注意哪些点
 - 4、团队人员的组成，开发规模有多大，平时是如何迭代开发一个需求的（从产品经理到最终上线的流程）
-

第二部分：jdk 源码相关

- 1、volatile 的作用和底层原理、项目中或哪些地方 (JDK 或中间件) 用到了 volatile、volatile 的使用场景、double check 中 volatile 的作用是什么
 - 2、java 内存模型
 - 3、cas 的原理
 - 4、ThreadLocal 的底层原理，使用 ThreadLocal 需要注意什么问题，怎么去解决
 - 5、死锁发生的原因以及如何去避免死锁
 - 6、synchronized 实现原子性的原理
-
- 7、ReentrantLock 和 synchronized 的区别
 - 8、ReentrantLock 是怎么实现原子性的

9、aqs 的原理（如何阻塞线程，如何唤醒阻塞线程，不同类型节点

Exclusive、Shared 之间唤醒模式有什么区别）

10、aqs 在并发包的应用

11、自旋锁(tryLock)

12、什么场景下用读写锁(ReentrantReadWriteLock)

13、ConcurrentHashMap 实现线程安全的原理（put 和 get），对比 jdk 1.8 和

jdk 1.7 的加锁粒度

儒猿-石杉架构课

<https://sournl.cn/bFTskZ>

14、手写 ArrayBlockingQueue 的 take() 和 put()

15、synchronized 的 wait(), notify(), notifyAll() vs. Reentrantlock 对

应 Condition 的 await(), signal(), signalAll()

16、公平锁和非公平锁

17、jdk 都有哪些常见的线程池，什么情况下我们需要自定义线程池

18、线程池的构造函数的几个参数都是什么意思

19、线程池的执行流程（什么时候创建 core 线程、什么时候入队列、什么时候扩容到 max 线程、core 线程和 max 线程在线程池任务队列 take 任务时有什么区别）

第三部分：《从 0 开始带你成为 JVM 实战高手》

1、jvm 的内存区域划分，不同内存区域都有什么作用

2、双亲委派机制的作用

3、介绍下 cms 垃圾收集器的执行流程

4、为什么 full gc 会影响系统性能

第四部分：《从 0 开始带你成为消息中间件实战高手》

1、消息中间件的使用场景如何实现消息 0 丢失

2、如何避免消息重复消费

3、不同消息中间件之间的技术选型，都有什么特点和区别

4、如何实现消息顺序消费

第五部分：《Kafka 消息中间件内核源码深度剖析试看版》

1、kafka 的网络通信架构的设计细节

【基于 nio 实现的 reactor 网络通信架构(Acceptor 线程 + Processor 线程 + Handler 线程池 + RequestChannel)】

2、kafka 磁盘高性能读写实现细节

(mmap + FileChannel + 写 os cache + 顺序写磁盘，定期 flush，consumer 基于零拷贝读)

<https://sourl.cn/bFTskZ>

3、kafka 是如何查找一条消息的（基于 .index 稀疏索引文件的二分查找过程）

4、kafka 顺序写磁盘什么情况下会退化为随机读写磁盘

5、kafka 在磁盘中都有什么文件？(.timeindex + .index + .log)

6、kafka 什么地方用到了 ZooKeeper

第六部分：《从零开始带你成为 MySQL 实战优化高手》

1、介绍下 acid 是什么，InnoDB 是如何实现 acid 的？mvcc

2、B+树、聚簇索引、普通索引、回表

3、索引的优点和缺点

4、联合索引使用过程中需要注意的问题(什么情况会索引失效)

5、Innodb 是如何实现行锁的

6、分库分表

7、面试公司

小厂面试：

儒猿-石杉架构课

<https://sourl.cn/bFTskZ>

小厂面试主要就是提前帮助进入面试状态，挑选一些自己压根就不会去的公司

热热身；

小厂面试问的问题大部分都比较直接，就是直接考察一个技术的实现，还有一些

常见的技术方案；

面试的两家小厂都发了 offer，后面因为不想浪费别人时间直接拒掉了；

当小厂开始收 offer 之后，可以开始投规模稍微大一点的公司，独角兽和大厂

之类的。

中大厂面试：

心态变化：当开始面大厂之后，一个很明显的变化就是，压力自然而然就上来了，会很紧张，特别是被问到不会的问题的时候。

字节：字节二面写算法题的时候，写不出来，紧张地全身冒汗，最后挂了。

蚂蚁：蚂蚁投过三次简历，第一次是简历没写好的情况下投的被筛掉了；第二次修改好之后投了一次，又被筛掉；还是没放弃，继续投了第三次，最后才拿到了面试机会；

儒猿-石杉架构课

蚂蚁总共面了 4 轮，3 轮技术和 1 轮 hr 面，最终拿到 offer。蚂蚁的面试官非常专业，问的问题水平也很高，技术方面都是深入结合简历中的项目，建立在业务背景下来去讨论的，不是脱离业务背景下的发问。

会针对项目中的一些极端场景去发问，比如聊到了缓存与数据库双写一致性方案时，如果大量阻塞的读请求达到了超时阈值，此时高并发地直连数据库查询，怎么解决这种问题？？这种情况下怎么去优化数据库连接池？

爱奇艺：当时投的是中间件岗位，做的是开源项目的二次开发，抱着学习交流的心态试试看的，一共面了 3 轮技术面，后续没消息了，挂了。

中间件团队主要都是聊一些源码，jdk 并发包的源码，eureka（整体架构设计、优点、缺点、二次开发的思路）、kafka 源码、hystrix 源码、ZooKeeper 源码，手写实现 ArrayBlockingQueue 的 put 和 take，还有一道算法题；

达达：当时远程面了一轮，后续 hr 问到还有什么公司在面试时，说了在面蚂蚁、爱奇艺时，就没有安排远程二轮面试了。

携程：面了两轮技术还有智商和英语口语测试，当时因为有蚂蚁 offer，推掉了第三轮交叉面试，整体面试体验还是不错的。

8、面试总结

（1）面试技巧

- a、确保自己听懂面试官要问的问题，如果没听清楚，一定要多和面试官沟通，不要在没搞清楚问题情况下就开始说，会大大扣分的。
 - b、回答问题的时候先讲重点，然后再是详细描述里面的细节，千万不要上来就是一堆细节，比如面试官让你讲 volatile 的作用时，你说了可以实现可见性和有序性，如果面试官还想听更底层的实现，此时你就可以继续往深了说硬件层面 mesi 缓存一致性协议和内存屏障那一套。
-

c、回答问题的时候，需要注意面试官的反应，确保讲的方向是正确的，如果不对需要及时和面试官沟通。

d、保持耐心，如果面试官多次打断你，或者面试官讲的不太正确，不要去和面试官抬杠。

e、有机会先提前找小伙伴进行模拟面试，帮助热身进入状态，也可以提前查漏补缺。

儒猿-石杉架构课

f、面试如果遇到不会的问题，千万不要不懂装懂，不懂的问题可以记录下来，
<https://sourl.cn/bFTskZ>
或者请教面试官，很有可能同样地问题会被反复问到多次，比如当时我就不知道每次发送的 key 数据库都不存在的情况下缓存穿透问题的解决方案，面试官就给我介绍了通过布隆过滤器实现的方案，后面面试别的公司时候就被重复问了 2 次。。。

g、如果面试没有通过，千万不要灰心，认真总结原因，是哪里没做好，查漏补缺。

(2) 面试总结

平时的学习和积累很重要，不存在突击的情况，技术都是靠平时一点点累积起来的。

面试遇到不会的，总结思考原因，是好事情。

面试挂了，别灰心，太正常不过了。

相信自己，自信心很重要。

表达能力也很重要，老师平时的大白话讲课方式，在面试给面试官阐述技术和

设计方案的时候起到了很大的帮助。

保持耐心，不要浮躁。

<https://sourl.cn/bFTskZ>

9、学习方法

先调整好心态，不要指望一遍学完就彻底掌握，这是不现实的想法。

核心是重复学，因为第一次学习的情况下，很多东西都是很陌生的，一遍下

来，其实只能够了解个大概讲了个什么东西，有点像是预习的感觉。

在第一遍学完之后，再重复地学第二遍，此时就可以慢慢地明白个大概所以然

了，当然第二遍学习肯定还是会有一些不懂的东西，没关系，抓大放小过掉就

好了。

第三遍学习的时候，就可以站在一个更高的角度去思考当前所学习课程了，然后第三次学的时候，就会体会到老师为什么要这么安排课程的章节，会发现其中大有玄机。

学习的时候得做好完整的笔记，个人用的笔记软件是 Notion。

代码跟着老师一行一行敲，源码分析课程跟着老师在源代码上面写注释，看视频不倍速，原速播放。

儒猿-石杉架构课

老师画图的时候，自己也跟着在 ProcessOn 上面画图，加深印象。

<https://sourl.cn/bFTskZ>

如果课程感觉自己没学懂，可以重新看视频，如果本身就学懂的，就没必要重

新看视频浪费时间了，直接看之前自己画的图，根据图来复习对应源代码，你

会发现刷刷刷的就回忆起来了。

定期复习课程，知识旧了不复习一定是会忘记掉的。

10、总结

课程非常好，帮助我不断地突破不同阶段下的天花板，而且老师讲课的思路不

仅仅是授人以鱼，还有授人以渔。

个人在学习架构班课程中，感觉到几个阶段的蜕变。

蜕变一：能够设计和开发较为复杂的模块，不再写出屎一样的代码。

没有学习架构班之前，至少业务逻辑稍微复杂一点，就容易把代码写成一坨

屎，在经历了项目一的学习之后，学习到了软件工程的流程，知道怎么去做模

块设计，还有遇到复杂业务逻辑时候，怎么合理的使用设计模式，抽象代码，

不让代码变成一坨屎。

儒猿-石杉架构课

蜕变二：能够读懂开源项目的源代码。

<https://sourl.cn/bFTskZ>

没有老师的课程前，尝试看过别人讲的源码课，不到几节课时间，就开始打瞌

睡了，压根就看不下去，学习曲线过于陡峭，频繁被劝退。

从阶段二 spring cloud 源码开始，老师站在我们初学者的角度，一步步引导我

们去阅读源码，抓大放小，学会连蒙带猜，屁股猜想论，阶段性画图总结，源

码写注释等等方法

不但教会了我们一门开源框架的源代码，还教会了一套如何去阅读源代码的方

法论，用在后续独立学习一门新技术，读源码尤为有效。

蜕变三：能够体会到不同开源框架的设计思路、优缺点。

课程学习到后面，尤其到了小文件和自研微服务注册中心的时候，其实就是对之前阅读过的 hdfs 源码、kafka 源码、ZooKeeper 源码的精华的抽象。

比如小文件就深度吸收了 hdfs 中 master-slave 架构设计（namenode 和 datanode），网络通信架构吸收了 kafka 的 reactor 网络通信的三层架构支持高性能、高并发；

自研微服务注册中心深度吸收了 zk 集群的 bio 网络通信架构，客户端与服务端之间的 nio 网络通信架构，吸收 zab 协议自己实现了一套 leader 选举机制，吸收 redis cluster 的 hash slot 机制自己设计了一套集群元数据存储机制等等。。。。。
