\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*第0篇\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

快手后端Java岗社招面筋分享  
  
编辑于 2020-07-19 23:59:00  
  
 一面：   
  
 1、先做下自我介绍吧；   
  
   
 2、说一说你最近做的那个项目？然后针对简历提了几个问题。   
 3、我看你简历里面说做过mysql的优化，说说你都做过哪些优化或者优化的思路？   
 4、数据库的索引了解吗？说一下索引的原理？   
 5、聚集索引和非聚集索引了解吗？了解mysql的回表吗？   
 6、mysql实现分布式锁了解吗？还有没有其他更好的方式？   
 7、说一下事务的一些东西？你对事务的了解有哪些？   
 8、说说数据库的乐观锁和悲观锁？   
 9、Redis有哪些持久化方式？你们在项目中一般怎么做持久化？如何实现集群和高可用？   
 10、Java中有哪些锁？   
 11、synchronized与Lock有哪些区别？什   
 12、么是公平锁和非公平锁？他们的底层怎么实现的？   
 13、AQS原理了解吗？能不能详细介绍一下？   
 14、说一下线程池的原理？ExcutorService下的四种线程池分别用在什么场景下？   
 15、为什么单线程池和固定线程池使用的任务阻塞队列是LinkedBlockingQueue()，而缓存线程池使用的是SynchronousQueue()呢？   
 16、说一下violate关键字吧？刚才你提到可见性？他是如何保证可见性的？   
 17、说一下GC吧，什么时候进行Full GC呢？你了解哪些收集器？CMS和G1。详细谈谈G1的优点   
 18、最后写了一个编程题：给定一个二叉搜索树, 找到该树中两个指定节点的最近公共祖先。   
   
 百度百科中最近公共祖先的定义为：“对于有根树 T 的两个结点 p、q，最近公共祖先表示为一个结点 x，满足 x 是 p、q 的祖先且 x 的深度尽可能大（一个节点也可以是它自己的祖先）。”   
 例如，输入: root = [6,2,8,0,4,7,9,null,null,3,5], p = 2, q = 8   
 输出: 6   
 解释: 节点 2 和节点 8 的最近公共祖先是 6。   
 示例 2:   
 输入: root = [6,2,8,0,4,7,9,null,null,3,5], p = 2, q = 4   
 输出: 2   
 解释: 节点 2 和节点 4 的最近公共祖先是 2, 因为根据定义最近公共祖先节点可以为节点本身。   
   
 二面：   
   
 1、面试官上来就让设计了一个小型的社交系统。从数据库设计开始，然后设计架构。这个花了很长时间。   
 2、然后问了下dubbo的原理，有没有看过dubbo源码？dubbo的容错机制？   
 3、给定一个整数数组，其中第 i 个元素代表了第 i 天的股票价格 。   
 设计一个算法计算出最大利润。在满足以下约束条件下，你可以尽可能地完成更多的交易（多次买卖一支股票）:   
 你不能同时参与多笔交易（你必须在再次购买前出售掉之前的股票）。   
 卖出股票后，你无法在第二天买入股票 (即冷冻期为 1 天)。   
 示例:   
 输入: [1,2,3,0,2]   
 输出: 3   
 解释: 对应的交易状态为: [买入, 卖出, 冷冻期, 买入, 卖出]   
   
 4、最后还有什么想问的？

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*第1篇\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

【Fake面经】腾讯、快手社招服务端开发面经  
  
精  
  
  
编辑于 2020-06-28 21:41:33  
  
  
原始面经链接：  
https://www.nowcoder.com/discuss/444262?type=0&order=1&pos=8&page=1&channel=2000&source\_id=discuss\_center\_0  
  
  
 以下是mock回答：   
  
 面经   
 腾讯   
 一面   
 疫情期间，电话面试。   
  
 大数据问题。给定4G内存，以及16亿个QQ号，这些QQ号里面有重复的，找出重复次数排名前100个QQ号。 经典的大数据topn问题，采用分治再归并，每个qq号大约是10位左右，大约占10个字节，4G=4000MB=4000000KB=4^9B=40亿B，可以容纳4亿个QQ号 将16亿个QQ号分为四组，每组在内存内进行count，然后分两个策略   
  
 count结果全量的持久化到磁盘上，优点是精准，缺点是慢   
 count结果取前一千万放在内存里，剩余的丢弃，优点是快，但是可能不太准   
 让我写我会选择综合两个，先剩1KW放内存，然后四组数据互相join，没有匹配到的数据再从磁盘里捞出来计算   
 聊了下微服务，服务注册发现是怎么做的？微服务的监控是怎么做的？

服务注册：   
  
 直接和中心服务器（比如SpringCloud当中的eruka）使用长连接进行心跳保持   
 使用http进行非长连接的注册和留活   
 服务发现：   
  
 在中心服务器维护当前服务的所有真实服务数据   
 在每个consumer中维护所调用服务的副本，在请求时都走本地缓存进行请求路由   
 在真实服务数据发生变更（机器上下线，版本变动等）操作的时候，由中心服务器向consumer进行元数据的推送   
 consumer设置一个定时的后台进程每隔N分钟重新请求   
 监控：   
  
 机器监控 在监控中心维护某provider的所有机器IP，然后去拉去其sar的数据并可视化   
 服务监控   
  
 监控机器的服务提供成功率   
 全链路trace调用监控   
 实现思路都是一样的，打日志然后从监控中心拉机器IP去采日志，成功率通过常规日志即可获得，全链路跟踪在获取全链路id后可生成调用链（此处注意调用链不可依赖时间戳，应该有自己的逻辑调用先后顺序）   
 sc里也有能监控endpoint的组件（具体记不太清了）   
   
 实现一个线程安全的阻塞队列。 阻塞队列用法：FIFO，在队列满的时候让线程等待 则将put方法加一个ReenterLock的ConditionA，然后维护一个volatile的isFull字段，在put之前判断volatile是否true，若为true则直接ConditionA.await，put成功后尝试唤醒pull的condition 将pull方法加一个ReenterLock的ConditionB，然后维护一个volatile的isEmpty字段，在pull之前判断是否空，若空则ConditionB.await，pull成功后尝试唤醒put的condition   
 给定两个数组，每个数组中都有重复的数字。不用类库函数，对这两个数组排序。 两个数组append一下直接快排不就完事儿了吗= =，还是说要加个SET去重   
 多路复用是什么？怎么用？ 多路复用定义：一个线程监听多个端口or文件描述符，减轻cpu线程切换压力 select:监听多个IO任务，但是存在描述符个数限制，所有存在一个上限，并且在每次有事件发生的时候需要遍历持有的描述符列表来确定具体是哪个描述符发生了事件 poll:去除了文件描述符的限制 epoll:监听的描述符完事儿后会回调注册到就绪队列，可以直接从中获得对应的描述符，并且使用了mmap进行zerocopy提升性能，而select/poll都存在在内核空间与用户空间的拷贝   
 Linux中的文件节点是什么？（这个不太会） 我也不太会- -   
 聊了下项目架构 这个各凭本事吧，主要是考虑以下几个点   
  
 对于大流量是否有应对措施   
 对于异步同步怎么考虑的   
 异步丢消息怎么处理   
 在什么地方存在提高性能的trick   
 异常情况怎么考虑，怎么做补偿   
 数据库设计、缓存设计、其他数据源的设计   
   
  
 二面   
 视频面试   
  
 给定一个数组，求该数组所有的自子数组 lc原题，DFS可解   
 去掉一个字符串中的所有空格（送分题） 送分   
 两个系统A和B，如果A调用B的时候发生超时，这个时候A会重试，那么怎么确保B只调用一次？ 幂等，设计考虑要点   
  
 是否完全幂等，即是否要保存这次调用记录但是不改变状态   
 幂等键的定义，该幂等键是否只match这个请求，幂等键又该如何存储   
 返回值的定义，是返回创建失败还是成功，若是成功，是否要告知该次是幂等成功   
   
 项目中的数据库分表是怎么做的？ 水平：根据某个key进行水平拆分，设计要点   
  
 如何规避跨库事务   
 分表键如何设计   
 如何防止热点   
 是否有库可以不拆   
 UUID如何生成   
 垂直：一库拆多库，schema不同，使用库间外键进行关联，涉及分布式事务的时候很恶心，可以使用DDD进行拆分，每个库的核心是一个聚合根   
  
 三面   
 电话面试   
  
 给定一个数组，元素的大小0~25，有重复元素。按出现频次的高低输出所有的数字。 送分，直接map+sort，on扫两编   
 聊一下项目中HBase的RowKey是怎么设计的？ 设计原则   
  
 需要保证唯一，若没用到基于timestamp的version字段则每次查询相同key都会返回最新值，所以要在业务上保证唯一   
 是否满足查询条件，rowkey实际上充当索引的功能   
  
 rowkey只支持全名的get或者range查找，在java客户端中可以只用prefix的filter   
 若进行复杂查询，可以新建一张表做索引表   
 考虑查询热点，需要分散热点   
  
 md5   
 hash   
 取模   
 +随机数   
   
   
 考虑变的更短来节省存储空间：比如用编码代替字符串   
 rowkey冲突：使用占位符或者特殊分隔符   
   
 项目中的事务消息是怎么做的？ RocketMq中的事务消息:   
  
 producer发送一个half消息   
 执行本地逻辑   
 本地逻辑执行失败：rollback   
 本地逻辑执行成功：commit   
 在commit之后这个消息才算是投递成功并且能被consumer所消费   
   
 对加班怎么看？表示自己可以996 钱到位人到位   
  
 四面   
 电话面试，这里好像已经到腾讯的面委会了   
  
 给定一个二叉树，依次打印出每一行。 层序遍历   
 聊一下Redis的zset？实现原理是什么？为什么不用红黑树来实现？ hashtable/skiplist+关联一个double的score 红黑树是特殊二叉平衡树，插入场景下会需要旋转树造成额外开销   
 redis哨兵怎么选举主节点？   
  
 任意一个发现主服务器客观下线之后向所有的sentinel请求成为主节点   
 每个sentinel只认最早来发起请求的主sentinel   
 过半投票投向某个sentinel则成为主   
 若在这次配置纪元中不存在过半投票的则在一段事件后发起新的一轮选主   
   
 为什么IP分组到达的时间不一样？ 这啥啊没看懂   
 有什么要问我的？ 加班吗，干啥活的，为什么招人，我进去后做啥业务   
  
 五面   
 电话面试，我真的已经忘记聊什么了。很快，没聊什么技术。聊了下为什么想换工作？以及自己的职业规划   
 快手   
 一面   
 视频面试   
  
 leetcode 2 2sum，set+一边for循环，on   
 自我介绍 show一下学历、经验以及主要owner的业务和团队定位   
 线程池实现原理，用法 核心设计思想：   
  
 持久化的维护一些线程在内存当中，减小初始化以及关闭线程的损耗（大部分的线程池、连接池的核心思路）   
 存在三级结构来maintain线程   
  
 核心线程池：该线程池当中的线程不会消亡，所有任务优先使用里面的线程   
 任务队列：当核心线程池满的时候添加到任务队列当中，队列满了则会开启新的线程（无界队列除外），队列种类   
  
 LinkedBlockingQueue:基于链表   
 ArrayBlockingQueue:基于数组，吞吐量比linked高（因为基于数组嘛）   
 SynBlokcingQueue:单元素阻塞   
 PriorityBlockingQueue:无限阻塞   
   
 最大线程池大小：在阻塞队列满了之后开启新的线程池进行处理   
   
 使用考虑要点：   
  
 核心线程数的设定：   
  
 若是IO密集型，可以将线程数多设一些（比如2\*cpu核数），因为大量的线程会hang在等待IO上面，不会争抢cpu时间片   
 若是计算密集型，则将其设计为cpu核数，这样保证在没有其他任务运行时能全量抢占cpu，减少上下文切换   
   
 重写线程工厂类：在线程工厂中可以对线程进行命名以及打印Log等操作   
 提交任务的时候可以使用Future进行提交，可以获取线程的运行结果，Future当中的Callable实现类可以自行拓展，主要考虑继承主线程的一些属性（全链路TraceID，RPC上下文，ThreadLocal等，ThreadLocal可以用InheritedThreadLocal进行继承）   
   
 JVM内存结构，垃圾回收机制 内存结构：虚拟机栈、native栈（这俩在hotspot里其实是同一个东西）、方法区（1.8后改为meta space，里面存了方法签名、class信息、hot code cache等）、堆、计数器 GC器： serial：串行 parnew：并发串行 CMS：并发清理，不会始终stw G1：分块清理 GC机制： ROOT：对应上文则是线程栈上的对象、native方法栈上的对象、静态变量、堆中的常量、class锁机制中的monitor、system class loader以及bootstrap class loader中的加载的类 YGC：分s1s0以及eden，比例一般是1:1:8，对象放到eden当中（过大的直接升级到old），当eden满了之后和s1当中的对象merge并且执行清理然后放到s0当中，每一次清理都stw并且会把对象的age+1，若age抵达设置的age则晋升old区 FGC：由YGC晋升来的对象触发，当old区在FGC之后还是无法容纳需要晋升上来的对象则会直接OOM，此处的常规清理流程以CMS举例 CMS：   
  
 并发初始标记（stw）：将GCROOT以及young区关联到Old区的根对象找出来   
 并发标记：以1中的根对象为根，将所有存活对象进行标记   
 预清理：可以设置次数，因为在并发标记阶段会有并行的对象改动，改动完的对象会被标记为dirty，预清理会重新检查这些对象以及相关的对象   
 最终标记（stw）：检查所有dirty对象   
 并发清理：清楚不需要存活的对象   
 reset：将程序初始化等待下一轮   
 优点：只在标记阶段进行stw，在最终的标记前采取预清理机制，多线程标记提速 缺点：多线程会增加cpu负担 附赠小知识：在JVM做垃圾回收的时候并不会将内存即时归还给OS，频繁的申请和归还内存可能会导致堆内存的不稳定，所以在启动JVM时我们一般将xmx和xms设置成相同的值来稳定堆内存空间   
 Synchronized加锁原理。偏向锁、轻量级锁、重量级锁。 Syn的加锁原理通过占用和退出对象的monitor中，monitor的信息存储在对象头当中 锁升级过程：   
  
 在第一次获取对象的锁的时候直接偏向锁，即在monitor中保存当前的线程ID   
 有第二个线程来争抢的时候升级为轻量级锁并进入自旋获取该锁   
 若自旋超出一定次数没拿到这个锁则升级成重量级锁，线程进入等待状态，等待唤醒   
   
 AQS原理。公平锁和非公平锁。 AQS是一个双向队列，通过一个volatile的state来标志当前资源的状态，用当前ownerThread以及状态来确定是否共享以及排他 公平锁：若检测到当前资源被使用则直接进入等待队列 非公平锁：检测到当前资源被使用，先cas自旋一次，获取不到再等待   
 MySQL索引什么时候失效？ 最左前缀匹配失效或者mysql计算这次query根本不需要走索引   
 RocketMQ生产消息，存储消息，生成索引，消费消息全流程。   
  
生产消息： 在producer处组装消息，通过timestamp以及当前机器维护的一个AtomicInteger来生成msgid，拉取当前Topic可用的所有broker然后轮询发送给对应的某台Broker 顺序写入物理队列，开放逻辑队列给每个consumer消费，逻辑队列只允许一个consumer进行消费 优势在于使用了mmap的zerocopy来进行读写，减少IO开销，并且写入都是顺序的，提升刷盘性能 生成索引： 通过slot（就是hash）+链表的方式维护commitQueue到commitLog的映射 消费消息： push方式里，consumer把轮询过程封装了，并注册MessageListener\*\*\*，取到消息后，唤醒MessageListener的consumeMessage()来消费，对用户而言，感觉消息是被推送过来的。 pull方式里，取消息的过程需要用户自己写，首先通过打算消费的Topic拿到MessageQueue的集合，遍历MessageQueue集合，然后针对每个MessageQueue批量取消息，一次取完后，记录该队列下一次要取的开始offset，直到取完了，再换另一个MessageQueue。   
   
 ConcurrentHashMap扩容算法 先判断元素size是否抵达0.75倍的容量，超过阈值则进行扩容，扩容时先获取老table里面的队列，然后初始化一个forwardNode放到该队列的下面，然后将老table里的队列使用头插法迁移到新数组当中，在此期间如果其他线程的有读写操作都会判断head节点是否为forwardNode节点，如果是就帮助扩容，并block。   
 有什么要问我的？ 啥部门啥业务为啥招人   
  
 二面   
 视频面试   
  
 求N内的所有素数； 素数定义：无法被除了一以外的整除，奇数只需要除以奇数，偶数不需要判断，大于该数1/2的数不用除   
 给定一个乱序数组，求数组内最大连续的数； 基础滑窗   
 自我介绍   
 聊项目。项目中的难点是什么？如何解决，我讲了RocketMQ的调优 各凭本事   
 分布式ID实现，不准用UUID 持久化取号+内存atomicInteger   
 MySQL间隙锁的机制？主要解决的问题是什么？ 锁一个range，解决幻读问题（RR隔离下）   
 课余时间怎么学习？ 11点下班熬夜学习到两点   
 有什么要问我的？ 同上   
  
 三面   
 视频面试   
  
 讲项目，redis的作用，HBase RowKey的设计。项目主要做了些什么？有什么难点？ 各凭本事 redis做缓存的话注意击穿、穿透、雪崩以及缓存失效流程（先改持久层再更新redis）与更新失败的补偿机制 redis做分布式锁设计的话注意锁的可重入，续约，跨线程释放时的错误释放问题以及淘汰机制（禁止淘汰、过期随机淘汰、过期LRU淘汰、过期TTL淘汰、全量随机淘汰、全量LRU淘汰） hbase的rowkey设计同上   
 Leetcode 146 基础LRU   
 Leetcode 470 拒绝采样   
 有什么要问我的？ 同上

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*第2篇\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# 【社招Java】腾讯、阿里、美团、快手、字节等10公司面经

编辑于 2020-08-28 17:21:52  
  
2020 年春节后回到北京，准备开始找工作，因为疫情影响，整个 2 月都是在家上班，这也给远程面试提供了便利，免去了请假的尴尬和奔波的麻烦。  
年前就准备开始面试的，但总以为需要刷很多题，一直没时间刷就一直没行动。幸好临近过年有一个同事要离职了，他跟我传授经验，让我先面小公司积累经验，之后再去面大公司。我受他的鼓舞，终于决定了在年后踏上面试之路。  
根据同事的经验，我修改了“脉脉”上的的状态，切换到“找工作”。然而还没来得及找小公司“练手”，就被各种猎头消息淹没了，于是就走上了面试之旅（不是脉脉软文啊……）。  
事实证明“练手”真的很有必要。我一共面过阿里、蚂蚁、字节、Shopee、OPPO、华为、美团、快手、猿辅导、腾讯，城市涉及成都、北京、深圳。一开始我连自我介绍都不会，所以前面面的阿里、蚂蚁、字节、Shopee 都跪了；中间面的 OPPO、华为面了一半失去音讯；后面面的美团、快手、猿辅导、腾讯都拿到了 offer。吐槽一下华为，机试的时候试题都能出错，后面重试的时候全部 AC 也能判断我机试挂掉……还有 OPPO，两面面完发了一个邀请函，不知道干什么用的，问了内推人也没有了后续。  
这些公司每家偏重点不同，难度也不同，有的中规中矩，有的挖尽细节。事后总结了一下，大体都能分为下面列举的 7 个问题，列出来跟大家分享一下。  
  
补充一下，我是 16 本 Java 社招，项目问得比较多，可能是我比较菜，问到的算法 easy 居多，最多出现个 medium。下面也没分一二三四面，没分公司，就是提取了共性。  
  
1 自我介绍  
远程面试，视频接通后，面试官可能会简单介绍下自己，然后就直接让你做一个自我介绍，这个没有例外。  
我一开始心里想这不都是简历上写着的么？后来感觉可能是考察你的表达能力吧。这个时候说说姓名、学校、专业、毕业时间、工作经历、主要项目、技术栈，应该就差不多了。  
说完之后，有的面试官会认真听，并且就你说的找他感兴趣的问你，当然也有的面试官就走个过场，你说完他就不管了，直接问下一步。  
1.1 工作经历  
通常说到工作经历的时候，会问你：  
  
毕业第一份工作为什么选择了那里？   
为什么看机会？上次换工作为什么？   
有了解过我们部门做什么的么？感兴趣么？   
工作这些年，主要的收获是什么？   
  
最好别老抱怨之前的公司，可以说说自己的规划之类的。  
1.2 项目经历  
说到项目经历的时候，有些面试官会从你说的当中跳他感兴趣的问你，所以自我介绍的时候可以有倾向性的说说你比较拿手的项目。  
1.3 未来规划  
我在被问到这个问题的时候通常是一头懵逼，因为我之前没有好好想过，大家可以准备一下。  
  
未来的规划（城市）   
技术上的追求   
长期方向（行业）   
  
1.4 日常行为  
  
平时有看什么技术相关的书，博客？   
最近有在学习什么新技术么？讲讲   
怎么对待 995 的情况？   
怎么处理同事不配合的情况？   
  
2 最有挑战的项目  
自我介绍之后，如果面试官没有从你说的项目里主动问你，一般就会让你自己选一个项目说。  
通常问法是：  
  
在上次工作中，哪些项目最有成就感，最有挑战？   
最有挑战的一个工作，遇到的困难，采取了什么方式，取得了什么结果   
遇到的最难的项目？难在哪里？怎么解决的？得到了什么收获？   
基础的一点的技术接触过什么？感觉枯燥么？碰到什么困难？举一个最难的例子   
举例两个做的比较好的项目   
  
这一块因为每个人都不一样，所以不细写了，但是很重要。社招项目问得是比较细的，事前好好准备一下吧，不要像我一样说到自己做的项目，因为时间比较久了，有些细节都忘了。面试前将自己做过的项目再熟悉熟悉。  
3 基础知识  
基础知识这块是跟你简历上提到的技术栈相关的，每个人不一定相同，但 Redis 和 Kafka 都是没有被错过的。这一块就是考察你掌握是深度了，不一定全都要了解到底层，每家要求不一样。  
3.1 Java  
1）基础  
  
int float short double long char 占字节数？   
int 范围？float 范围？   
hashcode 和 equals 的关系   
深拷贝、浅拷贝区别   
java 异常体系？RuntimeException Exception Error 的区别，举常见的例子   
lambda 表达式中使用外部变量，为什么要 final？   
  
2）集合  
集合这块关注一下 1.8 的实现吧，和 1.7 相比变化了很多  
  
Collection 有什么子接口、有哪些具体的实现   
简单介绍下 ArrayList 怎么实现，加操作、取值操作，什么时候扩容？   
讲一下 hashMap 原理。hashMap 可以并发读么？并发写会有什么问题？   
讲一下 concurrentHashMap 原理。头插法还是尾插法？扩容怎么做？   
堆是怎么存储的，插入是在哪里？   
集合在迭代的过程中，插入或删除数据会怎样？   
  
3）并发  
  
线程  
   
进程和线程的区别？并行和并发的区别？了解协程么？   
进程间如何通信：进程 A 想读取进程 B 的主存怎么办？线程间通信？   
线程的生命周期有哪些状态？怎么转换？   
wait 和 sleep 有什么区别？什么情况下会用到 sleep？   
怎么停止线程？   
怎么控制多个线程按序执行？   
   
线程池  
   
会用到线程池么？怎么使用的？用什么实现的？   
常用的线程池有哪些？用的哪个线程池？什么情况下怎么选择？   
ThreadPoolExecutor 有什么参数？各有什么作用？拒绝策略?   
一个任务从被提交到被执行，线程池做了哪些工作？   
   
锁  
   
讲一下锁，有哪些锁，有什么区别，怎么实现的？   
ReentrantLock 应用场景   
死锁条件   
   
AQS  
   
了解 AQS 么？讲讲底层实现原理   
AQS 有那些实现？   
讲讲 AtomicInteger 的底层实现   
   
volatile 关键字有什么用？怎么理解可见性，一般什么场景去用可见性   
讲一下 threadLocal 原理，threadLocal 是存在 jvm 内存哪一块的   
  
4）IO  
IO 这块我不熟，没有多讲  
  
了解 NIO 么？讲讲   
NIO 与 BIO 有什么区别？   
了解 Netty 原理么   
  
3.2 JVM  
1）内存与 GC  
  
jvm 内存区域分布？gc 发生在哪些部分？   
介绍一下垃圾回收过程。   
垃圾回收算法的了解。现在用的什么回收算法？   
现在使用的什么垃圾回收器？知道哪些？讲讲 G1   
容器的内存和 jvm 的内存有什么关系？参数怎么配置？   
  
2）异常与调优  
  
线上有什么 jvm 参数调整？   
oom 问题排查思路   
线上问题排查，突然长时间未响应，怎么排查，oom   
cpu 使用率特别高，怎么排查？通用方法？定位代码？cpu高的原因？   
频繁 GC 原因？什么时候触发 FGC？   
怎么获取 dump 文件？怎么分析？   
  
3）类加载器  
  
怎么实现自己的类加载器？   
类加载过程？   
初始化顺序？   
  
3.3 Spring  
  
spring 介绍一下   
讲一下 ioc、aop   
ioc 怎么防止循环依赖   
aop 的实现原理、动态代理过程   
tomcat 与 spring、controller 的关系   
spring boot starter 自加载是怎么实现的？在生命周期哪个阶段？   
Spring 处理请求的过程？   
  
3.4 MySQL  
  
数据仓库与 mysql 区别？hive 和 mysql 有什么区别？spark 和 hadoop 区别？mapreduce 互相等待，怎么解决？   
acid 含义？事务隔离级别？幻读怎么解决的？   
用过 mysql 的锁么？有哪些锁？   
MyISAM、InnoDB 区别？为什么不用 MyISAM？   
mvcc 原理？多版本数据存放在哪？   
mysql 脏页？   
redo log，undo log？   
索引  
   
innodb 的索引结构是什么？什么是聚簇索引？   
b+ 树与 b 树的区别？   
b+ 树与二叉树区别，优点？为什么不用红黑树？   
多列索引的结构   
字符串类型和数字类型索引的效率？数据类型隐式转换   
主键与普通索引的联系？存储上的区别？   
   
sql  
   
join 和 in 怎么选择？有什么区别？   
union 和 union all 有什么区别？怎么选择？   
怎么处理 sql 慢查询？   
索引用得不太正常怎么处理？同时有（a，b）和（a，c）的索引，查询 a 的时候，会选哪个索引？   
   
跨库分页的实现？   
分库分表有哪些策略？怎么保证 id 唯一？   
对 uuid 的理解？知道哪些 GUID、Random 算法？   
主键选随机 id、uuid 还是自增 id？为什么？主键有序无序对数据库的影响？   
主从复制的过程？复制原理？怎么保证强一致性？   
  
3.5 网络  
  
tcp  
   
tcp 有哪些机制确保可靠性？拥塞控制怎么实现？   
close\_wait 太多怎么处理？为什么会出现这种情况？   
讲讲三次握手，四次挥手   
   
http  
   
http 2 有了解过么，新增了哪些功能，现在用的什么版本？1.1？   
http 缓存机制都有哪些？什么是 cdn？header 中涉及到缓存的字段有哪些？   
cookie session 介绍一下   
html 页面，怎么与后端交互？流程是什么？涉及到哪些组件？   
http 协议，报文格式？   
keepalive 有什么用？   
Https 原理？   
知道哪些 http 状态码有哪些？   
http 有哪些请求方法？put、post 实现上有什么区别？   
   
前后端分离与不分离的区别？各有什么优缺点？   
常见 web 攻击有哪些？了解 csrf 攻击么？   
restful 的作用？有哪些优点和缺点？   
nginx 达到上限了怎么办？怎么对 nginx 负载均衡？dns？   
nginx 负载均衡有哪些算法？各自有什么优缺点？   
  
3.6 Redis  
  
Redis 数据结构、对象，使用场景   
Redis 内存淘汰策略   
缓存的热点 Key 怎么处理？redis 缓存穿透，怎么避免？   
redis keys 命令有什么缺点   
主从同步原理，新加从库的过程   
RDB 和 AOF 怎么选择，什么场景使用？   
redis 的 zset 的使用场景？底层实现？为什么要用跳表？   
怎么实现 redis 分布式锁？   
  
3.7 Kafka  
  
用 kafka 做了什么功能？   
kafka 内部原理？工作流程？   
Kafka 怎么保证数据可靠性？   
怎么实现 Exactly-Once？   
  
3.8 分布式  
  
有哪些分布式组件是你最熟悉的，简单聊一聊。   
cap 是指什么？mysql 满足 cap 中哪些？   
分布式锁有哪些方式可以实现？各有什么优缺点？   
什么是一致性 hash？自己实现一致性 hash，会用什么数据结构？   
  
3.9 微服务  
  
微服务用的什么体系？   
讲一下熔断概念？熔断原理？令牌桶？熔断三个状态关系？   
熔断会影响性能么？有遇到过线上发生熔断么？不加会怎样？   
什么是 RPC？怎么实现幂等性？   
微服务有什么优缺点？   
配置中心有哪些选项？apollo 的架构？怎么无感实现已加载数据更新？   
  
3.10 设计模式  
  
工厂方法和抽象工厂的区别   
装饰器和代理区别   
单例  
   
对于单例，你知道哪些实现方法？   
实现一个懒加载单例   
双重校验锁为什么需要双重校验？   
   
  
4 算法题  
基本都会问时间复杂度  
  
有哪些常用排序算法？   
手写快排，快排时间复杂度是多少？   
快排算法是不是稳定的？什么是稳定性？   
给定一个字母组成的字符串，找出不含有重复字符的最长子串的长度。   
9 个硬币中有一个劣币，用天平秤，最坏几次？   
深度优先，广度优先   
数字转中文   
括号匹配，时间复杂度，空间复杂度   
单向链表的归并排序   
链表做加法   
1g 大小文件，里面每行是最大 16k 的单词，限制内存 1m，统计单词频率最多的 100 个单词   
手写 100 位带小数的大数字的减法   
全排列。进阶：给数组长度 n，求第 k 行的排列   
  
5 系统设计  
  
一天爬一千万条文章，怎么做设计？怎么并行协调？100 台服务器怎么尽可能负载均衡？   
怎么设计一个秒杀系统？怎么解决大并发？   
设计一个抢红包系统，要注意哪些点   
设计一个微博社交系统，怎么更高效，索引怎么设计、提高效率，查询扫描行数，缓存设计   
在 java 里手动实现一个生产者、消费者模型   
设计一个视频上传的流程。表设计？文件上传服务器的原理？cdn？高qps怎么处理？上传和请求？缓存怎么加？   
有什么分布式 id 生成方法？各自的优缺点是什么？   
反羊毛怎么做？   
设计一个简单的智能家具系统，比如说加湿器和温湿度传感器关联，怎么设计？考虑哪些点？   
设计一个登陆过程。md5 的原理？可逆么？   
  
6 你有什么想问的？  
  
部门定位？岗位职责？   
目前在做什么业务？   
加班情况？   
薪酬福利介绍？   
  
7 Offer 选择  
  
目前手头的 offer 情况？   
还有哪些公司的面试流程？   
这些公司你自己有什么倾向性呢？最看重什么因素？   
你的预期薪酬是多少？目前的薪资情况？   
想加入一个怎么样的团队？   
最近一年的绩效情况？   
  
补充一下 offer 情况，最后拿到了美团基础架构、快手主 app、猿辅导斑马、腾讯 csig 的 offer。面试前的“练手”还是很重要的，前期“裸面”浪费了阿里、蚂蚁、字节、Shopee 的面试机会。所有开始面试之前一定要准备好啊，不然也是耽搁面试官和自己的时间。  
  
2021校招开始了  
点击 此处 ，马上开始上传简历吧！听说今年开发了 5000+ 岗位  
（内推二维码被吞了，点击上面链接吧）  
   
  
首发于 这里

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*第3篇\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

快手社招一二面面经  
  
编辑于 2019-12-26 11:54:36  
  
 一面   
  
 1.自我介绍   
 2.简单介绍一下自己最近做的项目  
 3.针对简历上有的技术以及项目进行提问  
  
  
  
  
  
 具体问题：   
 1.我看你简历里面说做过mysql的优化，说说你都做过哪些优化或者优化的思路？  
 2.数据库的索引了解吗？说一下索引的原理？聚集索引和非聚集索引了解吗？了解mysql的回表吗？  
 3.mysql实现分布式锁了解吗？还有没有其他更好的方式？  
 4.说一下事务的一些东西？你对事务的了解有哪些？说说数据库的乐观锁和悲观锁？  
 5.Redis有哪些持久化方式？你们在项目中一般怎么做持久化？如何实现集群和高可用？  
 6.Java中有哪些锁？synchronized与Lock有哪些区别？什么是公平锁和非公平锁？他们的底层怎么实现的？  
 AQS原理了解吗？能不能详细介绍一下？  
 7.说一下线程池的原理？ExcutorService下的四种线程池分别用在什么场景下？为什么单线程池和固定线程池使用的任务阻塞队列是LinkedBlockingQueue()，而缓存线程池使用的是SynchronousQueue()呢？  
 8.说一下violate关键字吧？刚才你提到可见性？他是如何保证可见性的？  
 9.说一下GC吧，什么时候进行Full GC呢？你了解哪些收集器？CMS和G1。详细谈谈G1的优点  
  
 最后写了一个编程题。  
 给定一个二叉搜索树, 找到该树中两个指定节点的最近公共祖先。  
  
 例如，输入: root = [6,2,8,0,4,7,9,null,null,3,5], p = 2, q = 8  
 输出: 6  
 解释: 节点 2 和节点 8 的最近公共祖先是 6。  
  
 示例 2:  
  
 输入: root = [6,2,8,0,4,7,9,null,null,3,5], p = 2, q = 4  
 输出: 2  
 解释: 节点 2 和节点 4 的最近公共祖先是 2, 因为根据定义最近公共祖先节点可以为节点本身。  
  
  
  
  
 紧接着就约了二面,二面比较注重设计   
  
 二面 面试官上来就让设计了一个小型的社交系统。从数据库设计开始，然后设计架构。这个花了很长时间。  
 然后问了下dubbo的原理，有没有看过dubbo源码？dubbo的容错机制？  
 然后又是写题了，是股票买卖的一道题  
 给定一个整数数组，其中第 i 个元素代表了第 i 天的股票价格 。  
 设计一个算法计算出最大利润。在满足以下约束条件下，你可以尽可能地完成更多的交易（多次买卖一支股票）:  
 你不能同时参与多笔交易（你必须在再次购买前出售掉之前的股票）。  
 卖出股票后，你无法在第二天买入股票 (即冷冻期为 1 天)。  
  
 示例:  
  
 输入: [1,2,3,0,2]  
 输出: 3  
 解释: 对应的交易状态为: [买入, 卖出, 冷冻期, 买入, 卖出]  
 最后还是有什么想问的？

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*第4篇\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

快手 java 社招面经  
  
精  
  
  
编辑于 2019-11-09 16:50:02  
  
快手java一面面经  
 1.要不先做下自我介绍？说一说你最近做的那个项目？然后针对简历提了几个问题。你哪年生的啊？我看你简历上没有。  
  
 2.我看你简历里面说做过mysql的优化，说说你都做过哪些优化或者优化的思路？  
 数据库的索引了解吗？说一下索引的原理？聚集索引和非聚集索引了解吗？了解mysql的回表吗？  
 mysql实现分布式锁了解吗？还有没有其他更好的方式？  
  
 3.说一下事务的一些东西？你对事务的了解有哪些？说说数据库的乐观锁和悲观锁？  
  
 4..Redis有哪些持久化方式？你们在项目中一般怎么做持久化？如何实现集群和高可用？  
  
 5.Java中有哪些锁？synchronized与Lock有哪些区别？什么是公平锁和非公平锁？他们的底层怎么实现的？  
 AQS原理了解吗？能不能详细介绍一下？  
  
 6..说一下线程池的原理？ExcutorService下的四种线程池分别用在什么场景下？为什么单线程池和固定线程池使用的任务阻塞队列是LinkedBlockingQueue()，而缓存线程池使用的是SynchronousQueue()呢？  
  
 7.说一下violate关键字吧？刚才你提到可见性？他是如何保证可见性的？  
  
 8.说一下GC吧，什么时候进行Full GC呢？你了解哪些收集器？CMS和G1。详细谈谈G1的优点  
  
 最后写了一个编程题。  
 给定一个二叉搜索树, 找到该树中两个指定节点的最近公共祖先。  
  
  
 百度百科中最近公共祖先的定义为：“对于有根树 T 的两个结点 p、q，最近公共祖先表示为一个结点 x，满足 x 是 p、q 的祖先且 x 的深度尽可能大（一个节点也可以是它自己的祖先）。”  
  
 例如，输入: root = [6,2,8,0,4,7,9,null,null,3,5], p = 2, q = 8  
 输出: 6   
 解释: 节点 2 和节点 8 的最近公共祖先是 6。  
  
 示例 2:  
  
 输入: root = [6,2,8,0,4,7,9,null,null,3,5], p = 2, q = 4  
 输出: 2  
 解释: 节点 2 和节点 4 的最近公共祖先是 2, 因为根据定义最近公共祖先节点可以为节点本身。  
  
  
 这个是leetcode原题。  
  
 写完面试官问方便来北京现场面试吗？你还有什么想问的？后续有什么消息hr会通知你。  
  
 紧接着就约了二面  
  
 二面比较注重设计  
  
 面试官上来就让设计了一个小型的社交系统。从数据库设计开始，然后设计架构。这个花了很长时间。  
 然后问了下dubbo的原理，有没有看过dubbo源码？dubbo的容错机制？  
 然后又是写题了  
 是股票买卖的一道题  
 给定一个整数数组，其中第 i 个元素代表了第 i 天的股票价格 。  
  
 设计一个算法计算出最大利润。在满足以下约束条件下，你可以尽可能地完成更多的交易（多次买卖一支股票）:  
  
 你不能同时参与多笔交易（你必须在再次购买前出售掉之前的股票）。  
 卖出股票后，你无法在第二天买入股票 (即冷冻期为 1 天)。  
  
 示例:  
  
 输入: [1,2,3,0,2]  
 输出: 3  
 解释: 对应的交易状态为: [买入, 卖出, 冷冻期, 买入, 卖出]  
 最后还是有什么想问的？

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*第5篇\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

社招-一年经验-Java开发-知乎/快看漫画/快手/头条面经  
  
精  
  
  
编辑于 2019-10-31 22:29:33  
  
 总结   
  
面试了 知乎/快看漫画/快手/头条 四家公司，拿到了其中两家offer  
  
  
感觉面试难度整体比校招简单一些，可能是由于有正式项目经历了，基础知识问的少了些  
  
  
感觉行情不大好，不过好好准备还是能得到心仪的 offer  
  
  
 二年没面试了，突然出去面试会很紧张，可以先找一个公司试试手，如果拿到了 offer 会极大的增加自信心   
  
 项目经历   
  
这个地方我写的不是很好，因为参与的项目大都已经稳定，更多的是维护工作和业务修改。  
  
  
要写最能体现能力的项目，如果只是简单参与了一个厉害的项目，但是你能说的清楚，有信心不被面试官问住，我觉的可以写上。  
  
  
因为很多面试官看完我的项目后，觉得没啥可问的，让我自己选择一个最熟悉或有啥可讲的项目，不局限于简历。  
  
 基础知识   
  
面试的知识点大都还是问简历上提到的，如果自己没接触过的，感觉也没必要特意去学习。  
  
 缓存一致性问题   
 数据先写 redis？还是先写 MySQL？ 写入失败怎么办？ redis 或者 MySQL 挂掉怎么办？   
 自己没准备好，多次提问总是能问住我。。。   
 场景设计题   
 有 10g 的文件，里面是 ip-访问次数 格式的数据（同一个ip 可能随机出现多次），给你1g内存，让你按访问次数倒排   
 发微博 @朋友的时候，优先显示最近@的十位朋友，怎么实现   
 MySQL   
 索引的数据结构   
 b+tree 基础知识   
 b+tree 与 b-tree 的区别，为啥要用 b+tree   
 简述通过索引查找数据的过程（非主键索引呢？）   
 delete 与 truncate drop   
 事务隔离级别，MySQL 默认的隔离级别   
 Kafka   
 Kafka 可以保证数据不丢失吗？如果保证的   
 Kafka 可以保证 Exactly-once-semantics 吗？如何实现的？   
 topic/broker/consumer/producer/consumer group 各个概念的含义及相互之间的关系   
 Redis   
 基本数据类型   
 持久化的方式   
 内存淘汰策略   
 使用场景   
 JVM   
 简述 G1，可预测停顿时间，是怎么做到的？   
 遇到过线上事故吗？ 怎么查问题，怎么解决？   
 简述内存运行时数据区   
 多线程   
 j.u.c   
  
lock synchronized volatile 的区别  
  
  
 简单了解内存屏障   
  
 读写锁   
 AQS 好像简单提了下   
 解释下 dcl   
 为啥要用线程池，线程池的参数含义   
 集合框架   
 hashmap   
 ConcurrentHashMap   
 ElasticSearch   
 一次检索请求的流程   
 倒排索引   
 Shard 和 Replicas 的含义   
 Segment 文件会一直增加吗？   
 为什么 ElasticSearch 是近实时的？   
 网络编程   
 tcp/ip 协议   
 https 是啥？   
 nio 是啥？   
 分布式锁   
 zk 与 redis 分别是怎样实现的，区别与适用场景   
 算法   
  
手写归并排序。  
   
两个有序数组合并。  
  
  
一个二维数组，每一列的数字从左往右增大，每一行从上往下增大，求一个指定的数字在这个数组中的位置。  
  
  
一个二叉搜索树，找出某两个节点的公共祖先。  
  
  
给出两个链表的头结点，找出这两个链表的交点。