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

百度后端Java面经分享  
  
发布于 2020-08-09 18:08:17  
  
 一面   
   
 1、刚来就是写代码：一个单链表的算法题目   
 2、随后面试官发问在你的项目中spring是如何管理zookeeper的？   
 3、了解zookeeper的leader选取算法吗，讲一下它的流程。   
 （1）自增选举轮次。Zookeeper规定所有有效的投票都必须在同一轮次中，在开始新一轮投票时，会首先对logicalclock进行自增操作。   
 （2）初始化选票。在开始进行新一轮投票之前，每个服务器都会初始化自身的选票，并且在初始化阶段，每台服务器都会将自己推举为Leader。   
 （3）发送初始化选票。完成选票的初始化后，服务器就会发起第一次投票。Zookeeper会将刚刚初始化好的选票放入sendqueue中，由发送器WorkerSender负责发送出去。   
 （4）接收外部投票。每台服务器会不断地从recvqueue队列中获取外部选票。如果服务器发现无法获取到任何外部投票，那么就会立即确认自己是否和集群中其他服务器保持着有效的连接，如果没有连接，则马上建立连接，如果已经建立了连接，则再次发送自己当前的内部投票。   
 （5）判断选举轮次。在发送完初始化选票之后，接着开始处理外部投票。在处理外部投票时，会根据选举轮次来进行不同的处理。   
   
 ·外部投票的选举轮次大于内部投票。若服务器自身的选举轮次落后于该外部投票对应服务器的选举轮次，那么就会立即更新自己的选举轮次(logicalclock)，并且清空所有已经收到的投票，然后使用初始化的投票来进行PK以确定是否变更内部投票。最终再将内部投票发送出去。   
 ·外部投票的选举轮次小于内部投票。若服务器接收的外选票的选举轮次落后于自身的选举轮次，那么Zookeeper就会直接忽略该外部投票，不做任何处理，并返回步骤4。   
 ·外部投票的选举轮次等于内部投票。此时可以开始进行选票PK。   
   
 （6）选票PK。在进行选票PK时，符合任意一个条件就需要变更投票。   
 · 若外部投票中推举的Leader服务器的选举轮次大于内部投票，那么需要变更投票。   
 · 若选举轮次一致，那么就对比两者的ZXID，若外部投票的ZXID大，那么需要变更投票。   
 · 若两者的ZXID一致，那么就对比两者的SID，若外部投票的SID大，那么就需要变更投票。   
   
 （7）变更投票。经过PK后，若确定了外部投票优于内部投票，那么就变更投票，即使用外部投票的选票信息来覆盖内部投票，变更完成后，再次将这个变更后的内部投票发送出去。   
 （8）选票归档。无论是否变更了投票，都会将刚刚收到的那份外部投票放入选票集合recvset中进行归档。recvset用于记录当前服务器在本轮次的Leader选举中收到的所有外部投票。   
   
 （9）统计投票。完成选票归档后，就可以开始统计投票，统计投票是为了统计集群中是否已经有过半的服务器认可了当前的内部投票，如果确定已经有过半服务器认可了该投票，则终止投票。否则返回步骤（4）。   
   
 （10）更新服务器状态。若已经确定可以终止投票，那么就开始更新服务器状态，服务器首选判断当前被过半服务器认可的投票所对应的Leader服务器是否是自己，若是自己，则将自己的服务器状态更新为LEADING，若不是，则根据具体情况来确定自己是FOLLOWING或是OBSERVING。   
   
 4、知道zab算法吗，讲一下它的流程。   
   
 二面   
   
 主要是两个方面，一个jvm，一个并发编程。   
 1、jvm主要是性能调优，jstack的使用，full GC和minor GC的分析等   
 2、并发编程问的很细。多个线程达到同一个状态然后再一起执行，达到某一个状态之后再继续并发执行，这种怎么实现？   
 答：jdk5之后有一个CyclicBarrir,通过这个来实现，它可以重用等   
   
 3、读写锁中加读锁后如何避免写线程饿死？   
 4、如何实现控制线程在某段时间内完成，不完成就撤销。   
 答：实现Callable接口，返回FutureTask类或者Future接口，然后去触发撤销操作。   
   
 5、然后开始问项目，描述一下对账系统，数据量大吗，并发量高吗，数据库怎么设计的等。   
   
 6、最后问我业务，你在第三方支付公司，能不能讲一下支付的技术流程，然后把自己理解的支付流程讲了一下。   
   
 三面   
   
 1、问异步处理的幂等性。   
 2、唯一性索引或者开发分布式锁。   
 3、你觉得你来百度能给百度带来什么样的价值。   
 4、你希望你加入的百度的团队是一个什么样的团队，你如何和产品经理沟通，假如你和其他研发接口制定无法统一你该如何解决等。

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

百度社招面经分享（Java）  
  
编辑于 2020-08-04 09:57:41  
  
 一面   
   
 1、首先是项目部分，问的比较细；   
 2、如果一个对象有多个方法加了synchronized，那么该对象有几把锁   
 答：对象锁是在一个类的对象上加的的锁，只有一把，不管有几个方法进行了同步。这些同步方法都共有一把锁，只要一个线程获得了这个对象锁，其他的线程就不能访问该对象的任何一个同步方法。   
   
 3、NIO与AIO的区别以及各自的作用；   
 （1）Java NIO ：同步非阻塞，服务器实现模式为一个请求一个线程，即客户端发送的连接请求都会注册到多路复用器上，多路复用器轮询到连接有I/O请求时才启动一个线程进行处理。NIO方式适用于连接数目多且连接比较短（轻操作）的架构，比如聊天服务器。   
 （2）Java AIO(NIO.2) ： 异步非阻塞，服务器实现模式为一个有效请求一个线程，客户端的I/O请求都是由OS先完成了再通知服务器应用去启动线程进行处理。AIO方式使用于连接数目多且连接比较长（重操作）的架构，比如相册服务器。   
   
 4、IOC的实现原理；   
 Spring的IOC实现原理就是工厂模式加反射机制，通俗来讲就是根据给出的类名（字符串方式）来动态地生成对象，这种编程方式可以让对象在生成时才被决定到底是哪一种对象。把IOC容器的工作模式看做是工厂模式的升华，可以把IOC容器看作是一个工厂，这个工厂里要生产的对象都在配置文件中给出定义，然后利用编程语言提供的反射机制，根据配置文件中给出的类名生成相应的对象。   
   
 5、反射的原理；   
 反射首先是能够获取到Java中的反射类的字节码，然后将字节码中的方法，变量，构造函数等映射成 相应的 Method、Filed、Constructor 等类。   
   
 6、数据库调优思路的思路；   
 7、开放性的问题，如何提高系统QPS。   
   
 二面   
   
 1、redis做分布式锁怎么做的？   
 2、Redis能做分布式锁的原理？   
 答：Redis为单进程单线程模式，采用队列模式将并发访问变成串行访问，且多客户端对Redis的连接并不存在竞争关系。   
   
 3、JUC下面主要用哪些东西？   
 4、说一下AQS？   
 5、ReentrantLock非公平锁和公平锁的实现原理？   
   
   
 三面   
   
 1、redis分布式锁如何保证原子性？   
 2、用过ThreadLocal吗？简单说一下？   
 答：ThreadLocal采用的是以空间换时间的方式，为每个线程提供一份变量副本。每一个线程都可以独立的改变自己的副本。   
   
 3、Java8之后的ConcurrentHashMap, 舍弃分段锁   
 4、介绍下cms收集器   
 5、redis缓存与数据库一致性问题？   
 答：一致性问题主要出现在数据更新的时候，通常在更新时采取删除缓存而不是更新缓存。具体一点，先淘汰缓存，再写数据库。因为如果先写数据库可能出现如果有读请求发生，可能导致旧数据入缓存，引发数据不一致。   
  
  
  
  
 6、  
 dubbo有哪几种负载均衡策略？   
  
   
 7、介绍下一致性hash？   
 8、介绍下mysql的回表和覆盖索引？   
 9、说下模板方法模式？

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

百度后端Java社招面经  
  
编辑于 2020-07-30 09:45:10  
  
 1、volatile和synchronized原理；   
 2、TCP 三次握手，四次挥手，TCP在OSI七层模型哪一层   
 3、HashMap，ConcurrentHashMap；   
 4、epoll，select，poll；   
 5、有没有用过异步I/O，说一下select、poll、epoll的区别   
 6、异步I/O和同步I/O的区别   
 7、jvm内存分布，垃圾收集，eden区；   
 8、jvm调优；   
 9、java对象头里有什么？（只答了age，monitor，其他的忘了）   
 10、mysql主从复制；   
 11、围绕项目问了一些问题，根据项目出了场景题；（dubbo，推荐框架，spark streaming）   
 12、为什么用mongo；   
 13、mysql集群；   
 14、爬虫查重过程；   
 15、推荐冷启动；   
 16、推荐怎么做的？   
 17、生产者消费者队列；   
 18、volatile用在什么场景？答了个单例模式；   
 19、场景题：数据量很大的订单商家统计排序；

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

百度、瑞修得Java工程师社招面经分享  
  
编辑于 2020-07-28 16:48:36  
  
 百度   
   
 面试过程：   
 1、java jvm的构成和组件   
 2、jvm如何优化   
 3、java 面向对象都有什么特点   
 4、设计数据库，1个名字空间，1个表，表里面的某一个ID是自增的，如何设计   
 5、如何实现多台缓存服务器数据的负载均衡   
 6、多线程的经验   
 7、如何给系统调优，例如系统提供服务，本来机器性能是可以跟上的，但是由于程序原因，本来可以并行10000个但是最后只能100个。   
 8、zk宕机，dubbo服务还能不能用？   
 答：能用，因为dubbo在连zk的时候会本地缓存一份接口数据和地址   
 9、hasecode的算法是怎么样的？   
 10、你以前工作中碰到过什么难题，请问你怎么解决的。   
   
 瑞修得   
   
 面试过程：   
 1、先问了些项目的具体情况，及有些并发问题的解决方案；   
 2、数据库分库分表；   
 3、Spring IOC 实现；   
 4、java classloader；   
 5、java的oop思想。（这只是一个思想，面相对象设计，例如java有三大特性：比如：封装，继承，多态和抽象。）   
   
 5、创建线程的方式；   
 答：有三种方式可以用来创建线程： 继承Thread类 实现Runnable接口 应用程序可以使用Executor框架来创建线程池   
   
 6、redis的key可以存什么样的类型。   
 答：二进制的序列，空值也行。   
   
 7、hashMap相关的知识，一致性hash；   
 8、冒泡和二分，数据排序，遍历   
 9、上传一个EXCEL后，EXCEL中的数据插入数据库需要比较长的时间，用户需要看到进度，怎么解决？   
 答：定时扫描已插入的条数，除以总条数得到比例，用异步返回给前端展示。

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

小米/百度/青鸟软通社招面经分享（Java工程师）  
  
编辑于 2020-07-13 12:31:33  
  
 小米   
   
 面试过程：   
 1、先是自我介绍，然后让我讲讲做项目的流程；   
 2、紧接着又出了一题编程题，从前后两边查找指定字符出现的个数；   
 3、怎么判断一棵树是不是二叉搜索树；   
 答：可以判断给定二叉树中序遍历之后是不是有序的；   
   
 4、面试官接着问我怎么判断这个序列是不是有序的；   
 答：判断遍历得到的序列和这个排好序的序列是不是一个序列（各值相等）   
   
 5、你对自己的职业规划是什么样的？你认为跳槽的年限大概多久？   
 6、你对小米有什么看法？你认为你加入小米能给小米带来什么？   
   
 百度   
   
 面试过程：   
 1、自我介绍了一番后，问到之前的工作经历以及负责的主要工作内容，用过哪些计算机编程语言。   
 2、然后是一些比较基础的问题，Jquery、多线程、Struts、Spring等等；   
 3、接着问了关于list,set,底层实现的源码。   
 4、还问了如何把mysql表中的数据，行转列，列转行。   
 5、你对待财富的看法；   
 6、为什么从上一家公司离职？   
 7、期望的薪资；   
   
 青鸟软通   
   
 面试过程：   
  
 1、问  
 oracle相关知识点；  
 orcale数据库中读取的数据怎么样转变成  
 date类型   
  
  
 答：  
 todate()  
  
 2、关于spring，springMVC；   
 3、根据工作经验给你一个场景问如何设计解决方案；   
 4、还有互联网的一些开发模式等；   
 5、讨论以前做过的一些项目情况，所担任的职责；   
 6、问了关于职业规划，个人优缺点等；   
 7、对于薪资的期望；

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

百度/微元数据/汇通天下Java社招面经分享  
  
编辑于 2020-05-27 17:28:55  
  
 百度   
 面试过程：   
 1、首先详细询问了最近一份工作项目的架构和工作内容；   
 2、然后围绕工作中用到的组件和中间件技术来扩展，考察掌握程度；   
 3、MySQL InnoDB存储的文件结构；   
 4、索引树是如何维护的？   
 5、数据库自增主键可能的问题；   
 6、Redis的并发竞争问题如何解决？   
 7、了解Redis事务的CAS操作吗？   
 8、分析线程池的实现原理和线程的调度过程；   
 9、动态代理的几种方式；   
 10、Spring AOP与IOC的实现；   
 11、为什么CGlib方式可以对接口实现代理？   
 12、RMI与代理模式；   
 13、Dubbo的底层实现原理和机制；   
 14、描述一个服务从发布到被消费的详细过程；   
  
 15、算法方面考察了一个简单的数组就地去重问题，用丢弃数组尾部元素的方式实现了。   
  
  
  
  
 微元数据科技有限公司   
 面试过程：   
 1、问java基础，锁；   
 2、阐述一下自己的项目；   
 3、分布式系统怎么做服务治理；   
 4、接口的幂等性的概念；   
 5、Maven出现版本冲突如何解决；   
 6、JVM垃圾回收机制，何时触发MinorGC等操作；   
 7、新生代和老生代的内存回收策略；   
 8、Eden和Survivor的比例分配等；   
  
 9、Synchronized和Lock的区别；   
  
  
  
  
 汇通天下科技物联有限公司   
 面试过程：   
 1、Java基础知识+分布式框架，考察架构分析能力；   
 2、问你常用的技术框架架构；   
 3、还问了集合、输出流、框架底层原理、数据库、前端。   
 4、说明职位的背景情况，了解过去的工作经历；   
 5、对薪资的期望；

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*第6篇\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

一些Java社招面试经历  
  
编辑于 2020-05-09 22:25:23  
  
 楼主是非科班自学的java，毕业后进入卡中心，现在是两年多的开发经验。去年辞职，20年年初海投了很多公司，先后面了很多家，现在将自己的经历分享出来，希望能帮到大家。   
   
 华院分析技术有限公司   
 面试过程：   
 1、做过的项目，自己的两个优点两个缺点。然后根据我的回答深入问。   
 2、技术方面根据我自己用到的技术深入了解：   
 •spring: spring技术的用处？   
 •hibernate:hibernate如何映射数据库才能达到最好效果？   
 •sqlserver：怎么提升数据库查询速度？   
 •若用户反映一个功能很慢，怎么办？   
 3、问了我有什么疑问？   
 4、需要经常出差，全国范围内，最长半年，每月可回家一次。   
   
 美通社(亚洲)   
 面试过程:   
 1、针对你的工作经历问一些问题，你是怎么设计的，难点在哪，如何考虑的等等，如果你是自己动手做的项目，就不会有任何问题。   
 2、之后他会介绍自己的项目，哪些地方是瓶颈，讨论怎么改进等 。   
   
 百度糯米   
 一面:   
 简单了解个人状况和工作经历。   
   
 笔试:   
 主要是SQL、英语水平测试、java、算法。   
   
 二面:   
 详细了解工作及其项目经历，并且会考英语口语、听力。   
   
 三面:   
 谈薪水、入职时间等问题 。   
   
 卓望信息技术有限公司   
 面试过程：   
 问我又什么规划，我说做架构和核心代码编写。后面有问了很多问题，最后说做开发没有问题，做架构他没有能力判断。就让我回去了。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*第7篇\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

百度Java社招经历分享  
  
编辑于 2020-05-09 10:58:40  
  
 接上一篇，例行背景介绍：   
 从3月初开始面到4月底，社招一年多年经验, 先后面了阿里、百度、网易。在看机会的时候在牛客网刷到一些面经感觉很有收获，现在也把自己的面试百度的经历分享出来，希望能够帮助到正在找工作的Java同学。   
   
 1、例行自我介绍，介绍自己的项目经验，以及项目上用到的技术栈。   
 2、Spring中创建对象的注解有哪些，分别介绍下区别。(即声明spring的bean的注解)   
 3、Ioc容器的加载流程。   
 4、Bean的生命周期，需要说下涉及到的一些接口名。   
 5、Bean的作用域有哪些？   
 6、spring常用的类有哪些，BeanFactory.   
 7、Spring中用到的一些设计模式，举例具体的场景。   
 8、SpringMVC的优势是啥，它的出现解决了一些什么问题？   
 9、Spring的单例Bean是线程安全的吗？   
 10、介绍一下线程池，使用线程池的好处，参数有哪些？   
 11、线程池有哪几种，优劣是啥？   
 12、线程的参数有哪几个？   
 13、线程池的拒绝策略有哪些？   
 14、线程池execute 和 submit的区别   
 15、synchronized 和 volatile   
 16、ThreadLocal的特性和应用场景   
 17、JVM的内存模型   
 18、JVM加载class的流程   
 19、算法题：（讲解思路，分析时间复杂度，本地IDEA实现）   
 （1）给一个整形数组，以及一个整形数字n，获取数组中那两个数字加起来等于结果n。   
 （2）链表合并：给出n个有序的链表，将他们合并为一个有序链表。   
   
 感觉答得不是很好，后面结果出来果然是凉了。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*第8篇\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

四面百度基础平台开发工程师社招  
  
精  
  
  
编辑于 2020-04-30 18:07:09  
  
  
 楼主18年毕业，渣硕一枚，19年经同学内推，面试了百度基础平台开发工程师岗位，两面技术+总监+经理面后被pass，面试完后自己有记录，这次放到牛客上回馈牛友们。   
  
  
 上学时一直帮导师做项目，搞的是.NET，毕业去公司做的是Java，一直是搞业务的，这次面试基础平台开发工程师，也是有想转型的意愿，毕竟业务搞得久了，   
  
  
 也想搞搞纯技术.....   
  
  
 废话不多说，上面筋：   
  
  
 一面：   
  
  
 自我介绍，基本是上学时的经历，研究方向，做的一些东西，然后是工作的一些项目经验。   
  
  
 HashMap结构，原理，为什么线程不安全，resize为什么会导致死循环，然后就是CurrentHashMap的结构，分断锁原理等等；   
  
  
 刚刚说了HashMap结构有红黑树，讲一讲红黑树的结构，balabalabala。。。。。   
  
  
 B树、B+树什么的结构知道吗？   
  
  
 balabalabala。。。。。   
  
  
 用的什么数据库？我说Mysql和SqlServer，然后就问mysql的隔离级别，mysql有几种引擎，各有什么特点？数据库索引，最左匹配原则。   
  
  
 InnoDB和MyISAM索引有什么区别？聚集索引和非聚集索引？各有什么应用场景，为什么InnoDB用的多？   
  
  
 数据库锁知道吗，有几种，加锁和解锁的场景，给一句SQL和隔离级别，能分析加什么锁吗？索引下推原理。。。   
  
  
 （数据库这块因为提前做过准备，就一直引导面试官问，聊的比较多。）   
  
  
 然后手撕了一个堆排序；   
  
  
 七层网络的协议，网络传输流程，ARP协议等等，三次握手，四次分手，拥塞控制，快重传和满开始等等。   
  
  
 到这基本上结束了，数据库方面聊的比较好，其他有些没答上来。   
  
  
  
  
  
 二面：   
  
  
 这次应该是技术leader了，还是先自我介绍，balabalabala......   
  
  
 海量数据的处理问题，网上很多，不多说。   
  
  
 看你用过Dubbo，讲下原理，额！然后RPC是什么，和restful的http有什么区别，什么场景下使用，socket和他们有什么关系，是那部分的封装。   
  
  
 常见的远程通信框架有什么？远程通信协议有哪些？序列化协议、传输方式？动态代理有几种？   
  
  
 dubbo默认使用什么传输协议？mina和netty知道吗？   
  
  
 BIO、NIO、AIO区别，原理是什么，有哪些实现，我说redis是NIO，然后开始怼redis。   
  
  
 redis poll、epoll，持久化，缓存一致性怎么实现，具体说说。。。。   
  
  
 zookeeper了解吗，能完整叙述下zookeeper启动和容错的选举流程吗？这个真的不会。   
  
  
 redis集群搭建，投票容错机制、高可用等等，redis问了蛮多，其实还有像事务，消息队列，除了五种数据结构其他的，redis模块等等，答得不好！   
  
  
 然后问了公司做的项目，聊了聊项目架构，技术选型，你觉得有哪些可以改进，我说路由要重新设计，就问我怎么设计？给出方案.....   
  
  
  
  
  
 三面：   
  
  
 经理面，主要聊了聊工作经历，为什么想来百度，以前是做业务的，现在换成基础平台开发，会不会转型难度大，了解k8s吗？然后又问了在学校学习情况，为啥没拿过本科没拿过奖学金（本科没好好学呗）。   
  
  
 感觉经理觉得我转型不靠谱，于是又安排总监面了一次。   
  
  
  
  
  
 四面：   
  
  
 总监面，还是聊了聊为什么想来，工作中有没有超额完成任务，有没有平时想主动优化代码，做过哪些工作范围外的创新或者业绩，，能不能接受加班。   
  
  
  
  
  
  
  
  
 最后还是被pass了，但其实并没有不开心。整个面试下来，感觉我的确有点低于他们期望值——能加班，技术还6.....

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*第9篇\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

记百度Java岗社招的一次面试  
  
编辑于 2020-04-26 10:56:32  
  
  
 一面:   
  
  
 1.写代码，一个单链表的算法题目。   
  
  
 2.在你的项目中spring是如何管理zookeeper的？   
  
  
 3.了解zookeeper的leader选取算法吗，讲一下他的流程   
  
  
 4.知道zab算法吗，讲一下他的流程   
  
  
  
  
  
 二面:   
  
  
 1.问了jvm，主要是性能调优，jstack的使用，full GC和minorGC的分析等   
  
  
 2.多个线程达到同一个状态然后再一起执行，达到某一个状态之后再继续并发执行，这种怎么实现［jdk5之后有一个CyclicBarrir,通过这个来实现，它可以重用等等］   
  
  
 3.读写锁中加读锁后如何避免写线程饿死。   
  
  
 4.如何实现控制线程www.qiangeseo.com在某段时间内完成，不完成就撤销。［实现Callable接口，返回FutureTask类或者Future接口，然后去触发撤销操作。］   
  
  
 5.描述一下对账系统......数据量大吗，并发量高吗，数据库怎么设计的等等   
  
  
 6.你在第三方支付公司，能不能讲一下支付的技术流程？   
  
  
  
  
  
 技术总监面:   
  
  
 1.异步处理的幂等性［唯一性索引或者开发分布式锁］   
  
  
 2.你觉得你来百度能给百度带来什么样的价值？   
  
  
 3.你希望你加入的百度的团队是一个什么样的团队   
  
  
 4.你如何和产品经理沟通   
  
  
 5.假如你和其他研发接口制定无法统一你该如何解决

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*第10篇\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

百度、优酷、搜狐和阿里java社招面试总结  
  
编辑于 2020-04-13 10:10:46  
  
一、百度  
 •先详细询问了最近一份工作项目的架构和工作内容  
 •MySQL InnoDB存储的文件结构  
 •索引树是如何维护的？  
 •数据库自增主键可能的问题  
 •Redis的主键争用问题如何解决  
 •了解Redis事务的CAS操作吗  
 •分析线程池的实现原理和线程的调度过程  
 •动态代理的几种方式  
 •Spring AOP与IOC的实现  
 •为什么CGlib方式可以对接口实现代理？  
 •RMI与代理模式  
 •Dubbo的底层实现原理和机制。  
 •描述一个服务从发布到被消费的详细过程  
 •数组就地去重问题，用丢弃数组尾部元素的方式实现了。  
 •分布式系统怎么做服务治理  
 •接口的幂等性的概念  
 •Maven出现版本冲突如何解决  
 •JVM垃圾回收机制，何时触发MinorGC等操作  
 •新生代和老生代的内存回收策略  
  
 •Eden和Survivor的比例分配等  
 •Synchronized和Lock的区别  
  
  
  
 二、阿里巴巴  
 •JVM内存分代，  
 •Java 8的内存分代改进  
 •深入分析了Classloader，双亲委派机制  
 •JVM的编译优化  
 •对Java内存模型的理解，以及其在并发中的应用  
 •指令重排序，内存栅栏等  
 •HashMap的并发问题  
 •了解LinkedHashMap的应用吗  
 •在工作中遇到过哪些设计模式，是如何应用的  
  
 三、优酷  
 •Java基础  
 •Spring原理  
 •Java NIO  
 •并发和集合框架  
 •TCP/IP协议  
 •长连接与短连接  
 •mapreduce过程  
 •多路归并的时间复杂度  
 •海量url去重类问题  
 •Java NIO使用  
 •倒排索引的原理  
 •对分词技术的了解  
  
  
 四、搜狐  
 •首先针对笔试和简历提问了一些基础问题  
 •提问了项目  
 •消息中间件如何解决消息丢失问题  
 •Dubbo的服务请求失败怎么处理  
 •重连机制会不会造成错误  
 •对分布式事务的理解  
 •深入分析几个设计模式  
 •不定长字符串转为定长字符串的问题

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*第11篇\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

百度云java社招三面面经  
  
精  
  
  
编辑于 2020-01-03 01:11:33  
  
        项目部分就不具体说了。问了怎么排查问题，都遇到哪些难题什么的。  
         1、redis分布式锁如何保证原子性？  
         可以了解下这个命令：set key value [EX seconds] [PX milliseconds] [NX|XX]  
  
         2、用过ThreadLocal吗？简单说一下？  
         ThreadLocal采用的是以空间换时间的方式，为每个线程提供一份变量副本。每一个线程都可以独立的改变自己的副本。具体例子可以自己去网上找。  
         3、Java8之后的ConcurrentHashMap, 舍弃分段锁  
         通过  JDK 的源码和官方文档看来， 他们认为的弃用分段锁的原因由以下几点：  
         加入多个分段锁浪费内存空间。  
         生产环境中， map 在放入时竞争同一个锁的概率非常小，分段锁反而会造成更新等操作的长时间等待。  
         为了提高 GC 的效率。  
         4、介绍下cms收集器  
         CMS收集器是老年代收集器，可以配合新生代的Serial和ParNew收集器一起使用。使用的是标记清除算法，容易产生内存碎片。  
         4个步骤： 1.初始标记--》并发标记--》重新标记--》并发清除  （初始标记、重新标记）仍需STW。但初始标记仅仅只标记了一下GC Roots能直接关联到的对象，速度很快。  
         而重新标记则是修正并发标记期间因用户程序继续运行而导致标记产生变动的那一部分对象的标记记录，虽然一般比初始标记阶段稍长，但要远小于并发标记时间。  
  
         5、redis缓存与数据库一致性问题？  
         一致性问题主要出现在数据更新的时候，通常在更新时采取删除缓存而不是更新缓存。  
         具体一点？先淘汰缓存，再写数据库。因为如果先写数据库可能出现如果有读请求发生，可能导致旧数据入缓存，引发数据不一致。  
  
         6、dubbo有哪几种负载均衡策略？  
         1.RandomLoadBalance:按权重随机调用，这种方式是dubbo默认的负载均衡策略  
         2.RoundRobinLoadBalance：轮询，按公约后的权重设置轮询比率  
         3.LeastActiveLoadBalance：最少活跃次数  
         4.ConsistentHashLoadBalance：一致性hash  
         5.自定义负载均衡策略  
  
         7、介绍下一致性hash？  
         先说下普通hash有机器宕机或者新加机器的后果。然后  
         介绍一致性hash。  
         hash值是个整数非负数值，所有的hash值形成一个闭圆环  
         对集群的的节点的某个属性求hash值，放到环上  
         数据key求hash值，也放到环上。  
         数据的hash值按顺时针找到离它最近的节点，放在该节点上。  
  
         8、介绍下mysql的回表和覆盖索引？  
         回表简单来说就是数据库根据索引（非主键）找到了指定记录所在行后，还需要根据主键再次到数据库里获取数据。  
         如果一个索引包含(或覆盖)所有需要查询的字段的值，称为‘覆盖索引’。即只需扫描索引而无须回表。  
  
         9、说下模板方法模式？  
         所谓模版方式模式：把不变的行为搬到超类，去除子类中重复的代码来体现他的优势；当不变的和可变的行为在子类实现中混合在一起的时候，  
         不变的行为就会在子类中重复实现，我们通过模板方法模式把这些行为搬移到单一的地方，这样就可以帮助子类摆脱重复不变行为的纠缠。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*第12篇\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

百度云社招java 二面  
  
精  
  
  
编辑于 2019-12-31 18:46:34  
  
 首先还是怼项目。问的比较细，有时候面试官会把条件改下，问假如现在需要你实现这个功能，你怎么来实现？ 中间也会穿插一些中间件和基础原理的问题。redis做分布式锁怎么做的？ 先拿setnx来争抢锁，抢到之后，再用expire给锁加一个过期时间防止由于程序crash等原因没有释放。最后线程操作结束之后，释放分布式锁。Redis能做分布式锁的原理？ Redis为单进程单线程模式，采用队列模式将并发访问变成串行访问，且多客户端对Redis的连接并不存在竞争关系。JUC下面主要用哪些东西？ CountDownLatch、Cyclicbarrier 。。。。。等等。其核心是AQS说一下AQS？AQS是JUC中很多同步组件的构建基础，简单来讲，它内部实现主要是状态变量state和一个FIFO队列来完成，同步队列的头结点是当前获取到同步状态的结点，获取同步状态state失败的线程，会被构造成一个结点（或共享式或独占式）加入到同步队列尾部（采用自旋CAS来保证此操作的线程安全），随后线程会阻塞；释放时唤醒头结点的后继结点，使其加入对同步状态的争夺中。  
AQS为我们定义好了顶层的处理实现逻辑，我们在使用AQS构建符合我们需求的同步组件时，只需重写tryAcquire，tryAcquireShared，tryRelease，tryReleaseShared几个方法，来决定同步状态的释放和获取即可，至于背后复杂的线程排队，线程阻塞/唤醒，如何保证线程安全，都由AQS为我们完成了，这也是非常典型的模板方法的应用。AQS定义好顶级逻辑的骨架，并提取出公用的线程入队列/出队列，阻塞/唤醒等一系列复杂逻辑的实现，将部分简单的可由使用者决定的操作逻辑延迟到子类中去实现。  
网上很多博客，不了解的可以去找找看。\*ReentrantLock非公平锁和公平锁的实现原理？ \*  
  
 final boolean nonfairTryAcquire(int acquires) {  
 final Thread current = Thread.currentThread();  
 int c = getState(); //获取同步状态  
 if (c == 0) {  
 //如果同步状态为0，表明没有线程获取锁，或者队列中的节点正在获取锁。  
 //此时非公平锁会直接去尝试获取锁。  
 if (compareAndSetState(0, acquires)) {  
 setExclusiveOwnerThread(current);  
 return true;  
 }  
 }  
 else if (current == getExclusiveOwnerThread()) {  
 int nextc = c + acquires;  
 if (nextc < 0) // overflow  
 throw new Error("Maximum lock count exceeded");  
 setState(nextc);  
 return true;  
 }  
 return false;  
 }  
  
  
 protected final boolean tryAcquire(int acquires) {  
 final Thread current = Thread.currentThread();  
 int c = getState();  
 if (c == 0) {  
 //公平锁的话。会先判断队列中是否有等到的线程  
 if (!hasQueuedPredecessors() &&  
 compareAndSetState(0, acquires)) {  
 setExclusiveOwnerThread(current);  
 return true;  
 }  
 }  
 else if (current == getExclusiveOwnerThread()) {  
 int nextc = c + acquires;  
 if (nextc < 0)  
 throw new Error("Maximum lock count exceeded");  
 setState(nextc);  
 return true;  
 }  
 return false;  
 }  
  
说下MVCC？数据库调优的思路？ 在读已提交的级别下，都会通过MVCC获取当前数据的最新快照，不加任何锁。存在的问题：在每次select之间有其他事务更新了我们读取的数据并提交了，那就出现了不可重复读。可重复读：MVCC版本的生成时机， 即：一次事务中只在第一次select时生成版本，后续的查询都是在这个版本上进行，从而实现了可重复读。 1.慢查询的开启并捕获 2.explain+慢sql分析 3.show profile查询sql在mysql服务器里面的执行细节和生命周期情况 4.sql数据库服务器的参数调优。  
kafka怎么保证消息的顺序性？zookeeper的选举原理？ 这个可以自己百度一下，网上很多。  
 Java远程调试的原理：这个真不知道。。。有知道的可以说下。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*第13篇\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

百度云java社招一面面经  
  
精  
  
  
编辑于 2019-12-23 20:03:29  
  
百度云java社招一面面经与个人总结  
 1、首先是项目部分，问的比较细。这部分没什么参考价值，忽略。  
 2、如果一个对象有多个方法加了synchronized，那么该对象有几把锁   
 对象锁是在一个类的对象上加的的锁，只有一把，不管有几个方法进行了同步。  
 这些同步方法都共有一把锁，只要一个线程获得了这个对象锁，其他的线程就不能访问该对象的任何一个同步方法。  
 3.NIO与AIO的区别以及各自的作用  
 Java NIO ：同步非阻塞，服务器实现模式为一个请求一个线程，即客户端发送的连接请求都会注册到多路复用器上，多路复用器轮询到连接有I/O请求时才启动一个线程进行处理。  
 Java AIO(NIO.2) ： 异步非阻塞，服务器实现模式为一个有效请求一个线程，客户端的I/O请求都是由OS先完成了再通知服务器应用去启动线程进行处理，  
 NIO方式适用于连接数目多且连接比较短（轻操作）的架构，比如聊天服务器。  
 AIO方式使用于连接数目多且连接比较长（重操作）的架构，比如相册服务器。  
 4.IOC的实现原理  
 Spring的IOC实现原理就是工厂模式加反射机制，通俗来讲就是根据给出的类名（字符串方式）来动态地生成对象，这种编程方式可以让对象在生成时才被决定到底是哪一种对象。  
 把IOC容器的工作模式看做是工厂模式的升华，可以把IOC容器看作是一个工厂，这个工厂里要生产的对象都在配置文件中给出定义，然后利用编程语言提供的反射机制，根据配置文件中给出的类名生成相应的对象。  
 Spring支持三种依赖注入方式，分别是属性（Setter方法）注入，构造注入和接口注入。  
 5、反射的原理  
 反射首先是能够获取到Java中的反射类的字节码，然后将字节码中的方法，变量，构造函数等映射成 相应的 Method、Filed、Constructor 等类  
 6、数据库调优思路的思路。  
 1）.慢查询的开启并捕获  
 2）.explain+慢sql分析  
 3）.show profile查询sql在mysql服务器里面的执行细节和生命周期情况  
 4）.sql数据库服务器的参数调优  
 7、开放性的问题，如何提高系统QPS。  
 这个问题我感觉大致可以从这几个方面  
 1）、单机版能承受并发的能力是有限的，我们可以进行系统拆分，分开部署在不同的机器上。  
 2）、用消息队列削峰。系统不至于因为瞬间的流量挂掉。并且可以配合使用限流与服务降级。  
 3）、用redis什么的做缓存。  
 4）、数据库分库分表，建立合适的索引。