|  |
| --- |
| 上午完成排序整理工作。整理好以后，就把书还掉。不再借这本书了。 |
| Tab键就是新增加一行。 |

今天开始关注tensorflow的github然后开始进入学习状态。

文章：

<http://www.cnblogs.com/jingmoxukong/p/4329079.html>

找到了TensorFlow游乐园。这个可以帮助我们在线学习tensorflow挺好玩的。要自己学习了。

热爱写代码。

面试—自我介绍—项目准备—基础知识—缓存—安全（注入攻击？）。

（认真负责，乐于协作。）

优势展现。

|  |
| --- |
| <http://blog.csdn.net/kjfcpua/article/details/8579422>对象池，池化的思想。  采用对象池化的本意，是要通过减少对象生成的次数，减少花在对象初始化上面的开销，从而提高整体的性能。然而池化处理本身也要付出代价，因此，并非任何情况下都适合采用对象池化。  所以在創建過程中需要花費大量時間的對象，才需要池化。  思路是：  线程池中有两个集合，一个集合A存放空闲中的对象，一个集合B存放使用中的对象，  线程从A中拿到对象使用，并放入B中，当线程使用完对象之后从B中取出放回A中，  有状态的对象在使用之前先恢复为初始化状态，  当线程池中所有的对象都是使用中时（即都属于集合B），线程等待，当有线程返还对象时，线程唤醒。 |
|  |
|  |

|  |
| --- |
| 1Nosql解决高并发读写（读还可以，写不行。）  2海量数据的高效存储和访问。Huge storage。  3.关系型数据库做不到横向扩展（Nosql也可以解决。）  Redis 是内存的（key value）  Mongoldb是存在硬盘上的。文档，数据结构不严格，性能不好，缺少统一的sql语法、  hbase（列存储）  **nosql 的特点**。数据之间没有关系的。易扩展，高可用，灵活数据类型，大数据量，高性能。 |
| Redis发展：  2008年开始才开始开发Redis。现在12%，知乎，微博都在用redis。  用c语言开发。  最主要的应用场景是  1缓存，2任务队列（秒杀抢购。）3网站访问统计 4数据过期处理 5 分布式集群架构中的session分离。 6应用排行榜 |
|  |
|  |

**技术人员如何建立自己的个人品牌 优雅的程序员列传**

[**http://www.jianshu.com/p/d29cba7934c9**](http://www.jianshu.com/p/d29cba7934c9)

**这个问题就好学。**

|  |
| --- |
| Java研发岗：本人东北大学本科妹子一名，能力一般，内推到百度一面，但是应该是挂了，特来写篇面经，纪念一下  一面内容：  MySQL索引底层实现，效率  MySQL数据是怎么存储的  MySQL崩溃了怎么办  NIO和IO的区别，NIO: 1024字节，传了512后，下次会接着传吗  内存泄漏的工具  interrupt()对线程的影响，应用场景  Http1.0,1.1,2.0的区别  TCP的TIME-WAIT讲一下，时间是多久  读写锁讲一下，说一下应用场景  找两个链表的公共节点  翻转how are you  为  you are how  tomcat崩溃了怎么办  nginx了解吗  死锁的条件和原因  thrift(项目中的内容)  状态码：302，304代表啥，受什么参数影响  spring IOC AOP讲一下  差不多就记得这些了  面试官建议：因为我完全没有接触过分布式，所以面试官建议我学习一下分布式  经验总结：多说自己了解方向的词，面试官可能就会跟着问了，第一次校招面试没经验，以后会进步的 |
| servlet生命周期  servlet的工作流程  jsp和servlet的区别  spring的功能 Ioc 容器 ioc是什么意思 怎么实现 反射怎么用  spring aop aop 事务管理 声明式 怎么编写  aop用什么实现 动态代理 手撸了使用代码  springmvc有什么了解 直接说没什么了解使用struts 又问两者优劣....  面向对象特征 4个 分别解释 重载 继承 有什么情况下不能继承 final jdk中哪些类是final String为什么要这么实现final string stringbuilder stringbuffer 之间的区别  类加载机制  set中的对象需要重写什么方法  hashCode和equals  序列化怎么用 不想序列化的属性  Transient  java中的gc机制  多线程 3种方式  并发同步的方法 两种方法有什么区别  lock可以有什么高级的控制  线程有几种状态  什么方法造成状态的转换 wait sleep分别导致线程进入哪种状态  可能是简历里有 两次面试都问了 socket 服务端 撸代码 socket使用nio还是bio nio和bio的区别 优势  什么叫非阻塞 后来反应过来为什么问 面试官的意思是服务端socket接收数据线程池的实现可以用nio代替，一个buffer对应多个channel  撸代码 给个文件 打印出全是数字的行 一面近一个小时，二面30+分钟 ，  hr近一个小时，感觉hr面跪了，常给你一个问题描述整个过程。比如学习能力，你有学过什么东西，然后做了什么努力，达到什么样的成果，有什么样影响.还问了投了哪些公司…在项目中组员间出现过的矛盾，如何解决..... 只记得这么多了，第一次记下来，希望做好事有好运吧～ ps：我不小心带了支笔，后面的兄弟，我只能做到这里了^ - ^ |
| **一面 40分钟左右**   * 自我介绍 * 项目相关的组织 * Tomcat组织结构 * Servlet是否是线程安全的？   servlet是否线程安全是由它的实现来决定的，如果它内部的属性或方法会被多个线程改变，它就是线程不安全的，反之，就是线程安全的。首先什么是线程安全？  引用概念：如果你的代码所在的进程中有多个线程在同时运行，而这些线程可能会同时运行这段代码。如果每次运行结果和单线程运行的结果是一样的，而且其他的变量的值也和预期的是一样的，就是线程安全的。 我们都知道servlet是多线程的，同时一个servlet实现类只会有一个实例对象，也就是它是Singleton的,所以多个线程是可能会访问同一个servlet实例对象的。 每个线程都会为数据实例对象开辟单独的引用，那么servlet会是线程安全的吗？ 要判断是否是线程安全，我们需要知道线程安全问题是由什么引起的。 搜索得到答案：线程安全问题都是由全局变量及静态变量引起的。   * HTTP请求中get、POST的区别？HEAD请求的作用？ 【**1**.GET请求的数据会附在URL之后，POST是没有大小限制的，HTTP协议规范也没有进行大小限制 * Spring MVC 的工作流程？ * Spring MVC 中的常用注解？RequestMapping的常用配置属性，分别有什么作用？ * Spring的主要特性？ * Java如何使用多线程？线程池相关？ * Java的解决多线程冲突的方法，锁的类型？ * ReentrantLock、synchronized的区别和各自特点? * Java如何序列化一个对象，如何自定义序列化？如何在序列化的过程中排除指定的属性？ * Java如何读写文件（多种方案）？如何进行随机读写？字节流和字符流的区别？ * Mysql的索引的类型？有哪几种索引？怎么优化查询？ * Mysql的隔离级别？Mysql是如何消除幻读的？ * 你有什么要问我的？   **二面**   * Java的基本类型有哪些？ * 介绍一下Spring Boot？ * Spring Bean的声明周期？ * Spring MVC的工作流程？ * 最短路径的算法？能不能写一下？ * Mysql的隔离级别，分别解决什么问题？ * Mybatis中的SessionFactory的作用？ * volatile 保留字的作用？ * ThreadLocal的作用？ * git的操作指令？   **HR**   * 自我介绍 * 为什么来杭州？ * 你在实习公司做的什么工作？ |
| 网易一面：  1.conCurrentHashMap中的取Size操作的实现原理？如果是你来做，你怎么怎么实现？  2.conCurrentHashMap相对于HashMap在并发方面做了哪些优化？  3.Hashmap同时get put会发生什么？  4.Mysql中有几种搜索引擎，讲一下他们的区别？  5.SpringMvc的流程  6.在url进入dispatchServlet之前相对url做一些改变，你有什么办法？  7.jvm垃圾回收的范围，内存模型，垃圾回收机制，怎么判断对象死亡？计数法和可达性分析法的区别？  8.分布式redis怎么实现？  9.redis的数据类型？  10.分布式环境下redis怎么保证线程安全？  11.redis的底层实现看过么？  12.项目一套~  13.还有好多忘了~  网易二面：  1.Long和long怎么比较？从jvm层面分析  为什么是Long拆箱而不是long装箱？  long Long问题是拆箱不是装箱可以这样回答吗：若用==比较两个Long的值是否相等那就没有办法比较两个Long 对象是否相同了，即使装箱之后再比较也需要比较Long 对象中的value值，多两步操作没有意义。好难答...  2.java中for中remove对象会发生什么？  在for（）中，不容许调用remove（）函数。  http://blog.csdn.net/truong/article/details/39178807  3.解释一下fast-fail机制，怎么实现的，一定会抛异常么？  4.讲一下反射，反射中私有变量怎么处理？newInstance如果抛异常的话，是什么原因？  5.了解序列化么？不同序列化方案的有缺?  6.索引什么时候建立？好处和坏处？  7.怎么判断索引的效率？  8.索引失效的情况有哪些？范围查询一定会失效么？  9.生产者消费者中消费者怎么来监听事件？有哪些方法？  看用什么了，最原始的办法就是wait/notify，细粒度的控制就是ReentrantLock里的condition，await/signal 轮询  10.注解的范围？  11.线程互斥的方法？  12.sychnorized知道么？synchronized(静态对象)和synchronized(实例对象)的区别？  13.sychnorized和lock接口的区别？  14.threadLocal的实现？  15.看过那些开源框架的代码，讲讲。  16.项目~  17.还有一些问懵逼的问题已经忘了是什么了~  三面HR~  总体感觉还是问的比较深的，基础还不够扎实。 |
| 引导新人如何在互联网上获取优质的学习资源  <https://github.com/ziwenxie/netease-dl>  一个看起来不错的小项目。  <https://www.ziwenxie.site/>  这是他的个人博客，个人品牌经营的不错。  比较会用github。  在github的搜索上 awesome+关键字，就可以搜索到很多比较好的资源。  <https://github.com/ziwenxie/snippet/blob/master/chrome/vim-surfingkeys.md>  chrom浏览器一个插件的使用。  <http://www.jianshu.com/p/a22bdc2d5f23>  <https://github.com/521xueweihan/HelloGitHub#%E5%BE%80%E6%9C%9F%E5%9B%9E%E9%A1%BE>  <https://hellogithub.com/>  http://linuxtools-rst.readthedocs.io/zh\_CN/latest/base/index.html一些入门项目。  **分享 GitHub 上有趣、入门级的开源项目。**  这是一个面向**编程新手**、**热爱编程**、**对开源社区感兴趣** 人群的月刊，月刊的内容包括：**各种编程语言的项目**、**让生活变得更美好的工具**、**书籍、学习笔记、教程等**，这些开源项目大多都是非常容易上手，而且非常 Cool。主要是希望大家能动手用起来，加入到**开源社区**中。  [**http://www.jianshu.com/c/e2a2a0073e2d**](http://www.jianshu.com/c/e2a2a0073e2d)  **github上有趣的资源。** |
|  |
|  |
|  |
| **hashmap 并发put引发resize导致死循环的例子我一直无法理解。。或者说复现。。。**  这个是jdk1.7的历史遗留问题，主要是jdk1.7是头插数据，而jdk1.8是尾插式数据，一个线程在resize（）集合时，头插式会导致两个节点的循环。这就是所谓的infinite loop问题  线程A调用HashMap的put方法触发resize的时候，resize还没执行完，被挂起，线程B同时调用put方法也引起resize，此时，等到b线程resize完毕之后，线程A被唤醒，继续执行resize，极易引起死循环导致oom。改用红黑树，节点插入不会倒序，不易引起死循环  <http://github.thinkingbar.com/hashmap-infinite-loop/> 这个是我认为讲解 infinite loop最透彻的一篇 可以看看 |
|  |
|  |
|  |
| **三次握手总结：**  <http://www.cnblogs.com/Jessy/p/3535612.html> 注明了各种缩写代表什么意思。  [**http://blog.csdn.net/whuslei/article/details/6667471/**](http://blog.csdn.net/whuslei/article/details/6667471/) **这一篇写的非常的通俗易懂。便于复习。**  位码即tcp标志位,有6种标示:  SYN(synchronous建立联机)  ACK(acknowledgement 确认)  PSH(push传送)  FIN(finish结束)  RST(reset重置)  URG(urgent紧急)  Sequence number(顺序号码)  Acknowledge number(确认号码)  **客户端TCP状态迁移：** CLOSED->SYN\_SENT->ESTABLISHED->FIN\_WAIT\_1->FIN\_WAIT\_2->TIME\_WAIT->CLOSED 服务器TCP状态迁移： CLOSED->LISTEN->SYN收到->ESTABLISHED->CLOSE\_WAIT->LAST\_ACK->CLOSED |
| 拥塞控制就是防止过多的数据注入网络中，这样可以使网络中的路由器或链路不致过载  [**http://blog.csdn.net/sicofield/article/details/9708383**](http://blog.csdn.net/sicofield/article/details/9708383)  **慢开始**与**拥塞避免**  如下图：  http://img.blog.csdn.net/20130801220615250?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvc2ljb2ZpZWxk/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast |
| **Volatile关键字**  [**http://www.importnew.com/18126.html**](http://www.importnew.com/18126.html)  **cpu并不是把计算的值直接存入到内存，因为这个过程太慢，所以是先存在cpu的缓存之中的，而其他的线程访问一个还没写入到内存的共享变量的时候，这会让读不一致的问题。加入volatile就可以用保证，cpu处理这个变量的时候就是把这个变量的值从高速缓存写入到物理内存中。**  **一.内存模型的相关概念**  大家都知道，计算机在执行程序时，每条指令都是在CPU中执行的，而执行指令过程中，势必涉及到数据的读取和写入。由于程序运行过程中的临时数据是存放在主存（物理内存）当中的，这时就存在一个问题，由于CPU执行速度很快，而从内存读取数据和向内存写入数据的过程跟CPU执行指令的速度比起来要慢的多，因此如果任何时候对数据的操作都要通过和内存的交互来进行，会大大降低指令执行的速度。因此在CPU里面就有了高速缓存。  也就是，当程序在运行过程中，会将运算需要的数据从主存复制一份到CPU的高速缓存当中，那么CPU进行计算时就可以直接从它的高速缓存读取数据和向其中写入数据，当运算结束之后，再将高速缓存中的数据刷新到主存当中。举个简单的例子，比如下面的这段代码：   |  |  | | --- | --- | | 1 | i = i + 1; |   当线程执行这个语句时，会先从主存当中读取i的值，然后复制一份到高速缓存当中，然后CPU执行指令对i进行加1操作，然后将数据写入高速缓存，最后将高速缓存中i最新的值刷新到主存当中。  **在并发编程中，我们通常会遇到以下三个问题：原子性问题（不可分，不可打断，回滚），可见性问题（立即课件，用volatile或者syncronized或者lock可以做到。），有序性（**语句1和语句2谁先执行对最终的程序结果并没有影响，那么就有可能在执行过程中，语句2先执行而语句1后执行。这就是【指令从排序】**）问题。**  自增操作不是原子性操作，而且volatile也无法保证对变量的任何操作都是原子性的。· |
| **ImportNew 中会经常用到一些github源代码。还有很多好的开源项目推荐。**  **Stactoverflow 问答很很好。英文不是问题。多用。** |