

一： 基础知识

计算机基础

1. 操作系统

1.1 进程和线程的区别？

面试官问这个问题，并不是只想问这一道题，而是想根据这道题看看面试者，对进程和线程的了解情况。并引出一些别的问题。

如果说内存结构不同，进程和进程是不能共享内存的，线程之间是可以共享内存的。那么面试官有可能会问一些内存寻址方面的问题。还有线程共享内存的实现方法等等。

如果说线程间通信是很方便的。但是进程之间通信要复杂一些了。那么面试官可能会问一些通信方法的优缺点。

一个电脑上可能存在很多个进程，而这个进程里面又包含了很多个线程。系统会分给每一个进程一段内存，而所有的线程就是共享这一段内存的。系统分给第一个进程其实是个内存的寻址空间。系统为什么会有32位和64位的呢？因为32位系统的寻址空间是 2^{32} 次方，就是四个G。而进程就是占用的这其中的一段寻址空间。

而我们自己的进程可以去读取别人进程中的内存内容吗？每个进程的寻址已经分配好了不能调用。

2. 网络

3. 数据库

编程语言基础

1. 数据类型 2. 装箱与拆箱

二： 编程能力

编码技巧

- 1. 递归控制
- 2. 循环控制
- 3. 边界控制
- 4. 数据结构
- 5. 树的遍历

面向对象思想

1.类与对象 2.接口与实现 3.继承与封装 4.不可变类型 5.泛型

设计模式

1. Singleton 2.变继承关系为组合关系 3.对象如何创建

三： 高级知识点

1.并行计算 2.多线程 3.资源管理

四： 微服务相关知识点

五： webflux 相关知识点

六： 项目介绍

电商项目介绍

商品录入模块

这个模块主要是让后台管理人员，对商品的CRUD操作。为了让用户购买方便，我们对商品进行了分类。我们的分类总共有三级，如休闲食品下面有糖果类，饼干，肉食类。而糖果类下有硬糖，软糖，口香糖。肉食类下面有牛肉，羊肉。而当我们选则一类的时候，就可以选择商品规格。根据用户的选择，就会筛选出用户想要的商品。 为了实现这个功能，我们使用了模板，SPU,SKU的概念来实现。

SPU就是一个最小的商品，比如苹果手机 +型号 就是一个SPU。如一个5S。 三星S9。而这个是 无法订价的，因此就有了一个SKU ， S9+黑色+ 32G内存 就是一个SKU 可以订价了。

我们将一个三级分类，设计成一个模板。具体的规格信息是以json的形式保存到模板表中。这样方便我们在页面中展示，而我们还创建了字典表来保存规格信息。

而一个sku是我们的最小销售单元。用户可以选择不同的规格来组成一个最小销售单元。这里的表名是tb_item。里面主要保存一些类目和规格组成的title字段，最小的销售单元就存在这里面。卖点字段，价格，库存数量，商品图片，商品所属类目，商品状态，这个sku是否是一个默认的，卖家ID，商品ID等等。

在实现这个功能的时候，我们使用了一个小的技术点是深度克隆。

我们是多商家入驻的平台，在对商品做一些操作的时候，商家只能看到自己的信息。

我们商品也是分类的，比如一些审核通过，未审核，审核未通过之类的。为了提高用户体验，像类似于这些的分类项，我们都是将数据保存到redis缓存中。查询的效率就会比在mysql数据库中高出很多。

关于商品的上下架，我们是只有运营商通过审核之后，在由商家去上下架。一般运营商做审核主要是审核一些图片是符合格，词语有没有反动的言论。

广告管理模块

我们的广告分为好几类，有首页的轮播图，今日推荐，猜你喜欢，楼层广告等等。实现这个功能我们有广告分类表主要是存储广告有哪些分类，广告表里面主要有广告分类ID，广告标题，广告链接是跳到哪一个商品上，图片地址，状态，排序。（注：关于表的这一块由于没有什么亮点可以不说，如果对方问的话，在进行回答）。

运营人员在设置添加广告的时候，可以选择在哪一个广告类型的第几个位置的广告。由于页面大小是有限的，只有设置有限的几个广告，多余的就会被撤除。

我们在做猜你喜欢的时候，做了一个推荐功能。就是根据用户注册填写的信息来推荐一些广告商品。我们在设置广告的时候会有一些选项，就是一些广告受众的特点。如性别，年龄区间，爱好，职业等等信息。当用户登录成功之后，就会拿用户的这些信息作为关键字去找相应的广告。如果找不到相应就按照默认的显示。

后来发现当用户刷新页面的时候还是不停的请求数据库服务器，不止是会给数据库造成压力而且也对用户体验不是很好，容易出现短暂的卡顿。当时为了解决这个问题，我们就想着把这些数据也添加到缓存中，就向我们将字典表数据添加到缓存中一样。这样效率就很会高了。我们先将默认的广告缓存到数据库中，如果用户没有登录就直接显示这些数据就行了。如果用户登录了就按照我们制定的规则，我们就将数据添加到缓存中。这些数据也不会一直保存到数据库中的，我们将数据保存时间设置为48小时，这样可以减少占用缓存。我们的保存数据的格式为使用hash做的，大key是用户名加下划线加content这个广告字来做的就，比如用户账号为 tianshuai 那么保存的key就是 tianshuai_content。小key就是保存的广告类型，如猜你喜欢。

当时我们本来是想将数据都保存到redis中，在通过redis模糊查询来查询出关键字来，获取里面的内容，可是后来发现redis模糊查询key虽然可以，但是会对服务器的性能造成一定

的影响，因此还是使用了刚才的方法来做的。

商品展示模块

我们在这一个模块主要展示商品的详情，好让用户更了解商品。我们在上线第一版的时候，是使用的动态网页，就是用户点击一个商品的时候，去数据库中查询数据的信息。在传到前台页面中来展示信息，后来我们在第二版中对这个进行了改进。因为第一版是有缺点的，就是说数据对数据库访问的次数增多了。那样就服务器压力太大，就会造成访问速度变慢。我们就使用了页面静态化，freemarker 这个就是一个模板引擎。我们使用这个模板引擎和生成静态的页面。放到数据库中，用户访问就可以直接访问一个静态页面了，不会在对数据库进行操作。

我们在生成静态页面的时候，会将sku显示到页面中。因为商品在创建sku的时候会选择一个默认的。这里我们在生成静态页面的时候就会展示默认的信息。如果sku的状态是不可用的状态，在页面中就不会显示添加购物车和立即购买的按钮。

我们这些操作都是在前台使用的js进行操作的。虽然我们是使用的freemarker生成的静态页面，但并不是说这个页面是完全静止的，我们只是想让一些固定的内容提前生成，不会在用户查看页面的时候，去数据库中去访问，即影响了用户体验，也会给服务器造成压力。当页面中需要和数据库进行交互的时候，我们是可以使用js来异步加载一些数据的。

我们是在商品审核通过的时候生成页面的，而运营端的服务是集群化的，我们不能让页面分布在不同的服务器上，因此我们使用了mq来实现这个功能，当审核通过的时候调用mq，发送一个商品的ID在MQ服务器上，发送的消息是广播模式。消费端就会在服务器上创建相应的HTML页面。我们使用了NGINX来做为静态页面的web容器。

登录模块

采用的是单点登录，因为我们是使用的是分布式系统，必须使用单点登录。我们是使用第三方框架来完成这个功能的。casserver，就是一个单独的服务，需要用一个单独的tomcat来启动他。默认的情况下，只要一关闭浏览器，就会退出。他还默认使用https来作为服务协议，因为我们对安全要求没有那么高。就只使用了http协议。

: 面试流程和技巧分享