# 云计算 1607C 刘笑阳

# JAVA工程师

1. 项目介绍(按照实训二做的项目写)

|  |
| --- |
| 本项目主要用于广告投放，服务于广告主和媒体方，广告主在平上进行投放广告，我们平台进行数据处理，然后提供给媒体方，进行广告展示，并且对媒体方进行相应的计费。项目整体框架是由Spring Cloud+Spring Boot+Spring Data JPA的Spring全家桶微服务架构，由Maven管理，接口遵循RESTful规范开发，数据检索使用Elasticsearch作为搜索引擎，在一些高并发以及流量大的场景，使用Apache Kafka作为消息队列减轻服务器压力。热点数据使用Redis做缓存 |

1. 功能模块介绍(按照自己得实际项目写)

|  |
| --- |
| 我个人主要负责登陆系统和检索系统的核心业务开发，首先说一下登陆的流程：广告主登陆的方式是手机号加动态验证码的，这里的话，为了短信接口的可用性，我们接入了两个短信接口，分别是阿里大于和容联。前台广告主点击发送短信之后，后台拿到手机号，生成一个随即的6位验证码，然后调用短信接口发送短信，如果发送成功，则将验证码存到redis，并设置过期时间，过期时间为两分钟，然后广告主会根据手机接收到的短信输入验证码，传到后台之后我们从redis取出并进行判断，如果没有过期并且一致，就说明登陆成功，然后会查看数据库里有没有这个手机号的用户，如果有，则代表是登陆，如果没有，则代表是注册，就会添加广告主信息到数据库用户表，我们将广告主的IP+手机号+用户名+时间添加到登陆日志表里，然后根据广告主的手机号生成token，并且设置有效期，一般都是7200S,然后返回给前台，前台每次向后台发送请求都会经过zuul网关，而且要带着token请求，然后zuul网关进行拦截并且验证token，如果token有效并且正确，则验证通过，开始转发请求，进行处理业务，如果token为null或者是伪造的，则返回相应的错误信息，并且不会进行转发请求，这是登陆所涉及的逻辑，然后我负责的另一个模块就是检索系统，主要任务就是向媒体方提供数据，也就是向曝光系统提供数据，并且作为消费者监听Kafka，当有提供者，也就是投放系统将广告的数据发送到Kafka中时，检索系统会收到消息并且拿到数据，然后进行数据处理，存到Elasticsearch 搜索引擎和数据库里，当媒体方要拿数据的时候，会经过曝光系统调用检索系统，用的是SpringCloud的Feign组件，然后检索系统会从Elasticsearch 里面查询数据，返回给曝光系统，曝光系统进行数据处理，将合理的广告信息返回媒体方，进行展示广告，每个广告信息都会有多个回调函数，比如展示之后的回调函数，和点击之后的回调函数，会调用到计费系统进行相应的计费策略，  计费策略有CPM、CPC、CPA、CPD等策略，  首先Kafka是一个高吞吐，高流量的消息中间件，适用于数据量非常大的情况下，Kafka会自带一个zookeeper ，用于kafka服务器的注册，包括生产者和消费者，元数据信息都会被注册到zookeeper里，用与监控和提供服务器IP地址。生产者为生产消息的一方，发送消息共有三种策略，①没有返回值，消息发送出去即可，这种策略不安全，一般不会用，②会当场得到返回值，从发送消息到接收返回值期间，生产者会一直等待，这会造成线程阻塞的情况，而且会浪费时间，但是会保证数据的安全，能保证消息肯定能发送到kafka,并且能被处理。③消息发送出去之后，会立即得到响应，可以执行之后的逻辑，但不会立即得到返回值，等到消费者处理完消息之后，会执行回调函数，并且有返回值，返回值和策略②的相同。但是不会造成阻塞，并且能保证消息的安全性。性能非常高，一般都会使用这个策略。此为生产者发送消息的三个策略。发送的消息要经过序列化处理，只需要设置序列化策略就行，取出的时候也是设置反序列化策略，不需要手动反序列化。  消费者处理消息是以消费者组的形式存在的，一个分区不可以被一个消费者组中的多个消费者同时处理，只要在一个消费者组里面，每个消费者处理的分区都是不一样的，但是多个消费者组可以同时处理一个的分区。订阅的体现只是在消费者组与kafka之间，不是消费者与kafka之间。  消费者接收消息有五个策略 |

1. 开发中遇到的技术难点以及解决办法

|  |
| --- |
| 难点1：在检索系统前期，让使用map存储索引  解决方法1：首先，索引分为两个，分别是全量索引和增量索引，全量索引就是每次项目启动的时候都会将日志文件里的数据全部查询出来，然后将索引存放到Map里面，增量索引就是监听日志文件，每当数据库有变化，都会出发一个event事件，然后我们根据事件的类型来操作索引map，就实现了增量索引，代码中，全量索是在indefileloder中的init方法上加了个@postconstruct标签，标签的意思就是程序启动的时候执行此方法，先用files的newBufferedReader方法，传入文件名，然后读取文件中的每一行数据，然后收集到list集合里，就是String类型的List，每一行都对应数据库里的一条数据，然后遍历将String转换为相应的实体对象，然后以id为键，对象为value，存到map集合中，map选用线程安全的ConcurrentHashMap。用的是正排索引，但是在加载关键字和地域限制的时候用的是倒排索引，在倒排索引的创建时，我们会判断一下我们所传进来的关键字的单元ID是否存在，如果存在就收集到set集合里面，这就是倒排索引，以上是用map做全两加载的流程。增量索引就是，会有一个BinlogRunner类，这个类实现了CommandLineRunner类，当继承这个类之后，就会在程序启动之后，执行这个类重写的run方法，在这个run方法里面，我们调用了一个方法，然后方法里创建了一个线程，并且是一个长链接，这个方法里面监听了mysql的binlog日志，每当数据库数据被改变，程序都会监听到，然后判断是什么操作，然后对map进行相对应的操作。这就是增量索引的流程。  难点2：广告项目内部微服务相互调用解决方案  解决方法2：使用feign组件进行调用，我们会将微服务的元数据注册到eureka中，  包括application.name 如果application.name一样，则表示是用一个服务的集群  通过feign进行调用，feign默认负载均衡方式是使用ribbon  。。。。。。。。。。。 |

1. 项目技术亮点分析

|  |
| --- |
| 1. 项目框架：项目采用得spring boot+springcloud微服务框架，好处是SpringCloud 提供了好多轻量级的组件，供我们使用，并且这些组件组成了一个完整的微服务架构，并且Spring Boot可以使我们简化代码，简化配置，并且内嵌tomcat，所以是非常便捷的 2. kafka**技术:**项目的消息中间件是使用Kafka，首先kafka是一个高吞吐高流量的消息中间件，比较适合消息量非常大的场景，我们项目中使用到kafka的场景就是在广告主投放广告的时候，会将消息发送到消息队列中，然后又相应的消费者组进行消费，比如mysql组，es组，每个组都会有多个消费者，消费者是以组的形式存在的，我做的检索系统就负责es的操作，所以我的检索系统就部署了三台，是属于同一个消费者组的，同时监听kafka，有消息之后就会拿到消息进行操作es。当曝光系统要取数据的时候，他会调用检索系统的接口，存取数据\ 3. Elasticsearch：Elasticsearch主要是在我的检索系统中用到，因为我的检索系统主要是对存储和提供数据的作用，当检索系统启动的时候，回去监听到kafka消息队列，每当投放系统往kafka中我所监听的主题发送消息的时候，我都会去将消息取出，并且将数据存到Elasticsearch中，然后当曝光系统要取数据的时候，我就相应的数据取出来返回，让曝光系统处理，Elasticsearch主要就是数据储存和提供的功能，因为Elasticsearch是一个性能高的搜索引擎，他可以自动帮我们创建索引，我们只需要将数据存进去即可，所以非常方便。 |

1. 项目答辩中亮点简历审核评定标准

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **考核内容** | **简历样式是否规范** | **项目内容是否完整** | **项目中是否有亮点** | **技术难点解决办法是否高效** | **存在问题** |
| **项目经理 （评定）** |  |  |  |  |  |
| **就业处干事（评定）** |  |  |  |  |  |
| 简历是否合格 | | | | 合格 | 不合格 |

1. 项目答辩成绩评定标准

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **考核内容** | **亮点简历与项目描述是否匹配** | **项目流程是否清晰** | **学生讲解是否流畅** | **项目介绍是否到位** | **技术亮点考核是否合格** | **存在问题** |
| **项目经理 （评定）** |  |  |  |  |  |  |
| **就业处干事（评定）** |  |  |  |  |  |  |
| 项目答辩是否合格 | | | | 合格 | 不合格 |  |