# 云计算 1607C 朱连杰

# JAVA工程师

1. 项目介绍(按照实训二做的项目写)

|  |
| --- |
| **荔枝便民平台** 是一个集交友、问答、招聘等多功能的一个平台，用户可以发布一些生活、学习、工作中的一些问题，或者对该问题进行解答，根据自己喜爱的条件对平台用户进行搜索，对于感兴趣的用户进行加关注。用户还可以对一些事情在此平台上进行吐槽，点赞等功能。该项目整体采用了整个项目是采用了**Spring Boot + Spring Cloud**的框架。后端使用 **Spring Data JPA** 持久层框架与数据库进行交互。使用**Eureka** 注册中心对项目各个微服务进行管理，使用网关**Zuul**对发送到服务器的请求进行一些预处理操作。采用关系型数据库**MySQL**对一些用户的重要信息进行保存，用到的非关系型数据 库有**Redis**、**MongoDB**。**Redis**主要用在用户根据Id查询时将返回的数据存到缓存中，或者是记录当前用户的状态，下次查询直接从缓存中返回，防止数据库压力过大。**MongoDB**主要用在储存回复、评论等具有层级关系的情况中。搜索系统通过使用 **ElasticSearch** 对问答信息、招聘信息、交友信息、租房信息等信息进行搜索并展示。使用**postman**进行接口测试。 |

1. 功能模块介绍(按照自己得实际项目写)

|  |
| --- |
| **1文章微服务：**用户根据遇到一些问题，进行发布，由其他用户进行解答，互动的一个模块。  **2基础微服务：**这个微服务主要功能是对用户地址，以及用户所喜爱的文章标签的一个数据字典。  **3交友微服务：**实现添加好友，删除好友的功能。  **4活动微服务：**实现发布活动，根据城市不同显示该城市的活动，对活动的一些基本CRUD的实现。  **5搜索微服务：**用ElasticSearch 进行对吐槽和文章的搜索。  **6用户微服务：**实现了新用户注册，用户登录和管理员登录的模块。  **7注册中心微服务：**使用**Eureka** 注册中心对项目各个微服务进行管理。  **8问答微服务：**根据用户感兴趣的标签不同，显示的问题也不一样，比如我们查询关于java的问题，这个问题下又分为三类回答：最新回答、热门回答以及等待回答。回答表中又有这三种回答类型的字段最新回答是根据时间，热门回答根据回复数，等待回答根据回复数等于0来判断。标签表与问题表是多对多的关系。所以他们有一个中间表，根据问题id与标签id两种id联合主键。问题与回答关系为一对多，点击问题（浏览量+1），根据问题id去查询答案，进行列表展示。但用户进行回答问题时，把问题表中回复数+1，最新回复数与最新回复时间更新，待用户解决问题后，点击解决问题按钮，改变‘是否解决’字段，这个字段默认肯定是未解决的。  **9吐槽微服务：**用户可以开启吐槽模式，开始疯狂吐槽。吐槽这个模块相对来说并不是很重要的，即使丢几条吐槽数据也不没有关系，所以我们采用**MongoDB** 非关系型数据库，非关系型数据库的一个特点就是，吞吐量大，可以满足我们这个需求。我们在设置**MongoDB** 的主键的时候是\_id，他不会像**MySQL**一样任何字段都可以设置主键，他的主键必须有下划线。我们在表中还设置了一个父级id这样做是为了让用户可以回复其他用户吐槽的内容，提高用户体验度，也因为这个特性，我们表设计的结构是树形结构。吐槽还有一个点赞的功能，首先，用户必须为登录状态才能进行点赞，在数据初始化的时候我们点赞数就初始化为0，一个用户只能点赞一次，如果我们使用**MySQL**数据库实现点赞效果，我们得先去数据库查点赞数量，然后进行点赞数修改，每次要与数据库进行两次交互。而使用了**MongoDB**数据库后，我们利用**mongoTemplate**对象把**MongoDB**原生命令转为java代码这样我们每次只需要对数据库进行一次交互，降低服务器的压力。用户点赞的时候把用户的id与吐槽的id组成键值对存入**Redis**，当用户进行点击点赞按钮时，先去**Redis**中进行判断用户是否点赞，如果没有，就进行点赞操作，如果已经点赞，则进行取消点赞操作，取消点赞操作成功后把**Redis**中该用户的id进行删除。  **10网关微服务：**网关作为微服务的入口，其并发压力都在网关，当网关的并发能力无法支撑用户量的时候我们就需要部署多个网关，然后在网关的前面加一个负载均衡服务器**nginx**。网关首先判断用户的**Token**，看用户是不是正常进行访问，当用户不是正常访问时进行拦截，如果用户是正常访问，那就进行转发。  **11短信微服务：**调用第三方短信接口（支付宝，百度），在用户注册时候发送验证码给用户的手机，把第三方生成的验证码与用户的id组合为键值对，保存到**Redis**缓存中，并给这个键值对设定过期时间，当用户输入验证码进行提交操作后，根据用户id进行查询，把**Redis**中保存的验证码与用户的验证码进行比对，如果一样，进行添加用户的操作，同时保存用户手机号，以及把手机号的状态码加入到用户的状态码中，意义为用户已经进行了手机号绑定，用户可以使用手机号或者用户名进行与密码进行登录，也可以使用手机号与验证码进行登录，当然执行地逻辑还是一样的。 |

1. 开发中遇到的技术难点以及解决办法

|  |
| --- |
| 难点1：**荔枝便民平台**项目用户注册模块短信验证码有90s时效？  解决方法1：将第三方发送给用户的验证码，按照用户id与验证码组合成一个键值对存入Redis中，设置改数据过期时间。  难点2：**荔枝便民平台**项目如何保证接口的安全性？  解决方法2：用户使用用户名密码登录后服务器给客户端返回一个Token（通常是UUID），并将Token-UserId以键值对的形式存放在缓存服务器中。服务端接收到请求后进行Token验证，如果Token不存在，说明请求无效。Token是客户端访问服务端的凭证。  难点3：**荔枝便民平台**项目如何处理高并发？  解决方法3：（1） ****图片服务器分离：****对于**Web**服务器来说，图片是最消耗资源的，于是我们有必要将图片与页面进行分离，这样的架构可以降低提供页面访问请求的服务器系统压力，并且可以保证系统不会因为图片问题而崩溃   1. ****缓存：****把一些热数据，不经常改变的数据，放到**Redis** 数据库中，用为**Redis**是储存在内存中的，读写速度是比较快的，我们这些热数据放到**Redis**中 会减少频繁去**MySQL**数据库查询的压力。   难点4：**荔枝便民平台**项目如何处理雪崩？  解决方法4：由于网络等原因，服务并不能保证 100% 可用，如果单个服务出现问题，调用这个服务就会出现线程阻塞，此时若有大量的请求涌入，Servlet 容器的线程资源会被消耗殆尽，导致服务瘫痪。由于服务与服务之间存在依赖，故障会在调用链路上传播，导致整个微服务系统崩溃，这就是服务故障的“雪崩”效应。为了解决这个问题，Spring Cloud 提供了对 Hystrix 断路器的集成，当服务调用失败的频次达到一定阈值，断路器将被开启，降级的策略可以开发者制定，一般是返回一个固定值。这样就能够避免连锁故障。此外 Spring Cloud 还提供 Hystrix Dashboard 和 Hystrix Turbine，帮助我们进行监控和聚合监控。  难点5：**荔枝便民平台**项目吐槽模块层级回复？  解决方法5：我们在表中设定了一个为父id的一个字段，因为我们使用的MongoDB数据库所以我们把表设计为树状结构的表，我们使用其中父链接结构，通过MongDB的命令进行快速添加，查询，删除完美解决我们的问题。  难点6：**荔枝便民平台**项目中微服务网关宕机了怎么办？  解决方法6：网关作为微服务的入口，其并发压力都在网关，当网关的并发能力无法支撑用户量的时候我们就需要部署多个网关，然后在网关的前面加一个负载均衡服务器，**nginx：**第一，如果可以把加权轮询算法分为先深搜索和先广搜索，那么nginx采用的是先深搜索算法，即将首先将请求都分给高权重的机器，直到该机器的权值降到了比其他机器低，才开始将请求分给下一个高权重的机器。 第二，当所有后端机器都down掉时，nginx会立即将所有机器的标志位清成初始状态，以避免造成所有的机器都处在timeout的状态，从而导致整个前端被夯住。  难点7：**荔枝便民平台**项目中**Redis**宕机了怎么办？  解决方法7：我们采用了**Redis**集群：两台为一组，一台主机一台备机 ，平时主机工作，备机不工作 ，备机每隔几秒就会给主机发送一个ping ,主机在正常工作的时候会发送一个ping 给备机 ，如果不返回ping ，可能会连续ping三次，都不返回那么备机就会上，备机里有主机的全部内容，这个叫做心跳检测机制（高可用的作用），主从热备这种方式来解决**Redis**宕机。 |

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......  1：注册中心，spring boot+spring cloud使用Eureka作为服务的注册中心，早dubbo+zookeeper的微服务中也常使用zookeeper作为服务的注册中心  2：服务治理，和服务的调度  3：客户端负载均衡（包括常用的三种负载均衡策略）  4：服务远程调度的容错保护，即电路设计中所说到的保险丝（熔断机制）  5：spring cloud的声明式远程调度  6：服务路由和服务的转发     1. **redis技术**：项目中点赞模块使用到**redis**了，具体是这样使用的:在用户注册时候发送验证码给用户的手机，把第三方生成的验证码与用户的id组合为键值对，保存到**Redis**缓存中，并给这个键值对设定过期时间，当用户输入验证码进行提交操作后，根据用户id进行查询，把**Redis**中保存的验证码与用户的验证码进行比对，如果一样，进行添加用户的操作，同时保存用户手机号，以及把手机号的状态码加入到用户的状态码中，意义为用户已经进行了手机号绑定，用户可以使用手机号或者用户名进行与密码进行登录，也可以使用手机号与验证码进行登录，当然执行地逻辑还是一样的。 2. **MongoDB**数据库**:**我们使用它主要因为以下三点   ①弱一致性（最终一致），更能保证用户的访问速度：  举例来说，在传统的关系型数据库中，一个COUNT类型的操作会锁定数据集，这样可以保证得到“当前”情况下的较精确值。这在某些情况下，例 如通过ATM查看账户信息的时候很重要，但对于Wordnik来说，数据是不断更新和增长的，这种“较精确”的保证几乎没有任何意义，反而会产生很大的延 迟。他们需要的是一个“大约”的数字以及更快的处理速度。但某些情况下MongoDB会锁住数据库。如果此时正有数百个请求，则它们会堆积起来，造成许多问题。我们使用了下面的优化方式来避免锁定：每次更新前，我们会先查询记录。查询操作会将对象放入内存，于是更新则会尽可能的迅速。在主/从部署方案中，从节点可以使用“-pretouch”参数运行，这也可以得到相同的效果。 使用多个mongod进程。我们根据访问模式将数据库拆分成多个进程。  ②文档结构的存储方式，能够更便捷的获取数据。对于一个层级式的数据结构来说，如果要将这样的数据使用扁平式的，表状的结构来保存数据，这无论是在查询还是获取数据时都十分困难。  ③内置GridFS，支持大容量的存储。GridFS是一个出色的分布式文件系统，可以支持海量的数据存储。内置了GridFS了MongoDB，能够满足对大数据集的快速范围查询。  (4)**Eureka** 注册中心:1.Eureka是Netflix开发的服务发现框架，本身是一个基于REST的服务，主要用于定位运行在AWS域中的中间层服务，以达到负载均衡和中间层服务故障转移的目的。SpringCloud将它集成在其子项目spring-cloud-netflix中，以实现SpringCloud的服务发现功能。  2.Eureka包含两个组件：Eureka Server和Eureka Client。  3.Eureka Server提供服务注册服务，各个节点启动后，会在Eureka Server中进行注册，这样EurekaServer中的服务注册表中将会存储所有可用服务节点的信息，服务节点的信息可以在界面中直观的看到。  4.Eureka Client是一个java客户端，用于简化与Eureka Server的交互，客户端同时也就是一个内置的、使用轮询(round-robin)负载算法的负载均衡器。  在应用启动后，将会向Eureka Server发送心跳,默认周期为30秒，如果Eureka Server在多个心跳周期内没有接收到某个节点的心跳，Eureka Server将会从服务注册表中把这个服务节点移除(默认90秒)。   1. Eureka Server之间通过复制的方式完成数据的同步，Eureka还提供了客户端缓存机制，即使所有的Eureka Server都挂掉，客户端依然可以利用缓存中的信息消费其他服务的API。综上，Eureka通过心跳检查、客户端缓存等机制，确保了系统的高可用性、灵活性和可伸缩性。   (5)**Spring Data JPA** :1、JPA是Java Persistence API，Java持久化API，是SUN公司推出的一套接口，一套标准Hibernate是一个具体的ORM的持久层框架，实现了JPA接口  2、Spring Data是Spring开发团队提供的一套标准API和不同持久层整合技术实现Spring Data的出现就是为了简化、统一持久层的各种实现技术APISpring Data在项目里以spring-data-commons这个jar存在  3、Spring Data JPA既实现了Spring Data接口，又实现了JPA接口，也是为了简化持久层的开发Spring Data JPA在项目里以spring-data-jpa这个jar存在。  4、Spring AOP 的思想a .拿到SimpleJpaRepository的所有方法 具体执行操作的类SimpleJpaRepository b.拿到自定义的接口的所有方法。 |

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 |  |