# 云计算 1607C 王顺虎

# JAVA工程师

1. **项目介绍(按照实训二做的项目写)**

|  |
| --- |
| **面试官您好，现在我给您介绍一下我在上家公司做的一个关于社交和招聘租房项目，在这个项目中呢，用户可以在平台进行交友，找工作，租房，等之类的，整个项目主要是采用SpringBoot+SpringCloud微服务框架实现的，通过eureka微服务注册中心实现多个微服务之间的调用，另外这么多的微服务不可能说是访问不同的微服务就使用不同的端口，这样显然是不现实的，所以我们就使用的Zuul网关，实现对请求的拦截和转发，然后数据库方面我们用的是MySQL关系型数据库和redis，MongoDB非关系型数据库，使用SpringDataJpa实现与数据库进行交互。搜索方面使用的是es，es可以实时的同步数据库中的数据，当然我们也不会让他几分钟就执行一次，而是给他设置一个定时器，半夜用户访问相对较少的时间在进行同步，这样不会影响用户的体验度，也能保证es中的数据能和MySQL中的一致性，这就是我对上家公司的项目的一个简单的介绍，您还有什么要问的吗？** |

1. **功能模块介绍(按照自己得实际项目写)**

|  |
| --- |
| 1、用户微服务：用户这个模块呢？主要是登录和注册功能，注册的话有一个短信验证的功能，我们用的是阿里大于进行的发短信，用户注册的时候点击获取验证码，后台生成一个6位随机数，一式两份Redis中一份，Mq 中一份，发短信模块呢，进行监听，监听到之后会调用阿里大于发短信的接口，用户接受短信填写验证码，跟Redis中进行对比，验证通过就是在数据裤用户表中加一条数据，登录这块，首先就是先判断这个用户是不是存在，如果是数据库有着一条数据，那么就是通过JWT生成一个token令牌，将这个token返回给前台 ，如果用户访问哪个微服务的时候，而当前操作是需要用户的权限的，那么这个微服务就是需要配置拦截器，但是这个拦截器不是对前进行拦截，而是对请求中的token进行解析，判断角色字段，将用户和管理员分别放在作用域中，如果访问当前的方法需要用户或者是管理员权限，就通过获取作用域中的token，判断这个token是否存在，不存在就不让执行抛出异常，存在就放行，程序继续执行  2、问答微服务：问答模块呢？主要是实现让用户实现在平台可以发布在面试中遇到的问题，请求帮助，对于好的回答可以进行点赞，这里用到了Redis防止重复点赞，用户也可以对已经有的问题进行答复，因为问答或者评论数据量是很大的并且是属于一层层的数据，所以我们没有用MySQL，而是用到了MongoDB这个文档型数据，后台我们在创建回答的时候会先判断他的上级Id是否存在，如果是存在的话上级的用户的回复数加1，MongoDB提供了一个inc方法这个方法是实现了当前字段的自增。  3、搜索微服务：搜索这块呢我们用的是es，es呢它有两个端口，分别是9200和9300，用户插件也就是可视化图形界面操作es的时候使用的是9200，java代码操作的时候使用9300端口，es 的数据结构是索引、类型、文档。索引就是相当于MySQL的数据库，类型就是相当于一张表，文档是相当于一行数据，es中有一个IK分词器，配置这个IK也是为了让它支持中文,需要在跟数据库中对应的实体类中的属性配置一个ik\_max\_word ，ik\_max\_word表示的是为最细粒度划分，还有一个是ik\_smart表示最少切分，这两个的区别呢，就是前者中的汉字可以被重用，然后就是将数据库中数据同步到es中，用的是logstash，用户搜索的话就是直接从es中查询返回。  4、交友微服务：用户点击交友这一按钮，会列出一些其他用户，对于自己喜欢的人，可以对其进行关注，对于不喜欢的人呢可以对其进行拉黑，后台操作的话就是判断是否喜欢，喜欢向好友喜欢表中添加后双方的Id，还要判断是否已经喜欢和是否互相喜欢，若互相喜欢该两条记录的状态为1表示相互喜欢，如果是不喜欢，还需要判断是否已经是好友，若已经是好友，还要判断是否是相互喜欢，是则删除单向喜欢数据，修改相反喜欢的状态为0，再向非好友中添加数据。如果不是好友，直接向非好友表中添加，对应好友粉丝数和自己的关注数要进行修改。  **金融项目这个项目的话就是一个提供借款，投资的一个p2p平台，用户可以通过平台进行借款也可以进行投资，但是这些操作后台都是需要审核的，我在这个项目中主要负责是模块是用户投标模块和用户招标模块，**  **招标模块就是用户通过平台进行借款，但是首先是要通过审核，审核规则就是每个用户注册后都有一个信用积分，规定就是信用积分必须超过30才可以进行借款，当然用户可以通过上传一些自己的一些证件，比如省份证，房产证，有没有 车啊，会对用户的这些信息进行评估，当信用积分达到30的时候就可以让他进行借款，这时会让他填写自己的一些基本信息，比如结没结婚啊，有没有子女啊后台审核通过才可以进行招标，**  **投标模块就是投资人对标的进行投资，可以手动投资，也可以设置自动投资，设置之后平台会进行监控，只要有标的就会进行按规定的最少投资金额进行投资，当然前提是投资人的账户的资金是充足的。否则就是投资失败。** |

1. **开发中遇到的技术难点以及解决办法**

|  |
| --- |
| 难点1：对分布式事务进行解决  解决方法1：本地事务  分布式事务两阶段提交协议  使用消息队列来避免分布式事务  不断重试，保证最终一致性。出错通过人工处理  难点：redis解决缓存失效和缓存击穿问题  首先先说下缓存失效，在同一时间有大量的key同时失效，瞬间加大了mysql的压力。解决办法就是存储key时失效时间要错开。  缓存击穿：他指的是用户访问的数据从redis里面查不到，从mysql里面查，如果重复访问，会造成mysql的压力加大。所以就要要求我们第一次从mysql里面查的时候，如果没有数据，就在redis把对应的value设置为null值。  难点：对分布式事务进行解决  首先项目主要用到的是微服务架构和分布式架构,项目中就会遇到有关分布式场景的一些问题,例如分布式事务,分布式锁,跨域等问题.分布式事务产生的原因是在多个系统中数据库垂直分割,数据库之间不能跨数据库进行访问,所以每个数据库的事务,都是在自己数据库独立进行管理的.这样就需要解决分布式事务的解决方案.而解决分布式事务的原理则是利用了Java的JTA,而java的JTA则是java操作XA接口,XA相当于协调者提供资源管理器与事务管理器之间进行通信的标准接口。而他们遵循2pc,3pc协议而我们绝大部分采用柔性事务都遵循CPA和Base理论,可以暂时不一致,但是最终要保证数据的一致性而我们公司最开始的探讨研究过分布式的解决方案第一种利用的是java Automatic框架,最开始感觉java Automatic有一定的缺点他遵循的是两段提交协议,他可能出现延时,当别人没有进行通知时可能进行卡死而我们公司发现Automatic用于多数据的情况下,用于传统的项目,而我们公司采用的是分布式项目所以不适合采用.在当时我们有采用另外一种方案,采用MQ解决分布式事务,但是由于我们的项目利用的是Springcloud,用MQ会比较麻烦所以最后又采用新的方案支付宝回调方式,就是类似于支付方式去做分布式事务管理但是又要自己写补偿机制和重拾机制还有要解决一些幂等性问题还是比较麻烦的,最后在github上找到一个新的框架tcc,lcn框架,发现lcn框架写的比较好,然后我们自己研究了一番,发现lcn底层原理的最大核心我不管理事务,而只是事务的搬用工lcn的原理是发起方调用接口,去xml创建事务分组,将事务分组的id传入,参与方接收到id时,操作一个假的关闭,不提交事务,当发起方代码执行完成以后,通知协调者,然后转发给参与方,看事务回滚还是提交.这是我们分布式事务的一个解决方案.  难点：对分布式锁进行解决  第二个难题就是分布式锁,什么是分布式锁,我们当时在项目当中遇到一种场景,需要做一个全局的id,但是全局id需要一个时间戳,如果在集群的情况下,时间戳因为服务器,可能在时间戳产生相同的情况下,当时想了好多的办法去解决全局id,之后利用分布式锁去解决.分布式锁相当于在多个服务器之间进行通信时,能够保证JVM性能的安全性问题,分布式锁跨Jvm,分布式锁在多个jvm进行通信时,能够保证多个JVM只有一个进行执行.在当时也有两种解决方式一个是利用reids,但是redis的有效期是比较麻烦的,有可能产生死锁,另外一种通过zookeeper的节点机制,首先zookeeper通过临时节点,当别人可以将临时节点创建成功时,就可以拿到我们这个锁了,因为节点是不允许重复的,当别人创建节点时,是不可创建的,需要去等待,等锁释放完成以后才可以拿到锁,那么zookeeper里面是怎样管理锁的释放了,zk有一个比较好的概念,就是有一个临时节点的特征.临时节点相当于session会话一关闭的情况下,整个连接将会断开,断开以后节点清空.所以锁就会被释放.  难点：跨域问题  在前后端分离的情况下难免会出现跨域问题,在当时前端通过Ajax发请求时访问不到,跨域问题产生于浏览器方面,不是在服务端,而在当时我们就在服务器端解决跨域问题,同时也考虑过客户端解决,但是客户端解决有一些缺陷,通过jsonp,jsonp本身原理是通过标签进行转发,有这样一个缺陷,这个缺陷在当时我们解决方式是因为它只支持get请求,不支持post请求,所以我们当时就搭建网关的方式,网关方式相当于全部请求到网关,然后进行拦截,将部分分发到具体服务器上,从而保证域名相同,或者我们可以使用httpclient请求进行转发,但是这个比较占内存,或者使用设置请求头进行跨域这是我在项目中要到的问题,分布式事务,分布式锁,和跨域问题,这几个问题的遇到让我有很大的提升. |

你最擅长什么：

MySQL 性能优化

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**spring boot**快速开发，快速整合，配置简化、内嵌服务容器。SpringCloud是一套非常完整的微服务解决方案，俗称“微服务全家桶”，几乎内置了微服务所使用的各种技术，可以不必集成第三方依赖。  (2)**redis技术**：项目中问答模块使用到redis了，具体是这样使用的用户对问题进行回答或者评论，其他用户就可以对评论进行点赞，但是又不能重复点赞，所有就用到了Redis，用户点赞后会在Redis中存一个当前用户的一个状态，再次点赞的时候会判断Redis中有没有当前用户的状态，有的话不让重复点赞。验证码也是放在Redis中，因为Redis可以设置过期时间。  **RDB持久化**  RDB 是在某个时间 点将数据写入一个临时文件，持久化结束后，用这个临时文件替换上次持久化的文件，达到数据恢复。  优点：使用单独子进程来进行持久化，主进程不会进行任何 IO 操作，保证了 redis 的高性能  缺点：RDB 是间隔一段时间进行持久化，如果持久化之间 redis 发生故障，会发生数据丢失。所以这种方式更适合数据要求不严谨的时候。  **AOF持久化**  Append-only file，将“操作 + 数据”以格式化指令的方式追加到操作日志文件的尾部，在 append 操作返回后(已经写入到文件或者即将写入)，才进行实际的数据变更，“日志文件”保存了历史所有的操作过程；当 server 需要数据恢复时，可以直接 replay 此日志文件，即可还原所有的操作过程。AOF 相对可靠，它和 mysql 中 bin.log、apache.log、zookeeper 中 txn-log 简直异曲同工。AOF 文件内容是字符串，非常容易阅读和解析。  优点：可以保持更高的数据完整性，如果设置追加 file 的时间是 1s，如果 redis 发生故障，最多会丢失 1s 的数据；且如果日志写入不完整支持 redis-check-aof 来进行日志修复；AOF 文件没被 rewrite 之前（文件过大时会对命令进行合并重写），可以删除其中的某些命令（比如误操作的 flushall）。  缺点：AOF 文件比 RDB 文件大，且恢复速度慢。  **RabbitMq**：是一套开源的消息中间件，通过rabbitmq在项目中实现了消息的生产和消费，有三种模式，包括直接模式、分裂模式以及topic模式。项目中采用的是直接模式。通过异步的方式获取消息，调用阿里云接口获取验证码。  (3)**MongoDB技术**：MongoDB 将数据存储为一个文档，数据结构由键值(key=>value)对组成。MongoDB 文档类似于 JSON 对象。字段值可以包含其他文档，数组及文档数组。，可以在MongoDB记录中设置任何属性的索引，如果负载的增加（需要更多的存储空间和更强的处理能力） ，它可以分布在计算机网络中的其他节点上这就是所谓的分片  (4)**JWT：**这JWT呢首先它是由三部分组成的头部、载荷、签名，头部就是jwt的基本信息，例如类型和签名所使用的算法，然后就是对头部信息进行加密，形成第二部分。载荷：就是存放有效信息的，比如说是当前用户的id和名字，还有角色，然后也是对信息进行加密，形成第二部分，然后将第一部分和第二部分进行拼接形成一个字符串，对这个字符串在进行加密这样就形成第三部分，将这三部分用.拼接就形成了token。  (5) **es:** es，相对于solr的好处就是可以试实时建立索引，而solr实时建立索引会发生堵塞，不足之处就是维护团队没有solr的维护团对那么完善，用户对问答、好友、招聘信息进行搜索的时候就是先直接去es中查询，直接返回，es会定时同步数据库中的数据的，保证es中的数据是最新的。 |

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 |  |