

自我介绍+项目介绍

自我介绍

你好我叫王欢,今年29岁,有5年多的java开发经验,

之前待过的公司主要有, 来伊份, 怪兽充电和爱回收。

业务的话主要涉及电商和供应链。

怪兽:

在怪兽的话主要负责建设资产数据体系, 就是整个充电宝, 柜机基础数据。他的一个难点的话主要在于, 数据量大, 他的数据膨胀特别快。所以需要持续的去引进一些方案, 做一些优化。我这边用到或者说引进的一些技术主要包括, sharding-jdbc, sharding-proxy。去做一些分库分表的方案。然后引进DTS去做数据同步和数据镜像, 用elasticsearch去做一些复杂的列表查询, 用clickhouse去做一些数据报表数据大屏。

然后以及整个过程中回去做一下性能优化,慢查询,深度分页,数据镜像时间过长,es数据同步延迟

再怪兽这块整个做的还是比较好的, 服务的稳定性,服务的性能.还有整个的一个技术方案.

爱回收

现在是在爱回收这边, 在这边主要是负责卖家端的一些相关业务, 主要负责的工作 会负责公司一些重点项目的设计跟进,然后代领组内的人一起开发.

会参与代码评审,复盘会议,总结编码规范. 然后制定一些技术规范的文档包括像设计文档,代码评审的文档.

然后平常会主动去协助组长 想一些方案,解决一些技术难题. 假如说像我刚来的时候这边的时候线上问题会比较多, 然后我提出组织大家轮流每周五对服务做一个定期巡检, 看看 错误日志,慢查询,服务的健康情况 cpu,内存,jvm. 另外的话 就是基于复盘和代码评审制定一些代码规定. 效果还是比较明显的.

然后最近的话 是做了一个es的查询引擎,主要解决es查询的代码的一个简化,减少工作量,另一个就是实现es查询的降级,假如说 es出现了 异常, 为了保证服务的可用性,可以降级到mysql查询.

es延迟

DTS集成的kafka是基于单分区的，所以同一时刻只能有一个消费者消费,在某些情况可能会产生消息堆积，导致消费延迟

使用redis 队列

项目

es查询引擎

为什么要去做这个东西呢:

1)原生的客户端使用起来没有那么方便, 业务代码需要些大量的判断,常量或者枚举, es的查询条件又比较多, 代码看起来非常冗余不是很优雅.

2)有一些核心的业务, 假如说app 物品列表, 如果es出问题了,可能会导致服务出现不可用. 那我们期望做成什么样子呢, 就是出来es列表查询之外,会用mysql再来做一个兜底. 就是会写两套,一套是es的查询,一套是mysql的查询,然后会长配置中心做一个开关, es有问题,可以直接切换到mysql. 这个方案是没什么问题的.但是做起来非常的麻烦, 你需要去写两套代码,测试的工作量也会增加.修改也要修改两套,增加了系统的复杂度也容易出错.

基于以上两个原因,我就去调研然后封装了一个es查询引擎.

项目介绍

然后我具体说一下 他的一个主要功能和实现的一些思路:

1)基于注解的一个结构化查询, 再入参的对象上面对应的字段上 加上对应的es注解, term terms from to exist prefix sort . 主要会用到反射,把注解解析出来之后, 封装成es的查询语句. 然后去执行查询. 主要会用到策略模式,term查询有一个term对应处理的handler.

2)通过动态代理结合spring扩展 去实现跟 jpa 一样的那种你只写一个接口的定义,然后我会自动给他生产代理类,代理类会调用的前面说的那个 解析查询的. 然后去执行查询. 这个主要会用到spring的一个扩展点, BeanDefinitionRegistryPostProcessor BeanDefinitionRegistry的后处理器. 扫描继承了BaseESRepository的对象 也就是我们声明的接口. 然后会以他的 simpleName 作为key, 然后已我们创建的一个动态代理对象作为value 注册到BeanDefinition当中. 动态代理的方法里面就会实现具体的es查询逻辑 (动态代理invoke里面可以拿到他的 方法, 入参)

3)支持sql查询, 就像jpa sql原生查询那样定义了一个注解,注解里面写sql和对应的占位符, 然后我会解析, 封装成http请求的sql查询格式, 查询es,再将查询结果解析一下. 可以像jpa一样在注解里面写个sql去执行查询.

4)整合外部的框架, 目前的话整合了mybatis,前面已经实现了sql查询, 那么如果我们能把mybatis执行过程中产生的sql给它拦截下来,然后再把这个sql拿到es去执行 返回结果. mybatis本身他支持拦截器, 会通过mybatis的拦截器实现 拦截, 判断方法上面是否有自定义的注解, 调用es查询并返回结果. 然后其中有两个需要处理的地方, 第一个就是 我们平常es的结构都是多个表结合的,

也就是说 mybatis的 查询sql大多数情况下可能是一个连表查询,如果直接去es执行的话可能不对, 需要对sql做一个改写,这里会用到sql解析器jsqlparser 来对sql进行一个解析改写.

改写的思路:

1)第一会去掉所有的join,把表名替换成 去获取我们mapper接口上自定义注解里面标识的 索引名

2)条件,分组,排序,having ,里面的字段可能是带有别名的 假如 t.id , d.status 我们要把 前面的别名点去掉.

另外的话会加一个开关,然后执行查询的时候去判断一下是否要走开关. 最终达到的一个效果就是直接在mybatis Mapper 接口上去添加一个注解就可以实现es的查询, 不用开发两套代码, 然后降级的话直接也比较方便.直接在配置中心加一个开关,去判断是否需要降级. 执行的sql都是一样的,基本上也不会有太大的问题.

业务

整体业务,

统货操盘

9点业务上班 ,9点半之前业务要操作, 10点开场. 所以我只有半个小时时间, 而且越快越好

创建规则,针对不同的品类,仓库,型号. 选品创建统货包,打包售卖

选品->锁库->分包->创场 每个节点入库

流拍, 自动降价 低于底价 运营处理 ,打散,统货重新定价

, 打散 , 暗拍, 明拍, 一口价