齐 齐 哈 尔 大 学

毕业设计（论文）

题 目： 基于Spark网易云音乐数据分析设计和实现

学 院： 计算机与控制工程学院

专业班级： 软件141班

学 号： 2014023002

学生姓名： 边琪

指导教师：

成 绩：

2018年6月18日

# 郑 重 声 明

本人呈交的学位论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用内容外，本学位论文的研究成果不包含他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文的知识产权属于培养单位。

本人签名： 日期：

# 摘 要

随着互联网发展，大批大型互联网公司如雨后春笋发展壮大，诸如淘宝，网易，百度公司随着规模的扩大，在运维中产生的大量的运营数据，根据公司积累的海量的数据，市场策划部以及运维部去调整公司各个部门发展战略显得尤为重要，同时数据可视化帮助公司高层领导更清晰的洞察整体发展现状，了解公司的发展现状，此外网站的规模不断扩大，用户激增，用户对客户端体验度提升，包括CDN部署提高用户的访问网站的速度，以及为了更好提高用户的粘性度，构建人物画像，为推荐系统提供数据，提升公司市场竞争力是每个互联网公司亟待提升方向。

关键词：搜索引擎；Spark；ElasticSearch；Logstash；Flume； Hadoop；Azkaban

Abstract

The resources in the Internet are very rich, but how to efficiently search and access to resources is an extremely difficult thing. This paper aims to provide users with a convenient, fast and efficient way to obtain network resources by setting up a network resource search engine based on more than 10 million shared resources collected from Baidu network disk.

The main functions of the search engine based on cloud computing and large data include reptile module, query service module and data analysis module. System Implementation The WebMagic web crawler framework was used to write network robots, using ElasticSearch's highly extensible and open source full-text search and analysis engine to build index libraries, and Logstash to import local resource data into the index library. Using JavaEE to build and develop B/S-based search engine system, to provide users with a variety of search interface, simple and responsive response. System pages using Bootstrap, HTML5, CSS3 development, responsive interface beautiful atmosphere, simple and pleasing to the eye.

The search service provided by the system can quickly and near real-time retrieval and analysis of massive data, can provide users with the most massive, most efficient, the most diverse and most valuable data, to the user a wonderful search experience.

**Key words:** Search engine; WebMagic; ElasticSearch; Logstash; Bootstrap

目 录

[摘 要 I](#_Toc515111066)

[Abstract II](#_Toc515111067)

[第1章 绪 论 1](#_Toc515111068)

[1.1 课题背景 1](#_Toc515111069)

[1.2 国内外的发展现状 2](#_Toc515111070)

[1.3 课题实现的主要目标 2](#_Toc515111071)

[第2章 系统相关技术 3](#_Toc515111072)

[2.1 系统开发及运行环境 3](#_Toc515111073)

[2.2 技术简介 3](#_Toc515111074)

[2.2.1 Spark简介 3](#_Toc515111075)

[2.2.2 Hadoop简介 4](#_Toc515111076)

[2.2.3 ElasticSearch简介 4](#_Toc515111077)

[2.2.4 AmazeUI简介 4](#_Toc515111078)

[第3章 系统可行性和需求分析 5](#_Toc515111079)

[3.1 可行性分析 5](#_Toc515111080)

[3.2 业务描述 6](#_Toc515111081)

[3.3 功能需求 7](#_Toc515111082)

[3.3.1 功能需求分析 7](#_Toc515111083)

[3.3.2 系统功能需求 7](#_Toc515111084)

[3.4 非功能需求 9](#_Toc515111085)

[3.4.1 需求分析阶段的架构性能分析 9](#_Toc515111086)

[3.5 UML统一建模 9](#_Toc515111087)

[第4章 系统设计 11](#_Toc515111088)

[4.1 系统设计目标概述 11](#_Toc515111089)

[4.2 系统技术架构设计 11](#_Toc515111090)

[4.2.1 架构设计目标 11](#_Toc515111091)

[4.2.2 Spark离线计算引擎系统 12](#_Toc515111092)

[4.2.3 系统的网络架构 13](#_Toc515111093)

[4.3 系统功能架构 15](#_Toc515111094)

[4.3.1 爬虫模块 16](#_Toc515111095)

[4.3.2 数据下沉模块 18](#_Toc515111096)

[4.3.3 数据分析模块 18](#_Toc515111097)

[4.3.4 数据展现模块 20](#_Toc515111098)

[4.4 数据展现模块数据库表设计及ES实体设计 20](#_Toc515111099)

[4.4.1 数据库的实体ER图 20](#_Toc515111100)

[4.4.2 ES实体映射设计 21](#_Toc515111101)

[第5章 系统详细设计与实现 23](#_Toc515111102)

[5.1 UI页面设计 23](#_Toc515111103)

[5.1.1 功能简介 23](#_Toc515111104)

[5.1.2 设计思路与实现过程 23](#_Toc515111105)

[5.1.3 界面截图 23](#_Toc515111106)

[5.2 数据存储模块 25](#_Toc515111107)

[5.2.1 功能简介 25](#_Toc515111108)

[5.2.2 设计思路与实现过程 26](#_Toc515111109)

[5.2.3 界面截图 27](#_Toc515111110)

[5.3 数据分析 27](#_Toc515111111)

[5.3.1 功能简介 27](#_Toc515111112)

[5.3.2 设计思路与实现过程 27](#_Toc515111113)

[5.3.3 界面截图 30](#_Toc515111114)

[第6章 系统测试 32](#_Toc515111115)

[6.1 测试目的 32](#_Toc515111116)

[6.2 测试原则 32](#_Toc515111117)

[6.3 系统测试的实施 32](#_Toc515111118)

[6.3.1 架构测试 32](#_Toc515111119)

[6.3.2 数据传输测试 34](#_Toc515111120)

[6.3.3 系统预发布测试 35](#_Toc515111121)

[结 论 36](#_Toc515111122)

[参考文献 37](#_Toc515111123)

[致 谢 38](#_Toc515111124)

# 第1章 绪 论

当今世界是一个信息高速发展的年代，信息流通越来越大，人与人之间的交流越来越密切，生活购物越来越方便，互联网越来越渗入到每一个人的生活中的每一个时刻，无时无刻我们每个人都在产生数据，大数据在这个背景下由此而产生了，云计算时代的到来更加激增了大数据的火热程度，大数据的应用也是在这个背景下，如火如荼的展开来，他所占的领域包括电子商务，O2O，音乐视频，物流，互联网公司在运营中生成，积累了大量的用户网络操作行为，这些数据非常庞大，不在可以使用GB，TB来衡量，淘宝天猫当天一天产生的数据就在几百PB，甚至是EB级别，这些数据背后蕴含着大量信息对于公司战略调整以及提高公司效益有着巨大的潜力。伴随着技术成熟和我们现代设备的计算能力的强大，离线和实时数据分析也逐渐成为了一种可能。

## 1.1 课题背景

随着互联网的不断发展，各大互联网公司积累了大量的数据，数据中蕴含大量的信息和价值，这些数据可以为公司未来的发展和最大决策提供巨大的参考价值，很多互联网公司忽略了这一部分数据价值。但是对项目和决策进行全方位的科学分析和评估项目决策的可行性，为公司投资和决策提供科学严谨的的依据，减低公司决策和项目投资的风险。数据分析报告是企业决策可行性的重要判断，任何欣欣向荣的企业的发展壮大都是建立在数据分析基础上的。如何才可以确定发展战略的可行和优化，如何才可以了解用户的留存呢？音乐网站QQ，网易云音乐，酷我音乐各大运营商对自己很多年发展做出大量调整，网易云音乐为提高用户的粘度，推出个性化推荐，个性化推荐的背后需要用户对不同风格音乐的分类等等。所以尤其是在网站运维发展离不开数据分析报告，与此同时企业的管理层迫切需要一个统一的，规范的数据报告来衡量企业项目开发可行性和企业发展方向合理度。因此专业的数据分析报告在国内变得炙手可热。越来越大的互联网尤其是音乐运营商也选择项目数据分析报告为他们公司的发展做出科学，合理的分析，以此来正确决策项目，更多的大型互联网公司的投资方将数据分析报告作为他们投资公司和企业项目是否合适，是否值得投资的重要根据。因此说：公司的未来的发展离不开数据分析，公司决策更加离不开数据分析报告，投资方考虑是否投资也是离不开对于数据分析报告。

## 1.2 国内外的发展现状

数据分析报告在近几年大数据和机器学习不断火热程度背景下，国内：如阿里淘宝双十一总控制台实时刷新双十一交易量，以及后期对网站流量数据分析，对于用户偏好点击网页的那些部位，用户对那些商品更感兴趣。在音乐视听领域，游戏领域，近期比较火热的绝地求生刺激战场，腾讯利用大数据分析鲸鱼用户，以及用户的留存进一步对用户奖励做出修改等等，国外社交网站例如Facebook，在社交方面，用户的粉丝关注度对于权重最初分析，对一些鲸鱼用户定期做出一些鼓励性奖励更多吸引大量的用户来提升网站总体用户。国外最大的代码托管仓库GitHub利用用户点击日志向用户推荐用户感兴趣的项目。但是相对来说很多的数据分析报告都是处于初级的发展阶段。数据分析仍然存在巨大的潜力。

## 1.3 课题实现的主要目标

本课题主要研究的是基于Spark网易云音乐数据分析，音乐网站运营离不开数据分析，对于公司的领导层也需要查看网站近期运维状况。课题主要研究目标为采集网易云音乐平台数据利用大数据框架实现人物画像分析，网站流量时间段使用频率，用户权重，以及用户年龄分布。根据网站用户使用地区呈现网站用户的地区分布从而合理部署CDN。大致分为以下几点：

（1）用户人物画像任何一个网站都是以用户为中心，对于用户喜欢听什么风格的音乐，用户是喜欢在哪一个时间段登录网站使用APP听音乐，用户是哪一地区等等。为网站的推荐系统提供非常有利的数据。例如A用户位于苏州喜欢听伤感和怀旧的音乐，B用户也位于苏州也是喜欢听伤感音乐。即可为B用户尝试推荐怀旧风格的音乐。利用人物画像结合协同过滤算法提高用户对本网站粘稠度。

（2）歌曲分类预测任何一个音乐网站都会用大量的歌曲，对于海量歌曲的分类如果交给人工分类是一个繁琐的过程。采用分类算法多层感知器对歌曲的进行分类。使用训练好的模型，再不断进行分类。为网站首页分类歌曲提供一定的依据。

（3）挖掘网站鲸鱼用户是很多网站都需要的，Google早在2012年对网站排名就使用了图计算。课题实现目标之一就是通过用户关注和被关注度使用图计算挖掘出本网站鲸鱼用户。

（4）对某一首歌和整体歌曲采取时间段以及所有歌曲整体热度分析。例如通过分析刘若英的后来这一首歌，我们可以观察到近期这一首歌再次火热于近期后来的我们上映有很大的关联。对所有歌曲热度的整体进行分析，音乐平台可以在用户初次注册的时候为用户推荐排名度最高的歌曲，因为大多数用户喜好都是基本上一直的。

（5）随着网易云音乐网站用户突破3亿，网站压力也是越来越大，这就对于网站架构的改造例如动静分离，CDN部署显得尤为重要。但是CDN的部署投入的资金是非常庞大的，这就对于网站运维人员合理部署CDN提出更高的要求，因此数据分析报告平台分析用户的在国内地理位置分布为运维人员合理部署CDN提供了很高的数据价值。更加提高了网站的高可用性。

# 第2章 系统相关技术

## 2.1 系统开发及运行环境

本课题制作的是基于Spark网易云音乐数据分析报告是基于B/S架构的互联网应用，采用PowerDesigner和Visio进行系统架构图、流程图等的绘制，在IDEA及Eclipse集成开发环境中使用Java和Scala语言进行网易云爬虫系统开发和数据分析报告平台开发。采用AmazUI前端框架以及BootStrap进行系统界面的开发，和样式的美化，使用原生JS提供用户平台交互。

由于本课题研究的基于Spark网易云数据分析报告，系统依赖Spark集群，Hadoop集群，Flume及LogStash采集日志，以及Azkaban提供调度平台，Mysql数据库存储Spark处理后的数据同时为Azkaban提供元数据数据库。ElasticSearch(以下简称ES)提供搜索。开发中采用Vmware启动十一台虚拟机其中三台虚拟机搭建Spark集群，一台Master，两台Worker，其中三台虚拟机搭建Hadoop集群一台NameNode，两台DataNode,其余五台服务器分别为一台Flume服务器，一台Azkaban调度平台服务器，一台MySql服务器,一台数据平台展现服务器。一台部署ES和LogStash服务器。

## 2.2 **技术简介**

本系统搭建和开发中采用了大量的如今非常流行的大数据技术和框架，以下我对部分大数据流行框架做一个简单介绍。

### 2.2.1 Spark简介

如今在Hadoop生态圈，引领大数据风向的框架莫过于Spark框架，众所周知Spark框架是加州大学伯利克分校实验室的项目，性能方面：Spark在内存的计算速度上比起MapReduce要快100倍，在硬盘上比起Spark要快10倍。Spark可以运行在Hadoop，Apache Mesos，Kubernetes，独立版云平台，同时它可以访问不同的数据源。功能方面：经过这么多年发展Spark在RDD，DataFrame，DataSet，Spark Streaming，Spark Structured Streaming,Spark mllib，Spark graphX，Spark SQL等等。架构方面：包含Spark Master，Spark Worker各大大型互联网公司数据分析平台逐渐向Spark计算平台靠近。

### 2.2.2 Hadoop简介

Hadoop是Google三大剑客之一GFS的开源实现，后期逐渐发展成为Apache顶级项目之一。是一个分布式文件存储系统，其强大的容错和存储，可以让许多的中小型公司在廉价的机器平台上搭建大型的分布式存储系统，用户的和开发者可以在不太了解底层的情况下，迅速开发属于自己的业务的分布式程序，充分利用集群威力实现快速计算和存储，分布式文件存储系统又称HDFS，全称Hadoop Distributed File System。NameNode和DataNode为HDFS俩大核心组件。NameNode负责元数据存储，DataNode负责真实数据存储，Hadoop的优势在于高可靠性，高扩展性，高效性，高容错性，低成本。在HDFS引导下逐渐形成了Hadoop生态圈MapReduce Hbase Pig Spark 等等一系列优秀的大数据框架。

### 2.2.3 ElasticSearch简介

ElasticSearch作为ELK技术栈中的一个。以Lunce为基础，是一个开源分布式实时搜索引擎，其强大的分析能力和快速的执行速度成为了许大型中性公司的首选，其中京东天猫都是在ES的github主分支上Fork自己的分支，维护自己的代码。社区也是相当活跃。PR接近几千。架构方面：cluster代表一个集群，集群中有多个节点，其中有一个为主节点，这个主节点是可以通过选举产生的，主从节点是对于集群内部来说的。es的一个概念就是去中心化，字面上理解就是无中心节点，这是对于集群外部来说的，因为从外部来看es集群，在逻辑上是个整体，你与任何一个节点的通信和与整个es集群通信是等价的。Shards：代表索引分片，es可以把一个完整的索引分成多个分片，这样的好处是可以把一个大的索引拆分成多个，分布到不同的节点上。构成分布式搜索。分片的数量只能在索引创建前指定，并且索引创建后不能更改。recovery代表数据恢复或叫数据重新分布，es在有节点加入或退出时会根据机器的负载对索引分片进行重新分配，挂掉的节点重新启动时也会进行数据恢复。

### 2.2.4 AmazeUI简介

AmazeUI是国内第一个开源 HTML5 跨浏览器跨屏幕的流行的框架。转为移动设备而产生。Amaze UI 以移动优先（Mobile first）为理念，从小屏幕一步一步移动到大屏幕，这一切都是为移动互联网趋势的适配器。 模块化组件：惊人的UI有近20个CSS组件，20个JS组件，并且有多个包含不同类型的Web部件，可以更好地快速构建界面，体验最佳的跨屏页面，大幅提高开发效率。 本地化支持：在外国对AmazeUI的青睐情况下，根据User Agent调整字体以获得更好的英文排版效果; 考虑到国内主流浏览器和应用程序内置的浏览器兼容性支持。 轻量级，高性能：惊人的UI开发和使用HTML5 CSS3动画交互，平滑，高效，更适合移动设备，使Web应用程序加载速度更快。

# 第3章 系统可行性和需求分析

在系统进行开发的前期阶段，必须对系统可行性和需求进行一系列的衡量，否则系统开发将变得毫无意义和价值，音乐网站在不断做大做强同时，网站的运维和决策人员急切需要观察网站的运维状况和近期的发展现状，公司领导层也非常关注。系统极大的方便了网站管理人员的对系统清晰的了解。比起传统的BI系统，后台基于强大的Spark计算集群和Hadoop集群对数据处理功能强大，在可行性和需求方便都是可以的。

## 3.1 可行性分析

（1）经济可行性

经济方面：系统采用10台VMware，一台Tomcat服务器，全部在window10笔记本电脑架设虚拟机，考虑到内存不足，为笔记本电脑添加16G内存条，经济上不需要巨大的开销。开发环境和部署环境都是采用开源免费的产品。经济上没有太大的需要，所以说在经济上是可行的。

（2）时间可行性

时间方面：大四时时间是非常充足的，前期爬虫开发加上系统架构搭建包括Scala代码的开发以及展现平台的开发，时间大概在5到6个月左右，在时间上时间是非常充足的，所以说时间方面是可以的。

（3）技术可行性

技术方面：系统中所采用的大量的主流的大数据技术Spark和Hadoop，ES，Azkaban，Flume都是大三时候就学习过的技术，但是涉及到Scala代码的开发，因为Scala天生表达性非常强大，难度比较高，所幸有StackoverFlow，github国外技术高手可以帮助到自己。当然了在开发中也难免遇到一些棘手的问题，身边可以有QQ群之类的高手帮助自己因此说技术上是可行的。

（4）操作可行性

操作方面：作为软件就是为方面用户使用，程序处理后的报表全部在展现平台展现，数据报告平台采用AmazeUI页面简单大方，JS动态交互方便用户操作，包括首页数据呈现，搜索页面的简单方便，以及使用了echarts，各种交互，让用户有一种美感。在操作方面用户不仅简单而且会系统上这套系统。

（5）法律可行性

法律方面：系统开发中所采用的全部软件都是开源免费不存在侵权问题，系统所使用的框架都是遵循Apache2.0协议。开源免费，这套系统开发中也不存在版权问题，网上关于这方面的系统也是甚少。在法律方面不存在纠纷。

## 3.2 业务描述

本系统作为一套数据分析报告平台，主要用户为网站的运维人员网站决策人员以及公司的高层领导，需要运维人员配合数据运行，用户使用时仅仅也要点击左边菜单栏观看自己所关注的部分，系统数据采集的部分为爬虫，数据传输采集和传输采用Flume和LogStash,分析系统为Scala代码提交到Azkaban调度在Spark大数据平台上运行，数据处理结束后自动导入的MySQL数据库中。最后在BI平台展现。

下面描述了数据报告展现平台的整体功能结构如图3-1所示。

网站运维和管理人员可以查看音乐网站的目前使用人数，歌曲数量，歌曲评论数量，一天内网站的运行压力，总体热歌排行榜，某一首歌评论词云，图计算中鲸鱼用户和屌丝用户，用户关联三角形数目，及用户在全国整体分布，用户使用年龄分布，用户星座分布，用户性别比例分布，以及歌曲的分类和评论分类包括人物画像分析等等。



图3-1 数据展现平台结构图

## 3.3 功能需求

整个系统涉及到网站运维人员和网站决策以及公司高层领导层，系统脚本的部署集群的管理以及数据的呈现部分功能都是需要公司内部专业的人员操作，系统BI数据展现平台的使用用户不需要了解系统后台脚本集群是如何运行的。整个系统前台对网站公司决策人员和领导是完全黑盒的。

### 3.3.1 功能需求分析

（1）用户类别

整套系统开发的目的为了那些急需了解网站公司运行状况的，更加清晰生动的展现网站数据，公司高层领导对于数据展现生动形象的要求是不可忽视的。同时数据从HDFS获取而来，运行脚本，需要网站管理人员干预，在脚本部署这个过程方便便捷更是显得非常重要。从数据呈现平台公司高层领导，到集群运维监控，到脚本部署网站维护人员，所有的功能各司其职。

（2）系统需求

系统开发前期对市场做出调查，对于大部分的音乐网站数据分析做出比较少，对于很多公司的高层领导对于网站系统，公司运维状态没有一个清晰的认识和感性直观的认识。这套系统对于海量的数据分析报告音乐网站公司整体运行状态，用户的存留配合其他开发部的数据支持。业界淘宝也是近几年才开始进行数据BI呈现，许多新型的小公司开始青睐这一部分市场。

（3）开发脚本部署

系统中脚本的部署是非常重要的关系到系统数据的可靠性，在运行脚本时，需要网站专业人员部署到azkaban中，执行定时任务或者是延时任务，需要专人负责，这套系统架构中这一部分对于部署人员技术要求是比较高的。

由于。

### 3.3.2 系统功能需求

由上面的系统的大致需求分析后，下面对于各类用户使用系统的用例做出一定得分析，系统上有爬虫，中间有Spark和Hadoop集群，下层有Azkaban调度系统平台，这种数据下层到Mysql数据中，分为3类用户，不同的用户有不同的用例，不同的使用方式。第一类系统集群维护人员，第二类系统的脚本部署人员，第三类观测数据领导和决策人员。系统集群维护人员用例图如图3-2，系统脚本部署人员用例图如图3-3，系统数据呈现使用人员如图3-4。



图3-2 集群维护用例图



图3-3部署维护用例图



图3-4 BI平台使用者用例图

## 3.4 非功能需求

### 3.4.1 需求分析阶段的架构性能分析

本课题研究是互联网数据分析，海量的数据分析依靠可靠的强大的计算平台，系统整体架构所使用到的Spark集群和Hadoop集群都是社区和业界公认的性能强大的计算平台，稳定安全。大数据分析离不开计算资源。系统开发限于使用笔记本但是三台Spark集群每台3G内存足以运行Spark处理数据。因为是离线分析，时间反而不是问题，计算内存低，可能需要更多的时间。所以综上在时间上架构性能上，是我们可以承受的，性能方面可以接受。

## 3.5 UML统一建模

统一建模语言UML，现代系统开发中，无论是系统的架构还是系统用户需求都离不开系统建模，可视化建模已经成为了工业界离不开的统一的标准，在系统设计的初期和后期，都可以帮助用户缩短设计时间，减少改进的成本。

在系统开发中，系统的整体全部架构图如图3-5所示。



图3-5 系统整体架构图

类图是是个系统中一定不可缺少的部分，类图可以清晰的展现出系统中的各个类之间的协作关系。



图3-6 数据呈现模块类图

# 第4章 系统设计

## 4.1 系统设计目标概述

本系统的设计目标为了公司整体业务方向的调整和配合数据部门推荐开发部门的同学以及为领导提供报表，所以系统的设计的初衷就是为了挖掘出数据中的价值，优化公司的业务流程，以及达到公司对某一项投入的结果的预测，同时为提高客户的对产品的体验度离不开数据可视化和数据分析平台提供的数据，推荐部门需要分类数据为歌曲分类提供数据依据，提高用户的APP体验度以及用户对APP的粘度。获得更多的忠实的用户。

## 4.2 系统技术架构设计

本节将对系统的大致架构做出分析和介绍，音乐网站的大数据都有数据源，本系统的数据源来自于爬虫系统，强大的多线程爬虫系统爬取采用Flume和Logstash收集数据下沉到HDFS，提交Scala代码到Azkaban，Azkaban调度Scala到Spark计算平台，Spark计算完毕后把数据导入到MySQL数据库中，数据BI展现平台展现MySQL数据库中的数据。

### 4.2.1 架构设计目标

在设计本系统之前做出了大量的思考和资料的查询，参考各大现行的大型互联网公司的大数据架构，参考了2015年的唯品会的离线部门采用的离线计算平台，作为国内的第三大电商平台的设计的计算架构具有实用性和稳定性。爬虫通过自己编写，其他的从Flume采集数据到ES搜索到Spark计算到HDFS存储都是性能高，系统稳定，达到音乐公司的系统数据所需。

作为离线数据分析平台架构设计需要达到以下的目的：

1. 离线平台性能：Flume，Logstash日志采集工具，高性能的日志采集Logstash对于数据的丢失都做出了兼顾。Spark平台根据Spark官网做出了各种性能优化，采用Spark2.2.2系统，框架从内部做出了各种优化，除了这些JVM的内存做出优化，数据展现平台Tomcat参考官网做出优化，包括从BIO更换成NIO，平台数据HDFS存储可以加近期采用read balance。数据进行平分切割存储。
2. 系统安全性：由于系统是为公司内部服务的，部署到公司的内网，安全方面无需考虑的太多的东西，但是出于安全问题，数据存储问题在代码提交方面经过检验才可以提交到Azkaban提交，否则可能导致集群崩溃，所以在集群运维方面安全要自己内部注意。
3. 数据可视化展现友好性：可视化的平台采用AmazeUI以及大数据可视化平台的Echarts和Highchart,本身数据呈现就非常炫酷，Echarts作为百度开源的，业界非常优秀的JS框架，除了这一点在移动端的兼容方面也是非常大方美丽。
4. 离线计算平台的稳定性：HDFS数据存储海量数据丢失是可能无法避免，为了避免数据出现安全问题，HDFS采用了三套数据备份，数据的丢失几乎不存在，Spark采用RDD算子计算，计算稳定性在Spark2.2.0分布已经过去半年多稳定性也是可以的，Spark社区异常的活跃如果有稳定性问题可以在社区的到很快的解决。在稳定性方面这套离线计算平台非常不错的。

### 4.2.2 Spark离线计算引擎系统

Spark计算底层采用RDD进行迭代式计算，RDD（Resilient Distributed Dataset）叫做分布式数据集，是Spark中最基本的数据抽象，它代表一个不可变、可分区、里面的元素可并行计算的集合。RDD具有数据流模型的特点：自动容错、位置感知性调度和可伸缩性。RDD允许用户在执行多个查询时显式地将工作集缓存在内存中，后续的查询能够重用工作集，这极大地提升了查询速度。RDD属性如下：

（1）一组分片（Partition），即数据集的基本组成单位。对于RDD来说，每个分片都会被一个计算任务处理，并决定并行计算的粒度。用户可以在创建RDD时指定RDD的分片个数，如果没有指定，那么就会采用默认值。默认值就是程序所分配到的CPU Core的数目。

（2）一个计算每个分区的函数。Spark中RDD的计算是以分片为单位的，每个RDD都会实现compute函数以达到这个目的。compute函数会对迭代器进行复合，不需要保存每次计算的结果。

（3）RDD之间的依赖关系。RDD的每次转换都会生成一个新的RDD，所以RDD之间就会形成类似于流水线一样的前后依赖关系。在部分分区数据丢失时，Spark可以通过这个依赖关系重新计算丢失的分区数据，而不是对RDD的所有分区进行重新计算。

（4）一个Partitioner，即RDD的分片函数。当前Spark中实现了两种类型的分片函数，一个是基于哈希的HashPartitioner，另外一个是基于范围的RangePartitioner。只有对于于key-value的RDD，才会有Partitioner，非key-value的RDD的Parititioner的值是None。Partitioner函数不但决定了RDD本身的分片数量，也决定了parent RDD Shuffle输出时的分片数量。

（5）一个列表，存储存取每个Partition的优先位置（preferred location）。对于一个HDFS文件来说，这个列表保存的就是每个Partition所在的块的位置。按照“移动数据不如移动计算”的理念，Spark在进行任务调度的时候，会尽可能地将计算任务分配到其所要处理数据块的存储位置。

Spark的wordcount RDD计算过程如图4-1所示。



图4-1 Spark wordcount计算过程

### 4.2.3 系统的网络架构

系统整体集群网络架构，分为爬虫，Spark集群，Hadoop集群，Azkaban调度器，Flume日志采集工具，logStash日志集采工具，以及ES的集群，mysql数据，Tomcat采用Nginx作为代理服务器，三台Tomcat服务器提供服务，如下是前台展现web图的tomcat的架构图，如图4-2Tomcat结合Nginx架构图,如图4—3为Spark集群网络架构图，如图4-4为Hadoop集群网络架构图。



图4-2 Nginx结合tomcat网络架构



图4-3 Spark高可用网络架构



图4-3 Hadoop集群网络架构图

## 4.3 系统功能架构

系统的设计遵循了模块化设计，大体分为爬虫模块，采集模块，数据清洗模块，ES搜索模块，数据下沉HDFS模块，脚本调度平台模块，数据分析模块，数据展现模块。其中爬虫模块负责不同业务数据采集，采集后的数据通过Log4j对接Flume以及logstash下沉的HDFS以及ES，Azkaban调度Scala代码提交到Spark集群处理数据后的数据导入到mysql呈现数据。系统功能模块图如图4-4所示。



图4-4 系统的功能架构图



图4-5 系统功能模块图

### 4.3.1 爬虫模块

随着网络的发展，万维网成为了信息的主要载体，如何有效的提取有用的信息，为了开展某些业务，网络爬虫应用而生，网络爬虫是有目的性的，有选择性的抓取特定的数据，随着计算资源的强大，现代新型的爬虫也产生了，分布式爬取基本上取代了单机爬取，分布式爬虫最大的好处在效率上明显提高，除此外避免的访问速度过于快触发了对方的反爬虫策略，分布式爬虫本身就有良好的代理IP资源池。本系统采用的非分布式，单机式爬取机制。

系统中爬虫模块分为以下几个部分，网络访问部分，线程调度部分，代理IP资源池部分，启动程序部分。

1. 网络访问模块：其中分为不同的数据采集目标，有个人中心用户数据采集，音乐数据采集，评论数据采集。
2. 线程调度模块：采用生产者消费者的模式调度线程，主程序向线程池中添加线程判断线程池是否线程池已经满了，如果满了就暂停向线程池中添加线程，网络程序继续消费线程。
3. 代理IP资源池，采用一个新的线程从第三方网站抓取IP地址，由于第三方网站的IP地址的质量很低，所以没有办法过滤代理IP是必须所要做的。同时 考虑到第三方网站（我这里采用的是西刺代理）西刺代理网站对于访问也做出了限制，这里采用定时器线程池，每个十分钟重新更新一次代理IP资源池中的IP数据。
4. 启动程序模块：不同的数据抓取启动不同的程序。

系统的爬虫的整体模块结构图如图4-6所示：



图4-5爬虫结构图

### 4.3.2 数据下沉模块

Flume和LogStash采集文本数据对接HDFS和ES，为了考虑到数据稳定性，Flume采用通过Log4j远程Socket接受数据，LogStash也是采用Socket接受数据对接到ES及HDFS。Flume内部由agent传递数据，agent内部有三个组件

1. Source：采集源，用于跟数据源对接，以获取数据
2. Sink： 下沉地，采集数据的传送目的，用于往下一级agent传递数据或者往最终存储系统传递数据。
3. Channel：agent内部的数据传输通道，用于从source将数据传递到sink。

下图为Flume的连接HDFS的结构图：如图4-6：



图4-6 Flume连接HDFS内部结构图

### 4.3.3 数据分析模块

Spark作为主要的数据分析框架，系统分别使用到了SparkRDD,SparkSQL，SparkML，DataFrame，Graphx，SparkBroadCasts，数据分析模块同时配合Azkaban，Azkaban提交代码到Spark集群运行，由于Spark的底层大部分都是依赖RDD进行迭代式计算。如下对各个分析功能做出简要分析：

1. SparkSQL：我们都知道已经有Hive分析工具，Hive SQL转换成MapReduce然后提交到集群上执行，大大的简化了编写MapReduce程序复杂性，但由于MapReduce这种计算模型执行的效率比较慢，所有SparkSQL应运而生，他讲SparkSQL转换成RDD，然后提交到集群上执行，执行效率非常快。在分析数据时，分析评论词云的时候，首先创建SQLContext，SQLContext是通向SparkSQL的入口，StructType为数据库映射的到mysql数据库中的。SparkSQL在读取完HDFS后，处理之前先对mysql数据中的相应的数据库表数据清空，然后运行运行重新导入。
2. DataFrame：提到DataFrame我们不得不提SparkRDD，DataFrame是从RDD到DataSet发展而来的，与RDD类似，DataFrame也是一个分布式数据容器。然而DataFrame更像传统数据库的二维表格，除了数据以外，还记录数据的结构信息，即schema。同时，与Hive类似，DataFrame也支持嵌套数据类（struct、array和map）。从API易用性的角度上 看，DataFrame API提供的是一套高层的关系操作，比函数式的RDD API要更加友好，门槛更低。由于与R和Pandas的DataFrame类似，Spark DataFrame很好地继承了传统单机数据分析的开发体验。如图为dataFrame使用方式



图4-7 DataFrame映射方式

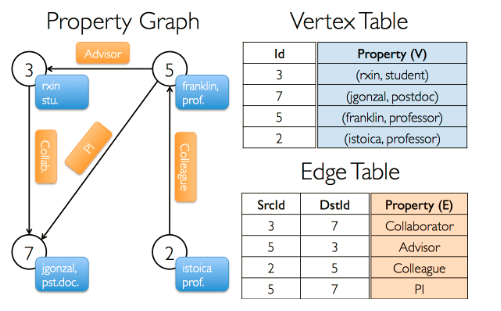
（3）Graphx：GraphX是Spark中的一个新组件，用于图形和图形并行计算。在高层次上，GraphX通过引入一个新的图抽象来扩展Spark RDD：一个有向多图，每个图的属性都附加在每个顶点和边上。为了支持图计算，GraphX公开了一组基本运算符（例如子图，joinVertices和aggregateMessages）以及Pregel API的优化变体。此外，GraphX还包含越来越多的图形算法和构建器，以简化图形分析任务。PageRank起初始于Google对网站网页的排名采用的计算方式，后来被用于各大社交网站分析用户的比例权重，PageRank用于测量图中每个顶点的重要性，假设从u到v的边代表v对u的重要性的认可。例如，如果Twitter用户跟随其他许多用户，则用户将被排名很高。 GraphX带有PageRank的静态和动态实现，作为PageRank对象的方法。静态PageRank运行的迭代次数固定，而动态PageRank运行直到队列收敛（即停止变化超过指定容限）。 GraphOps允许直接调用这些算法作为Graph上的方法。 GraphX还包含一个我们可以运行PageRank的示例社交网络数据集。

图4-8 图计算关联用户

（4）SparkBroadCasts：Spark在提高计算速率优化计算速度方面Spark采用了广播变量，大量的数据在和小型的数据进行聚合和分析是，Spark做出了将较小的数据集进行广播，分发到各个集群节点。

图4-9 广播变量广播评论数据

### 4.3.4 数据展现模块

数据展现模块采用框架AmazeUI,数据可视化采用echarts和heighcharts,其中数据呈现分为人物画像分析，图计算模块，评论，音乐模块，用户模块，机器学习模块，评论模块的词云采用了GitHub开源的项目，其中在用户的星座分布采用了heighcharts其他全部采用了echarts，整体风格大方，携带有网易云音乐的的红色风格。后台采用Spring+SpringMVC+Mybatis框架进行到MySQL数据库查询。

## 4.4 数据展现模块数据库表设计及ES实体设计

### 4.4.1 数据库的实体ER图

数据展现模块的数据库表分为十六张表如图下图所示：



图4-10 web展现平台数据库表

### 4.4.2 ES实体映射设计

ElasticSearch用于热门评论搜索，es作为一种数据库数据源也是需要设计，但是本身es自身携带@timestamp,@version,由于设计过程中需要去除这些字段，因此需要重新设计，通过评论关键字搜索评论同时搜索到这首歌，es实体中需要评论内容，评论时间，评论人，评论点赞量，评论这首歌歌曲的ID，歌曲的歌手，设计实体图如下：



图4-11 ES实体设计图

# 第5章 系统详细设计与实现

在本章将在第二章和第四章的基础上对系统做出更加详细的介绍，对系统中具体模块展现在本论文中，而不是仅仅浮现于文字本身，总体来说整个系统大体分为数据存储，数据分析，数据呈现几大部分 。

数据存储则采用HDFS及ES，在本机虚拟机上运行4台Centos虚拟机，数据分析Spark采用3台虚拟机，数据呈现运行与实体机Tomcat应用中，数据呈现和数据的分析都是采用笔记本，考虑到是离线分析，不需要时效性，只需要能够运行分析即可，没有必要租用云平台。

## 5.1 UI页面设计

### 5.1.1 功能简介

数据展现的系统UI界面直接关系到领导和网站决策人员的喜好度，UI设计的美观显得尤为重要，系统采用的响应式布局，移动端和电脑端全部适配，考虑到有时候不方便电脑端查看，如果没有响应式布局，手机端需要手动放大，给使用人员带来诸多不变。

### 5.1.2 设计思路与实现过程

采用AmazeUI实现了跨屏浏览，AmazeUI本身后台就是依赖Bootstrap，系统开发中首先引入Bootstrap所需要依赖的JS文件以及CSS文件然后，当然在Bootstrap之前 必须引入jquery.js。所有的库全部依赖jquery.js，AmazeUI本身自己开发了自己一套栅格系统，对于页面的响应式布局更加方便地操作。同时AmazeUI提供了大量的CSS，JS组件和插件。为开发人员开发更加友好的界面提供了极大的方便。系统同时采用了echarts生动的画面展现。

### 5.1.3 界面截图

电脑端chrome浏览器访问展现平台的首页如图5-1，移动端访问展现平台如图5-2。



图5-1 电脑端chrome浏览器首页图

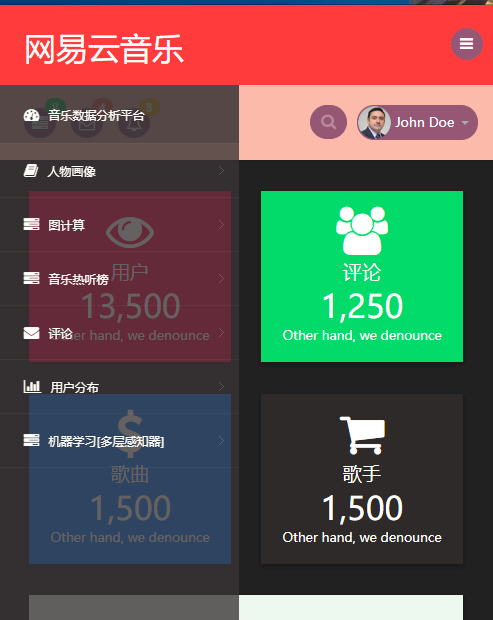


图5-2 移动端访问展现平台效果图

## 5.2 数据存储模块

### 5.2.1 功能简介

数据存储模块主要负责数据的可靠性存储和数据的检索，HDFS负责数据存储，ES负责数据的搜索，爬虫模块采集日志数据根据文件的大小128M一个一个以滚动的模式生成文件。由于Hadoop2.x分片大小为128M，所以滚动文件的大小128M最适应。

如下我们采用三种方式存储数据：

（1）日志服务器数据较小、压力较小，可以直接使用shell命令把数据上传到HDFS中（2）日志服务器数据较大、压力较大，使用NFS在另一台服务器上上传数据

（3）日志服务器非常多、数据量大，使用flume进行数据处理

如下图5-3为hadoop存储数据的DataNode的web监测平台，图5-4为Hadoop存储相关业务数据评论数据，用户数据，歌曲数据。

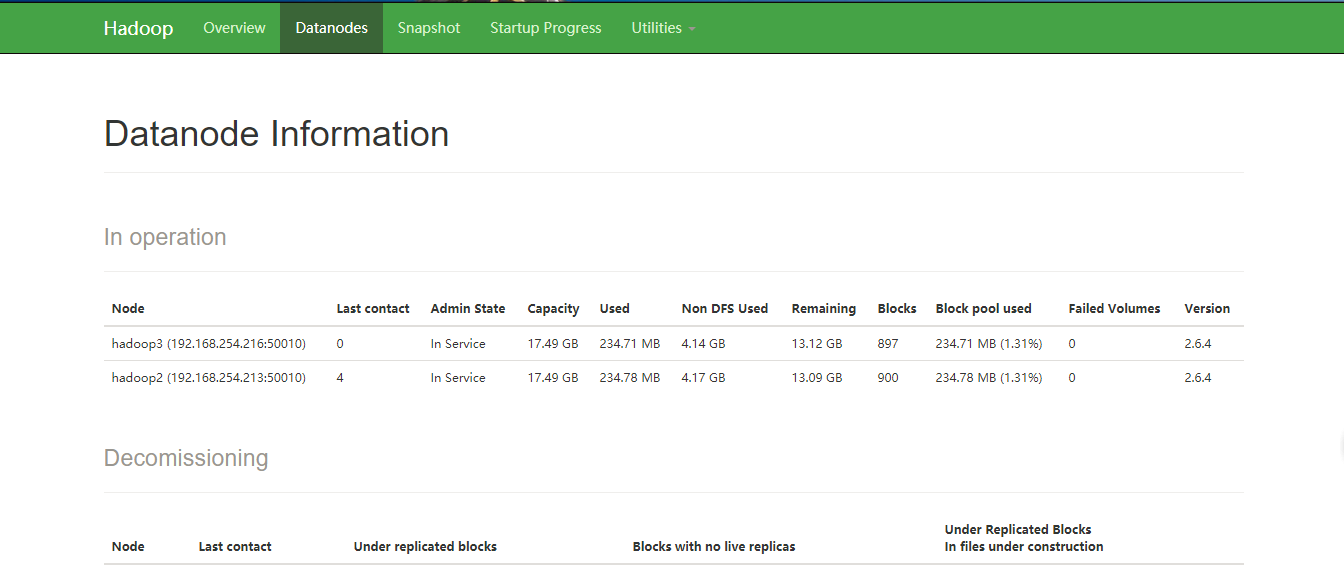


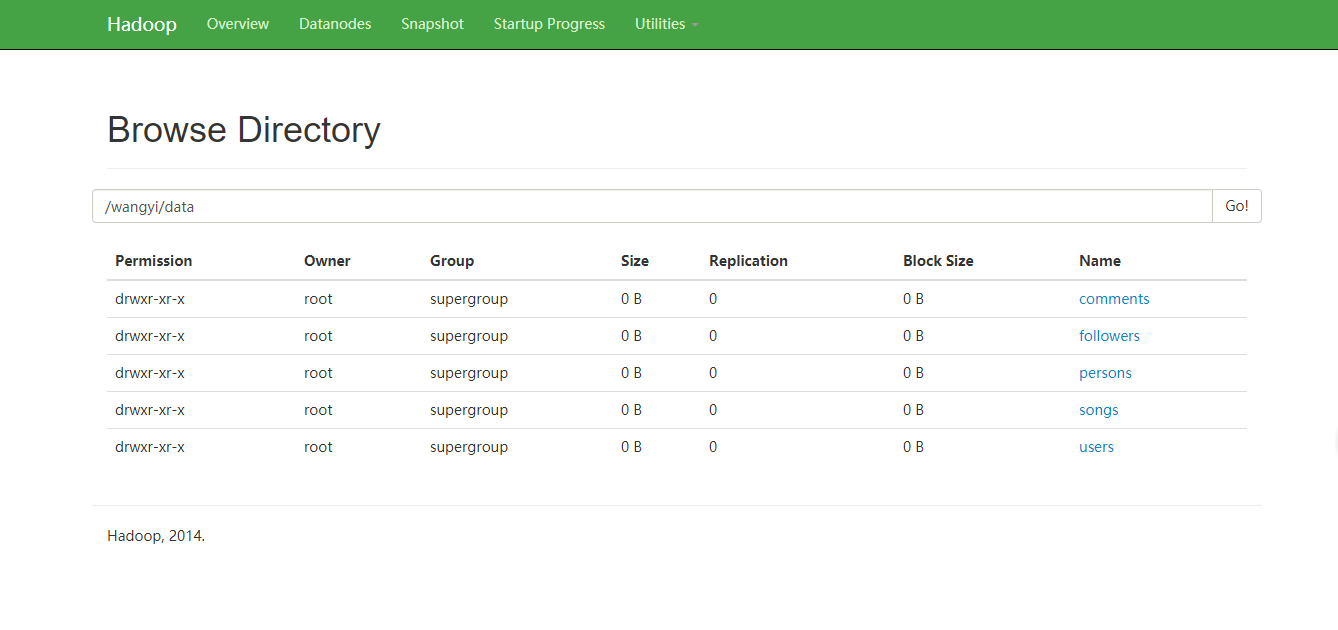
图5-3 Hadoop存储DataNode监测平台  


图5-4 Hadoop存储相关业务数据

### 5.2.2 设计思路与实现过程

存储模块作为整个系统中比较重要的一部分，系统最为核心的就是数据，但是数据的传输和数据安全问题是一个关系到网站的数据安全问题，不可以忽视，但是对于大数据我们关心的是数据的是否大部分到达了我们预期的存储的位置，数据是可以忍受数据少量丢失的。下面为Flume对接HDFS配置。Flume去硬盘采集数据通过内存转储到HDFS,没十分钟滚动一次，滚动的大小为20M。

|  |
| --- |
| # Name the components on this agent  a1.sources = r1  a1.sinks = k1  a1.channels = c1  a1.sources.r1.type = exec  a1.sources.r1.command = tail -F /wangyi/data/comments/comments.dat  a1.sources.r1.channels = c1  # Describe the sink  a1.sinks.k1.type = hdfs  a1.sinks.k1.channel = c1  a1.sinks.k1.hdfs.path = hdfs://hadoop1:9000/wangyi/data/comments/%y-%m-%d/%H  a1.sinks.k1.hdfs.filePrefix = comments-  a1.sinks.k1.hdfs.round = true  a1.sinks.k1.hdfs.roundValue = 10  a1.sinks.k1.hdfs.roundUnit = minute  a1.sinks.k1.hdfs.rollInterval = 3  a1.sinks.k1.hdfs.rollSize = 20  a1.sinks.k1.hdfs.rollCount = 5  a1.sinks.k1.hdfs.batchSize = 1  a1.sinks.k1.hdfs.useLocalTimeStamp = true  a1.sinks.k1.hdfs.fileType = DataStream  a1.channels.c1.type = memory  a1.channels.c1.capacity = 1000  a1.channels.c1.transactionCapacity = 100  # Bind the source and sink to the channel  a1.sources.r1.channels = c1  a1.sinks.k1.channel = c1 |

### 5.2.3 界面截图

后台终端数据查询HDFS上的数据，可以观察上传的关于用户的数据。从目录的角度我们可以观察到根据时间以及时间间隔滚动生成的数据样式。数据的样式是根据爬虫预订的方式存储的。如下图为终端观察到HDFS中的用户的数据。

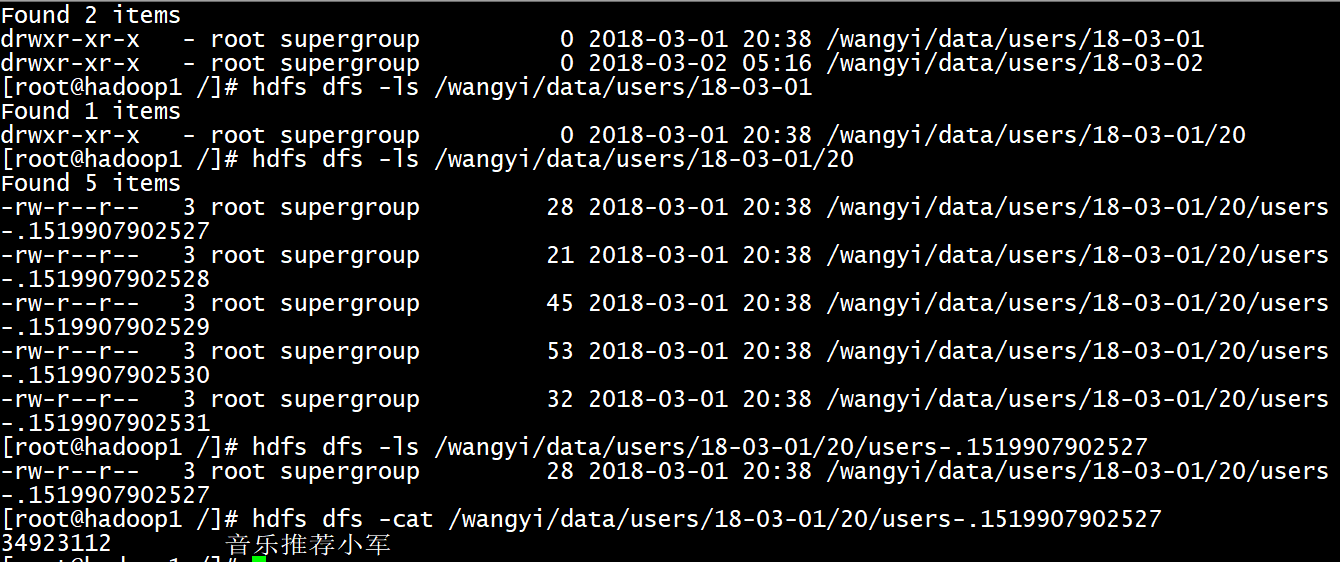


图5-3 xhell终端HDFS数据

## 5.3 数据分析

### 5.3.1 功能简介

数据分析是为系统最为重要的核心功能，采用Spark提供数据分析平台，Spark提供了R，Python，Java，Scala的用户接口。用户可以根据自己的业务编写R程序，Python程序以及Scala程序提交到Spark集群上运行。数据分析模采用scala语言编写分析程序。其中根据图计算分析用户权重用户的三角形数量，评论的词云，评论时间段，用户的年龄分布，用户的星座分布，全国地理分布，用户性别比例分布，评论分类，歌曲类型预测，以及评论top榜。热歌排行榜，人物画像分析。Spark数据分析时，打包成为jar文件采用azkaban提交到Spark，采用Spark submit提交程序提交，如果提交成功程序打印log日志可以观察到处理完成的数据已经完全导入到mysql数据库中。

### 5.3.2 设计思路与实现过程

数据分析模块设计到spark技术生态圈中的图计算，DataFrame,RDD，DataSet，Spark mllib，SparkSQL，广播变量。下面分为几个小模块依次分析是如何设计和实现。

1. 评论词云涉及到Spark最基本的RDD计算，如图5-4是数据预处理前数据样式。第一列为歌曲的ID，第二列为评论的ID，第三列为评论时间，第四列为评论处理完毕后分词的数据。第四列中的分词后的数据使用制表符进行分割方便spark在处理时使用制表符分割。首先某个词语出现一次记为一次，将出现的词语作为键然后根据键累加值。程序大体逻辑为典型的mapReduce的wordCount程序。导入mysql数据采用了SparkSQL导入。

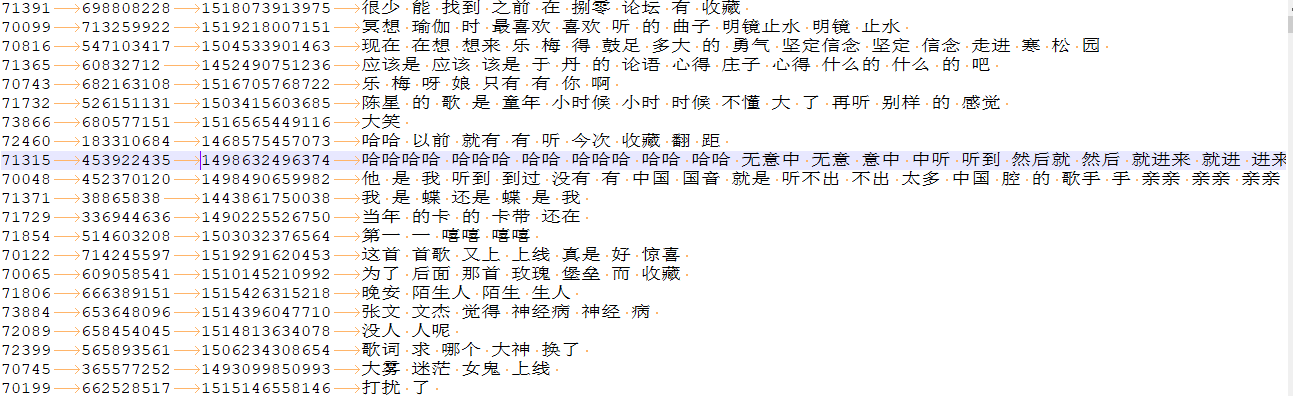


图5-4 预处理数据样式

1. 用户权重数据分析涉及到了Spark-Graphx，图计算在社交网站应用十分广泛。本系统其中一项分析用户的权重。根据用户的粉丝量和用户关注其他的数量确定该用户是否为活跃用户对于音乐网站的权重是否大，定期可以为改用户提供一定福利，吸引更多的用户。数据采用样式根据两部分为文件，follows.dat数据，users.dat数据文件。如图5-5为follows数据样式，5-6位users数据样式，follows为用户ID关联的关系，users为用户的ID以及用户的昵称。采集两种数据交由运行pageRank执行join操作使用SparkSQL导出处理完毕的数据到mysql数据库中，导出后mysql数据样式如图5-7所示。



图5-5 follwers数据样式



图5-5 users数据样式



图5-6 数据库数据样式

1. 评论时间段数据分析采用了Spark中广播变量，网易云音乐数据评论时间段预处理数据有评论时间和评论内容评论ID，同时评论时间处理时需要和预先写好的时间进行join操作，考虑到预先写好的时间段文件大小非常小，可以将其进行广播，广播到其他的集群节点上，然后各个集群节点和预先定义好的数据进行join，这样可以极大的加快运行的速度，Spark在运行中，Driver节点会首先分发小部分数据作为广播数据到其他节点，其他worker接受到广播数据开始运行。设计的评论广播数据如图5-6。



图5-6 广播变量数据样式

### 5.3.3 界面截图

用户权重的处理后报告平台展现的数据如下：



图5-7 图计算用户权重TOP

评论时间段处理后在报告平台展现如下：

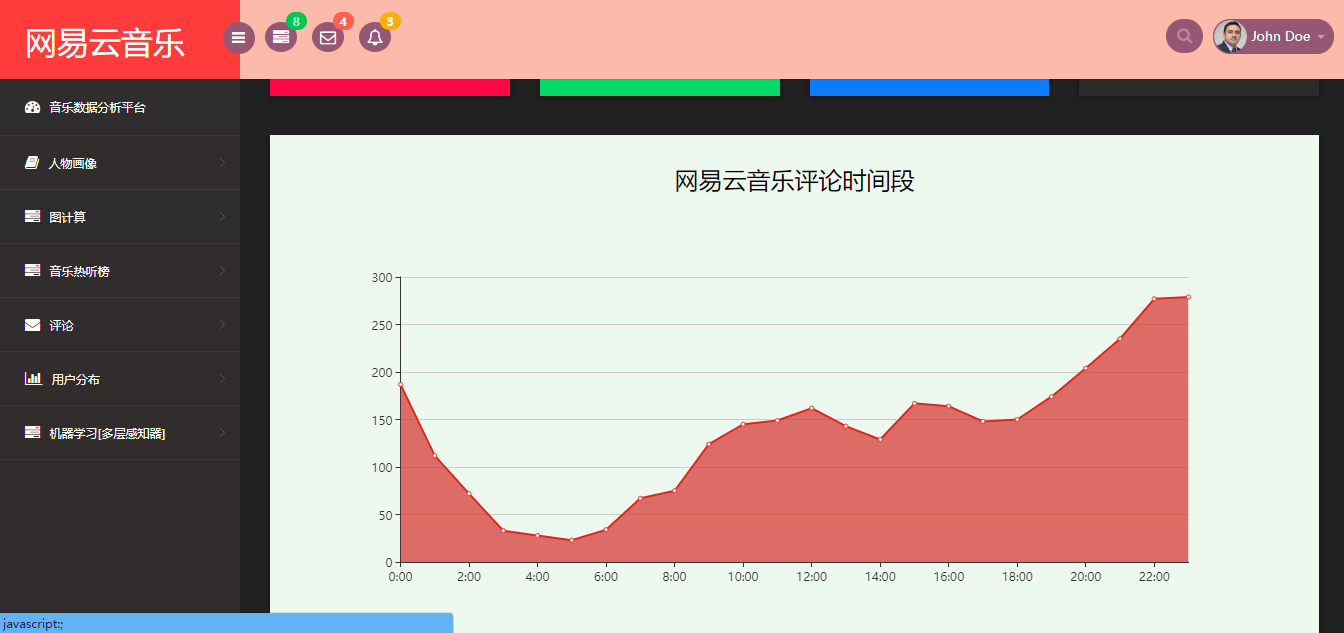


图5-8 网易云音乐评论时间段

通过Spark处理完毕后评论词云展现的数据如下截图：



图5-9 网易云音乐评论词云

# 第6章 系统测试

## 6.1 测试目的

系统在开发中，测试是非常重要的。我们在开发软件的过程中，采用了很多保证软件质量的方法和技术，但是在开发软件过程和开发完毕后依然存在大量的错误和缺陷，对于大型的软件和软件架构设计更是如此，因此：频率高的软件测试和严格的软件测试对于保证软件的质量以及稳定性具有非常重要的意义。在测试软件的过程中，我们的目的就是尽可能多发现软件开发的缺陷，从客观上讲测试的复杂性和软件的规模大小离不开，如果可以及早的排除这些错误，就可以为后期的开发和工作减少不必要的麻烦，也可以避免因此带来的巨大的代价，从而极大的提高了系统的开发的效率。具体来说当你写了一部分代码是，你就有必要进行用例测试，主要别人在使用你写的代码的时候发现BUG，也好调试。直接找到问题的所在。总的讲测试目的就是确保软件的质量。

## 6.2 测试原则

系统测试需要按照一定得原则展开，由于系统的开发规模庞大，各个组件的搭建和预运行测试，都是有必要的，系统根据以下的几点原则进行测试。

（1）从整体架构测试，由于系统使用了大量的开源项目，各个模块所使用的项目版本存在兼容性问题，在前期软件兼容性是测试一项重要的原则。

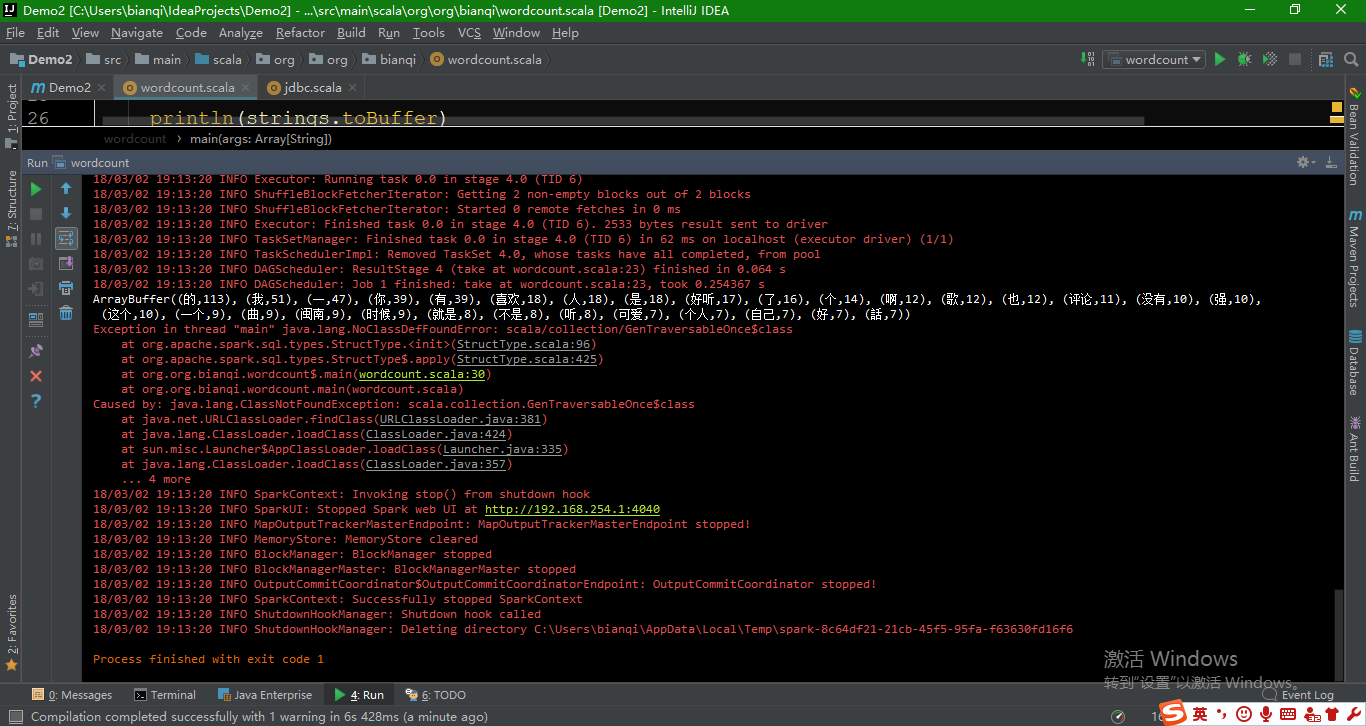
（2）系统开发中爬虫模块全部为自己编写各个组件需要经过单元测试，才可以整个到整个代码中，在开发中单元测试是非常重要的一部分，代码互相依赖度高，后期代码高度依赖 前期的代码。单元测试必不可少。

（3）系统上线测试试运行，系统部署与Linux操作系统上，在Vmware中运行，避免不了出现各种网络软件启动错误，测试各个模块数据传递是否正常也是测试非常重要的一部分。

## 6.3 系统测试的实施

### 6.3.1 架构测试

本系统进行软件测试的环境采用了Spark2.1.1，但是采用maven依赖SparkGraphx时，由于Spark2.1.1仅仅支持Spark-Graphx2.1，对于Spark-Graphx2.0不支持，在编译时为出现异常，运行才可以看到错误。如图6-1所示。调整Spark版本和Spark-Graphx版本兼容，系统运行代码时，测试通过。

  
 图6-1 Spark兼容错误图

为了方便系统拷贝直接拷贝虚拟机的目录，但是在系统启动后，在网卡启动失败，如图6-2为网络启动失败。解决办法移除克隆之后的网卡信息，重启机器然后系统会自动生成新的网络适配器，ifconfig查看自己的HWaddr，然后把HWaddr写入到eth-0网卡中，重启网卡启动成功，观察到IP地址分配成功。

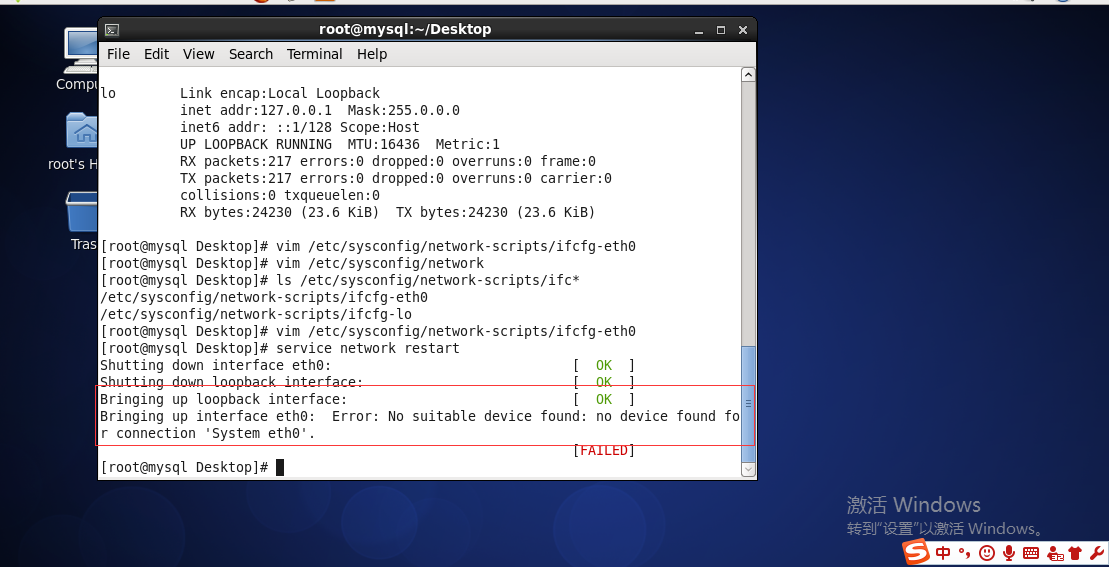


图6-2 网络启动失败

### 6.3.2 数据传输测试

数据在传输和落地之间，会存在各种各样的问题，譬如编码问题数据保存问题，数据传输稳定性问题等等，互联网数据避免不了数据会存在各种各样的数据格式，在传输过程以及保存到MySQL数据库中，我们会发现数据库编码有UTF8，以及UTF8mb4，现代许多表情符号都会使用到新型编码。在保存到MySQL数据库中，我们发现如下错误，如图6-3。解决方式去除如下特殊的编码，成功解决编码问题。

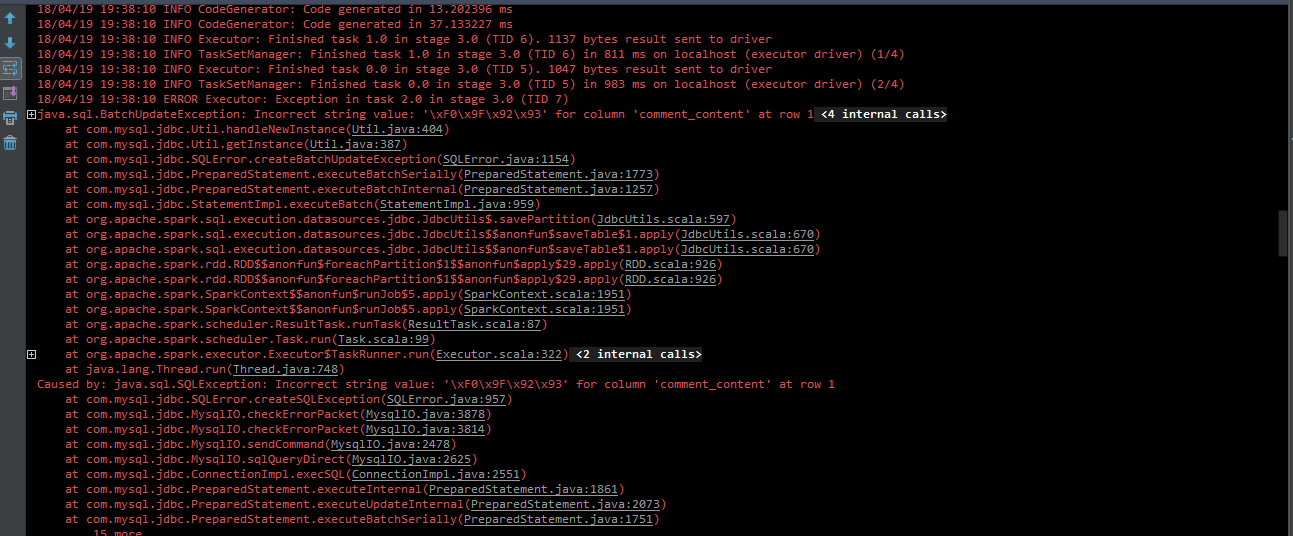


图6-3 编码错误

Jsp页面在测试过程中出现了如下错误，如图6-4，jsp页面的编码出现问题，初步判定是jsp页面仅仅支持UTF-8支持的编码的字，出现了UTF-8无法兼容问题，去除如下莫名其妙的字符，如图6-5。错误消失。测试通过。

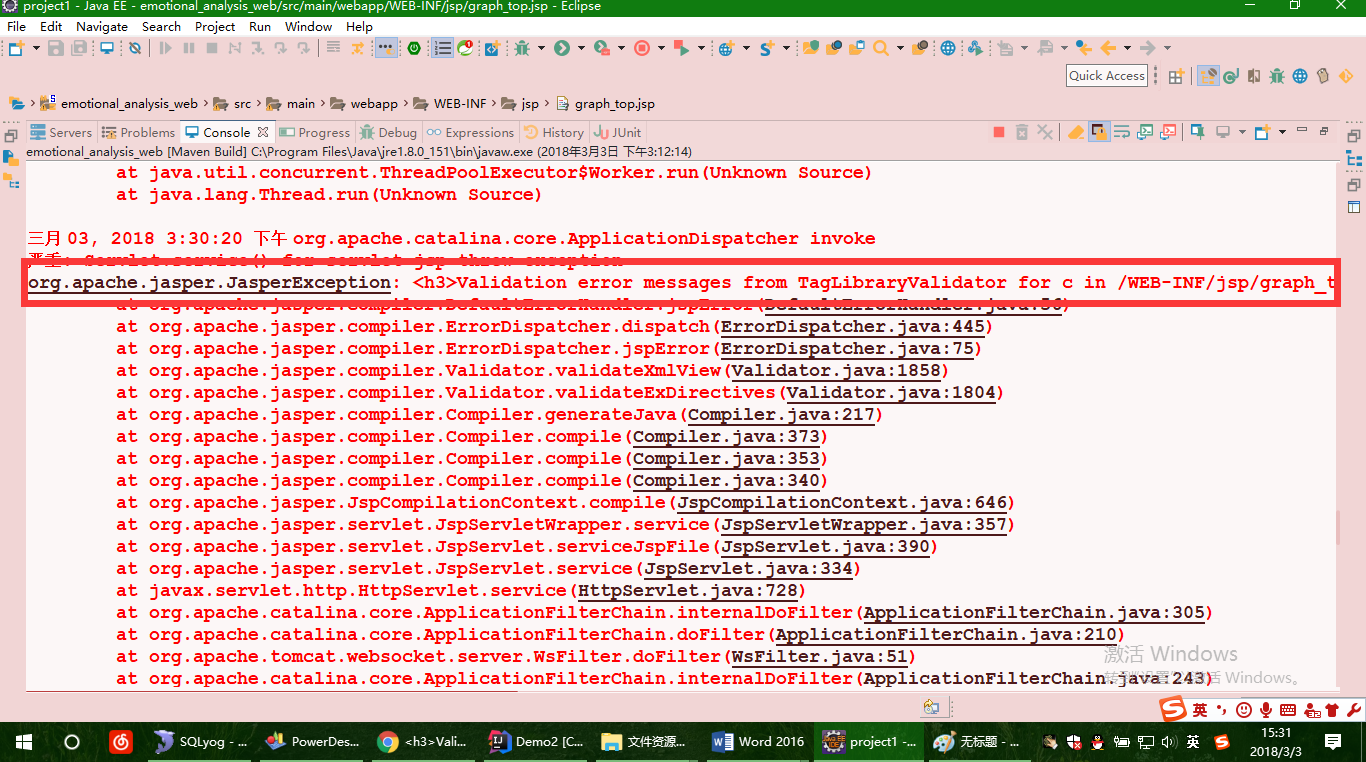


图6-4 jsp页面编码错误

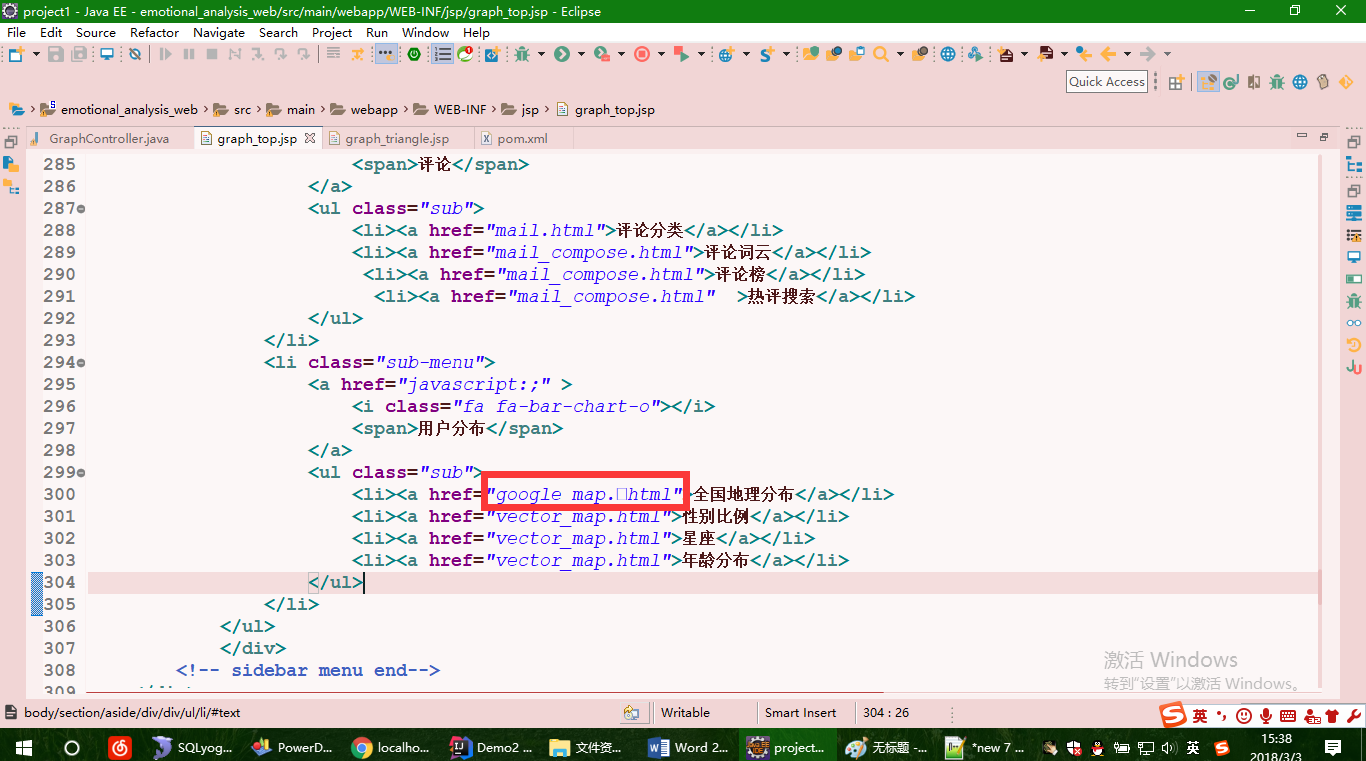


图6-5 jsp页面错误定位

### 6.3.3 系统预发布测试

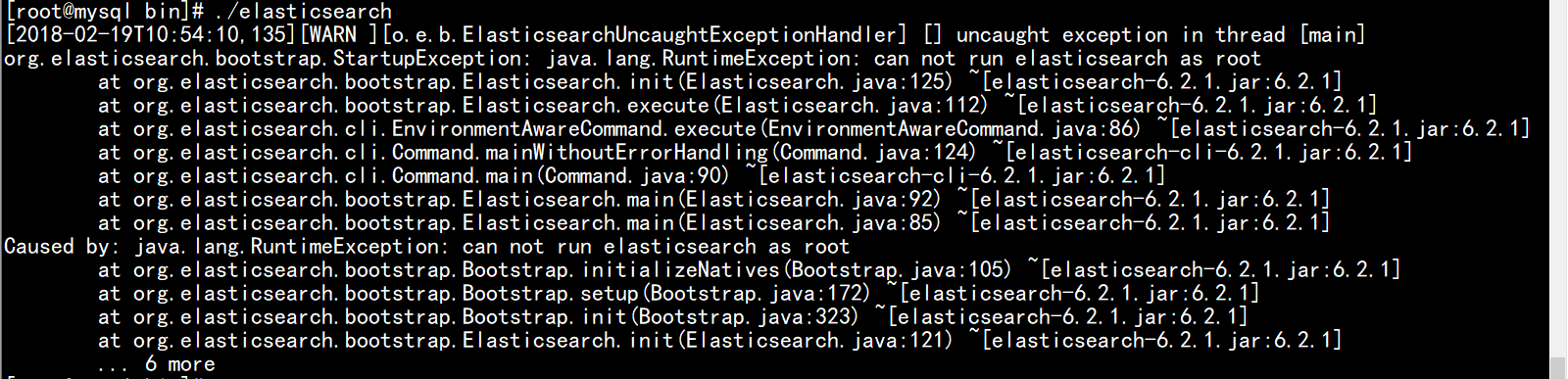
系统预发布即为系统即将上线，最后的部署测试，再次考虑到系统的安全性，ES专门做了安全问题考虑，不允许使用权限的用户启动ES，在启动过程中出现如下错误。如图6-6。ES3.x修改配置文件允许root身份启动ES，即可，ES高版本已经不支持修改配置文件启动。

图6-6 ES以root启动报错

# 结 论

本系统采用爬虫抓取数据，Flume以及LogStash采集数据，HDFS和ES存储数据，Akaban调度Scala脚本在Spark计算平台上运行，将处理完后的数据导入到MySQL数据库中，最后在WEB平台上展现。各个组件组件紧密衔接，展现平台采用各种业界成熟的报表框架，生动形象将企业级数据展现的淋漓尽致。但是系统也有很多的不足之处。分为以下几点。

1. 由于平台是在笔记本电脑上搭建，数据存储达不到几个T，由于硬盘的限制，后期考虑如果有资金足够可以迁移到云平台上。
2. 爬虫在抓取数据时，由于使用的是第三方平台的IP，第三方平台的IP经常是已经被拉黑的IP，以及IP质量很低。
3. Spark计算平台处理时，由于数据量低平台可以接受，如果呈现海量数据，显然计算平台计算资源是不够的。后期考虑阿里云平台购买计算资源。

# 参考文献

[1] 卡劳（Karau，H.）王道远. Spark快速大数据分析机器学习大数据算法参考书Spark作者及核心成员执笔[L]，科技信息, 2015(9):91-91.

[2] Tom White著 王海，华东，刘喻，吕粤海 译，Hadoop权威指南：大数据的存储与分析（第4版）.大数据于云计算 ,2017, 7(8):51-52.

[3] 范东来,Hadoop海量数据处理 技术详解与项目实战（第2版).计算机科学技术，2016,8(1):21-33.

[4] 王晓华,Spark Mllib机器学习实践.计算机科学技术，2015,12(34):12-22

[5] Reynold.Xin,Chief.Architect,Databricks,http://spark.apache.org/docs/latest/rdd-programming-guide.html,2012.

[6] 高凯 高莘,岳重阳,大数据搜索与挖掘及可视化管理方案 ——Elastic Stack 5：Elasticsearch、Logstash、Kibana、X-Pack、Beats （第3版),计算机开发与应用,2017:22-52.

[7] 哈里·史瑞德哈伦,马延辉,史东杰 Flume：构建高可用、可扩展的海量日志采集系统,大数据统计,2015:66-78.

[8] 凌峰,诚毅,http://echarts.baidu.com/tutorial.html%E5%B0%8F,数据可视化，2018:12-64.

[9] 杨开振,Java EE互联网轻量级框架整合开发SSM框架(SpringMvc+Spring+MyBatis)和Redis实现,计算机网络开发,2017:42-44.

# 致 谢

时光荏苒，四年的大学生活一转眼就过去了，就仿佛是昨天一样。四年太多的记忆和青春故事无法用语言讲述。一句古话讲的话，有缘千里来相会，无缘对面不相逢，感谢上天让我认识了我身边的每一位同学，是他们的鼓舞让我一路这么坚持下来。感谢思念老师的谆谆教导，成为了我黑暗中夜空中最亮的星，指引我前进。

最重要感谢四年的自己，对于每一个人最重要的是选择，最难得是坚持。感谢自己选择了齐齐哈尔大学大学，感谢自己四年选择了自己热爱的计算机并一直不断的坚持下去。最后致谢指导我完成毕业设计的各位老师。在这里诚挚的说一声你们辛苦了。谢谢。