目　录

摘　要：近些年来，随着我国信息技术的不断革新，越来越多的人热衷于从网络中获取信息、拓展人脉，社交网络取得了蓬勃发展。微博作为其典型代表，以其开放的社交环境、快速的传播机制和优秀的用户体验备受人们青睐，吸引了大量用户的参与。这些用户每天都在产生着海量的数据资源，这些数据资源背后蕴涵着丰富的学术研究价值、政府决策价值和商业参考价值。我们可以从不同维度研究和分析微博数据，将其中具有应用价值的信息挖掘出来，并将这些数据分析成果应用到实际生产生活中。因此，本文以新浪微博为研究对象，首先介绍了微博数据分析的研究和应用现状、大数据采集与分析等关键技术，接着围绕微博数据的采集、存储、分析和可视化这四大模块进行研究，运用网络爬虫、大数据框架、Spark 计算分析引擎、Flask Web 等技术对基于 Spark 的微博数据采集与分析系统进行设计和实现。最终实现的系统可根据不同的爬取条件对微博数据进行大规模爬取，并可对爬取到的微博数据进行多维度的分析以及分析结果的可视化展示。

关键词：微博；大数据；Spark；网络爬虫；数据分析

第 1 章 绪论

随着信息技术的迅猛发展以及智能手机的进一步普及，我国互联网已全面进入 Web2.0 时代，越来越多的人热衷于从虚拟网络中获取新闻信息、拓展人脉，社交网络取得了蓬勃发展。人们已经习惯于通过社交软件来从事社交活动，表达自己的观点和情感，并与其他人进行互动与分享。这些社交平台节约了社交时间和物质成本，满足了人们不同的社交需求，使时间、空间等因素不再成为人与人交往的障碍。

1.1 论文研究目的与意义

作为社交网络的典型代表, 微博以其开放的社交环境、快速的传播机制和优秀的用户体验备受人们青睐，吸引了海量用户的参与，其受众群体既有公众人物，也有草根平民，既有明星个人微博，也有政府机构和媒体，其内容覆盖领域广泛，囊括政治、军事、科技、体育、娱乐等诸多领域，社会影响力巨大。人们可以通过微博平台便捷地获取世界各行业的新闻信息，随时随地的抒发情感、表达自己的见解。这些用户活动每天都在产生着海量的数据资源，这些丰富的数据资源背后蕴藏着巨大的学术研究价值、政府决策价值和商业参考价值。我们可以从不同维度分析这些微博数据，将其中具有应用价值的信息挖掘出来，并把这些数据分析成果应用到政府与商业实践中，从而发挥微博数据的现实意义。

1.2 国内外研究现状

目前，微博作为中国现阶段第一大的广播式社交网络平台，曾吸引了国内许多学者的关注与研究。微博官方曾开发微博数据中心用于展示微博排行榜、微博粉丝分析、微博热点指数等不同类型的分析应用。除此之外，许多相关机构和第三方平台也参与到微博分析应用的开发中来，例如北京大学 PKUVIS、孔明社交管理和独到科技等都从不同角度设计开发了微博数据分析平台。在国外，关于社交网络的数据采集与分析一直是一个非常重要的研究方向，国外许多专家学者都曾把诸如 Twitter、Facebook 等在外国人生活中影响力较大的知名社交网络作为自己的数据采集与分析对象，并通过对此类数据的多维度分析，为国外政府以及相关企业机构提供了大量的有价值的数据集合和数据分析成果，其中比较知名的分析平台有 Sprout Social、Simply Measured 等。

然而随着大数据时代的来临，微博数据变得冗余且复杂，能实现大量获取微博数据并对其进行分析的难度越来越大。业界的主流方法开始将网络爬虫、HDFS 大数据存储和 Spark 等技术应用到微博的研究中来，利用分布式技术实现高复杂的计算任务。

1.3 关键技术综述

（1）网络爬虫

互联网发展迅速，大量的信息资源被存储在万维网中，为了能快速有效地获取这些资源，网络爬虫技术出现。它是一个可以通过一定规则和策略，自动下载网页并抽取网页中有用信息的程序，也是搜索引擎的重要技术支持。在诸多编程语言中，基于Python 的网络爬虫为我们提供了非常方便的扩展库用于实现爬虫程序，是工业和科研界数据采集技术的首选。

（2）HDFS

HDFS 是通过分布式存储技术来实现海量数据有效存储的文件系统，能提供高吞吐量的数据访问，通常可以部署在低廉的硬件设备。HDFS 一般采用主从架构，文件以数据块的形式进行存储。Namenode 负责客户端对文件的访问以及对数据块的创建和删除等操作进行管理。Datanode 作为数据节点，其主要任务是负责管理他自身节点上的文件存储，并定期向 Namenode 发送它所存储的数据块信息。

（3）Spark

Spark 是用来进行大规模数据快速处理的计算引擎。它的一个重要特点是基于内存进行运算。RDD 是 Spark 对数据的核心抽象，本质上是一个可分区、可并行计算的集合，是一种高容错性和并行的数据结构。Spark 主要通过 RDD 的创建、转换和行动来操作数据。除此之外，Spark 还可以部署在廉价的计算机集群中，具有伸缩性强、容错性高、处理迅速等特点。基于以上特点，Spark 比 MapReduce 的处理速度要快到10~100 倍左右。本文的微博采集与分析系统主要采用 Spark 来进行大数据的分析计算。

（4）Flask Web 框架

Flask 是基于 Python 语言编写的现如今最流行的 Web 框架。它核心简单，且易于扩展，非常适合中小型网站的快速开发。Flask 支持用扩展来给应用添加数据库集成、表单验证等各类功能。在 Flask 中，默认使用 werkzeug 来做路由分发，使用 Jinja2 来做模板渲染。本文系统主要采用 Flask 框架做 Web 功能的开发。

（5）ECharts

ECharts 是百度出品的开源可视化库，兼容当前绝大部分浏览器，可以提供折线图、柱状图、饼图、地图、热力图等各类形象直观、交互功能丰富的数据可视化图表。操作方便，可直接嵌入到前端页面，并与前端界面形成良好搭配。本文的微博数据采集与分析系统 Web 前端主要采用 ECharts 进行分析后数据的可视化渲染。

1.4 论文结构安排

本文主要是以新浪微博为研究对象，围绕微博数据的采集、存储、分析和可视化进行研究，运用网络爬虫、大数据框架、Spark 数据分析和 Python Web - Flask 框架等技术设计并实现基于 Spark 的微博数据采集与分析系统。论文的结构安排如下：

第一章为绪论。首先对论文研究的目的与意义进行阐述，然后对国内外社交网络的研究现状进行说明，接着对本文用到的关键技术进行综述，最后介绍了论文的结构安排。

第二章为需求分析。本章的重点工作是从用户角度出发，通过分析微博网站的设计与微博数据的特点，阐述了用户使用本系统的需求，其中包括功能性需求和数据分析需求等。

第三章为系统设计。本章的重点工作是设计基于 Spark 的微博数据采集与分析系统，具体从网络爬虫、数据存储、Spark 数据计算和数据可视化进行设计，本章将采用自顶向下的模块化思想，将系统划分为四大模块，并给出各个模块的详细设计方案。

第四章为系统实现。本章的重点工作是利用 Hadoop 大数据框架、Python 网络爬虫、Spark 数据计算引擎和 Flask Web 框架对系统的各大子模块进行实现。其中核心部分是利用 Python 网络爬虫对微博数据进行爬取，利用 Spark 计算引擎对微博数据进行分析。

第五章为系统测试。本章主要是对已经实现的系统的各大模块进行简单的功能性测试，并对系统运行时的各个模块的 Web 界面进行展示。

最后为总结与展望。本章主要是对论文的工作进行全面总结，并提出了系统设计和实现过程中的不足之处，并对接下来系统要继续研究的方向进行展望。

第 2 章 需求分析

本章主要介绍基于 Spark 的微博数据采集与分析系统的需求分析过程。具体分为系统概述、功能性需求和数据分析任务需求这三大部分。功能性需求将以模块划分的方式对每一个模块要实现的具体功能进行需求分析。数据分析任务需求按照微博数据特点制定三种不同的分析任务。

2.1 系统总体概述

基于 Spark 的微博数据采集与分析系统是一款集微博数据采集、存储、分析和可视化展示为一体的系统平台。系统涉及到多种大数据处理和分析技术的方法，包括网络爬虫、数据收集、分布式数据存储、Spark 计算分析、数据可视化等。最终面向用户使用的是一款集以上所有功能为一体的 Web 系统平台。用户可以通过 Web 页面的操作，对微博数据进行爬取、分析和可视化，从中获取用户自己想要的数据资源和分析结果，为生产和科研提供价值。

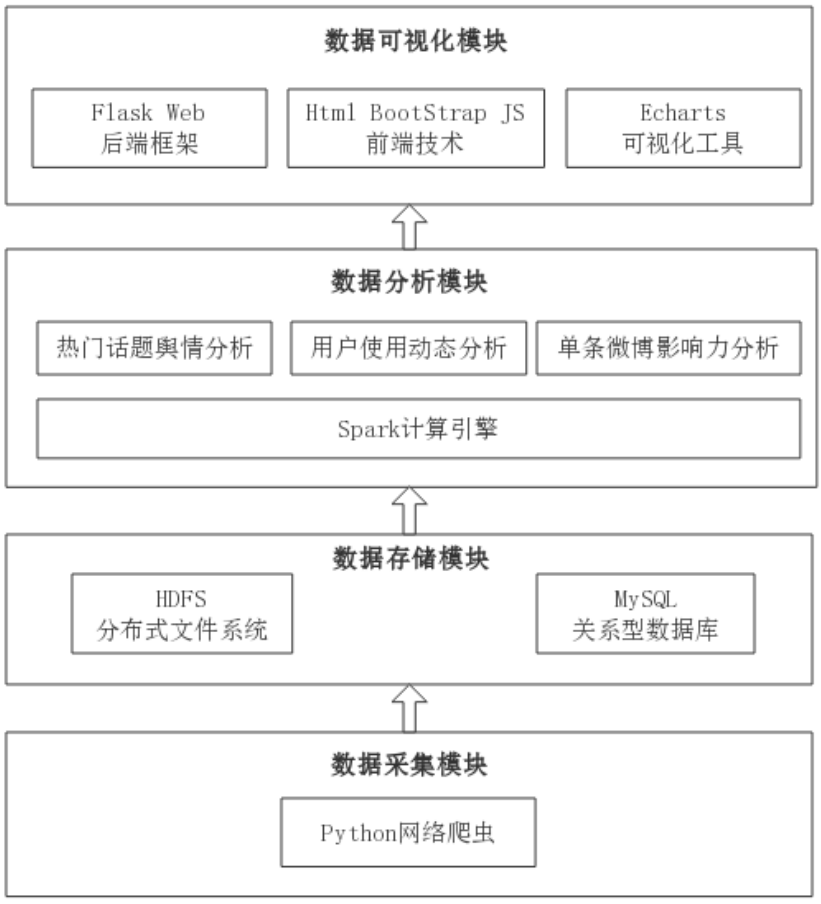

图 1 系统总体架构图

2.2 功能性需求

本系统主要面向已经注册本系统的普通用户使用。按照数据的处理流程以及一般用户的操作顺序，将本系统功能性需求化分为数据采集、数据存储、数据分析、数据可视化这四大模块。下面将从这四大模块对面向用户的功能性需求进行说明。

2.2.1 数据采集

用户在此模块里可以查看、增加、删除数据采集任务。用户增加采集任务主要通过自定义不同的搜索条件并调用爬虫程序，将该任务条件下的微博数据进行爬取。

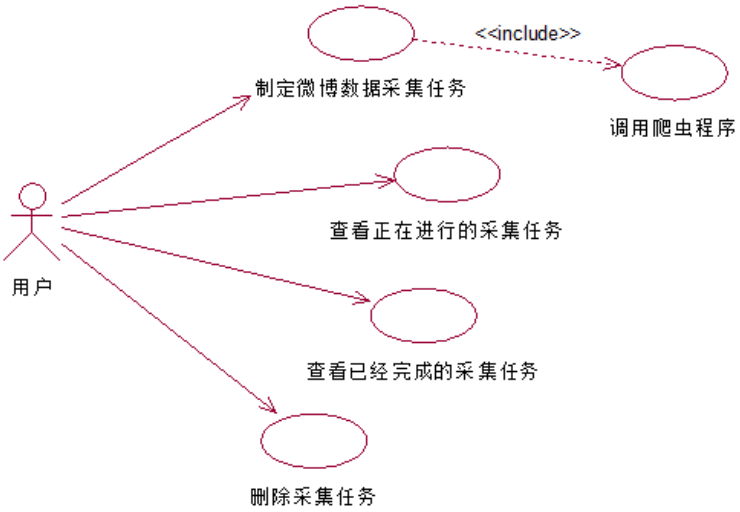

图 2 数据采集用例图

此模块是整个系统的数据源头，主要功能是负责所需微博数据的解析、下载、处理和收集。对大数据分析而言，数据量越大分析结果越精确，因此这就需要本模块能够快速、有效地获取到大量微博数据。用户可以自定义不同的爬取条件对数据进行收集。例如“通过关键字爬取某一时段下的全部热门微博以及每个微博涉及的用户信息”、“通过用户 ID 爬取用户的基本资料和所有已发微博信息”、“爬取某条微博的基本信息和全部评论信息”等。在微博数据采集的过程中，利用 Python 网络爬虫将爬取的微博数据持久化到本地或 HDFS 中。

2.2.2 数据存储

此模块主要负责用户信息、微博数据、采集分析任务信息和分析结果的存放，是整个系统的数据中转站。微博数据量巨大，需要妥善选择和设计合适的数据库对微博数据进行存储。用户信息、分析结果和任务信息涉及到的数据量较小，但需要与前端

交互或者频繁访问。所以基于以上两点需要根据其不同的任务特点选择合适的数据库。

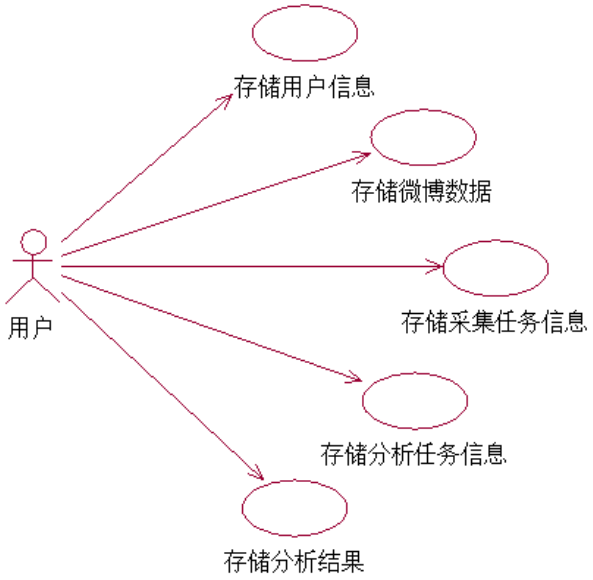

图 3 数据存储用例图

2.2.3 数据分析

用户在数据分析模块中可以查看、增加、删除分析任务。用户增加分析任务主要通过制定不同的分析条件并启动分析程序，对已经爬取的微博数据进行不同程度的分析。此模块是本系统的核心模块，负责对微博数据进行多维度下的处理分析。在分析过程中，我们需要从数据库中读取相应任务的微博数据文件，根据不同的任务特性对微博数据进行分析，最后将分析结果保存在数据库中供数据可视化模块使用。本模块连接数据存储模块和数据可视化模块，并为数据可视化模块提供所有的分析结果。

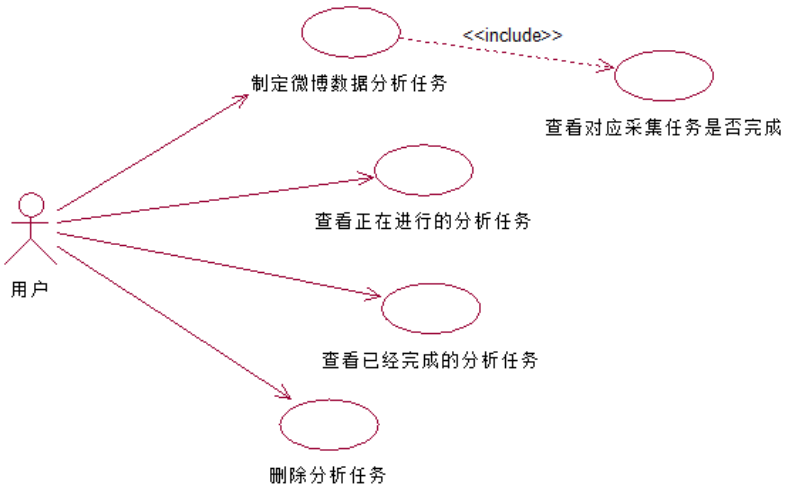

图 4 数据分析用例图

2.2.4 数据可视化

（1）用户使用概况可视化

用户使用概况可视化可以展示用户使用本系统情况的总览和明细，用户可以在此界面中查看用户自己的使用频率统计、登录总次数、详细个人资料、数据收集和分析的总次数以及详细的用户操作历史记录等。用例图如图 5 所示：

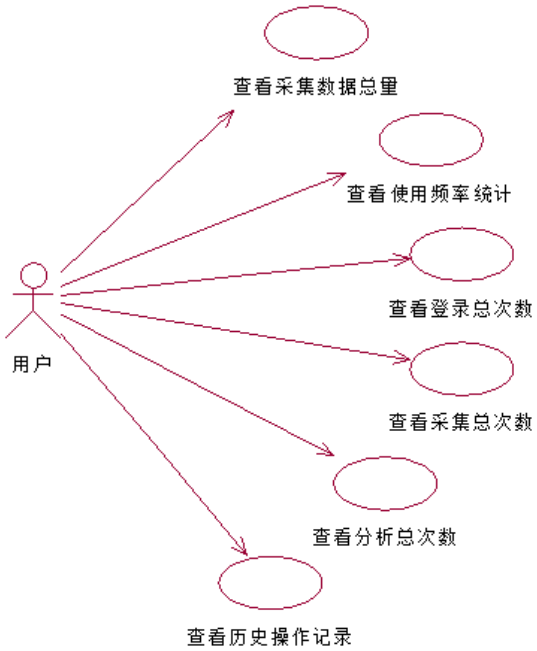

图 5 用户使用概况用例图

（2）数据分析任务可视化

用户在数据可视化模块中可以查看各项分析任务的具体情况以及经过数据分析后所生成的各项条件下的数据可视化的报表。用例图如图 6 所示：

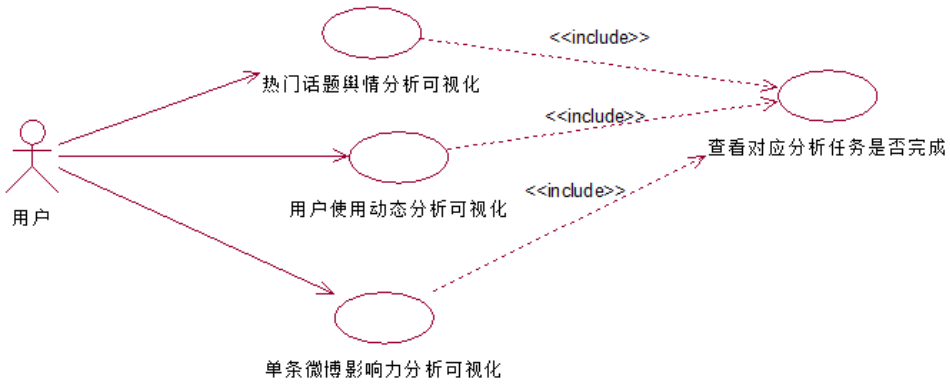

图 6 数据可视化用例图

此模块是本系统与用户的使用交互模块。主要内容是负责数据分析结果的对外可视化展示以及面向用户的图形界面操作，其实质是一个多功能的 Web 系统，为用户提供登录、注册、用户概况展示、用户资料修改、操作数据采集和数据分析模块的功能。可视化功能的各类图表通过前端可视化工具实现，将分析结果转化为各类形象、直白的柱状图、饼状图等。此模块是本系统中最终能够体现数据分析效果的重要模块，也是将整个系统各个子模块统一整合的用户主要的操作模块。

2.3 数据分析任务需求

根据微博数据的特点，本系统将会制定几个特有的分析任务，目前设计了“热门话题的舆情分析”、“用户使用动态分析”和“单条微博影响力分析”等三个典型的分析任务需求。下面将对这三个分析任务进行具体的需求说明。

2.3.1 热门话题的舆情分析

微博一直以来都是中国网民热门话题的聚集地。很多热门事件和重磅消息都通过微博热搜功能第一时间被大众所知。通过对微博热门话题进行舆情分析，有助于我们了解社会热门事件的传播动向、受众范围。

微博的热门话题一般都是由某一个关键词在一定时间内被发布和搜索的微博数量所决定。设计热门话题的舆情分析，其主要方法是首先为数据采集模块设置特定的热门关键字和时间段，对基于这个时间段的关于这个关键字的热门微博进行爬取，爬取的每条微博内容包括微博内容、发布此微博的用户信息、微博转发点赞评论数、微博发布时间等。然后对这些微博数据进行清洗和分析，其分析任务包括“此关键词下的微博数量”、“涉及用户数”“微博点赞数分布”、“微博转发数分布”、“微博评论数分布”、“涉及用户性别分布”、“涉及用户地域分布”、“微博发布时间分布”和“关键词词频、词云分析”等。

2.3.2 用户使用动态分析

微博作为中国使用率最高的社交应用之一，其活跃用户规模一直处于上升态势。对微博用户的使用动态进行分析，可以了解用户使用微博的规律和生活习惯，有助于商业平台针对不同的用户特点投放其感兴趣的广告，达到流量变现的价值。

设计用户使用动态分析，其主要方法是首先对用户的基本资料信息进行全方位的爬取。爬取内容包括用户名、用户性别、粉丝数量、关注数量、全部微博内容等。然后对这些微博数据进行清洗和分析，其分析任务包括“粉丝数”、“微博数”、“关注数”、“所有微博点赞数量”、“所有微博转发数”、“所有微博评论数”、“微博发布时间分布”、

和“关键词词频、词云分析”等。

2.3.3 单条微博影响力分析

微博作为中国开放的的社交应用之一，是政府、媒体、明星和老百姓发布信息的重要渠道。纵观微博的发展历史，很多情况下，一条有价值的微博能掀起整个中国社会一段时期内的舆论爆发和热议大潮。对某条有价值的微博进行分析，可以使我们对这条微博的影响力有一个详细的了解。

设计单条微博影响力分析，主要方法是首先对此条微博的基本信息、转发、点赞、评论信息进行爬取。然后对这些微博数据进行清洗和分析，其分析任务包括“微博点赞数量”、“微博转发数量”、“微博评论数量”、“点赞最高的 10 评论”、“评论最多的 10 位用户”和“评论中的关键词词频、词云分析”等。

2.4 本章小结

本章的重点工作是对本系统的需求分析进行了研究和说明，包括系统概述、功能性需求和数据分析任务需求三大部分。从用户使用本系统的角度出发，采取用例图的方式对面向用户的功能性需求进行展示。功能性需求主要是从四大模块入手，对每个部分所要实现的功能进行阐述。数据分析任务需求是根据微博数据的特点，制定了三种不同的数据分析任务。

第 3 章 系统设计

3.1 系统概要设计

3.1.1 数据采集模块

数据采集模块主要是对微博数据进行快速有效的爬取，收集和存储。微博数据爬取主要采用 Python 网络爬虫对微博数据进行爬取，用户可以自定义不同的爬取条件。爬取过程中采用多线程并行爬取，共同执行任务以确保微博数据的采集效率。网络爬虫负责微博网页的下载和解析，爬取时需要采用抓包工具获取微博网页接口，根据不同的爬取条件，设置不同的参数拼成可用 URL 对网页进行下载，并利用 Python 扩展库 BeautifulSoup 进行网页解析，将爬取条件下所需要的有用信息例如用户 ID、用户名、粉丝列表、微博具体内容等进行下载并进行数据处理与存储。

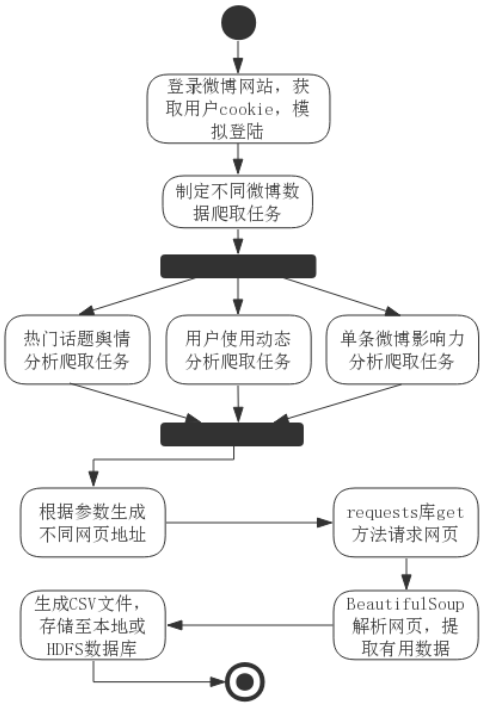

图 7 数据采集模块活动图

3.1.2 数据存储模块

数据存储模块主要负责微博数据的大规模存储以及用户信息、任务信息和分析结果的存储。针对不同的存储需求设计不同的数据库进行存储。微博数据存储主要运用到的是 HDFS。HDFS 凭借其分布式、高容错的特性可以很好的完成大规模数据的存储任务。用户数据、任务数据和分析结果主要采用 MySQL 进行存储。MySQL 体积小，

响应速度快，不需要像 HDFS 那样每次查询都需要对全盘进行扫描。使用 MySQL 为前端和其他模块提供查询和调用接口时，可以有效地减少时延。

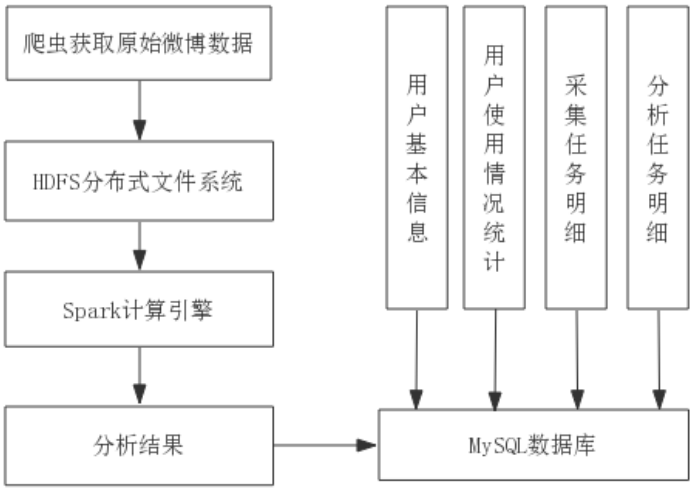

图 8 数据存储模块设计图

3.1.3 数据分析模块

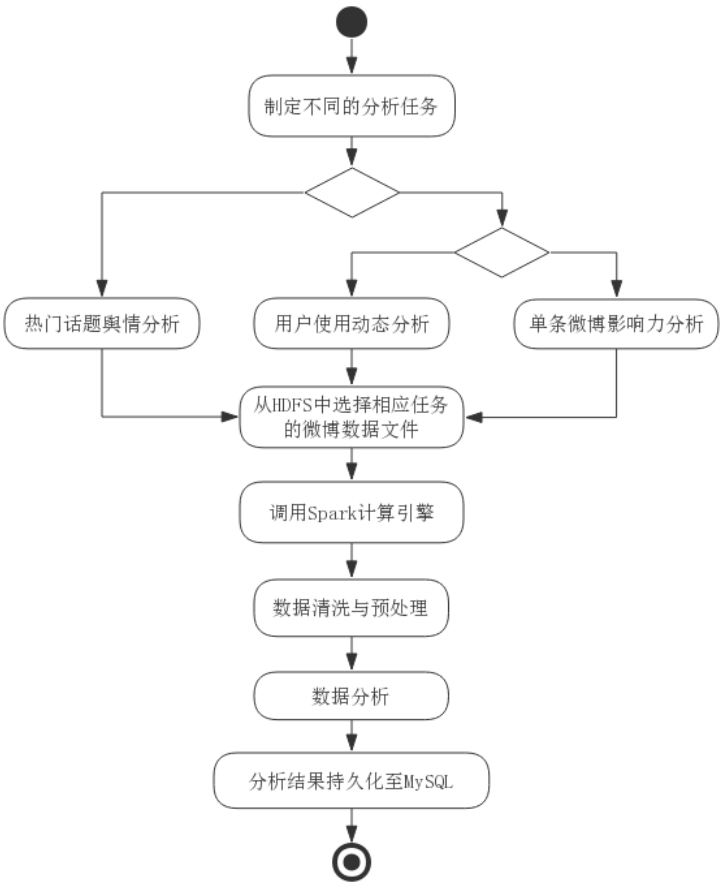

图 9 数据分析模块活动图

数据分析模块将会根据用户制定的不同的分析任务去寻找存储在 HDFS 中对应任务的原始数据文件，例如用户个人资料、微博基本信息和评论信息等，这些数据有些是冗余和错误的数据，需要对数据进行预处理，对重复数据、低质量的数据和没有意义的微博数据进行清洗。清洗之后 Spark 计算引擎会根据用户所要求的分析条件对数据文件进行数据计算与分析，分析结果将持久化到 MySql 数据库中进行存储。

3.1.4 数据可视化模块

数据可视化模块是系统与用户的交互模块。主要内容是负责数据分析结果的对外可视化展示以及面向用户的系统功能图形化操作，其实质是一个多功能的 Web 系统，用户通过登录进入本系统，通过系统的 Web 界面可以操作数据采集和分析模块的各项功能。

本模块采用 Flask Web 框架和可视化工具 ECharts 进行开发。Flask 主要负责实现 Web 系统后端的各项功能逻辑和业务流程，例如用户的登录、注册、用户使用概况展示，制定采集任务、制定分析任务、查看任务详情、查看数据分析报表和用户资料修改等功能。ECharts 主要负责优化数据可视化的效果，动态生成承载各项分析结果的报表，将抽象难懂的分析数据转化为形象易懂的多种可视化图表。此模块是本系统中最终能够体现数据分析效果的重要模块，也是将整个系统各子模块整合的用户的主要操作模块。

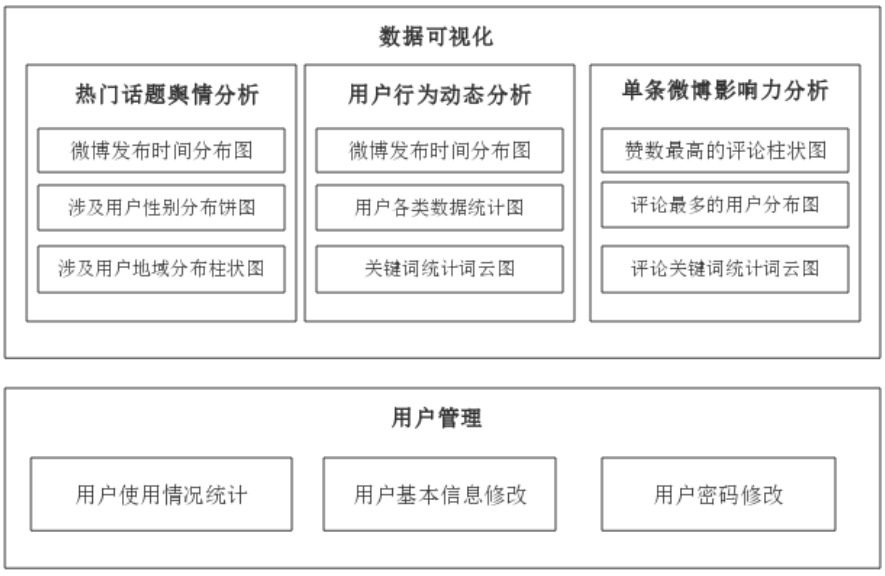

图 10 数据可视化模块架构图

3.2 系统详细设计

3.2.1 数据采集模块

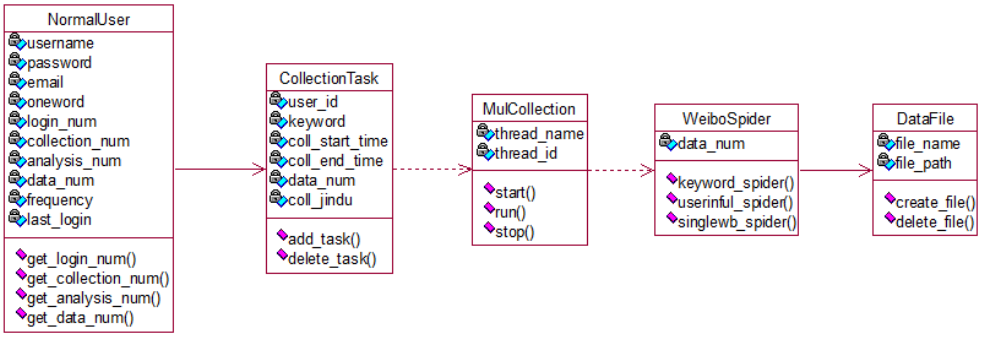

图 11 数据采集模块类图

NormalUser 为普通用户类， CollectionTask 为采集任务类，MulCollection 为爬虫线程类，WeiboSpider 为爬虫程序类，DataFile 为数据文件类。

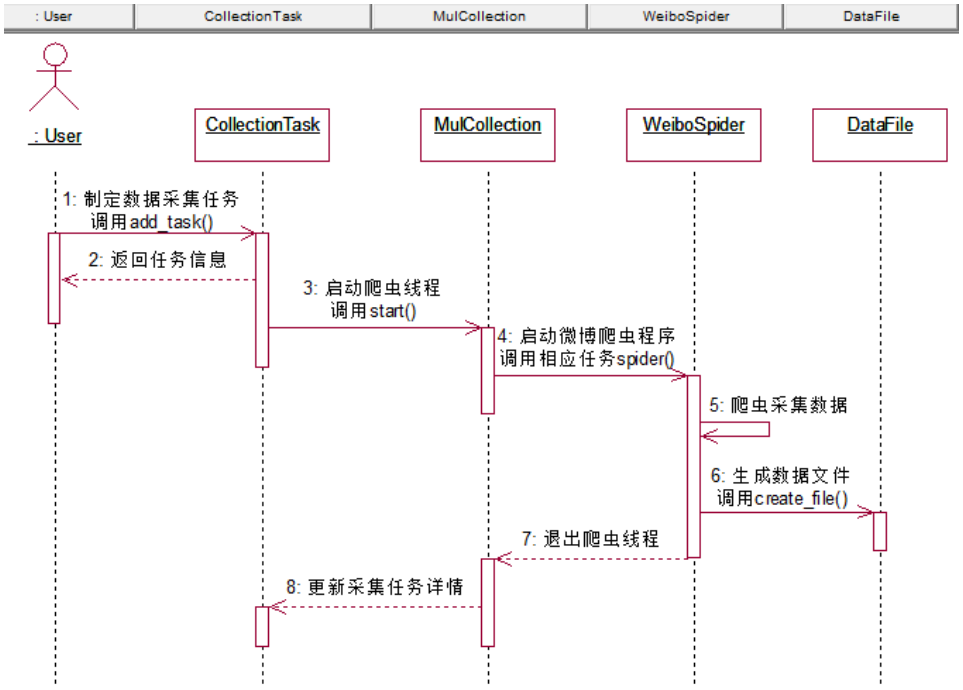

图 12 数据采集模块时序图

用户在数据采集模块中可以通过 CollectionTask 类的 add_task()制定采集任务，CollectionTask 类调用爬虫线程类 MulCollection 的 start()、run()方法开启新的爬虫线程，线程调用 WeiboSpider 类的相应爬虫任务方法进行微博数据爬取，爬取完成后调用 DataFile 类的 create_file()方法生成微博数据文件。

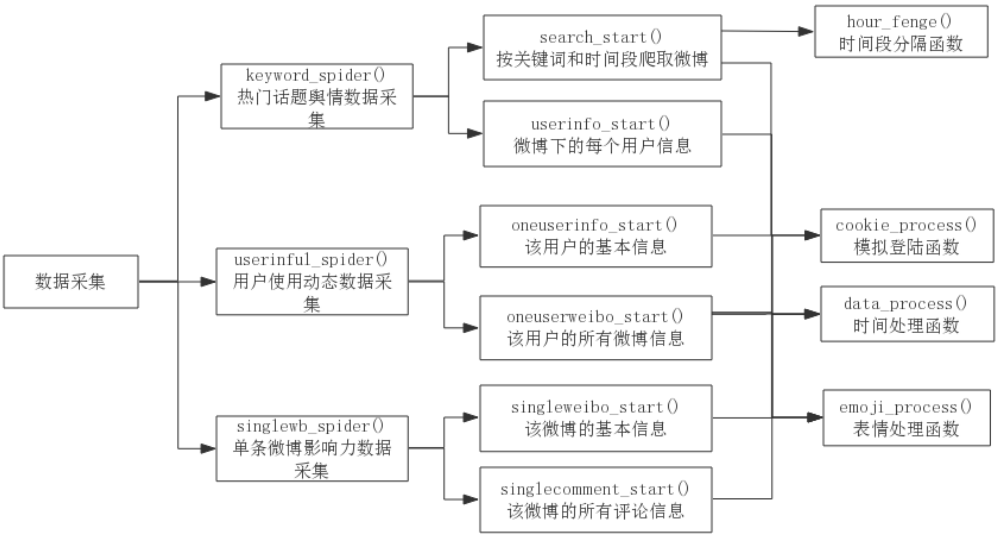

图 13 爬虫类中函数调用图

在爬虫类中，程序会分别调用三种函数来执行不同的采集任务：keyword_spider 为热门话题舆情数据采集、userinful_spider 为用户使用动态数据采集、singlewb_spider 为单条微博影响力数据采集。根据三种任务的不同需求，程序会调用相应的子函数爬取不同的微博数据。爬取过程中都会经过 data_process、emoji_process 等函数的过滤，以达到简单清洗数据的效果。cookie_process 函数负责模拟用户登录。

3.2.2 数据存储模块

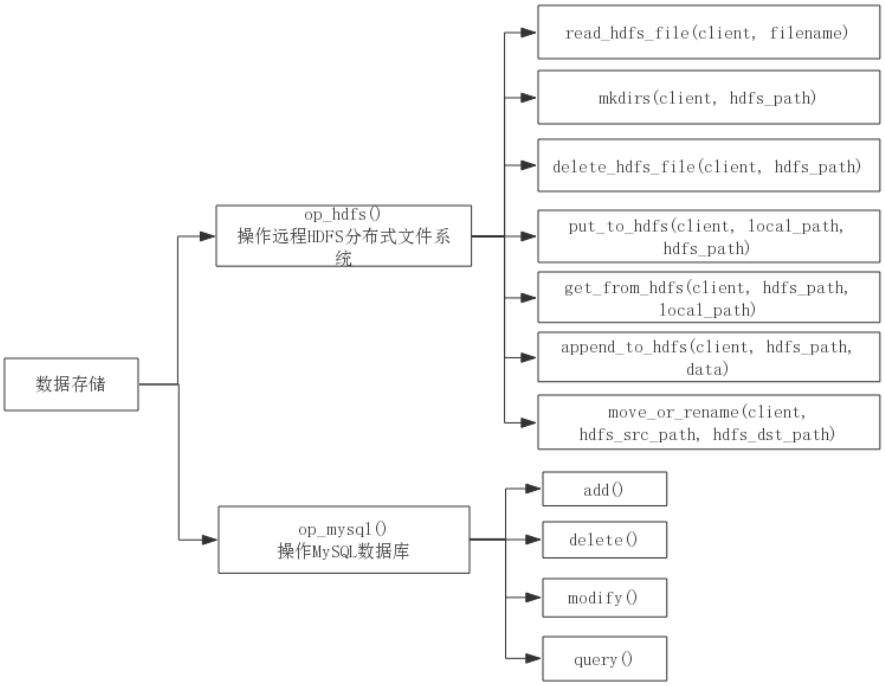

图 14 数据存储模块函数调用图

Python 操作 HDFS 函数调用图如图 14 所示。以下为操作 HDFS 的函数介绍，client 为连接对象、hdfs_path 为 HDFS 中路径、local_path 为本地路径、filename 为文件名：

read_hdfs_file(client, filename)：读取 hdfs 文件内容,将每行存入数组返回

mkdirs(client, hdfs_path)：创建目录

delete_hdfs_file(client, hdfs_path)：删除 hdfs 文件

put_to_hdfs(client, local_path, hdfs_path)：上传文件到 hdfs

get_from_hdfs(client, hdfs_path, local_path)：从 hdfs 获取文件到本地

append_to_hdfs(client, hdfs_path, data)：追加数据到 hdfs 文件

move_or_rename(client, hdfs_src_path, hdfs_dst_path)：移动或者修改文件

3.2.3 数据分析模块

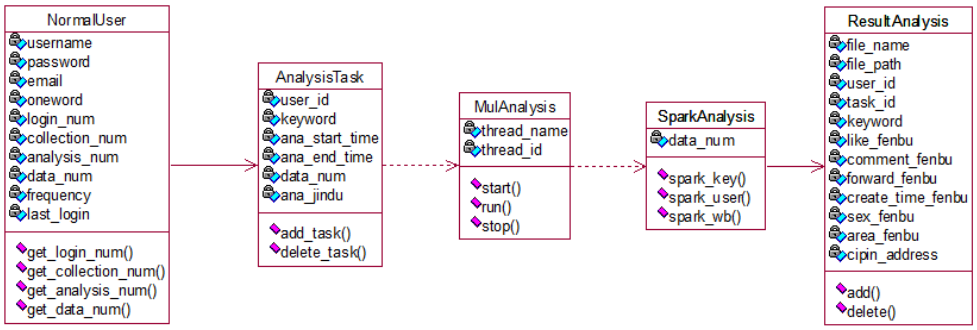

图 15 数据分析模块类图

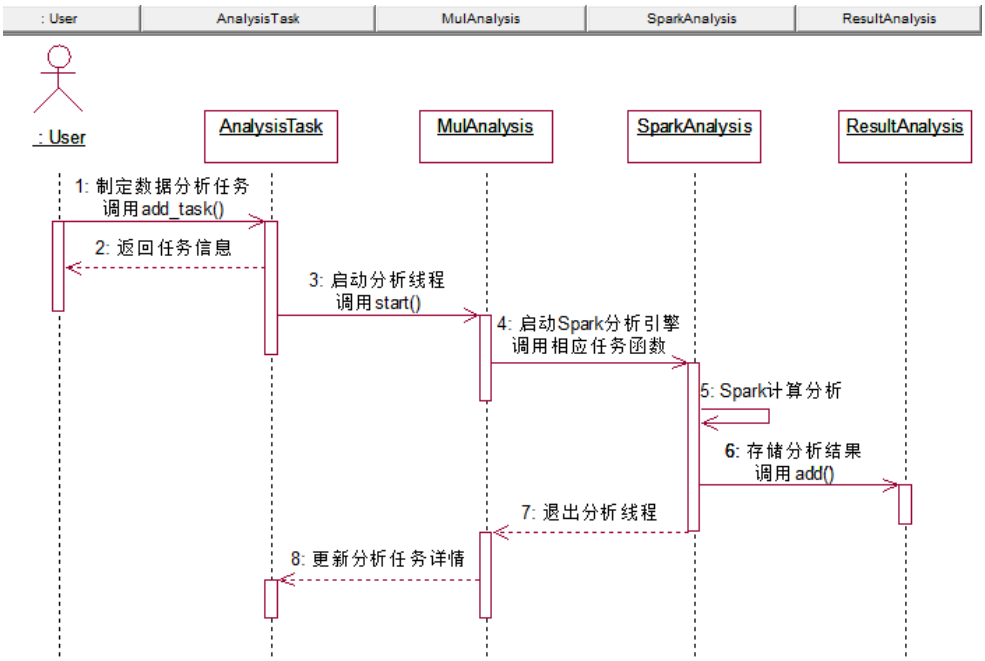

图 16 数据分析模块时序图

AnalysisTask 为分析任务类，MulAnalysis 为分析线程类，SparkAnalysis 类表示 Spark 计算分析程序，ResultAnalysis 为分析结果类。用户可以通过 AnalysisTask 类的 add_task()制定分析任务，AnalysisTask 类调用分析线程类 MulAnalysis 的 start()、run()方法开启新的分析线程，线程调用 SparkAnalysis 类的相应分析任务方法进行微博数据分析，分析完成后调用 ResultAnalysis 类的 add()方法添加分析结果进数据库。

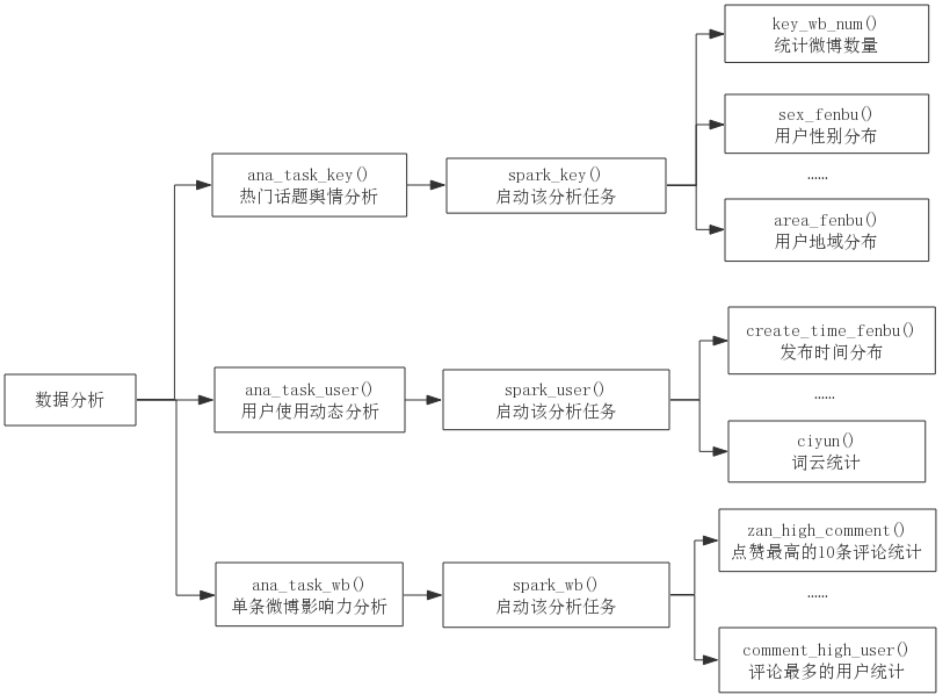

图 17 Spark 分析类中函数调用图

Spark 分析类中的函数调用图如图 17 所示， ana_task_key 为热门话题舆情分析、ana_task_user 为用户使用动态分析、ana_task_wb 为单条微博影响力分析。三种函数会启动不同的 Spark 线程获取相应微博数据，并针对不同的任务需求调用子函数进行具体分析。例如在热门话题舆情分析任务中，需要调用 sex_fenbu 函数用于用户性别分布统计、调用 area_fenbu 函数用于用户地域分布统计等。

3.2.4 数据可视化模块

数据可视化模块函数调用关系如图 18 所示。visual_key 用于热门话题舆情分析可视化、visual_user 用于用户使用动态分析可视化、visual_wb 用于单条微博影响力分析可视化，可视化之前调用各自相应的子函数获取对应任务的分析结果。用户管理中，get_user_situation 用于获取用户使用概况，modify_userinfo 用于修改用户基本信息，修改时调用 get_userinfo 函数展示用户历史信息，modify_pwd 用于修改用户密码。

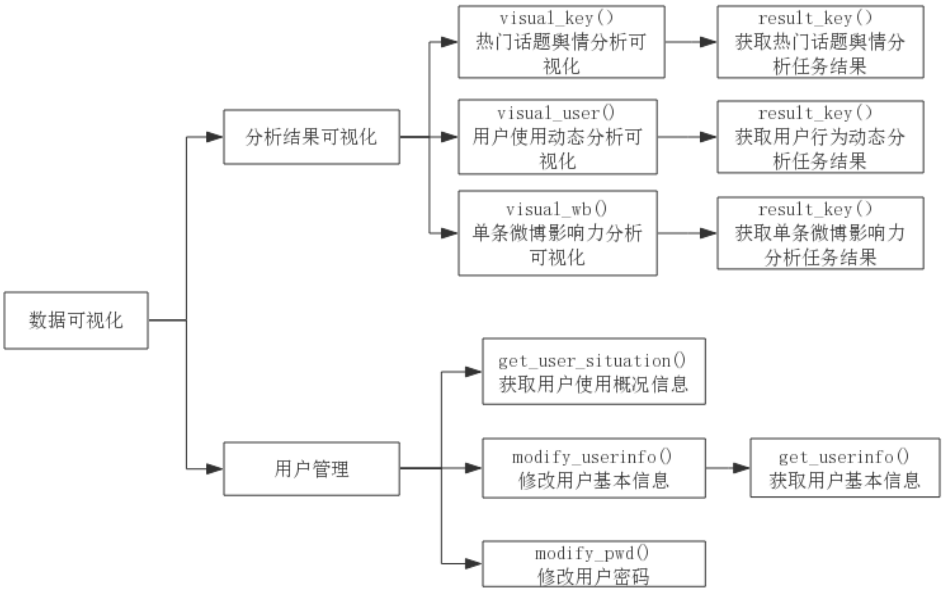

图 18 数据可视化模块函数调用图

3.3 E-R 图设计

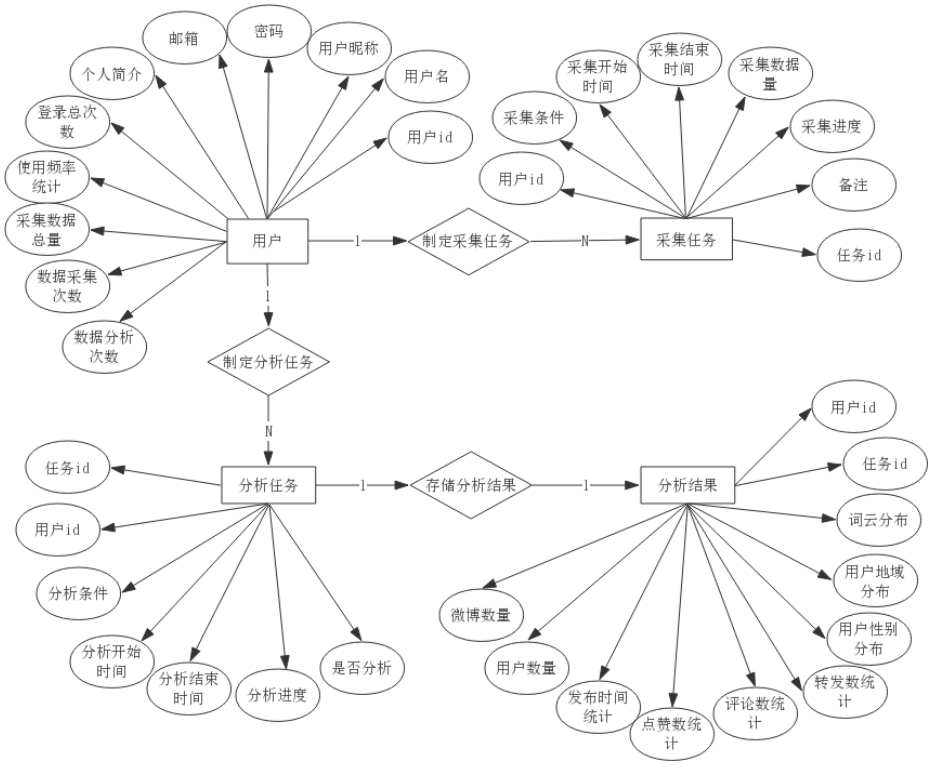

图 19 E-R 图

E-R 图如图 19 所示，主要展示出存储在 MySQL 数据库中的四类实体，包括用户、采集任务、分析任务和分析结果。用户实体包括用户名、个人简介、使用频率统计等

属性。采集任务实体包括采集条件、采集开始时间等属性。分析任务实体包括分析条件、分析进度等，分析结果根据三种不同的分析任务，相应的属性也不同。用户可以制定多个采集任务，也可以制定多个分析任务，一个分析任务只对应一个分析结果。

3.4 数据表设计

用户数据表如表 1 所示，其主要作用是存储注册本系统的用户的基本信息，包括用户名、用户密码和邮箱账号等以及用户使用本系统的概况统计，例如登录的总次数、登录频率统计、采集次数、分析次数、采集数据总量等。

表 1 用户数据表

字段名	数据类型	允许为空	字段说明
id	int(11)	NO	自增 id 索引
username	varchar(16)	NO	登录用户名
realname	varchar(32)	YES	用户昵称
password	varchar(18)	NO	密码
email	varchar(20)	NO	邮箱账号
oneword	varchar(36)	NO	个人简介
login_num	int(11)	NO	登录总次数
collection_num	int(11)	NO	数据采集次数
analysis_num	int(11)	NO	数据分析次数
data_num	bigint(20)	NO	采集数据总量
frequency	varchar(32)	NO	登录频率统计

表 2 所示为热门话题舆情分析任务详情表。任务详情表主要分为三种：热门话题舆情分析任务、用户使用动态分析任务、单条微博影响力分析任务。这里主要展示热门话题舆情分析任务的数据表，其他两个表类似。任务详情表主要是为了存储用户制定采集与分析任务时的明细，表内容主要包括制定本任务的用户索引、制定采集与分析任务的条件、采集开始及结束时间、分析开始及结束时间、采集及分析进度等。

表 2 热门话题舆情分析任务详情表

字段名	数据类型	允许为空	字段说明
id	int(11)	NO	自增 id 索引
user_id	int(11)	NO	用户 id 索引
keyword	varchar(64)	NO	话题关键词
start_time	varchar(16)	NO	关键词搜索开始时间点
end_time	varchar(16)	NO	关键词搜索终止时间点
data_num	int(11)	YES	采集数据量

续表 2 热门话题舆情分析任务详情表

字段名	数据类型	允许为空	字段说明
coll_start_time	varchar(32)	YES	采集任务开始时间
coll_end_time	varchar(32)	YES	采集任务结束时间
coll_jindu	int(11)	NO	采集进度
is_ana	int(11)	NO	是否分析
ana_start_time	varchar(32)	YES	分析任务开始时间
ana_end_time	varchar(32)	YES	分析任务结束时间
ana_jindu	int(11)	NO	分析进度

表 3 所示为热门话题舆情分析任务结果表。任务结果表与任务详情表对应，也分为三种情况，这里主要介绍热门话题舆情分析任务结果表。任务结果表主要是为了存储经 Spark 分析后的任务各项结果，以便于系统能从 MySQL 数据库中快速获取数据。热门话题舆情分析任务结果表主要包括制定本任务的用户 id 索引、与本任务对应的任务详情表索引、话题关键词、话题搜索的开始时间点及终止时间点、所有微博发布时间分布统计结果、涉及用户性别分布统计结果及用户区域分布统计结果等。

表 3 热门话题舆情分析任务结果表

字段名	数据类型	允许为空	字段说明
id	int(11)	NO	自增 id 索引
user_id	int(11)	NO	用户 id 索引
task_id	int(11)	NO	任务 id 索引
keyword	varchar(64)	NO	话题关键词
key_wb_num	int(11)	YES	话题下的微博数量
key_user_num	int(11)	YES	话题涉及的用户数量
like_fenbu	varchar(128)	YES	赞数分布统计
comment_fenbu	varchar(128)	YES	评论分布统计
forward_fenbu	varchar(128)	YES	转发分布统计
create_time_fenbu	varchar(256)	YES	微博发布时间分布统计
sex_fenbu	varchar(64)	YES	涉及用户性别分布统计
area_fenbu	varchar(128)	YES	涉及用户区域分布统计
cipin_address	varchar(64)	YES	词云图片地址

3.5 本章小结

本章主要是阐述系统的设计方法和思路。在概要设计和详细设计中将系统划分为四大模块，并对每个模块的设计方法和实现过程进行具体的阐述。

第 4 章 系统实现

本章将基于制定好的设计思路和方法，对微博数据采集与分析系统进行各功能模块的具体实现。大数据系统的开发经常需要软硬件设备的支持和分布式集群的搭建。本文将从系统分布式环境的搭建和每个功能模块具体的实现对整个系统的实现方法和过程进行阐述。

4.1 分布式集群环境搭建

4.1.1 Linux 服务器安装部署

分布式集群环境是大数据系统存储和计算的基础，本系统需要采用三台 Linux 服务器为分布式集群部署提供设备支持。三台服务器均安装 Centos6.8 版本的 Linux 操作系统。Java JDK 采用比较稳定的 1.8 版本，Python 版本为 3.6.8，Hadoop 版本为 2.8.5，Spark 版本为 2.4.1。集群采用主-从（Master-Slave）模式，三台服务器分别命名为 spark1、spark2、spark3 并通过修改/etc/hosts 将名字与服务器 IP 地址相对应，其中 spark1 为集群主节点（Master），spark2 和 spark3 作为集群从节点（Slave）。采用 Linux 系统的 ssh-keygen 命令对三台服务器实现相互之间的 ssh 免密码登录。在三台服务器上分别安装 Java、Python3，并配置相应的环境变量。

4.1.2 Hadoop 安装部署

本系统主要采用 Hadoop 的 HDFS 作为微博数据的分布式存储系统，Spark 代替 MapReduce 作为大数据分析引擎。

Hadoop 在部署时需要到 Hadoop 官网下载与 Linux 系统对应的 Hadoop 版本压缩包，传至 spark1 服务器，解压缩之后配置 Hadoop 相关环境变量。之后进入到 /hadoop/etc/hadoop 目录中对 Hadoop 配置文件进行修改，配置文件一般包括 core-site.xml、mapred-site.xml、hdfs-site.xml、yarn-site.xml、slaves 等。之后拷贝 spark1 上 hadoop 文件夹和系统配置文件至 spark2、spark3，利用 source 命令使各服务器配置文件生效。在主节点 spark1 上使用 hdfs namenode –format 命令格式化 namenode，使用 start-dfs.sh 命令启动 hdfs 集群。通过访问：spark1:50070 可查看 Hadoop 集群各个节点的运行状态。

4.1.3 Spark 安装部署

Spark 可运行于独立的集群模式中，也可作为计算模型运行于 Hadoop 中，并且可以访问 HDFS、HBase、Hive 等多种数据源。Spark 相对于 Hadoop 框架来说具有计算

上的巨大优势，但 Spark 主要用于替代 Hadoop 中的 MapReduce，并不能完全取代 Hadoop，数据存储方面依然需要借助于 HDFS 实现分布式存储。本系统主要采用 Standalone 模式对 Spark 进行集群部署，它也是典型的主从（Mater-slave）模式。

Spark 在部署时与 Hadoop 类似，首先需要进入到官网下载与 Hadoop 版本相对应的压缩包，上传至 spark1 服务器，进行解压缩安装和环境变量的配置。之后对 spark 中 conf 目录下的 spark-env.sh 配置文件进行修改，设置 JDK 和 Hadoop 的路径、设置主节点 IP、设置运行时资源分配的相关配置等，修改 slave 文件设置从节点。之后在每个节点上都依照 spark1 （Master）节点做同样的 spark 安装和配置。最后在 spark1 上 spark 目录下的 sbin 目录中执行./start.all.sh 启动 spark 集群，并可通过访问 spark1:8080 查看 spark 集群运行状态。

4.2 数据采集模块的实现

根据数据采集任务的要求并通过对微博网站数据进行解析，将微博数据爬虫任务基本分为以下三种：通过关键词和时间段爬取热门微博信息以及每一条微博对应的用户信息、通过用户 ID 爬取用户基本信息以及此用户对应的所有微博信息、通过微博 ID 爬取微博基本信息以及此微博下的每一条评论信息。

（1）微博爬取信息主要包括微博 ID、微博用户 ID、微博用户名、微博内容、微博创建时间、点赞数、转发数、评论数，微博 URL 等。

（2）用户爬取信息主要包括用户 ID、用户名、用户性别、用户地址、用户简介、用户认证、微博数量、粉丝数、关注数等。

（3）评论爬取信息主要包括评论用户 ID、评论用户名、评论赞数、评论内容、评论等。

数据采集模块主要依靠 Python 网络爬虫实现微博数据的采集。由于新浪微博 Web 端提供的数据资源丰富、网页结构简单，所以系统主要对新浪微博的 weibo.cn 和 weibo.com 进行网页解析并编写爬虫程序。BeautifulSoup 是一个可以从 HTML 或 XML 文件中提取数据的 Python 扩展库，本系统的网络爬虫主要依靠 BeautifulSoup 对微博网页进行解析和数据提取。

微博爬虫任务实现流程如图 20 所示：首先启动微博爬虫，读取本地配置文件，配置文件中主要存储可供使用的 IP 代理地址和用户 cookies。如果 cookies 已经失效，可登录微博网页，利用浏览器开发者工具进行抓包分析获取登录时的 cookies，在爬虫中请求网页时加 cookies 参数可实现微博爬虫的模拟登录。由于微博网站的反爬机制，

如不模拟登录将不能访问到有效的微博网页数据。之后根据用户给出的不同的爬取条件制定不同的参数与微博相对应的接口相拼接形成 URL，并将其加入到待爬取 URL 列表中，通过 Python requests 库的 get 方法对列表中的 URL 进行请求，获取返回的网页内容。此时的网页信息为繁杂的非结构数据，通过对前端页面结构的分析用 BeautifulSoup 设计出解析策略，提取网页中的微博数据并进行简单数据处理，之后作为结构化数据持久化成 CSV 文件保存至本地或 HDFS 中。

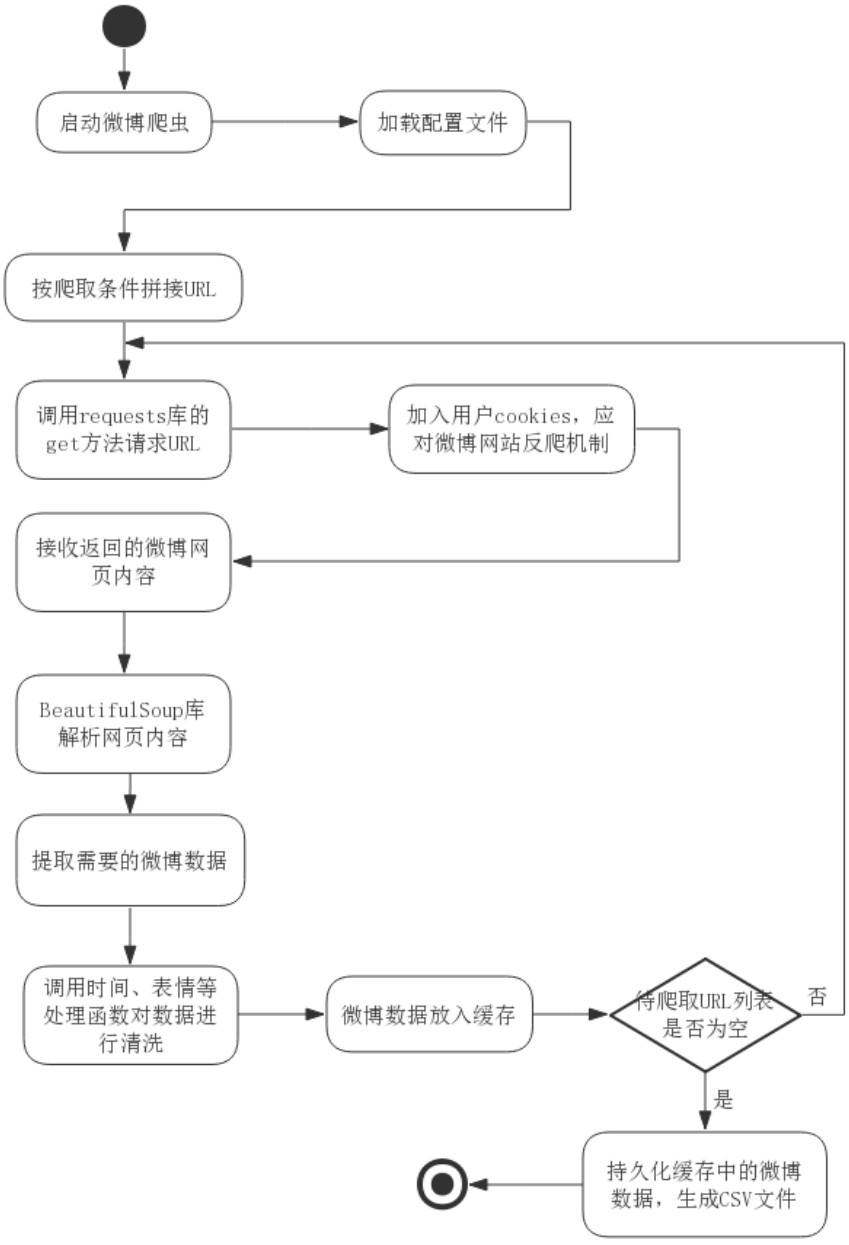

图 20 微博爬虫程序活动图

4.3 数据存储模块的实现

数据存储模块主要是对 HDFS 分布式文件系统和 MySQL 数据库进行操作。HDFS 部署已经在之前小节里加以阐述，这里主要介绍 Python 中操作 HDFS 的实现方法以及 Flask 框架中利用 SQLAlchemy 操作 MySQL 数据库的实现方法。

4.3.1 Python 操作 HDFS

Python 操作 HDFS 通过“HDFS”这一 Python 扩展库来实现。它可以连接远程 HDFS 集群，并提供读取 hdfs 文件、上传文件至 hdfs、下载 hdfs 文件、追加数据到 hdfs 文件、创建目录、移动和修改文件等方法。

首先创建一个 Client：client = Client("http://spark1:50070", root="/", timeout=100)，通过 client 可以连接到远程 HDFS 集群，然后根据各函数不同功能操作 HDFS。举例如下：（hdfs_path 为 HDFS 中路径、local_path 为本地路径、filename 为文件名）：

read_hdfs_file(client, ‘/home/共青团中央_wbdata’)，读取 hdfs 中 home 目录下的关键词为“共青团中央”的微博数据内容,将每行存入数组返回。

put_to_hdfs(client,‘d://123.txt, ‘/home’)，将本地 D 盘的 123.txt 文件上传文件到 hdfs 中根目录下的 home 目录。

4.3.2 SQLAlchemy 操作 MySQL

SQLAlchemy 是一款开源软件，可以用来操作 SQL，并且提供了 ORM 对象关系映射工具，采用了类似于 Java 里 Hibernate 的数据映射模型。ORM 可以让编程人员在操作数据库时跟操作对象一样，表抽象成类，数据抽象成该类的某个对象。Flask 框架主要采用 SQLAlchemy 作为数据库操作工具。以下为 SQLAlchemy 的具体操作：

（1）设置配置信息：在 config.py 文件中添加 MySQL 数据库配置信息。

（2）使用 SQLAlchemy 创建模型与表的映射：模型需要继承自 db.Model，映射到表中的属性需要写成 db.Column 的数据类型。Weibodata 表实例如下：

```
class Weibodata(db.Model):
    __tablename__ = weibodata
    id = db.Column(db.Integer,primary_key=True,autoincrement=True)
    wb_id = db.Column(db.String(12),nullable=False)
```

（3）通过 SQLAlchemy 对数据进行增、删、改、查：

```
# 增加一条微博数据举例：
data = Weibodata (wb_id='123',wb_content='I love you.')
db.session.add(data)
db.session.commit()
```

4.4 数据分析模块的实现

本章将根据数据分析模块的设计方法和思路，主要利用基于 Python 的 Spark 计算引擎完成数据处理和具体的微博数据分析任务。

图 21 为 Spark 计算引擎进行数据分析的大致流程图：

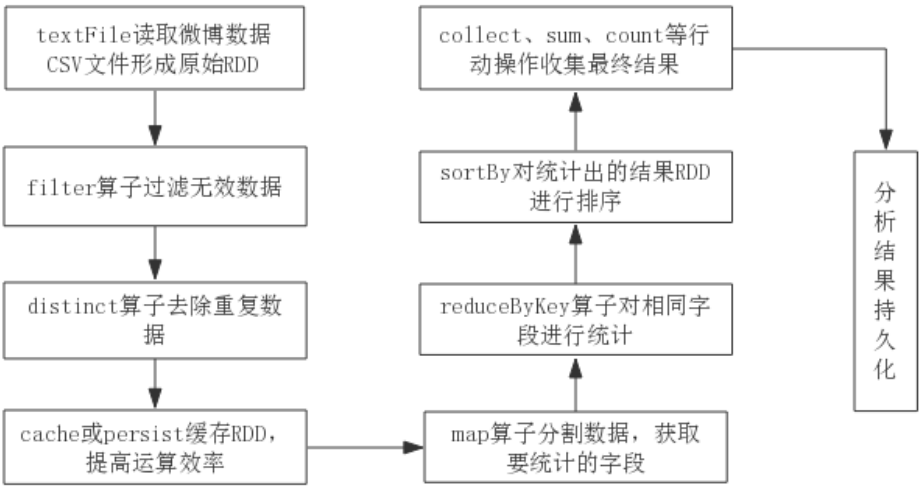

图 21 Spark 计算分析流程图

4.4.1 利用 Spark 完成数据预处理

微博数据预处理主要是通过读取存储在 HDFS 或本地的微博数据 CSV 文件，将原始数据转化为一个弹性分布式数据集，也就是一个 RDD，然后利用 RDD 的 map、filter、distinct、reduceByKey 等各类算子实现微博数据的预处理工作，主要目的是为了对微博数据进行过滤清洗，删除无效数据、去除重复数据等。特别说明的是，RDD 采用了惰性调用机制，在 RDD 做“转换”操作时不会触发计算，只有在 RDD 进行“行动”操作时才会触发真正的计算，这一机制提高了 Spark 的运行效率，避免了资源的重复浪费。

下面对数据清洗和分析所用到的 Spark 的主要算子进行介绍：

转换操作：

textFile()：程序可通过 Spark 的 textFile()方法读取 HDFS 或本地文件中的原始数据，创建一个可供 Spark 的其他算子进行处理的 RDD。

map(func)：map 可返回新的 RDD，此 RDD 是由每个输入元素通过 func 函数转换后组成。我们可以用 map 算子进行微博数据的分割，map 算子通常以行为单位，可以采用 split 方法以 CSV 文件中‘,’为分隔符对文件中每行的数据进行分割，从中提取

出分析任务需要用到的微博数据。

filter(func)：filter 可返回新的 RDD，此 RDD 通过 func 函数运算后由返回值为 true 的原始元素组成。我们可以用 filter 过滤掉一些文件中不满足分析任务需求、低质量、无效的微博数据。

distinct()：distinct 是专门去重的算子，它可以对源 RDD 去重后返回一个新的 RDD。

reduceByKey()：在键值对 RDD 上进行调用，指定的 reduce 函数，可以将相同 key 的值聚合到一起，并返回键值对 RDD。我们可以用 reduceByKey 完成微博数据中相同 key 值例如用户 ID、微博 ID 等的数据统计。

sortByKey()：在键值对 RDD 上进行调用，返回的键值对 RDD 按照 key 进行排序。我们可以用 sortByKey 来对微博分析任务中各类统计好的数据进行排序。

cache()或 persist()：RDD 缓存，可以将需要重复运算的 RDD 存储在内存中，以便大幅提升运算效率，cache 默认就一个缓存级别 MEMORY-ONLY ，而 persist 则可以选择缓存级别。

行动操作：

collect()：将数据集的所有元素以数组的形式返回。

take(n)：将数据集的前 n 个元素进行返回，返回一个数组。

count()：返回调用此方法的 RDD 的元素个数。

foreach(func)：对数据集的每个元素，通过 func 函数进行更新。

4.4.2 热门话题舆情分析任务的实现

热门话题舆情分析主要涉及到的分析任务为此关键词下的微博数量、微博点赞数分布、微博评论数分布、微博转发数分布、微博发布时间分布、涉及用户总数、涉及用户性别分布、涉及用户地域分布等。

下面对几个典型的分析任务实现进行说明：

（1）从文件获取原始 RDD

```
textRDD = sc.textFile(keywb_filename)
keywordRDD = textRDD.filter(lambda line: len(line.split(',')) == 12)
keywordRDD.cache()
```

通过读取相应的 CSV 文件，创建原始 textRDD，并利用 filter 对每行数据不满足 12 个数据字段的数据行进行过滤，并将过滤后的 keywordRDD 缓存到内存中。

（2）获取热门关键词下所有的微博数量：

key_wb_num = keywordRDD.count()

CSV 文件中每一行代表一条微博的信息，通过对 keywordRDD 进行 count()可直接统计文件中的行数，以获取此文件下的所有微博数量。

（3）微博创建时间分布情况分析

以下为 Python 实现代码，其中 lambda 是 Python 中用来表示匿名函数的表达式：

create_time_fenbu = keywordRDD.map(lambda line: line.split(",")[5]).map(lambda line: line.split(' ')[0]).map(lambda num: (num, 1)).reduceByKey(lambda a, b: a + b).sortBy(lambda a: a[1], False).collect()

对微博创建时间的分布进行分析大致流程如下：

首先从存储某关键词下所有微博的 CSV 文件中创建原始 RDD，进行简单过滤清洗后通过 cache 缓存到内存中，通过 map 算子提取原始 RDD 中经过 split 后的相应微博创建时间字段，之后利用 map 算子对每个字段的数量赋值为 1 生成键值对，然后利用 reduceByKey 将相同 key 的值相加获取不同分布时间的数据量，之后利用 sortBy 对创建时间的数据量进行排序，最后运行 collect 方法执行“行动”操作，触发真正计算，收集分析后数据集的所有元素。

4.4.3 用户使用动态分析任务的实现

用户使用动态分析主要涉及到的分析任务为此用户下微博发布时间分布、用户微博内容词云展示、此用户获取的所有转发数、此用户获取的所有评论数、此用户获取的所有赞数、此用户微博数、关注数、粉丝数等。

这里主要介绍获取用户所有微博的转发数量和用户微博内容词云展示的实现：

（1）此用户所有微博的转发数量

userwbRDD = sc.textFile(userwb_filename)

forward_all_num = userwbRDD.map(lambda line: int(line.split(',')[6])).sum()

首先从存储某用户下所有微博的 CSV 文件中创建原始 RDD，通过对原始文件的 RDD 进行 map 算子，获取转发情况对应的某字段，在 map 函数中将字段转为 int 类型，并利用 sum 方法对所有微博对应的转发数进行相加，获取最后结果。

（2） 词云展示

textRDD = sc.textFile(userwb_filename)

userwbRDD = textRDD.filter(lambda line: len(line.split(',')) == 9)

textRDD = "".join(userwbRDD.map(lambda line: line.split(",")[4]).collect())

cut_text = " ".join(jieba.cut(textRDD))

wCloud = cloud.generate(cut_text)

首先从存储某用户所有微博的 CSV 文件中创建原始 RDD，进行简单过滤清洗后通过 map 算子获取原始 RDD 中微博内容的相应字段，得到微博内容的 textRDD，接着将 textRDD 用 Python 扩展库 jiba 进行中文分词得到全部中文词语的统计 cut_text，之后用 Python 扩展库 WordCloud 对中文词语的统计结果进行词云图片的生成，WordCloud 会自动将文本中出现频率较高的“关键词”予以视觉化的展现。

4.4.4 单条微博影响力分析任务的实现

单条微博影响力分析主要涉及到的分析任务为此微博下点赞最高的 10 条评论、评论最多的 10 位用户、此微博的赞数量、此微博的评论数量、此微博的转发数量等。

这里主要介绍微博点赞最高的 10 条评论的实现方法，其他分析任务与以上类似。

textRDD = sc.textFile(userwb_filename)

commentRDD = textRDD.filter(lambda line: len(line.split(',')) == 7)

commentRDD.cache()

like_high_comment=commentRDD.map(lambdaline:(line.split(",")[4],int(line.split(",")[5]))).reduceByKey(lambda a, b: a + b).sortBy(lambda a: a[1], False).collect()[:10]

对微博下点赞最高的 10 条评论进行分析大致流程如下：

首先从存储所有评论内容的 CSV 文件中创建原始 RDD，利用 filter 进行简单过滤清洗后再利用 cache 缓存到内存中，通过 map 算子提取原始 RDD 中经过 split 后的两个相应字段，一个是评论内容，一个是对应评论的点赞数量，点赞数量在 map 函数中进行 string 转 int 类型操作，之后利用 reduceByKey 将相同 key 的值相加获取不同评论对应的点赞量，之后利用 sortBy 按照点赞量的大小进行从大到小排序，最后运行 collect 方法收集分析后的评论点赞数情况并进行分片，取前 10 条点赞数最高的评论。

4.5 数据可视化模块的实现

数据可视化模块主要分为三层，分别是界面展示层、业务逻辑层和数据访问层，整体架构如图 22 所示：

（1）数据访问层主要由 MySQL、HDFS 和 SQLAlchemy 组成。MySQL 负责存储整个微博数据采集与分析系统中用户的基本信息数据、用户使用情况的统计数据，采集任务以及分析任务的详情明细、分析结果的存储。HDFS 主要负责网络爬虫爬取到的大规模微博原始数据的存放。SQLAlchemy 负责与 MySQL 进行交互，将 MySQL 的

操作映射为类的操作，实现对 MySQL 的查询、存储、修改等操作。

（2）业务逻辑层主要由 Flask Web 框架来实现，负责系统各项功能业务有关的系统设计以及业务流程的具体实现，例如用户的登录、注册、用户使用情况的统计，制定采集任务、启动爬虫程序、制定分析任务、启动分析任务、各项任务详情的更新、查看数据分析结果和用户资料修改等。前端页面发送请求后，经过 Flask 的装饰器路由到 Flask 框架中的对应视图函数，各个视图函数来负责实现系统的各项业务逻辑。

（3）界面展示层是整个架构的最上层，由 HTML、ECharts 组成，它负责与用户交互，接收用户请求，并交给业务逻辑层进行处理，处理返回结果由 Jinja2 模板引擎负责渲染，在前端页面进行展示。ECharts 被嵌入至 HTML 界面中，通过初始化 ECharts 对象，配置 Option，传入分析结果参数，可以将分析结果转发为各类图表显示在前端界面中，实现数据分析的可视化。除此之外，界面展示层还使用了 CSS、JavaScript、Jquery、Bootstrap 等前端技术进行实现。

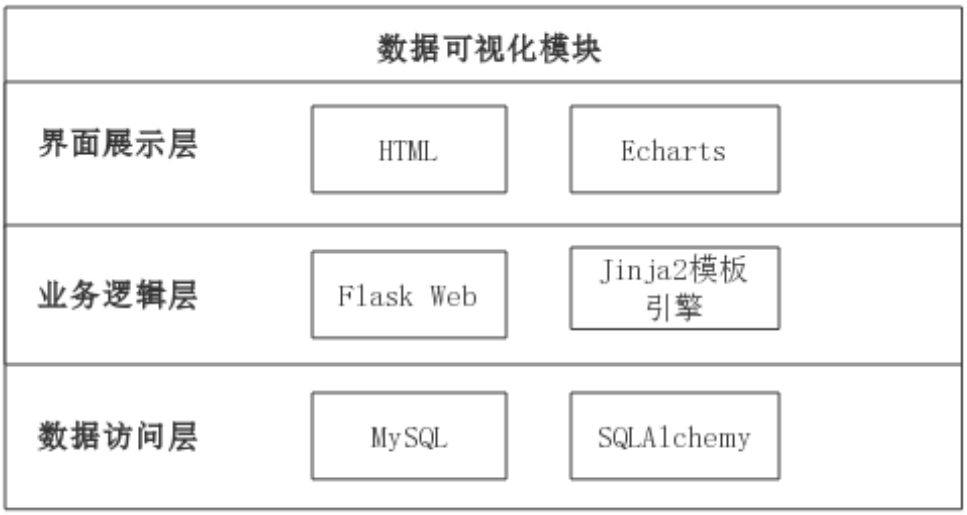

图 22 数据可视化模块技术架构图

4.6 本章小结

本章主要是对微博数据采集与分析系统的实现过程和方法进行详细的说明。首先对系统所用到软硬件环境进行阐述，接着说明了 Hadoop 集群和 Spark 集群的部署过程。之后利用网络爬虫、Spark、Flask 等技术对数据采集、分析、存储和可视化四大模块详细的功能设计进行具体实现，介绍了各模块的执行流程，并用流程图、架构图和代码片段的形式描述了重点功能的实现方法。

第 5 章 系统测试

为保证系统的正确运行，检测系统实现过程中出现的逻辑问题和代码漏洞。本章主要采用黑盒测试的方法对系统各功能模块进行测试。通过设计测试用例，输入相关数据对各功能点进行操作，检测各功能运行过程中的业务流程是否和预期一致。

5.1 用户基本操作测试

表 4 用户基本操作测试用例表

编号	测试用例	预期结果	实测结果
1	注册功能：输入用户名、邮箱、密码、密码二次确认，执行注册	如密码两次不一致、用户名已注册，提示注册失败；否则注册成功，转入登录界面	与预期结果一致
2	登录功能：输入用户名、密码，执行登录	如用户名、密码正确则转入主页，否则提示登录失败	与预期结果一致
3	在用户主页中，查看用户登录总次数、采集数据总量、采集及分析次数、最近任务明细等数据	用户各数据统计结果正确	与预期结果一致
4	点击采集、分析、可视化任务链接	进入相应的任务完成界面	与预期结果一致
5	修改密码：输入旧密码、新密码、密码确认	如旧密码错误或两次新密码输入不一致，提示修改失败，否则，修改成功	与预期结果一致
6	修改用户基本信息：修改界面显示历史数据，更新后执行修改	后台执行数据库更新操作，刷新界面，显示信息已经修改	与预期结果一致
7	点击用户注销功能	服务器清除用户相关 session，系统转到登录界面	与预期结果一致

图 23 用户主页界面

5.2 数据采集模块测试

表 5 数据采集模块测试用例表

编号	测试用例	预期结果	实测结果
1	输入采集条件，制定热门话题舆情分析、用户使用动态分析、单条微博影响力分析数据采集任务	后台启动爬虫程序，任务多时多线程并发爬取，爬取完成存入本地或 HDFS 数据库	与预期结果一致
2	查看正在采集和采集完成的任务详情	查看采集条件、采集任务开始时间、采集数据量等任务详情数据准确	与预期结果一致
3	删除已经完成的采集任务	提示是否确认删除，如确认，则删除任务，刷新当前页面	与预期结果一致
4	下载已采集完成的微博数据文件	后台自动找到对应的数据文件并通过浏览器下载到本地	与预期结果一致

图 24 爬虫程序运行演示图

图 25 已完成的采集任务界面图

5.3 数据分析模块测试

表 6 数据分析模块测试用例表

编号	测试用例	预期结果	实测结果
1	输入分析任务条件：制定热门话题舆情分析、用户使用动态分析、单条微博影响力分析任务	如果分析任务对应的采集任务存在且完成，则任务制定成功跳转到正在分析任务界面，否则系统返回提示信息	与预期结果一致
2	分析任务制定成功，启动分析	后台启动 Spark 引擎，读取 HDFS 文件，对各项任务进行分析，分析结果存入 MySQL	与预期结果一致
3	查看正在分析或分析完成的任务详情	查看分析条件、分析任务开始时间、分析进度等任务详情数据准确	与预期结果一致

```
此关键词下的微博数量:
48
此关键词下的点赞数分布:
[('82', 8), ('52', 2), ('502', 2), ('4', 2), ('3', 2), ('2', 4), ('16', 2), ('14', 2),
此关键词下的评论数分布:
[('35', 2), ('21', 8), ('2', 4), ('1', 4), ('0', 30)]
此关键词下的转发数分布:
[('4', 2), ('199', 1), ('198', 1), ('142', 8), ('14', 2), ('0', 34)]
此关键词下的发布时间分布:
[('2019-01-01', 44), ('2018-09-02', 4)]
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\WANGWE~1\AppData\Local\Temp\jieba.cache
Loading model cost 0.738 seconds.
Prefix dict has been built succesfully.
此关键词涉及用户总数:
16
此关键词下的用户性别分布:
[('男', 5), ('女', 11)]
[Stage 36:>                                                         (0 + 2) / 2]
```

图 26 Spark 引擎计算分析演示图

图 27 已完成的分析任务界面图

5.4 数据可视化模块测试

表 7 数据可视化模块测试用例表

编号	测试用例	预期结果	实测结果
1	可视化模块展示已经完成的各项任务的分析结果，用户可以点击可视化按钮查看分析结果	分析结果界面会根据不同的分析任务，呈现不同分析任务下的各项数据统计结果和 ECharts 图表	与预期结果一致

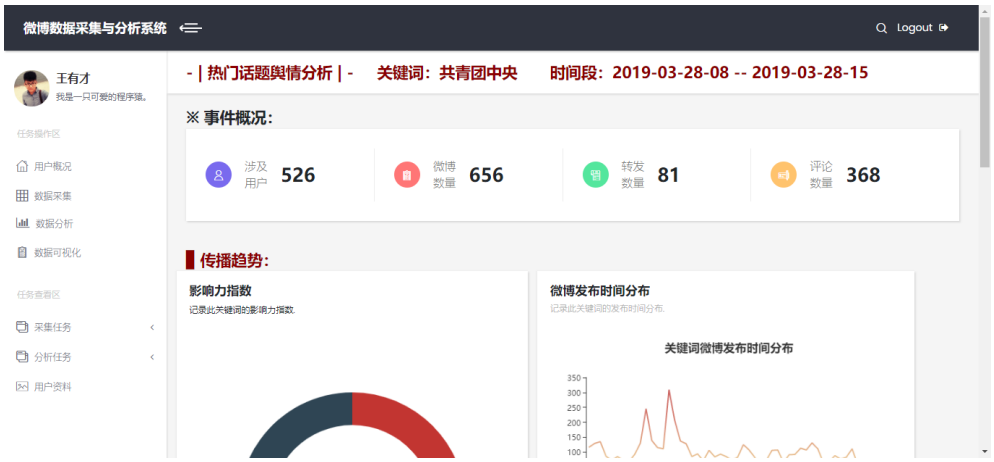

图 28 热门话题舆情分析可视化界面

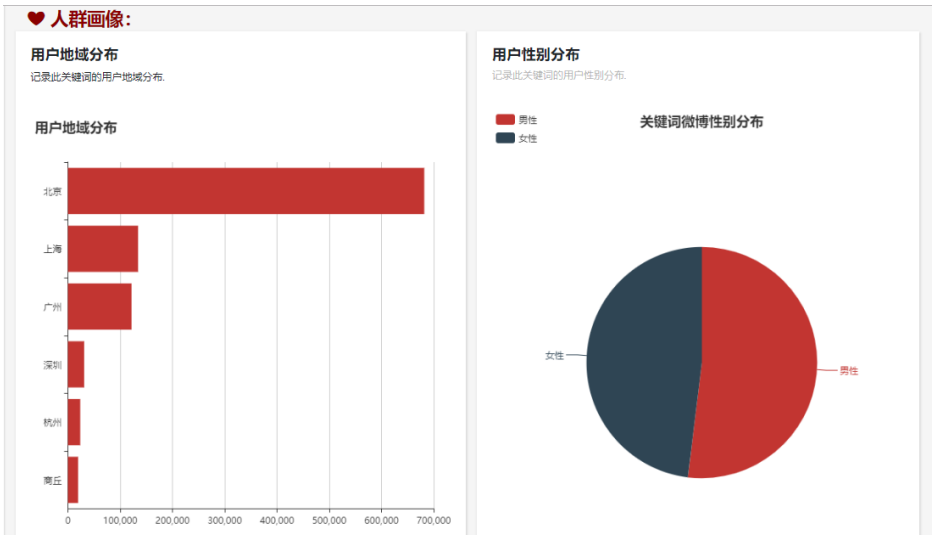

图 29 用户地域分布图和用户性别分布图

5.5 本章小结

本章主要对本系统各模块进行了简单的功能性测试，并对已经实现的系统界面和可视化功能图表进行了部分展示，系统测试结果与预期基本一致。

总结与展望

本文在充分的调查研究基础上，利用网络爬虫和 Spark 计算引擎等技术基本实现了微博数据采集与分析系统。本文的工作主要总结为以下几点：

（1）对微博数据采集与分析系统的现状进行调查研究。对微博网站中的业务流程和微博内容的结构进行分析，了解微博数据的格式和存在形式。对当下大数据技术、网络爬虫技术、可视化技术等进行研究，研究适合微博数据采集与分析的技术，学习各类技术的原理、特点和使用方法。对微博数据采集与分析系统进行需求分析。

（2）基于各类技术的适用场景，从具体需求出发，结合微博数据的特点和数据采集与分析技术的架构，进行系统设计，并制定出三类不同的爬取策略和三种不同的分析任务。按照高内聚、低耦合的方法设计数据采集、数据存储、数据分析和数据可视化等四大模块，并给出了具体的设计方法。

（3）对微博数据采集与分析系统进行实现。介绍软硬件需求、搭建系统环境，根据整体的架构和每个模块的设计思路进行具体实现。数据采集模块利用网络爬虫实现，数据存储模块基于 HDFS、MySQL 数据库实现，数据分析模块采用当下流行的 Spark 计算引擎，对大规模微博文件进行处理分析，获取分析结果。数据可视化模块利用 Flask Web 框架、ECharts、HTML 等技术实现系统的具体业务流程和分析结果的可视化展示。

本文基本实现了微博数据采集与分析系统，具备一定的数据采集、分析与可视化展示功能。但因开发环境有限、自身能力不足，系统的各项功能方面还有待完善。不足之处有以下几点：

（1）开发环境和存储设备不能满足要求，数据规模有限，没有达到海量数据的采集与分析，不能较好地在分布式环境下使用 Spark 引擎，测试方面主要使用单机环境，无法验证海量数据规模下的分布式数据分析效果。

（2）网络爬虫实现的方法比较简单，由于微博官方网站的限制，获取微博数据的速度较慢，没有制定出更优越的策略来应对微博官网的反爬机制。制定三种微博数据爬取任务有一定限制，不能完成微博全网的海量爬取。后续可进行爬虫程序的优化，实现快速全面的微博数据采集。

（3）数据分析任务较为简单，只制定了三种不同的分析任务，没有从更复杂的维度制定分析策略从而体现微博数据的价值性。本文的数据只限于文本，没有涉及图片和视频，后续可考虑运用人工智能方面的知识实现更复杂、关联性更强的分析任务。

参考文献

[1]刘铭宇．基于 Web 的数据可视化系统设计及应用[D]．北京邮电大学,2018
[2]邱德扬．基于 Spark 的海量数据分析与性能优化[D]．北京邮电大学,2018
[3]沈鹏飞．基于 Spark 的微博数据分析系统的设计和实现[D]．北京邮电大学,2018
[4]王颖．基于 Hadoop 的微博用户影响力分析[D]．山东大学,2018
[5]王海林．基于 Spark 的社交网络数据分析平台[D]．山东大学,2018
[6]王婧雅．微博数据挖掘可视化系统的设计与实现[D]．吉林大学,2017
[7]汤姆,怀特．Hadoop 权威指南:中文版[M]．清华大学出版社,2017
[8]王峰．基于新浪微博舆情采集与倾向性分析系统[D]．南京信息工程大学,2016
[9]王子毅,张春海．基于 ECharts 的数据可视化分析组件设计实现[J]．微型机与应用,2016,35(14)
[10]杨浩,曾兴斌,何加铭．基于 Hadoop 微博热点话题挖掘系统的设计与实现[J]．数据通信,2016
[11]卡劳．Spark 快速大数据分析[M]．人民邮电出版社,2015
[12]吴信东,李毅,李磊．在线社交网络影响力分析[J]．计算机学报, 2014,37(4)
[13]周中华,张惠然,谢江．基于 Python 的新浪微博数据爬虫[J]．计算机应用,2014,34(11)
[14]Jia Zhu．Research on data mining of electric power system based on Hadoop cloud computing platform[J]．International Journal of Computers and Applications,2019,41(4)
[15]Inberg M．Flask web development: developing web applications with python[M]．Reilly Media,2018
[16]Hua Xu,Weiwei Yang.Hierarchical emotion classification and emotion component analysis on chinese micro-blog posts[J]．Expert Systems with Applications,2015
[17]Shoro A G,Soomro T R．Big data analysis:Apache spark perspective[J]．Global Journal of Computer Science and Technology,2015
[18]Keim D,Qu H,Ma K L．Big-data visualization[J]．IEEE Computer Graphics and Applications,2013

Spark-based Microblog Data Acquisition and Analysis System

WANG Wei

Abstract: In recent years, with the continuous innovation of China's information technology, more and more people are keen to obtain information from the network and expand their contacts. Social networks have achieved vigorous development. Microblog, as its typical representative, is favored by people for its open social environment, fast communication mechanism and excellent user experience, attracting a large number of users to participate. These users are generating massive data resources every day, which contain rich academic research value, government decision-making value and commercial reference value. We can study and analyze microblog data from different dimensions, mine out the information with application value, and apply these data analysis results to actual production and life. Therefore, taking Sina Weibo as the research object, this paper first introduces the research and application status of microblog data analysis, and key technologies such as big data collection and analysis. Then it focuses on the four major modules of microblog data collection, storage, analysis and visualization, and designs and implements the Spark-based microblog data collection and analysis system using web crawler, big data framework, Spark calculation and analysis engine, Flask Web and other technologies. The system finally realized can crawl microblog data on a large scale according to different crawling conditions, and can analyze the crawled microblog data in multiple dimensions and visually display the analysis results.

Keywords: Microblog;Big data;Spark;Web crawler;Data analysis