

文本复制检测报告单(全文标明引文)

№:ADBD2018R_2016042609190020180428141426424850239014

检测时间:2018-04-28 14:14:26

检测文献: 程前+20140614204+管理科学与工程学院

作者:

检测范围: 中国学术期刊网络出版总库

中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库

中国重要会议论文全文数据库

中国重要报纸全文数据库

中国专利全文数据库

图书资源

优先出版文献库

大学生论文联合比对库

互联网资源(包含贴吧等论坛资源)

英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)

港澳台学术文献库

互联网文档资源

个人比对库

时间范围: 1900-01-01至2018-04-28

检测结果

总文字复制比: 2.5%

跨语言检测结果: 0%

去除引用文献复制比: 2.5%

去除本人已发表文献复制比: 2.5%

单篇最大文字复制比: 1%

重复字数: [367]

总段落数: [2]

总字数: [14631]

疑似段落数: [2]

单篇最大重复字数: [152]

前部重合字数: [0]

疑似段落最大重合字数: [215]

后部重合字数: [367]

疑似段落最小重合字数: [152]



文字复制比部分 2.5%
引用部分 0%
无问题部分 97.5%

指标: ☐ 疑似剽窃观点 ☒ 疑似剽窃文字表述 ☐ 疑似自我剽窃 ☐ 疑似整体剽窃 ☐ 过度引用

表格: 0 公式: 0 疑似文字的图片: 0 脚注与尾注: 0

2% (215) 程前+20140614204+管理科学与工程学院_第1部分 (总10901字)

4.1% (152) 程前+20140614204+管理科学与工程学院_第2部分 (总3730字)



(注释: 无问题部分 文字复制比部分 引用部分)

1. 程前+20140614204+管理科学与工程学院_第1部分

总字数: 10901

相似文献列表 文字复制比: 2%(215) 疑似剽窃观点: (0)

1	基于Node.js的网购交流系统设计与实现 王亚强; - 《电脑编程技巧与维护》- 2016-12-18	1.0% (105) 是否引证: 否
2	基于Mini2440的手机控制家电的简易实现 周柯柯 - 《大学生论文联合比对库》- 2015-03-24	0.6% (62) 是否引证: 否
3	14_wsx_wsx_wsx wsx - 《大学生论文联合比对库》- 2015-05-18	0.4% (44) 是否引证: 否

原文内容 红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

山东财经大学

本科毕业论文(设计)

题目: 气象大数据分析预测平台设计与实践

学院管理科学与工程
专业信息管理与信息系统
班级信息管理与信息系统1402
学号 20140614107
20140614204
20140614234
姓名傅腾达、程前、王丽
指导教师张贵杰
山东财经大学教务处制
二〇一八年四月

山东财经大学学士学位论文原创性声明

本小组承诺：所有提交的毕设内容均为本在指导老师的帮助下独立取得的研究成果。此外，本论文所有引用内容都已在下方注明来源，对该论文有贡献的个人或集体都已在下方注明并表达了感谢之意。除此之外，不再包含任何非原创内容。本声明所产生的法律结果由本小组成员共同承担。

学位论文作者签名：

年月日

山东财经大学关于论文使用授权的说明

本小组理解山东财经大学关于保留与使用学位论文的有关规定，本小组完全同意学校发布此论文的全部或部分内容，亦可以影音或者通过其他方式保存该论文。

指导教师签名：论文作者签名：

年月日年月日

实时气象大数据分析预测

摘要

随着信息化时代的进步，气象分析与预测越来越与人们息息相关。从上世纪二、三十年代开始，传统的无线电技术兴起，可以将高空中观测到的气象数据传递给地面，由此人类可以获取到很多观测资料，通过对这些数据的简单处理，就可以进行气象预测。对这些数据进一步统计处理分析，并根据流体学、动力学等物理学的原理，从而构建起一套基本的气象学科框架，并随着时代的进步，逐步对其进行完善。同时，伴随着大数据处理技术的快速发展，越来越多的人开始使用大数据分析工具进行气象分析。以日积月累的庞大气象数据集为基础，借助高效、先进的统计学算法，越来越多的气象规律被发现了。

在本篇论文中，我们将从头开始，使用分布式的储存系统、计算系统来分析数据，使用非阻塞、事件驱动的服务器提供WebAPI接口，使用MVVM设计模式的前端框架来实现数据可视化，一步一步的探索气象大数据分析平台的设计与实现。

关键词：实时，大数据，分析，预测

Financial Wallet Platform

ABSTRACT

With the progress of the information age, meteorological analysis and prediction are gradually becoming closely related to people. From 20 to 30 s of the last century, the traditional radio technology, the observed meteorological data can be high up in the air is passed to the ground, the man can get a lot of observation data, by means of the simple processing of the data, you can proceed to the weather forecast. Data being processing, and according to the fluid study, dynamics, such as the principle of physics, so as to build up a basic framework of meteorological science, and make progress with The Times, to improve step by step. Meteorological science, therefore, is established on the basis of observation data, and by meteorologists through a variety of advanced statistical methods in-depth scientific analysis and research, found that the internal mechanism, and finally

a set of perfect weather discipline theory system. Constantly enrich and perfect as the data is more and more big, the rise of artificial intelligence, a single computer has not enough to deal with so many meteorological data, distributed storage, the development of distributed computing, gradually introduced to the cloud.

Keywords : runtime weather、big data、analyse

目录

一、引言.....6

（一）市场分析.....6

（二）设计目标.....6

1.数据端.....6

2.后端.....6

3.前端.....	6
4.测试.....	6
（三）项目技术应用.....	6
1.数据技术栈.....	6
2.后端技术栈.....	7
3.前端技术栈.....	8
4.项目版本控制工具Git	9
二、系统分析.....	9
（一）系统工程时间预算.....	9
（二）需求分析.....	10
1.数据端.....	10
2.后端.....	10
3.前端.....	11
部分.....	11
需求名称.....	11
进度.....	11
前端.....	11
完成.....	11
完成.....	11
完成.....	11
完成.....	11
完成.....	11
（三）系统生命周期图.....	11
（四）系统用例.....	12
三、系统设计.....	13
（一）系统功能详细分析.....	13
1.系统组件图.....	13
2.系统流程图.....	13
3.系统实现关键代码.....	16
四、系统实施.....	24
（一）系统测试.....	24
1.前端测试内容.....	24
2.后端测试内容.....	24
3.数据端测试内容.....	25
（二）系统效果图.....	25
1.实时天气效果图.....	25
2.气象分析图.....	26
3.历时查看数据展示.....	28
（三）系统工程决算.....	28
五、结束语.....	29
六、致谢.....	29
七、参考文献.....	30

一、引言

（一）市场分析

目前我国互联网平台处于高速发展期，互联网+、大数据这些关键字成为互联网界的热点。伴随日气象数据的迅速增多，我们目前每天产生的气象数据已是PB级别，与上个世纪相比早已不是一个量级。同时在深改和经济转型的经济大环境下，互联网+天气的发展势头迅猛，国家发展气象信息的整体已经非常明确。同时，信息产业低资产运营，投资风险相对较小，气象创业公司已经成为各种资本市场的共同选择。可以预见，像提供更为精准的天气服务，精确到分钟，给物流、农业甚至金融都会带来极大的影响，因此未来气象大数据的市场必定会空间巨大。本次项目的设计充分体现了各个子项目解耦的特性，更符合当前云服务、微服务化的发展趋势。大数据BIGDATA处理技术已经日趋成熟的今天，大数据所带来的经济效益已经在各大互联网公司得到印证，但如何让大数据服务社会，实现其社会价值，是我们需要思考的，这也是本次项目的最终目的。

（二）设计目标

实现一个基于大数据的气象大数据平台，提供实时气象状况查询、历史气象查询、气象分析以及气象预测等功能。本项目将分四部分组成——数据端、后端、前端、测试。

1.数据端

主要负责数据的获取、处理、转换、导出，为前端提供各种个性化指标以及气象预测信息。

2.后端

主要负责提供高性能的Web API接口，为前端提供时间、地区等多维度的查询能力，是数据端与前端沟通的桥梁。

3.前端

主要负责数据的可视化展示，让使用者不仅能在其中获取数据，更能探索数据，让用户能够看到数据中蕴含的秘密。

4.测试

主要负责测试代码的稳定性、可用性，让接口在低密度IO的条件下，依然能够稳定、快速的工作。

(三) 项目技术应用

1.数据技术栈

数据架构：分布式文件系统Hadoop&快速大数据处理Spark

数据组件：hdfs+spark+spark-streaming+kafka

(1) 离线大数据储存——Hadoop

Hadoop是一个可扩展的、可靠的分布式的大数据储存与计算框架，这使得它成为当下云平台架构下的首选。Hadoop主要由两部分组成——HDFS和MapReduce，在本篇文章中，只用到了HDFS。借助Hadoop庞大而活跃的生态系统，我们可以方便的找到完善的文档以及已知问题的解决方案。同时，由于HDFS的高可靠、高容错、高扩展等性质，我们可以方便的在多台廉价的服务器上搭建起集群，从而使用非常低的成本便可以获取数倍提高的计算能力和容错能力。

(2) 快速大数据分析——Spark

Spark也是一种大数据计算框架，但相对于以Hadoop为架构基础的计算框架而言，Spark拥有更快的处理速度，更专业的语言支持，更活跃的社区氛围。因此，Spark已经开始逐渐替代Hadoop在计算领域的功能，而Hadoop则专注于数据的分布式储存和资源管理，二者相辅相成，共同构建了当下最出色的大数据处理架构。

本项目中，Spark的主要作用体现在各种气象指标的分析。

(3) 高性能消息中间件——Kafka

Kafka通常以一个消息中间件身份出现在大数据处理架构中。由于它具有高可靠、高容错、高吞吐、低延迟、持久性等特点，十分适合作为一个实时日志系统的消息队列，与Redis相比，Kafka拥有更加完善的生态系统。通过数据的生产、建立Topic、数据的消费等过程，完成数据日志的实时处理。

本项目中，Kafka的作用主要是模拟真实情况下，实时数据的生产与消费。

(4) 实时数据分析——Spark-Streaming

Spark-Streaming是用来处理实时数据流的框架。通常实时数据会先导入Kafka、Redis等高性能消息队列中等待被消费，然后再使用Spark-Streaming以极快的频率对数据流进行处理。并且，Spark-Streaming是一个非常出色的容错系统，具有极为高效的错误处理和恢复能力。

本项目中，Spark-Streaming的作用主要是实时消费Kafka中的气象数据。

2.后端技术栈

后端架构：nodejs+redis+mongodb

后端组件：express

(1) 高性能内存数据库——Redis

Redis是一个Key-Value内存数据库，支持string、list、set、zset等多种value类型，这些数据类型都支持push、pop、add、remove等操作，且这些操作都是原子性的。Redis可以方便的进行拓展，进行集群的部署，同时为了保证数据的安全性，Redis可以每个一段时间对数据进行一个备份，存到磁盘中以备使用，因此Redis是安全且高效的。

本项目中，Redis的作用主要体现在储存实时产生的气象数据

(2) 高性能海量数据存取数据库——MongoDB

MongoDB是一个NOSQL数据库，非常适合敏捷开发。虽然是一个文档数据库，但却支持像关系数据库那样的二级索引以及完整的查询系统。

此外，MongoDB具有极高的扩展性，我们可以方便的将他部署到集群上，从而得到性能上的成倍提升。与Redis最擅长的实时存取对比，MongoDB更适合进行数据的海量存储和查询。同时，MongoDB又是使用JS写的，可以与Node进行无缝对接，大大减少了我们的开发难度。

在本项目中，MongoDB的作用主要体现在储存所有产生的气象数据，以用于气象分析使用。

(3) 事件驱动非阻塞服务器——Node.js

Node.js是一个基于Chrome V8 JavaScript引擎的JavaScript运行时。nodejs使用一种事件驱动的、非阻塞的输入/输出模型，使其轻量级和高效。Node.js的包管理npm，是世界上最大的开源库生态系统。

本项目中，Node.js的作用主要体现在书写后端代码

(4) web 开发框架——Express

Express是一个基于Node.js的web开发框架，它灵活、极简，可以帮助我们快速创建各种APP和移动设备应用。同时它具有丰富的HTTP快捷方法和任意排列组合的Connect 中间件，让我们可以快速创建健壮、友好的API服务。Express不对Node.js已有的特性进行二次抽象，我们只是在它上面扩展了web应用所需要的基本功能。本项目中，Express的作用主要体现在提供web api接口

(5) Nginx高性能高并发http服务器与反向代理服务器

Nginx其特点是占有内存少，并发能力强，它底层在linux平台上基于epoll实现，epoll是linux上目前最先进的多路io复用方案，其原理是直接网卡驱动建立回调关系，当收到数据包时，是网卡最先收到，然后由驱动层去通知上层，epoll内部是一颗红黑树，用来管理所有关心的事件，也就是对应文件描述符的关联结构，红黑树是一颗近似平衡的搜索树，查找时具有高效的性能，接近于lg2n的时间复杂度，而且插入删除性能也较好。Nginx在这里我们用来管理我们的静态资源，比如html文件,图片文件，css样式文件等。

(6) Node进程管理工具——forever

Forever是一个Node进程管理工具，可以方便的管理、重启Node进程，保证我们Node服务的稳定性、健壮性。

3.前端技术栈

前端架构：Node.js+Vue

前端框件：ElementUI+Vuex+VueRouter+Axio

(1) 渐进式MVVM框架——Vue

Vue是一个渐进式的MVVM框架，它简单小巧，却不乏大将风范。通过Vue我们可以方便的构建一个基于MVVM设计模式的单页面应用程序。相对于JQuery生硬的手动修改DOM的方式，Vue使用数据双向绑定的方式来渲染页面，数据变化页面则变化，页面变化数据也会变化。此外，借助Vue官方的Vuex和VueRouter框架，我们可以专注于书写业务核心代码，极大提高我们的工作效率。

(2) 基于Vue2.0的桌面端组件布局框架——ElementUI

ElementUI是饿了么团队开发的一套UI框架，目前已经是Vue系列框架的佼佼者。目前ElementUI基于2.0开发，可以和Vue完美配合使用。它的设计符合一致性、反馈性、效率性、可控性的原则，可以帮助用户快速构建起一个后台管理系统。

(3) 全局状态管理框架——Vuex

Vuex通常作为一个全局状态管理工具体现在项目中，单页面应用最让人苦恼的问题就是全局状态共享，Vuex的出现解决这一痛点。我们通过配置Vuex，得到一个全局状态文件，其中属性的变化也会体现在Vue的数据双向绑定中，让每个组件都可以实时获取、更改当前的全局状态，从而让整个应用程序更加可控、代码更加优雅。

(4) 单页面应用路由框架——VueRouter

VueRouter是一个单页面应用的路由管理框架，使用Vue+VueRouter我们可以方便的创建单页面应用。我们只需要配以一下路由配置文件，便可以实现组件页面与路由的映射。

4.项目版本控制工具Git

Git是一个免费的开源分布式版本控制系统，它的设计目的是为了用于各种团队开发项目，管理代码版本，使开发拥有更快的开发速度和开发效率。Git很容易学习，很容易上手，几个命令就可以完成代码的提交，拉取，并且有一个很小的内存占用，并且具有闪电般的快速性能。它超越了诸如Subversion、CVS、Perforce和ClearCase这样的SCM工具，具有本地分支，在不能连接服务器时也可以提交代码、方便团队的协同合作开发。Git在我们系统中用于管理代码，我们三人写的代码，都是每人一个分支，写完一个功能后，在合并到主分支进行测试。

二、系统分析

系统设计是对整个系统的结构的规划以及宏观控制的关键环节，主要是根据系统需求，系统的功能，用例等，分析应该设计什么模块，大体规划使用什么框架去实现，是进行系统评估的重要环节。前期的分析和设计非常重要，性能和优雅是个需要权衡的东西，前期的分析与设计就要设计好，优雅的设计可以减少后期很多工作，使得系统更容易实现低耦合高内聚的设计目标，同时要确保了代码对内部修改的关闭，和对扩展的开放，符合OO的设计思想，模块化管理。

(一) 系统工程时间预算

系统的开发工作一般分做四个阶段：设计阶段，开发阶段，集成阶段和测试阶段。这四个阶段也可能交叉进行，比如开发阶段也会进行简单的单元测试，一个模块开发好了也会进行集成测试，中间开发也可能因为前期设计的不合理，而要进行重新设计。

(二) 需求分析

本项目由三部分组成：数据端、后端、前端，其中各自的需求分析如下：

1.数据端

获取原始气象数据，导入Kafka消息队列，准备消费。再使用Spark消费Kafka数据,进行ETL转化,转化成一份符合标准规范的日志,并持久化进Redis数据库，最后使用Spark、SparkStreaming分析日志，计算各种指标

部分需求名称进度

数据端获取原始数据集完成

搭建大数据分析服务器完成
原始数据使用Shell脚本导入HDFS 完成
使用SparkStreaming模拟实时数据的产生并导入Kafka 完成
使用Spark消费Kafka数据，进行ETL转化，并导入Redis 完成
完成实时数据的ETL 完成
完成预测数据的ETL 完成

表2-1 系统需求表

2.后端

使用Node+Express构建高性能WebAPI服务，借助Node的事件驱动、非阻塞IO的特性，为前端提供稳定的支持高并发IO的接口服务。基于数据端提供的格式化的数据，搭建起数据端与前端沟通的桥梁。此外，实时数据和历史数据是分别储存在Redis和MongoDB里的，这是为了充分利用这两个数据库的特性——实时查询和海量数据查询。

部分需求名称进度

后端搭建Node+Express服务完成
Redis、MongoDB数据库设计完成
完成Dao层代码设计完成
完成Route层代码设计完成
完成Controller层代码设计完成
完成城市列表接口完成
完成实时数据接口完成
完成不同时间筛选粒度的数据接口完成
完成预测数据接口完成

表2-2 系统需求表

3.前端

使用Node+Vue构建一个基于MVVM设计模式的气象大数据可视化平台。使用Vuex管理全局应用状态，是的每个组件都能方便的获取、更改全局状态。使用VueRouter管理路由与组件的映射关系。

部分需求名称进度

前端搭建Node+Vue服务完成
完成顶部水波动画完成
完成实时日期模块完成
完成实时气象状态模块完成
完成实时预测模块完成
完成实时风速模块完成
完成实时仪表盘模块完成
完成气象分析趋势图模块完成
完成气象分析区域筛选模块完成
完成气象分析指标同比对比表格模块完成
完成气象分析指标切换模块完成
完成历史数据数据总量模块完成
完成历史数据下载模块完成

表2-3 系统需求表

(三) 系统生命周期图

本系统的启动需要集群的支持，所以先要启动HDFS、Spark、ZooKeeper和Kafka，之后我们会分别运行数据生产者任务和数据消费者任务，到此数据端启动完毕。

数据端启动完毕后，便可以启动Node服务器了，同时启动Express WebAPI服务，到此后端启动完毕。

最后启动前端服务，浏览器便可以访问了。

图2-1 系统生命周期图

(四) 系统用例

图2-2 系统用例图

系统用例说明表

用例名称系统整体用例图

用例正流程 1.用户注册登录2.后端校验信息3.进入气象可视化平台4.进行Ajax请求5.后端接收请求，连接数据库，获取数据

6.根据数据渲染页面

表2-4 用例说明表

三、系统设计

功能流程是程序设计的重要工作环节，对于业务流程详细，合理的设计是确保系统稳定性，健壮性和友好性的重要保证。功能流程以用例规约表和活动图为主要设计手段，必要时以流程图为辅助。

流程设计的设计应力求简明扼要，清晰准确的描述系统功能的流程结构。以下为本系统的功能流程设计图表。

(一) 系统功能详细分析

1. 系统组件图

图3-1 系统组件图

2. 系统流程图

(1) 整体

从整体上来看，我们的后端作为一个桥梁沟通着数据端和前端。数据端会产生各种指标数据与预测数据存入数据库，同时后端会搭建WebAPI接口服务，响应前端页面的Ajax请求，对数据库进行增删查改的操作。

图3-2 系统组件图

(2) 数据端

在数据端内部，先把原始气象数据通过shell脚本存入HDFS中备用。之后，为了模拟实时数据的产生，我们会启动一个Spark任务去读取HDFS上的所有气象数据，并每隔3s导入到Kafka消息队列中。同时，我们会另起一个SparkStreaming任务去实时分析Kafka的weather topic中的数据，并实时导入Redis中，以此来模拟实时数据的产生。另一方面，为了满足各种个性化指标的展现要求，我们会使用Spark对HDFS的气象数据进行离线分析，通过特点算法得出各种个性化数据，这些数据也会被导入数据库中，供后端使用。

图3-3 系统流程图

(3) 后端

在后端内部，我们启动了一个Node服务器，使用Express做了一个WebAPI服务。它会实时接收前端的请求，并不断向Redis中读取实时数据，与此同时，读取的实时数据也会往MongoDB中导入一份，这是为了给气象分析使用。

图3-4 系统流程图

图3-5 系统流程图

(4) 前端

在前端内部，我们启动了一个Node热加载服务器和Vue单页面应用程序。整个前端页面采用了MVVM的设计模式，数据的改变会自动触发新的网络请求，从而更新当前页面的数据，数据会像水一样在程序内部流动，十分优雅。

图3-6 系统流程图

3. 系统实现关键代码

(1) 数据端

关键代码：分析数据导入Kafka

说明系统会启动多个线程，分别取模拟多个城市的气象站同时去获取实时气象数据。

片段1//提交5个线程cities.foreach { city => threadPool.execute(new ThreadTask(city)) }

片段2// 获取HDFS数据并导入Kafka
val reader = Try(HdfsUtils.readGZFile(fileSystem, s"/root/clouddata/\$year/\${city._1}-\$year.op.gz"))
reader match { case Success(r) => // 一行是一天
var lines = Stream.continually(r.readLine()).takeWhile(_ != null)
lines.toArray.slice(1, lines.length).foreach { line => logger.info(s"### sending kafka msg \$city ###")
val newline = line.replace(city._1.replace("-", " "), city._2)
producer.send(new ProducerRecord(topic, newline))
logger.info(s"### sent kafka msg: \$newline ###")
Thread.sleep(3000)
case Failure(f) => logger.warn(s"### year: \$year-\$city 获取BufferedReader失败 ###"); f.printStackTrace() }

表3-1 系统代码实现表

关键代码：实时消费Kafka数据

说明我们会启动一个SparkStreaming任务去实时消费Kafka数据

片段1 // 连接Kafka Streaming
val kafkaStream = KafkaUtils.createDirectStream[String, String, StringDecoder, StringDecoder](ssc, kafkaParams, topics)

片段2// 遍历streamData这个RDD集合并将数据导入Redis
streamData.foreachRDD { rdd => rdd.foreach { data => val jedis = pool.getResource
jedis.select(1)
logger.info(s"### writing into redis with data: \${new Gson().toJson(data.get)}")
jedis.rpush(data.get.city, new Gson().toJson(data.get))
jedis.close() }

表3-2 系统代码实现表

(2) 后端

关键代码：区间数据查询

说明进行区间数据的查询

片段1 // 连接Redis
client = redis.createClient(redisPort, redisIp);

片段2// 根据日期为index的方式进行区间查询
var startIndex = await insertValueSearch(startTime, false, true, lindexAsync,

keyName, count);if (startIndex == -1)throw "no find start in the range";var endIndex = await insertValueSearch(endTime, true, false, lindexAsync, keyName, count);if (endIndex == -1)throw "no find end in the range";ret = await lrangeAsync(keyName, startIndex, endIndex);

表3-3 系统代码实现表

关键代码：api路由注册

说明注册api路由关键代码

片段1 // 注册/query路由router.get('/query', async (request, response, next) => {let city = request.query.city;if (!city) {city = "济南";}let startTime = parseInt(request.query.startTime);let endTime = parseInt(request.query.endTime);if (isNaN(startTime) || startTime < 0)startTime = 0;if (isNaN(endTime))endTime = 20190101;var dataArray = [];let queryRes = await queryDataByRange(startTime, endTime, city)for (var i in queryRes) {var dataString = queryRes[i];dataArray.push(JSON.parse(dataString));}sendJson(response, 0, "invoke ok!", dataArray);}

表3-4 系统代码实现表

(3) 数据库

关键代码：数据库索引优化

说明对数据库的时间列进行索引优化，加快时间区间筛选的速度

Redis差值查找算法 // redis list中数据是按时间排序的，二分查找仍为log2n复杂度，目前使用了比二分还快的查找算法，即插值查找，按照数据值根据比例划分查找，达到秒查的效率。

指 标
疑似剽窃文字表述
1. Spark-Streaming是一个非常出色的容错系统，具有极为高效的错误处理和恢复能力。
2. 丰富的HTTP快捷方法和任意排列组合的Connect 中间件，让我们可以快速创建健壮、友好的API服务。Express不对Node.js已有的特性进行二次抽象，我们只是在它上面扩展了web应用所需要的基本功能。

2. 程前+20140614204+管理科学与工程学院_第2部分	总字数：3730
相似文献列表 文字复制比：4.1%(152) 疑似剽窃观点：(0)	
1 王智超 20130614235 管理科学与工程学院	4.1% (152)
- 《大学生论文联合比对库》- 2017-04-26	是否引证：否

原文内容 红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

//letlet partition = startIndex + parseInt((value - dataStartTime) * (endIndex - startIndex + 1) / (dataEndTime - dataStartTime));if (partition < startIndex)partition = startIndex;if (partition > endIndex)partition = endIndex;let partitionJson = JSON.parse(await lindexAsync(redisKeyName, partition));let partitionValue = parseInt(partitionJson.date);if (value < partitionValue) {endIndex = partition - 1;} else if (value > partitionValue) {startIndex = partition + 1;}else {findIndex = partition;break;}

MongoDB

表3-5 系统代码实现表

(4) 前端

关键代码：基于MVVM的设计模式

说明基于MVVM，筛选值变化会自动发起Ajax请求，重新渲染页面

片段1 // 监听筛选属性的变化watch: {result: {deep: true,immediate: true, // 立即执行handler: async function () {const data = {}data.startTime = this.result.startTimedata.endTime = this.result.endTimedata.city = this.result.city}}

片段2 // MVVM绑定<ve-line :data="chartData":settings="chartSettings"height="500px"></ve-line>

表3-6 系统代码实现表

关键代码：使用Vuex进行全局状态管理

说明 Vuex来管理全局的筛选条件，是的所有组件都能够方便使用

片段1 // 注册全局状态属性import Vue from 'vue'import Vuex from 'vuex'Vue.use(Vuex)const state = {filterData: {todayWeather: {city: '济南'},temperature: {city: '济南',timeType: 'date',startTime: new Date().getTime() - 3600 * 24 * 1000 * 7,endTime: new Date().getTime()}}...}

片段2 // 更改、获取全局状态computed: {city: {get () {return this.\$store.state.filterData[this.\$route.name].city},set (value) {this.\$store.commit('updateFilterData', {key: this.\$route.name,value,type: 'city'})}},

表3-7 系统代码实现表

四、系统实施

(一) 系统测试

系统测试是系统开发的重要工作环节，也是系统开发过程中必须要做的工作。本系统的系统测试内容主要分为黑盒测试和白盒测试，其中黑盒测试为主，白盒测试为辅。因篇幅问题，此处不再单独展示测试用例和相关代码，只提供相应的测试结果。

本系统测试将分为三部分：前端测试、后端测试、数据端测试。

1.前端测试内容

主要是测试Vue组件、Utils函数的可复用性、健壮性

前端测试计划表

components/TimeFilter.vue 测试TimeFilter对各个页面的支持情况，保证对每个页面都能表现出统一的性质

utils/fetch.js 测试fetch对象的拦截器是否正常工作，能否适当处理各种类型的请求

utils/timeData.js 测试timeData函数是否能根据不同的starttime、endtime、timeType产出正确的时间区间数据

表4-1 系统测试说明表

2.后端测试内容

主要是测试插值排序、实时接口性能测试、分析接口区间筛选正确性

后端测试计划表

routes/api.js 测试接口的并发性能以及稳定性

routes/api/insertValueSearch 测试insertValueSearch的性能、安全性、稳定性

routes/api/queryDataByRange 测试queryDataByRange的区间筛选是否能正确返回

routes/api/sendJson 测试sendJson的跨域设置是否正常

表4-2 系统测试说明表

3.数据端测试内容

主要是测试数据生产者、数据消费者的稳定性

数据端测试计划表

weather.analysis.RealtimeAnalysis 测试实时分析类（消费者）的稳定性

weather.producer.KafkaProducer 测试数据生产者的稳定性，同时消除内存泄漏的可能性

表4-3 系统测试说明表

本系统测试工作以确保系统可使用，保证系统健壮性和稳定性为目标。测试工作以单元测试为主要途径，黑盒测试和白盒测试相结合的方式进行测试。

（二）系统效果图

1. 实时天气效果图

图4-1 系统实现图

2.气象分析图

（1）按风速查看

图4-2 系统实现图

（2）按气温查看

图4-3 系统实现图

（3）按气压查看

图4-4 系统实现图

（4）按降水查看

图4-5 系统实现图

3、历时查看数据展示

图4-6 系统实现图

（三）系统工程决算

工程预算和决算时间表如下：

（1）设计阶段用例设计半日，流程设计半日。

（2）数据端开发预算15日，实际10日。

（3）前端开发预算20日，实际15日。

（4）后端开发预算5日，实际5日。

（5）开发阶段系统环境架构搭建1日。

（6）集成阶段前后端对接3日，实际6日。

（7）测试阶段共计7日

系统工程预决算表

工程阶段预算数决算数超标值

设计阶段 1 1 --

代码开发工作 40 30 --
前后端对接 3 6 100%
系统覆盖率测试 1 2 100%
系统测试 7 4 --

表4-4 系统预算表

五、结束语

本系统以气象大数据平台为设计目标，截止至2018年4月25日工程结束本系统已经初步具备数据分析系统平台架构以及包含基本模块。

通过张贵杰老师的指导，以及吸纳学院其他各位老师意见和建议，通过本次毕业设计项目，我们不仅在文档设计，需求分析，系统设计，编码以及系统测试、实施等一系列系统开发的全部生命周期流程都有极为深刻的学习和认识，并设计能力和开发能力也得到了提升，对于书面知识有着自己更深刻的理解。同时，我们也对大数据分析的需求和业务流程有一定的了解和认识。

通过本次毕业设计，我们三人对自己学习的领域有了更加深刻的认识。我们尝试了很多在IT领域新研的技术，没有了工作上的压力，让我们有了更大的勇气去尝试新技术、去探索各种高效的算法，这也使我们的开发能力和知识面得到了很大的提高，这是我们工作中极难得到的。

本次项目不免也有许多遗憾，由于资金有限，我们无法租到多台高性能主机，只能用一个2G内存的阿里云小服务器，这使得我们数据分析能力大打折扣。由于时间有限，有许多细节我们也无法顾及，但是我们还是完成了本次项目的核心代码。

总之，感谢本次毕设的经历！

六、致谢

本次毕业设计的从设计到开发离不开学院各位老师的帮助与教诲，尤其张贵杰老师对我们的工作进行了深入而有见地的指导，如果没有各位帮助支持，可能我们会走更多的弯路。本次毕业设计之所以取得我们预想的一个良好成果，与学院的各位老师的鼎力帮助支持密不可分，在这里，我们三个人首先向给予本次毕业设计指导帮助的各位老师，特别是我们的指导老师——张贵杰老师表达诚挚的感谢！

大学四年来，在校园中我们感受到了学院领导的教育以及各位老师的悉心栽培，四年的大学学习时光即将落幕，曾经遇到的挫折和受到的鼓励仍然历历在目，因此我们大学四年得以全面成长，这些离不开各位老师的密切找到。在理论、技术以及社会实践等方面获得全面的发展，以及更深的见解。相信我们走上工作岗位以后，我们也将继续承载着这份栽培与教诲，带着曾经在校园中收获的知识，不忘初心，继续前行，继续奋斗，努力。

最后，我们三个人再次向张贵杰老师以及为此次毕业设计提供各种支持帮助的学院老师们致以真诚的感谢！

感谢各位老师的悉心指导！

七、参考文献

[1] 朴灵.深入浅出Node.js [M]. 人民邮电出版社，2013（1）：10-188.
[2] 王嘉琳.大数据Spark企业级实战 [M]. 电子工业出版社，2015（1）：1-49.
[3] 陆嘉恒.Hadoop实战 [M]. 机械工业出版社，2011（9）：45-99.
[4] Tom White.Hadoop: The definitive Guide [M]. 人民邮电出版社，2013（1）：77-129.
[5] Ethan Brown.Node与Express开发 [M]. 人民邮电出版社，2015（1）：7-99.
[6] 梁灏.Vue.js实战 [M]. 清华大学出版社，2017（10）：129-243.
[7] Cay S. Horstman.快学Scala [M]. 人民邮电出版社，2017（7）：1-248.

指 标
疑似剽窃文字表述
1. 年来，在校园中我们感受到了学院领导的教育以及各位老师的悉心栽培，四年的大学学习时光即将落幕，曾经遇到的挫折和受到的鼓励仍然历历在目，因此我们大学四年得以全面成长，
2. 方面获得全面的发展，以及更深的见解。相信我们走上工作岗位以后，我们也将继续承载着这份栽培与教诲，带着曾经在校园中收获的知识，不忘初心，继续前行

说明：1.总文字复制比：被检测论文总重合字数在总字数中所占的比例
2.去除引用文献复制比：去除系统识别为引用的文献后，计算出来的重合字数在总字数中所占的比例
3.去除本人已发表文献复制比：去除作者本人已发表文献后，计算出来的重合字数在总字数中所占的比例
4.单篇最大文字复制比：被检测文献与所有相似文献比对后，重合字数占总字数的比例最大的那一篇文献的文字复制比
5.指标是由系统根据《学术论文不端行为的界定标准》自动生成的

6. 红色文字表示文字复制部分;绿色文字表示引用部分

7. 本报告单仅对您所选择比对资源范围内检测结果负责



 amlc@cnki.net

 <http://check.cnki.net/>

 <http://e.weibo.com/u/3194559873/>

“中国知网”大学生论文检测系统