文本复制检测报告单(全文标明引文)

№:ADBD2016R_20160524193951420601867238 检测时间：2016-05-24 19:39:51

检测文献： 慕课论坛爬取系统的设计与实现

作者： 贺翔宇

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
大学生论文联合比对库
互联网资源
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
优先出版文献库
互联网文档资源
图书资源
个人比对库

时间范围： 1900-01-01至2016-05-24

指导教师： 董永权

检测结果

总文字复制比： 6.3% **跨语言检测结果：** 0%

去除引用文献复制比：6.3% **去除本人已发表文献复制比：6.3%**

单篇最大文字复制比：3.6%

重复字数： [1069]	总字数： [17093]	单篇最大重复字数： [615]
总段落数： [6]	前部重合字数：[109]	疑似段落最大重合字数：[658]
疑似段落数：[4]	后部重合字数：[960]	疑似段落最小重合字数：[85]

指　标： ☐ 剽窃观点 ☑ 剽窃文字表述 ☐ 自我剽窃
☐ 一稿多投 ☐ 过度引用 ☐ 整体剽窃 ☐ 重复发表

表格：**0** 脚注与尾注：**0**

（注释： 无问题部分 文字复制比部分 引用部分）

1. 中英文摘要等

总字数：**1875**

相似文献列表 **文字复制比：**0%(0) **剽窃观点（0）**

原文内容

JIANGSU NORMAL UNIVERSITY

本科毕业设计

UNDERGRADUATE DESIGN

设计题目： 慕课论坛爬取系统的设计与实现

姓名： 贺翔宇

学院： 智慧教育学院(计算机科学与技术学院)

专业： 软件工程

年级 、 学号： 2012级、12267032

指导教师： 董永权

江苏师范大学教务处印制

设计原创性声明

本人郑重声明：所呈交的毕业设计，是在导师的指导下，独立进行研究所取得的成果，所有数据、图片资料真实可靠。除文中已经注明引用的内容外，本设计的研究成果不包含他人享有著作权的内容。对本设计所涉及的研究工作做出贡献的个人和集体，均已在设计中以明确的方式标明。本设计的知识产权归属培养单位。

本人签名： 年月日

设计版权使用授权书

本设计“慕课论坛爬取系统的设计与实现”是本人在校期间所完成学业的组成部分，是在江苏师范大学教师的指导下完成的，因此，本人特授权江苏师范大学可将本毕业论文的全部或部分内容编入有关书籍、数据库保存，可采用复制、印刷、网页制作等方式将论文文本和经过编辑、批注等处理的论文文本提供给读者查阅、参考，可向有关学术部门和国家有关部门或机构呈送复印件和电子文档。本毕业论文无论做何种处理，必须尊重本人的著作权，署明本人姓名。

作者签名： 指导教师签名：

年月日年月日

慕课论坛爬取系统的设计与实现

摘要

随着互联网Web技术的高速发展，以慕课为代表的在线课程获得了越来越多的关注，越来越多的教学资源由线下转移到了线上。而现今大量慕课系统的网站使用JavaScript及Ajax技术搭建，传统网络爬虫则面临运行网页上JavaScript脚本及加载异步网页内容等方面的挑战。面对这些挑战，能够处理页面脚本并渲染动态内容的爬虫应运

而生。本系统通过调用浏览器接口，使用JavaScript和Python编程语言，并结合PostgreSQL数据库开发，能够处理采用大量前端技术搭建的网站，并且结合任务队列和消息机制，能够实现一定程度的并行爬取。本系统同时实现了控制台及数据可视化功能，能够对爬虫活动进行实时监控，并分析、统计爬取的数据。

该论文有图16幅，表5个，参考文献20篇。

关键词：慕课论坛爬取系统论坛爬取系统爬取系统

Design and Implementation of A Crawling System for MOOC Forum

Abstract

With the rapid development of the Web technology, the online course represented by MOOC has gained more and more attention, and more and more teaching resources have been transferred from the offline to the online. And today a large number of MOOC system website using JavaScript and Ajax technology to build, the traditional web crawler is facing the challenge of running the JavaScript script on the web page and load asynchronous web content and other aspects. In the face of these challenges, can handle the page script and rendering the dynamic content of the crawler came into being. The system by calling the browser interface, using JavaScript and python programming language, combined with the PostgreSQL database development, are able to deal with large number of front-end technology to build a website, and the combination of task queue and message mechanism, to achieve a certain degree of parallel crawling. At the same time, the system can realize the function of the console and data visualization, and can monitor the activities of the reptiles in real time, and analyze the data.

Key Words: Crawling System for Mooc Forum Crawling System for Forum Crawling System

目录

图清单

表清单

变量注释表
currentPage 当前页面
SubjectCode 课程代码
mainInterval 主调度

---

2. 第1章绪论 总字数：**1299**

相似文献列表 **文字复制比：** 8.4% (109) **剽窃观点（0）**

1	服务器集群监控系统的设计与实现 朱瑞斌(导师：赵宏) - 《北京交通大学硕士论文》- 2015-05-01	5.0%（65） 是否引证：否

2	一场最重要的教育方式变革 陈玉琨; -《未来教育家》- 2014-02-06	3.0%（39） 是否引证：否

原文内容

第1章绪论

1.1 课题背景及研究意义

1.1.1 课题背景

慕课[1]（MOOC，massive open online courses）即大型开放式网络课程，是新近涌现出来的一种在线课程模式，它将在线学习管理系统与开放的网络课程资源综合起来，形成了一种新的课程开发模式。为了提升用户体验，如今很多慕课系统使用大量JavaScript技术进行开发，使得传统爬虫在应对这些网页时遇到很多困难，为了从慕课系统的大量课程及讨论资料中获得有价值的信息，一个针对慕课的爬虫和数据分析系统成为迫切需求。

1.1.2 研究意义

在当今时代，互联网的出现为教育改变提供了数字化的支撑，让优质的教育资源得以高效地传输，开放课程资源、推进教育公平势在必行。本系统通过对慕课系统大量课程的讨论资料进行整理、总结，帮助教育工作者获得课程的重点、难点，从而提高教学质量。

1.2 开发工具的选择及语言介绍

1.2.1 JavaScript简介

JavaScript[2]是一种高级的、动态类型的、弱类型的解释型语言，在ECMAScript语言规范中被标准化。它与HTML 、CSS一起，是World Wide Web最重要的3个核心技术。当前，绝大多数网站使用JavaScript，并且主流的浏览器可以无需插件支持JavaScript的运行。

JavaScript是一种基于原型的、支持头等函数语言，这些特点使之对多种编程泛型的都有良好的支持，如面向对象、命令式编程和函数式编程。

1.2.2 Python简介

Python[3]是一种面向对象、解释型的编程语言，拥有高效的高级数据结构，并且能够用简单而又高效的方式进行编程。

Python是一门扩展性很强的语言，可以很容易的使用 C 或 C++（其他可以通过 C 调用的语言）为 Python 解释器扩展新函数和数据类型。同时，它包含了一组功能完备的标准库，能够轻松完成很多常见的任务。

1.3 本文研究内容及主要贡献

本文主要介绍了慕课论坛爬取系统的背景、意义、整体设计思路以及相关技术等。

慕课论坛爬取系统能够有效的实现对网易云课堂讨论区的爬取，并且实现对爬取数据的持久化保存，以及对数据的整理、查询和可视化操作。系统采用B/S技术实现，用户不需要安装任何应用软件即可体验体统功能。系统在设计时，采用WebSocket技术，使用PostgreSQL数据库和Tornado服务器实现。

本系统包括爬取和数据可视化可视化两部分。可以实现对不同课程的选择性爬取，用户可以通过图表形式可视化地浏览数据，可以使用表单查询的方式对系统资源进行交互。

本文的章节内容安排如下：

第1章：绪论。主要详述了课题研究的背景、意义、开发工具的选则、本文的研究内容和主要贡献。

第2章：需求分析。主要介绍了系统功能上需求和性能上的需求。

第3章：系统总体功能模块设计。介绍了功能模块的划分以及数据库的设计。

第4章：系统实现过程。介绍了爬取模块和数据可视化模块的具体实现，以及相关的代码。

第5章：关键技术。介绍了本系统的安装和配置步骤。

指　标

剽窃文字表述

1. 时代，互联网的出现为教育改变提供了数字化的支撑，让优质的教育资源得以高效地传输，

3. 第2章需求分析

总字数：**1193**

相似文献列表　　**文字复制比：** 18.2% (217)　　**剽窃观点（0）**

1	数据库课程管理系统的设计与实现 邱盈盈 - 《大学生论文联合比对库》- 2015-05-19	18.4%（219） 是否引证：否
2	数据库课程管理系统的设计与实现 邱盈盈 - 《大学生论文联合比对库》- 2015-05-17	17.4%（207） 是否引证：否

原文内容

第2章需求分析

2.1 功能需求

2.1.1 爬取功能

1）课程主页

对课程主页的爬取要求获取慕课系统的所有课程信息，以及每个课程的开课情况，如正在开课、已经结束等等。

2）帖子列表页

对帖子列表页的爬取要求获得一个课程的讨论区的所有发帖，并记录帖子对应的超链接、页号，为进一步爬取帖子的详细信息做准备。

3）帖子详情页

对帖子详情页的爬取要求获得一个帖子的全部详细信息，如标题、作者、正文、创建日期、点赞数、评论数等等，所获取的数据要求写入数据库做持久化保存和之后进一步的分析和展示。

4）评论部分

对评论部分的爬取要求获得一个帖子的所有评论内容，包括正文、作者、创建时间、点赞数、子评论等，所获取的数据要求写入数据库做持久化保存和之后进一步的分析和展示。

2.1.1 数据可视化功能

1）爬虫活动可视化

该模块要求以一个较短的时间间隔实时地监控爬虫的运行状况和相关活动，并对爬虫的网络流量以可视化的形式做出展示。

2）数据统计

该模块要求对已经爬取的数据做出数量上的统计，如课程数、发帖数、评论数、用户数等。

3）活跃用户

该模块对慕课系统中活动频率较高的用户做出显示，并对相关用户做出活跃度排序、发帖数统计等。

4）词频统计

该模块对于用户输入的词语，生成该词语近期在评论区中的出现频率。

2.2 性能需求

2.2.1 系统的软件环境

* 数据库服务器。

PostgreSQL数据库、PostgREST提供REST API。

* Web服务器。

1） Nginx 1.8.1

2 ） Tornado 4.3

* 客户端计算机。

1 ） OS X 10.10

2 ） Google Chrome 50

2.2.3 系统的性能要求

1 ）并发需求：要求系统具有一定的并发爬取能力以充分利用硬件资源。

2 ）磁盘容量要求：本网站是基于B/S的架构，所以，在存储容量方面，网站部分所用空间不大。但是，爬虫的数据库需要较大的存储空间。

3 ）适应性要求：要求系统的功能模块清晰，模块之间具有较强的内聚性，较低的耦合性，能够使用户在很短的时间内熟悉系统的整个操作流程。

2.3 可行性分析

可行性分析是指在现有的组织环境下，分析一个系统的开发工作是否已经具备了必要的工作条件及资源。

2.3.1 系统业务流程调查

本系统的工作流程大致可以分为两部分: 一部分是从慕课论坛爬取数据存入数据库。另一部分是对数据进行可视化显示。

2.3.2 系统可行性调查

1 ）经济的可行性：经过开发测试，本系统可以在普通个人PC和一般的网络状况下运行，对机器性能的要求不高，且爬取效率较高，具有较高的经济可行性。

2 ）技术可行性：本系统主要采用前后端分离的方式设计开发。这种架构具有较好的可扩展性以及较低的耦合性，便于系统的开发与维护。

指　标

剽窃文字表述

1. 清晰，模块之间具有较强的内聚性，较低的耦合性，能够使用户在很短的时间内熟悉系统的整个操作流程。 2.3 可行性分析可行性分析是指在现有的组织环境下，分析一个系统的开发工作是否已经具备了必要的工作条件及资源。 2.3.1 系统业务流程调查本系统的工作流程大致可以分为两部分:
2. 可行性。 2 ）技术可行性：本系统主要采用前后端分离的方式设计开发。这种架构具有

4. 第3章系统总体功能模块设计　　总字数：**1119**

相似文献列表　　**文字复制比：** 7.6% (85)　　**剽窃观点（0）**

1	基于Android平台的O2O美发销售应用 黄少斌 - 《大学生论文联合比对库》- 2015-05-09	5.7%（64） 是否引证：否
2	无线路由器管理系统的研究 张小玲(导师：胡怡红) - 《北京邮电大学硕士论文》- 2013-12-07	3.9%（44） 是否引证：否
3	空间数据的分布式存储与管理的设计与实现 孙杰(导师：黄岚) - 《吉林大学硕士论文》- 2015-04-01	3.8%（42） 是否引证：否

原文内容

第3章系统总体功能模块设计

3.1系统功能模块的划分

系统的功能模块主要分为爬取模块和可视化模块，如图3-1所示，两个模块与统一的服务端连接，再连接至数据库。

图3-1系统模块图

3.1.1 数据爬取模块

数据爬取模块采用树状结构爬取讨论区内容，如图3-2所示。

图3-2爬取结构

3.1.2 数据可视化模块

数据可视化模块主要负责数据展示，可细分为爬取活动的实时监控、用户活跃度统计、词语趋势统计和主题模型统计，其模块结构如图3-3所示。

图3-3数据可视化模块结构图

3.2 数据库设计

本系统使用PostgreSQL[4]作为数据库，PostgreSQL是一个关系型数据库管理系统，强调可扩展性和与标准的兼容性。作为一个数据库服务器，其主要功能是安全地存储数据，并且为软件的数据获取请求提供服务。

本系统使用PostgREST作为数据库的RESTful API前端，达到方便开发的目的。

3.2.1 表设计

系统的表设计如表3-1至表3-3所示。

表3-1 课程表

列名数据类型允许空默认值说明

id（主键）

name text F课程名

state text F课程名（正在开课、已结束等）

code text F课程代号

表3-2 帖子表

列名数据类型允许空默认值说明

id（主键）

post_id integer F慕课论坛内部ID

subject_id integer F帖子对应的课程ID

title text F帖子标题

score integer F帖子评分

content text F帖子正文内容

user_name text T用户名

user_id integer T慕课论坛内部用户ID

created_at timestamptz F创建时间

表3-3 评论表

列名数据类型允许空默认值说明

id（主键）

post_id integer F帖子的慕课论坛内部ID

parent_id integer F上级评论的ID

score integer F评论评分

content text F评论正文内容

user_name text T用户名

user_id text T慕课论坛内部用户ID

replies_count integer F 0 回复数

created_at timestamptz F创建时间

3.2.2 数据库ER图

本系统数据库ER图如图3-4所示。

图3-4 数据库E-R图

3.3 本章小结

本章主要介绍了系统主要功能模块的划分，分为爬取模块和可视化模块，以及它们之间的组织与联系。

---

5. 第4章系统实现过程

总字数：**8845**

相似文献列表　　**文字复制比：**0%(0)　　**剽窃观点（0）**

原文内容

第4章系统实现过程

4.1 页面内容抽取

本爬取系统的网站页面抽取主要涉及三个部分：论坛列表页、帖子正文页和评论区页面。其中，评论区的抽取涉及到Ajax动态加载的异步分页技术，在实现时做了特殊的处理，以爬取到帖子的全部评论内容。

4.1.1 论坛页面的内容抽取

论坛页面的抽取内容主要包括页面上所有帖子的地址，在该页面的抽取并不涉及帖子的详细信息如标题、发帖人等，其页面结构如图4-1所示。

图4-1 论坛页面

该页面的抽取函数如下：

```
function describePage() {
var subjectCode = location.pathname.match(/\w+-\d+$/)[0];
var match = location.hash.match(/p=(\d+)$/),
currentPage = match ? parseInt(match[1]) : 1; // 获取当前页码
var getPosts = function() {
var posts = [];
$(".u-forumli").each(function(idx, el) {
// 获取页面信息
var post_id = $(el).find(".j-link").attr("href").match(/pid=(\d+)/)[1],
date = $(el).find(".f-fc9").text().match(/(\d+)年(\d+)月(\d+)/),
year = date[1],
month = date[2],
day = date[3];
var post = {post_id: post_id,
date: new Date(year, month, day)};
posts.push(post);
});
return posts;
};
// 等待页面完成渲染
var interval = setInterval(function() {
if ($(".zpgi").length > 0 &&
$(".u-forumli").length > 0) {
```

```
var totalPage = parseInt($(".zpgi").last().text());
var page = {subject_code: subjectCode,
current_page: currentPage,
total_page: totalPage,
posts: getPosts()};
// 传递当前页面内容
chrome.runtime.sendMessage({message: "describe-page",
page: page});
if (currentPage > 1) {
chrome.runtime.sendMessage({message: "close-me"});
}
clearInterval(interval);
}
}, 200);
}
```

4.1.2 帖子页面的内容抽取

帖子页面的抽取设计对应帖子的详细内容，包括标题、正文、作者等，其页面结构如图4-2所示。

图4-2 帖子页面

该页面的抽取函数如下：

```
function describePost() {
// 获取帖子相关信息
var postId = parseInt(location.hash.match(/pid=(\d+)$/)[1]),
subjectCode = location.pathname.match(/\w+-\d+$/)[0],
title = $(".j-title").text(),
content = $(".j-content").html(),
score = parseInt($(".f-thide.j-num").text());
var userName = null,
userId = null;
// 获取用户相关信息
if ($(".j-post .userName").length > 0) {
userName = $(".j-post .userName").text();
userId = parseInt($(".j-post .userName").attr("href").match(/\d+$/)[0]);
}
var post = {post_id: postId,
subject_code: subjectCode,
title: title,
content: content,
user_id: userId,
user_name: userName,
score: score};
// 传递帖子内容
chrome.runtime.sendMessage({message: "found-post",
```

```
post: post});
}
```

4.1.3 评论区的递归爬取

评论区使用Ajax动态加载分页，因此使用一个递归函数爬取所有分页中的内容，该页面的结构如图4-3所示。

图4-3 评论区页面

该页面的抽取函数如下：

```
function describeComments() {
var comments = [],
postId = parseInt(location.hash.match(/pid=(\d+)$/)[1]);
$(".j-reply-all .m-detailInfoItem").each(function(idx, el) {
var content = $(el).find(".j-content").text(),
score = parseInt($(el).find(".j-num").text()),
repliesCount = parseInt($(el).find(".j-cmtBtn").text().match(/\d+/)[0]),
userName, userId;
if ($(el).find(".userName").length > 0) {
userName = $(el).find(".userName").attr("title");
userId = parseInt($(el).find(".userName").attr("href").match(/\d+$/)[0]);
}
var comment = {
post_id: postId,
content: content,
user_name: userName,
user_id: userId,
score: score,
replies_count: repliesCount
};
comments.push(comment);
});
for (var i=0; i<comments.length; i++) {
var comment = comments[i];
// 传递评论信息
chrome.runtime.sendMessage({message: "found-comment",
comment: comment});
}
if ($(".znxt").length > 0 &&
!$(".znxt").last().hasClass("js-disabled")) {
$(".znxt")[$(".znxt").length - 1].click();
// 等待页面渲染后，获取下一页
setTimeout(function() {
describeComments();
}, 2000);
} else {
```

chrome.runtime.sendMessage({message: "close-me"});

}

}

4.2 任务队列与消息通信

4.2.1 任务队列的设计与实现

爬取程序的后端部分实现了一个简单的任务队列，并提供了一个可配置的控制并发的机制，其结构如图4-4所示。

线画整齐点

图4-4 任务队列

系统中定义了两种任务：

1.打开一个论坛页（图4.4中的蓝色框）

2.打开一个帖子页（图4.4中的绿色框），任意两个任务之间是次序无关的。系统通过一个定时器和选择分支结构以此处理任务队列中的任务，并控制当前并行的任务数量。

任务队列的处理流程如图4-5所示：

图4-5 任务队列处理流程图

任务队列的处理代码如下：

```
mainInterval = setInterval(function() {
if (tabs.length < 5 &&
tasks.length > 0) {
var task = tasks.shift();
switch(task.type) {
case "post":
// 任务类型为爬取一个帖子
var url = "http://mooc.study.163.com/learn/" + task.subject_code + "#/learn/forumdetail?pid=" + task.post_id;
chrome.tabs.create({url: url,
active: false,
selected: false}, function(tab) {
tabs.push(tab);
});
break;
case "page":
// 任务类型为爬取一个论坛页面
var url = "http://mooc.study.163.com/learn/" + task.subject_code + "#/learn/forumindex?t=0&p=" + task.page;
chrome.tabs.create({url: url,
active: false,
selected: false}, function(tab) {
tabs.push(tab);
});
break;
default:
console.error("Unknown task:", task);
}
```

```
}
}, 200);
```

4.3 实时数据监控

本系统实现了对爬虫活动的实时监控，该模块可以在支持HTML5技术的浏览器上使用SVG图形绘制爬虫流量，其界面效果如图4-6所示。

图4-6 爬虫活动监控面板

本系统通过实现一个实现一个Logger对象来进行日志记录，Log记录支持4个类别，分别为error、success、warning和info。

Logger类的具体实现代码如下：

```
var Logger = function() {};
Logger.levels = ["error", "success", "warning", "info"];
Logger.append = function(level, message) {
if (!message) {
message = level;
level = "info";
} else if (Logger.levels.indexOf(level) === -1) {
level = "info";
}
var container = document.getElementById("logger-content");
// 构造日志条目
var logItem = document.createElement("div");
logItem.className = "log-item log-" + level;
var now = (new DateTime()).formats.compound.myLog;
var textContent = level === "info" ? "["+now+"] " + message : message;
logItem.textContent = textContent;
// 写入日志条目
container.appendChild(logItem);
container.scrollIntoView(false);
};
window.Logger = Logger;
```

4.3.1 模块握手

爬虫和监控面板在进行消息传递前需要先进行握手操作，爬虫通过向服务器发送包含{type: "id_crawler"}的消息来声明身份，监控面板通过发送包含{type: "id_visual"}的消息声明身份。

监控面板的连接建立过程如下：

```
var wsAddress = "ws://localhost:8000/ws";
Logger.append("warning", "Connecting to WebSocket server: " + wsAddress);
var ws = new WebSocket(wsAddress);
// When the connection is open, send some data to the server
ws.onopen = function () {
Logger.append("success", "WebSocket Opened!");
ws.send(JSON.stringify({type: "id_visual"}));
};
```

```
// Log errors
ws.onerror = function (error) {
Logger.append("error", "WebSocket Error!");
};
// Log messages from the server
ws.onmessage = function (e) {
Logger.append(e.data);
onMessage();
};
ws.onclose = function (e) {
Logger.append("warning", "WebSocket Closed!");
};
```

爬虫的连接过程如下：

```
var wsAddress = "ws://localhost:8000/ws",
ws;
// 建立WebSocket连接
function openConnection() {
ws = new WebSocket(wsAddress);
ws.onopen = function () {
console.log("WebSocket opened!");
sendMessage({type: "id_crawler"});
};
}
// 发送消息
function sendMessage(message) {
// ensure connection is open
if (ws && ws.readyState === 0) {
console.log("WebSocket is connecting...");
return;
} else if(!ws || ws.readyState !== 1) {
console.log("WebSocket not connected...");
openConnection();
return;
}
ws.send(JSON.stringify(message));
// send our message
}
```

握手完成之后，服务器在爬虫与监控面板之间建立一个单向的数据传递。相关代码如下：

```
class SocketHandler(websocket.WebSocketHandler):
def check_origin(self, origin):
return True
# 连接开启
```

```
def open(self):
if self not in cl:
cl.append(self)
# 收到消息
def on_message(self, message):
global ws_crawler, ws_visual
message_json = None
try:
message_json = json.loads(message)
except:
pass
# 消息解析失败
if message_json == None:
print "Failed parse message:", message
return
message = message_json
msg_type = message.get("type")
if msg_type != None:
if msg_type == "id_visual":
ws_visual = self
elif msg_type == "id_crawler":
ws_crawler = self
else:
print "Ignore identity ack:", message
return
if ws_visual != self and ws_crawler != self:
print "Drop message:", message
return
print "Received message:", message
if ws_crawler == self:
if ws_visual != None:
try:
ws_visual.write_message(json.dumps(message))
except:
pass
def on_close(self):
if self in cl:
cl.remove(self)
```

4.3.2 消息传递与日志

爬虫与监控面板之间通过消息传递的方式进行数据通信。图4-7显示了控制台记录的爬虫消息。

图4-7爬虫控制台

4.4 数据统计

本系统使用Highcharts[10]作为图表显示库。Highcharts是一个流行的用于前端显示图标的JavaScript库，对各种图表具有广泛的支持，并且简单易用。

4.4.1 用户活跃度统计

本系统实现了对用户活跃度的统计功能，用户活跃度指标分为发帖活跃度和评论活跃度，如图4-8所示。

图4-8用户活跃度图表

图表绘制函数：

```
$('#chart-1').highcharts({
chart: {
type: 'column',
width: 740,
height: 300
},
title: {
text: 'Post User Activities'
},
xAxis:
type: 'category'
},
yAxis: {
title: {
text: 'Count'
}
},
legend: {
enabled: false
},
plotOptions: {
series: {
cursor: 'pointer',
point: {
events: {
click: function() {
openUserPostsModal(this.name);
}
}
},
borderWidth: 0,
dataLabels: {
enabled: true,
format: '{point.y}'
}
}
```

```
},
series: [{
colorByPoint: true,
data: {% raw user_post_counts %}
}]
});
```

4.4.2 词语趋势统计

词语趋势统计模块可以针对用户输入的特定词语，查询其在不同时间段内的出现频率，如图4-9所示。

图4-9词语趋势图

图表绘制函数：

```
$('#wf-chart').highcharts({
chart: {
type: 'column',
width: 740,
height: 300
},
title: {
text: 'Word Trend {% if word %}for "{{ word }}"{% end %}'
},
xAxis: {
title: {
text: 'Time'
},
type: 'category'
},
yAxis: {
title: {
text: 'Count'
}
},
plotOptions: {
series: {
borderWidth: 0,
dataLabels: {
enabled: true,
format: '{point.y}'
}
}
},
series: [{
colorByPoint: true,
data: chartData
```

}]

});

4.4.3 主图模型统计

主题模型统计模块运用主题模型对数据库中的所有帖子进行主题划分，如图4-10所示。

图4-10主题模型图

4.5 本章小结

本章主要介绍了系统的详细实现过程，包括爬取模块和数据可视化模块的设计与实现。

---

6. 第5章关键技术 总字数：**2762**

相似文献列表 **文字复制比：** 23.8% (658) **剽窃观点（0）**

1	数据库课程管理系统的设计与实现 邱盈盈 - 《大学生论文联合比对库》- 2015-05-17	14.3%（396） 是否引证：否
2	数据库课程管理系统的设计与实现 邱盈盈 - 《大学生论文联合比对库》- 2015-05-19	14.3%（396） 是否引证：否
3	Rproxy，一个基于Twisted的反向代理，以及锻炼身体工具推荐 | observer专栏杂记 -《互联网文档资源（http://www.360doc.co）》- 2015	8.1%（223） 是否引证：否
4	刘杰-专注医院网络营销 | 刘杰，现供职于北京德健医院管理有限公司，致力于医院信息化实施方案 -《网络（http://www.liujie.or）》- 2012	6.7%（185） 是否引证：否
5	分布式存储 CentOS6.5虚拟机环境搭建FastDFS-5.0.5集群_辛子奇_13252 -《网络（http://blog.sina.com）》- 2015	6.7%（185） 是否引证：否
6	开题，j2ee网站并发性能优化_熊勐 -《网络（http://blog.sina.com）》- 2013	6.3%（174） 是否引证：否
7	nginx+tomcat+session共享_accp -《网络（http://blog.sina.com）》- 2013	5.5%（153） 是否引证：否
8	Nginx+Tomcat配置_心碎逍遥 -《网络（http://blog.sina.com）》- 2012	5.4%（148） 是否引证：否
9	5127101361_彭秋源_微信机 器人的设计和实现 彭秋源 - 《大学生论文联合比对库》- 2014-05-14	4.5%（124） 是否引证：否
10	系统通过nginx实现tomcat集群_Deng海山 -《网络（http://blog.sina.com）》- 2012	4.5%（124） 是否引证：否
11	软件1010-1011610303-于炳哲 于炳哲 - 《大学生论文联合比对库》- 2014-05-20	3.5%（98） 是否引证：否
12	Web服务器Nginx常见的配置选项整理_操作系统教程 -《网络（http://www.3lian.com）》- 2012	3.4%（93） 是否引证：否
13	高性能Web服务器Nginx的配置与部署研究（11）应用模块之Memcached模块的两大应用场景 - 钟超 TechBlog | 柳惊鸿·Poechant - 博客频道 - CSDN.NET -《网络（http://blog.csdn.net）》- 2012	2.2%（62） 是否引证：否
14	nginx同IP、同端口、不同域名时的转发 - SoftWare - yaosansi's Blog -《网络（http://www.yaosansi.）》- 2011	2.2%（62） 是否引证：否
15	Nginx透传获取客户端IP地址--相关文章 -《互联网文档资源（http://www.360doc.co）》- 2015	2.2%（62） 是否引证：否

16	Nginx负载均衡基础知识--相关文章 -《互联网文档资源（http://www.360doc.co）》- 2011	2.2%（62） 是否引证：否
17	nginx负载均衡策略--相关文章 -《互联网文档资源（http://www.360doc.co）》- 2013	2.2%（62） 是否引证：否
18	lvs nginx-proxy nginx 取用户真实IP--相关文章 -《互联网文档资源（http://www.360doc.co）》- 2011	2.2%（62） 是否引证：否
19	自己反代 Google 字体库，实现国内/外均高速访问（其他公共库的原理相同）--相关文章 -《互联网文档资源（http://www.360doc.co）》- 2014	2.2%（62） 是否引证：否

原文内容

第5章关键技术

5.1 服务器模块

本系统中用于数据可视化的服务器模块使用Python编程语言，结合Tornado框架开发。

Tornado是一个Python的Web框架和异步网络库，通过使用非阻塞的网络IO，Tornado可扩展到数万网络连接，非常适用于长轮询、WebSocket，或者其他需要长时间保持网络连接的应用。

5.1.1 HTTP组件

通过tornado.web.RequestHandler类创建一个HTTP请求处理模块。

Nginx服务器配置：

```
upstream visual {
server 127.0.0.1:8000;
}
server {
listen 80;
server_name visual.local;
location / {
proxy_pass http://visual;
proxy_redirect off;
proxy_set_header Host $http_host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-Host $server_name;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}
client_max_body_size 4G;
keepalive_timeout 10;
}
```

HTTP服务器路由设定：

```
app = web.Application([
(r'/', ConsoleHandler),
(r'/statistics', StatisticsHandler),
(r'/words', WordsHandler),
(r'/user_posts', UserPostsHandler),
(r'/user_comments', UserCommentsHandler),
(r'/static/(.*)', StaticFileHandler, {'path': './static'})
```

], debug=True)
if __name__ == '__main__':
app.listen(8000)
ioloop.IOLoop.instance().start()
5.1.2 WebSocket组件
通过tornado.websocket.WebSocketHandler类创建一个WebSocket处理模块。
WebSocketHandler类内部的重要方法包括：
表5-1事件处理方法
方法名说明
WebSocketHandler.open 当Websocket开启时调用此方法
WebSocketHandler.on_message 处理入站消息
WebSocketHandler.on_close 当WebSocket关闭时调用此方法
表5-2消息输出方法
方法名说明
WebSocketHandler.send_message 发送一条消息
WebSocketHandler.close 关闭连接
WebSocket路由设定：
app = web.Application([
(r'/ws', SocketHandler),
], debug=True)
if __name__ == '__main__':
app.listen(8000)
ioloop.IOLoop.instance().start()
5.2 服务器搭建
5.2.1 安装服务器环境
Requirements.txt文件内容：
tornado
psycopg2
使用pip install –r requirements.txt安装必要的python库。
5.2.2 运行服务器
使用python server.py命令运行服务器程序，运行结果如图5-1所示。
图5-1程序运行结果
5.3 Chrome Extension 的创建
5.3.1 manifest.json文件
创建manifest.json文件
{
"manifest_version": 2,
"name": "Crawler",
"description": "No description",
"version": "1.0",
"browser_action": {
"default_popup": "popup.html"

},
"content_scripts": [
{
"matches": [
"http://*.study.163.com/*"
],
"js": ["jquery-2.2.0.js", "content.js"]
}
],
"background": {
"scripts": ["jquery-2.2.0.js", "background.js"]
},
"permissions": [
"activeTab",
"http://localhost:5000/"
]
}

5.3.2 使用扩展

在Google Chrome浏览器下，依次单击“Customize”、“Settings”、“Extensions”，进入浏览器的扩展页面。点击“Load unpacked extension…”按钮可以加载一个本地的插件。页面如图5-2所示。

图5-2加载插件程序

5.4 本章小结

本章主要介绍了系统的使用方法和关键技术。

参考文献

[1] A McAuley, B Stewart, G Siemens, D Cormier. The MOOC model for digital practice [J]. davecormier.com,2010.

[2] Regina Ob，Leo Hsu．PostgreSQL: Up and Running: A Practical Introduction to the

[3] 弗兰纳根(David Flanagan)．JavaScript权威指南[M]．北京：机械工业出版社，2012，04-01．

[4] 丘恩 (Wesley J.Chun)，宋吉广．Python核心编程[M]．北京：人民邮电出版社，208，07-01．

[5] 比伯奥特，卡茨，三生石上．jQuery实战[M]．北京：人民邮电出版社，2012，03-01．

[6] 罗刚，王振东．自己动手写网络爬虫[M]．北京：清华大学出版社，2010，10-01．

[7] 麦金尼 (Wes McKinney)，唐学韬. 利用Python进行数据分析[M]．北京：机械工业出版社，2014，01-01．

[8] Michael Dory，Adam Parrish，Brendan Berg．Introduction to Tornado [M]．USA：O'Reilly Media, Inc，2012，04-13．

[9] 陶辉．深入理解Nginx:模块开发与架构解析[M]．北京：机械工业出版社，2016，02-01．

[10] Joe Kuan．Learning Highcharts [M]．USA：Packt Publishing，2012，06-01．

[11] 库克 (Darren Cook)．HTML5数据推送应用开发[M]．北京：人民邮电出版社，2014，07-01．

[12] Masoud Kalali，Bhakti Mehta. Developing RESTful Services with JAX-RS 2.0, WebSockets, and JSON [M]．USA：Packt Publishing，2008，07-01．

Advanced Open Source Database [M]．北京：人民邮电出版社，2009，01-16．

[13] 科克伦，惠特利．Bootstrap实战[M]．北京：人民邮电出版社，2015，05-01．

[14] 道格拉斯•克罗克福德 (Douglas Crockford)．JavaScript语言精粹[M]．北京：电子工业出版社，2014，0

9-01．

[15] 扎卡斯 (Nicholas C. Zakas)．编写可维护的JavaScript [M]．北京：人民邮电出版社，2013，04-01．

[16] SM Mirtaheri, D Zou, GV Bochmann. Dist-ria crawler: A distributed crawler for rich internet applications [J]. IEEE，2013．

[17] A Heydon, M Najork．Mercator: A scalable, extensible web crawler [J]．Springer，1999.

[18] V Shkapenyuk, T Suel - Data Engineering．Design and implementation of a high-performance distributed web crawler [J]．Proceedings，2002．

[19] I Fette, A Melnikov．The websocket protocol [J]．tools.ietf.org，2011．

[20] Y Furukawa．Web-based control application using WebSocket [J]．2011 - accelconf.web.cern.ch s2011.

页码怎么不对??

致谢

本论文在董永权老师的悉心指导下完成。从系统的设计到论文的写作以及成稿，董老师给与我极大地帮助。在系统设计时期，老师对我的系统需求分析提出了很多宝贵的建议，为我开阔了思路。在后期，老师多次仔细审查并批注了我的论文，导师丰富的科研实践经验以及严谨的治学态度给与我极大的鼓励。导师对计算机科学研究事业孜孜不倦的追求精神，给了我无穷的启发与思考。在此，我要向董老师致以最诚挚的谢意。

同时，也要感谢我的舍友，感谢他们在后期对论文提出的意见和建议。

感谢江苏师范大学智慧与教育学院的全体老师，衷心的感谢他们能为我们提供良好的学习环境以及周到、细致的学习计划。

通过本次论文的撰写，我系统的学习了系统开发方面的理论知识并获得了宝贵的开发经验，这对于我今后的工作来说，无疑是个不可多得的锻炼机会。由于本人首次独立开发完整的爬虫系统，细节方面可能存在一些欠缺，论文中涉及到的一些技术介绍可能还存在不足，恳请老师们指正。