基于网络爬虫的技术博客分类系统设计与实现

课程名称：《Java 编程技能训练》

成员名单：

小组成员	姓名	学号
组长	王奕超	1501211000
组员	张晗	1501211029
组员	胡玉	1501210913

2015 年 1 月 7 日

一、 系统简介

在技术人员做开发的过程中，难免会碰到一些比较困难的问题，这些问题仅凭现在自己已有的知识并不能解决，此时就需要去网络上查询相关的资料进行学习，而技术博客则是目前为止最为技术人员青睐的学习之源。然而网络上的博客众多，相关的论坛也是分类复杂，资源不够集中，除此之外，页面上的各种广告也让人感觉眼花缭乱，以上这些原因都将导致查询资料的效率大大降低，最终导致技术人员开发周期的大幅延长以及开发质量的大幅下降。本系统基于以上需求，进行了详细的分析与设计，并最终实现了基于 HttpClient Fluent 的网络爬虫博客分类系统，通过在主流技术博客论坛爬取相关技术文章数据并进行精准有效的分类并以动态的形式向用户展示分类数据。有利于极大提高技术人员的检索效率，并以友好简洁的系统界面，提供给使用者良好的用户体验。

二、 涉及技术

按照数据处理的层面，将系统划分成数据获取，数据解析、组织及存储，数据展示三个部分分别进行设计实现。

1. 数据获取

网页数据抓取是系统数据的获取方式，系统基于 HttpClient Fluent 工具包实现网页数据的动态抓取，Fluent 是 Apache 开源项目 HttpClient 4.2 版本后推出的方便开发者利用 Java 实现网页数据的抓取，相比之前的版本 HttpClient 4.2 提供了一组基于流接口(fluent interface)概念的更易使用的 API，即 Fluent API。为了方便使用，Fluent API 只暴露了一些最基本的 HttpClient 功能。这样，Fluent API 就将开发者从连接管理、资源释放等繁杂的操作中解放出来，从而更易进行一些 HttpClient 的简单操作。

2. 数据解析、组织及存储

数据解析、组织及存储是系统数据的处理方式，系统将第一阶段数据抓取获取的数据进行解析，根据对获取数据的统计分析，将博客数据内容进行分类并存储。系统采用 HTMLParser 进行抓取页面的解析，抓取博客文章标题、网站分类、文章链接等数据，并将数据持久化到 MySQL 数据库。

HTMLParser 是一个纯 Java 写的 HTML（标准通用标记语言下的一个应用）解析的库，它不依赖于其它的 java 库文件，主要用于改造或提取 HTML。它将 HTML 页面中的标签按树形结构解析成一个一个结点，一种类型的结点对应一个类，通过调用其方法可以轻松地访问标签中的内容。

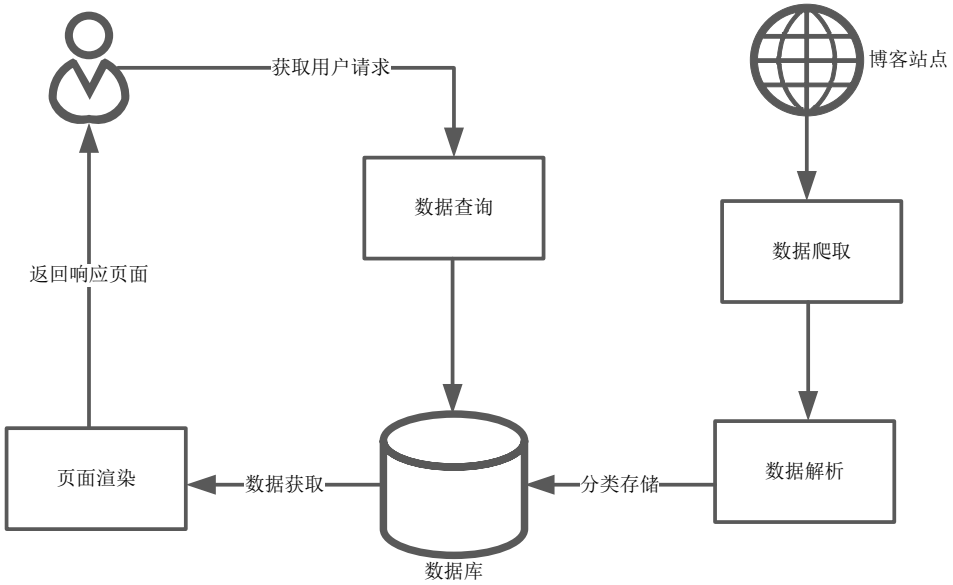
3. 数据展示

数据展示是系统数据可视化方式，系统采用 JSP + servlet 技术按照预设好的分类将数据以简洁的形式展现给用户，方便用户的查询检索。

三、 框架及介绍

1. 系统框架

本系统采用 JSP + JavaBean 简单架构和 MVC 架构相结合的架构模式，即在用户选择新的数据抓取站点的时候采用 MVC 架构，用户发送抓取请求并发送抓取站点链接，服务器获取用户请求，访问站点链接并进行数据抓取，将抓取到的数据分类进行持久化，用户点击相应分类，系统会根据用户点击选择，列出所选分类的文章标题及链接，用户点击即可访问该博客；当用户没有输入抓取站点 URL（即默认状态），系统基于数据库中已有数据信息，根据用户类别选择列出所选分类的文章标题和链接。系统整体架构图如图所示：



系统从网络中博客站点爬取博客信息并将所需数据元素提取，根据原始数据统计词频建立分类将数据分类存储，用户进入网站浏览分类选择分类或

文章点击发出请求，获取用户请求查询数据获取数据并渲染页面返回给用户。

2. 爬虫框架

Fluent 是 Apache 开源项目 HttpClient 4.2 版本后推出的方便开发者利用 Java 实现网页数据的抓取，相比之前的版本 HttpClient 4.2 提供了一组基于流接口(fluent interface)概念的更易使用的 API，即 Fluent API。为了方便使用，Fluent API 只暴露了一些最基本的 HttpClient 功能。这样，Fluent API 就将开发者从连接管理、资源释放等繁杂的操作中解放出来，从而更易进行一些 HttpClient 的简单操作。

3. 分析框架

本系统区别于其他现有系统的优势之一就是博客的分类方式，在大家搜索相关技术文章的时候，应该是在开发过程中遇到了问题，而正在开发也就必然正在使用某种编程语言，所以按语言查询是一种潜在的用户请求，特别是针对初学者。所以本系统首先将现阶段几大流行的编程语言作为第一级分类，并在此分类的基础上，对所爬去的技术博客内容进行分类并统计数量：

```
java : 3425
算法 : 564
Java web : 510
leetcode : 424
并发 : 391
设计模式 : 289
多线程 : 231
数据库 : 131
hibernate : 120
html : 118
浏览器 : 116
框架 : 115
对象 : 112
null : 111
function : 111
并发 : 108
数组 : 102
java web : 100
工具 : 96
spring : 91
jdk : 89
```

此图为将分类条件设定为 Java 相关的时候的词频统计结果，则根据结果可以发现博客文章的主题的热门程度，这个热门程度也同样反映着开发人员对于技术资料的主流需求，因此我们将提取靠前的主题作为第二级分类。最

后将所有的博客根据所属类别进行归类存储。

四、 实验结果

系统界面如下图所示：

系统将排名前六名的语言作为第一级分类（即第一级导航栏），用户在导航栏下方的搜索框中输入想要爬取数据的站点 URL，只要符合系统定义的爬取标准，系统就可以将指定站点的网页数据爬去下来存储到本地，并将有效信息（博客信息）存储到数据库。

系统的第二级分类在鼠标悬浮在第一级导航栏上方时就会自动弹出，在用户点击二级导航栏（第二级分类）时，系统界面中央区域就会显示出相关的技术博客文章的标题以及链接，用户可以点击文章标题阅读博客的具体内容。

五、 总结

根据系统设计与实现的整个过程，我们按照实验和系统两个方面进行总结。首先对于实验，我们将任务按照层面进行划分，每人负责其中的一部分，任务比较明确，每个人都能够积极的参与进来，并且在定义好数据接口后三个部分的实现同时展开，提高了实验的效率和质量。

另一方面，系统的整体效果比较符合预期，且具有较好的可拓展性（系统分层实现）和较低的耦合性，但是系统的部分代码的可重用性仍需改进。对于后期的系统开发有许多值得改进的地方，比如说系统可以进行参数化配置，这样用户可以自定义分类或增加子分类，支持更多的爬取站点等。

总之，通过实验，小组内成员都有很大的收获，从系统设计到最终系统的实现大家都参与进来，体验到了整个系统开发的过程，巩固了所学习的软件工程技术和 Java 编程技术。