****

**毕业设计(论文)**

( 2015届 )

论文题目 互联网信息定向采集系统

The Internet Information

Oriented Acquisition System

学 生 姓 名： 王琦 学 号： 110806129

二级学院名称： 新媒体学院 专 业： 信息管理与信息系统

指 导 教 师： 俞定国 职 称： 副教授 .

**郑 重 声 明**

我谨在此郑重声明：本人所写的毕业论文《互联网信息定向采集系统》均系本人独立完成，没有抄袭行为，凡涉及其他作者的观点和材料，均作了注释，若有不实，后果由本人承担。

**承诺人（签名）：**

**年 月 日**

# **1 绪论**

## 1.1 选题背景与意义

在一定的时间与空间范围内，民众、媒体通过互联网对该事件产生所发表的政治态度、意见称之为网络舆情。近年来，随着互联网行业的高速发展，网络舆情渐渐的对社会产生了更多更大的影响。

传统社会中，舆情散布在大街小巷,难以统一获取，只能借助社会明察暗访，民意调查等传统方式，不仅耗费人力物力，就结果而言，效率不高且获取样本数量也不多，随着互联网技术高速发展，大众越来越依赖网络这个平台，也更多的以信息化的方式在网络上发表自己的观点，相对传统获取舆情方式而言，网络舆情可以通过自动抓取等技术，十分方便的获取，摆脱传统人力浪费，使用计算机即可实现，效率高且抓取而来的信息是第一手资料，覆盖的范围也很广。

网络具有的开放性和相对自由的宽松度，使得民众发言摆脱了社会权利体制的管制和限制，可以畅所欲言无所顾忌的表达个人的观点、立场、情绪，民意表达更为畅通。但互联网也带来了相当大的隐患，匿名发言被误解为可以胡乱发言，不需要被规范。当今网络已经成为反映社会舆情的最主要载体,因此对网络舆情监测分析工作显得愈发重要。

互联网信息的定向采集器，主要解决数据的收集和解析，是基于云计算的电视节目舆论监测分析系统的前端和基础。系统将对指定的互联网站点及信息内容进行自动分类采集，并对采集的网页信息进行智能解析，对web页面进行智能解析，识别页面结构，将内容信息转换为结构化数据，并存储到数据中心，将不规则的海量数据转换为格式化数据，为后续舆情检测服务。网络信息采集的准确性、采集的深度与广度对舆情系统的性能有决定性作用，因此对互联网信息的定向采集和自动解析技术的研究具有重要的社会意义和现实价值。

## 1.2 毕业设计工作介绍

本系统使用java语言，开源的mysql数据库，设计、开发了一整套互联网信息定向采集系统。用户根据自己的需要，配置相关的任务信息，如：任务的初始站点、任务需要保留或剔除的域名、任务采集的深度以及数据库地址等一系列信息。启动任务后，系统将根据任务设定的条件，将用户需要的信息保存到指定的数据库，供用户使用。系统大致上可以分为：监控管理平台、定向采集器、智能解析器。相关概念定义如下

监控管理平台：用户通过监控管理平台，自定义配置采集任务，控制任务的起始、暂停状态，并通过界面实时反馈任务进度。

定向采集：即限定采集站点范围，通过系统配置限定采集对象。如采集站点设为“新浪”“搜狐”两个门户网站，系统将只采集这两门户网站的网页。

智能解析：系统将自动对已采集的网页文件进行分块、去噪，识别网页结构，自动抽取网页正文及其它关键信息（标题、文章出处、发布时间、作者等），并转换成结构化数据。

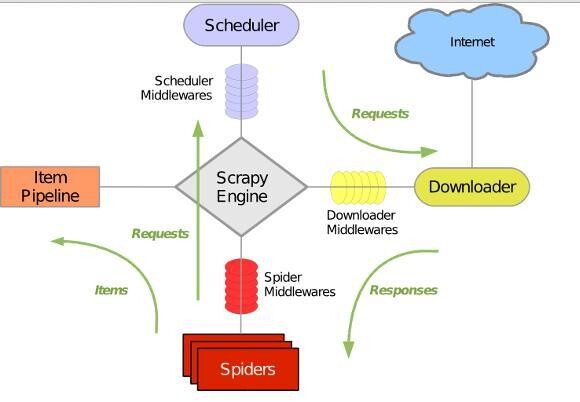
# **2相关知识及技术介绍**

## 2.1网络爬虫

## 2.1.1 什么是网络爬虫

网络爬虫简单的说，就是一个用于在互联网上抓取信息的程序，一个爬行互联网的程序，一个抓取[网页](http://baike.baidu.com/view/828.htm)内容的程序。网络爬虫可以做一个形象的比喻：一张张的网页就像节点，链接就好比一条条的线，爬虫沿着线爬到点，就像蜘蛛知网一样，网络爬虫由此得名。

网络蜘蛛通过分析[网页](http://baike.baidu.com/view/828.htm)内容，提取出页面里包含的URL地址来寻找新的网页，从一个种子页面开始，下载并且解析网页，找寻网页中存在的其它链接地址，再根据这些找到的链接地续，找到下一个网页，如此循环，直到抓取完所有网页为止。整个互联网好比是一大的个网站，爬虫就是通过这种方式抓取整个互联网上的[网页](http://baike.baidu.com/view/828.htm)，如图2-1所示。



**图2-1传统爬虫示意图**

## 2.1.2 网络爬虫的分类

**①通用爬虫**

一般爬虫框架流程为：从互联网的海量的页面抓取一些高质量页面例如导航等页面，抽取其中所有包含的url，并将这些URL放入待抓队列中，爬虫从队列中获取链接，然后进行一次dns解析转换ip地址，通过模拟浏览器打开ip地址返回页面的源文件。

成功获得源文件后，立刻对其建立索引，然后保留获取记录以免重复获取。

获得源文件后，会对源文件进行一次解析，解析出的URL放入待抓队列中，然后再从队列里获取URL打开，如此循环。

因为互联网上的信息量巨大，如今的爬虫，都会制定一些特定的抓取测量来抓取，不会全部都抓。

所以，我们可以大致上把互联网上的页面分为以下5各部分：

1)已下载页面集合

2)已过期页面集合

3)待下载页面集合

4)可知的页面集合

5)不可知的页面集合

②聚焦爬虫

聚焦爬虫顾名思义，针对某一类主题或者内容进行抓取。自动过滤一些不相干的链接，只保留需要的链接放入待抓队列中。对于保留下来的链接，系统会进一步跟进，完善本身的过滤策略。

聚焦爬虫除了需要有爬虫的基本机构外，还需要：

1) 对抓取目标的描述或定义

2) 对网页或数据的分析与过滤

3) 对URL的搜索策略

第一点是指定过滤策略的基础。而2，3点事爬虫行为的重要依据，所以这两个部分是紧密联系在一起的。

③智能爬虫

智能爬虫是垂直搜索引擎的核心技术，衡量一个[垂直搜索引擎](http://baike.baidu.com/view/7491.htm)的好坏主要分为以下几个标准。

1）数据的更新频率

爬虫间隔多久获取同一个网页上的信息。

2）覆盖的网站个数

覆盖越多的网站，搜索引擎就越强大。

3）单站的有效数据抓取率

对于同一个网页上获取有效信息的获取量，越高则爬虫的质量就越高。

垂直搜索引擎大约有两种形式。

1）爬虫定向采集，人工配置页面的模板，获得结构化数据存入。

优势：

更为准确的获取目标信息。比如文章的作者、时间、标题、正文等。

劣势：

难以覆盖所有的网站，因为模板需要人工编写，而且维护起来也相对麻烦，需要大量的人力物力的投入。

2）语义全网爬取，爬虫会根据语义识别，并自动进行信息格式化分析，并存储。

优势：

爬虫的覆盖面很广。不需要人工配置模板，节约了成本。

劣势：

准确率不高。

## 2.2 网页解析技术

## 2.2.1 网页解析技术的基本概念

信息抽取，就是把散乱的源文本进行整理，形成一种较为合理的信息格式表输出。信息从散乱到整齐，好处之一就是便于统一管理，只需要对相关信息进行整理即可。至于该信息是否有用，则需要进一步研究。

互联网上的信息，例如网页信息，并不是完全结构化的，甚至是完全动态的。如今，互联网已经不在仅仅是电脑上的，手机的发展带来的信息跨平台也增加了信息抽取的难度。

许多大型的网站例如新浪等，是由机器生成的网页，相对结构化，很多小型网站却没有一套相应的规范传统的信息抽取技术只能针对结构化的文本进行抽取，对于非结构化的抽取质量不高

## 2.2.2 网页解析技术的常用方法

## ①基于树结构的信息抽取技术

利用TagNode构造的网页树，可以把网页信息映射在网页结构树中信息。通过对不同网站的信息进行分析，发现这样的规律：抽取的手机型号和手机价格往往在同一个table，手机型号和手机价格间存在一对一或一对多的关系。可采用如下的启发式的规则进行网页信息抽取。启发式规则：待抽取的信息通常在同一个table中，并且它们之间存在一对一或一对多的关系。

单个网页源文件可以形象的比如成一棵树。相关的信息往往就在同一个节点上。因此，可以以此为基础建立相应的解析库，应用于信息抽取。

解析库里包含了网页的特征，在进行信息抽取时，网页结构树定位模式库中的修需要抽取的信息，进行匹配，如果成功，说明在同一个节点上。

②基于包装器的信息抽取方法

一个包装器往往只能处理一系列信息源。不同信息源中抽取信息,需要不同的包装器程序库。

与自然语言处理方式相比,包装器很少依赖全面的语法分析或者分词等一些列复杂的自然语言处理,会更注重于表格格式和文本结构分析。由于一个包装器只能处理一一系列网页,当出现一类新的网页或着旧的页面发生变化,原来的包装器就会无效,无法继续从数据源中获取数据甚至得到错误的数据,缺乏主动理解。

③基于Ontology的信息抽取

该方法主要依赖于一个完整的知识库。知识库里定义了每一个元素的抽取模式,以及内在的联系。抽取前,将包含数据的记录分隔开,依次对每一个记录进行信息抽取。抽取模式不依赖于特定文档分隔符或者词性的自然语言理解技术,而主要使用通用模式。这种方法不依赖于任何结构的表现,使用Ontology定位信息并使用这些信息造对象。这事先需要构造一个完整的Ontology库,这需要很长时间。

④基于去除html标签的网页正文提取算法

此方法先取出html标签，然后通过一定的算法计算出正文的位置，合并后输出正文。

第一步首先需要去除html标签和JS代码。采用正则表达式方法去除例如对于<script></scripte>这种标签，使用正则表达式<script>:?</script>去除，对于<div>…<div>…</div><./div>这 种类型，使用栈，按顺序取出html中的非标签文字。

去除了标签后的源文件，每次读取一行，读取这一行的字数，如果连续两行字数不为0，则保留，在下一次连续空行前判断字数是否达到了一定的值，达到了即为正文。然后与其他的符合条件的文字合并，直到文件结束。

## 2.3相关技术介绍

## ①Nutch

Nutch 是 Apach的一个开源项目，它原本的作用是 Lucene 项目的一个子项目，后来渐渐成长为一个独立的项目。使用Java 开发，基于 Lucene 框架，提供了 Web 网页爬虫和搜索引擎两部分功能。

Nutch主要分为两个部分：网页爬虫和搜索引擎。爬虫用于从网络上抓取网页并建立索引。搜索引擎主利用这些索引查找关键词产生结果。它提供了一种插件框架，能够对网页内容进行解析、采集、查询、过滤等功能方便的进行扩 展，正是由于此框架，使得 Nutch 的插件开发非常容易。

## Heritrix

Heritrix，是[www.archive.org](http://www.archive.org/) 上的开源产品，他出色之处在于可扩展性，扩展它的组件，实现自定义的逻辑，但是配置比较复杂。Heritrix在进行抓取的过程中可以获取完整、精确的内容复制。并包括文本和非文本信息，抓取并存储内容。缺点在于抓取速度上慢、需要大容量存储空间和宽速网络。重新爬行对相同的URL不进行替换。此外Heritrix 有 Web 控制管理界面，进行启动、监控。

## larbin

larbin是一种开源的网络爬虫，由Sébastien Ailleret独立开发，用c++语言实现。larbin能够跟踪页面url进行扩展抓取，为搜索引擎提供广泛的数据。 Larbin只是一个网络爬虫，也就是说larbin只能够抓取网页，不能完成解析工作。另外，存储到数据库以及建立索也不提供。

latbin最初的设计既是简单高效，由此我们可以看到，一个简单的larbin的爬虫可以每天获取500万的网页，实在是非常高效。利用larbin，我们可以轻易的获取单个网站的所有联结，甚至可以镜像一个网站；同时也可以利用它建立url 列表群，如针对网页进行 url retrive后，进行xml的联结。

# **3 系统设计**

## 3.1 总体功能结构设计

整个互联网信息定向采集与智能解析系统分三大部件：任务监控管理平台、网页信息定向采集器和网页内容智能解析器，其功能结构如图3-1所示。



**图3-1 系统功能结构图**

（1）任务监控管理平台

任务监控管理平台主要实现对整个采集和解析任务进行监控和管理。

（2）网页信息定向采集器

网页信息定向采集器主要实现网页的采集功能。

（3）网页内容智能解析器

网页内容智能解析器主要对采集的网页进行智能分块、去噪，抽取网页正文和其它关键信息等。

## 3.2 任务监控管理平台设计

## 3.2.1任务监管平台与网页采集（解析）器间的通信模式设计

任务监管平台与任务采集（解析）器间通过共享数据库数据进行信息通信，其通信模式如图3-2所示。



**图3-2 通信模式**

（1）任务监控平台将任务配置信息、控制指令等数据写入任务监管数据库，并从任务监管数据库中读取任务状态信息和网页采集的统计数据，并显示在任务状态页面中；

（2）采集解析监管服务程序是任务监管平台和网页信息采集和解析程序间的桥梁。采集解析监管服务程序定期（每隔3秒）扫描任务监管数据库中其对应任务的控制信息，并分析其控制指令，根据控制指令操作（启动、停止）网页信息采集和解析程序，并将相应的状态信息写入任务监控数据库。

## 3.2.2 任务监管平台的功能结构设计

任务监管平台的功能结构如图3-3所示。主要包含任务配置、任务状态浏览和任务控制三大功能模块。



**图3-3 任务监管平台的功能结构**

（1）任务配置

任务配置功能主要实现对采集任务的属性信息和该任务进行采集的网站站点信息进行配置、管理和维护。

任务属性信息主要包括：

任务编号、任务名称、任务描述、采集进程数、解析进程数、数据存放服务器、自动更新服务器、是否保留网页源文件、存放网页源文件服务器、采集间隔时间（单位为分钟）。

采集网站的站点信息包括：

任务编号、站点编号、站点名称、站点类别、站点归属地、站点优先级（高、中、低）、入口网址（可采用表达式匹配方式）、采集深度、采集宽度、用户名、密码

（2）任务控制

任务控制主要实现对指定任务进行启动、停止操作、清空本任务已经采集的数据。

（3）任务状态浏览

任务状态浏览功能主要实现两大功能：

A．任务状态查看：查看当前任务的状态：执行中、停止（完成、错误中断、用户中断）；

B．采集数据统计：统计显示本次任务采集的数据记录数和累计采集的数据记录数，最近采集活动启动时间、最近采集活动结束时间和最近采集活动持续时间。

## 3.2.3 采集解析监管服务程序的功能结构设计

采集解析监管服务程序主要功能包括采集程序控制和任务监听两大模块，其功能结构如图3-4所示。



**图3-4 采集解析监管服务程序的功能结构**

（1）任务监听

任务监听模块定期（每隔3秒）扫描任务监管数据库，分析任务指令数据的变化，根据指令数据调用爬虫控制模块控制网页信息采集与解析程序。指令数据主要分两类：

用户操作指令：启动、停止、清空数据；

自动操作指令：当最近采集活动结束时间已经超过采集间隔时间，自动启动采集任务。

（2）采集程序控制

采集程序控制模块主要控制网页信息采集与智能解析程序，主要控制动作包括：

（1）启动：启动网页采集与解析程序进行网页采集，并在任务监管数据库中记录本次采集活动开始时间，修改任务状态为“进行中”；

（2）停止：停止网页采集与解析程序进行网页采集活动，并在任务监管数据库中记录本次采集活动结束时间；修改任务状态为“已停止”，并记录停止原因；将本次采集的数据量和累计采集的数据量记录到数据库。停止前需备份各个内存数据库。

## 3.3网页信息定向采集器

## 3.3.1网页信息定向采集器功能描述

网页信息定向采集器是根据任务监控管理平台的采集任务安排，抓取指定URL的网页文件，对网页内的新URL进行提取，作为进一步采集的对象，并将采集到的网页文件传递给网页内容智能解析器进行网页正文和关键信息解析抽取。

## 3.3.2网页信息定向采集器流程设计

系统工作流程如图3-5所示，其中对获取的web网页进行URL解析（新URL抽取）工作的流程如图3-6所示。



**图3-5 系统工作流程**



**图3-6 解析新URL工作流程**

## 3.3.3. 网页信息定向采集器功能结构设计

网页信息定向采集器的系统功能结构如图3-7所示。



**图3-7 网页信息定向采集器的功能结构**

系统主要包括以下几大模块：

（1）任务初始化

任务初始化模块的主要功能是对几大URL列表数据的初始化，系统将根据任务启动的类别，初始化网页采集任务：

**系统完全启动：**

待抓URL列表从站点配置数据库中导入；

已抓URL列表和错误URL列表从内存数据库文件中恢复；

定向抓取下限定URL表从站点配置数据库中导入。

**从暂停和错误中断后启动：**

待抓URL列表从内存数据库文件中恢复；

已抓URL列表和错误URL列表从内存数据库文件中恢复；

定向抓取下限定URL表从站点配置数据库导入。

（2）采集任务控制

系统器采用多线程抓取任务分配，进行多线程网页抓取。采集任务控制模块主要实现对各采集进程的任务安排和采集过程监管，及其各进程间的数据共享控制等。系统将根据待抓URL列表数据和多线程限制数进行采集任务分配。当待抓URL列表为空时，并且分配给各进程的采集任务都已完成时，停止网页信息定向采集器工作，并向采集任务监管平台反馈相应信息（任务完成时间和抓取的数据记录数）。

（3）网页编码识别

网页识别模块将自动识别抓取网页的编码方式。系统通过httpclient 模拟浏览器方式访问网页，获取服务器返回的信息，从中获取网页编码方式。如编码为空，则获取网页源文件中head部分中页面编码，作为网页编码方式。

（4）网页下载

网页下载模块的功能是下载指定URL的网页文件。在实现上主要是利用 Java 中的 URLConnection 类来打开 URL 对应页面的网络连接，然后通过 I/O 流读取其中的数据。

（5）URL提取

URL提取模块的重要功能是分析已经下载的网页文件源码，解析出网页中的URL，并提取出来，主要包含URL抽取和URL解析两大功能。URL抽取就是分析网页的HTML源码，抽取其中的URL连接字符串；URL解析就是分析抽取出来的URL连接字符串，将相对地址的URL转换成绝对地址。

（6）URL过滤

URL过滤就是对解析出来的URL连接进行合法性分析，过滤非法和错误的URL，主要操作功能有非法性过滤、去重过滤和非定向过滤。

URL定向过滤就是将待抓URL限定在指定的网站范围内，即信息抓取只在限定站点内进行。为了把信息采集限制在一批站点范围内，需要截断指向站外的链接,保证爬虫总在站内执行,即准确地根据URL判断该链接是否指向站外。

URL去重过滤就是过滤掉已在待抓URL、已抓URL和错误URL列表中的ULR连接，采用布隆过滤器（Bloom Filter）技术进行URL去重过滤操作。

（7）URL存储

为了到达去重、定向抓取（限定抓取范围）、排除错误URL列表和多线程抓取目的，需要保存多个URL数据列表，分别是待抓取URL列表、已抓取URL列表、错误URL列表和定向匹配URL列表。URL存储模块主要实现这个4个URL列表的管理功能，待抓URL列表和已抓URL列表管理采用内存数据库模式进行管理，错误URL列表和定向匹配URL列表采用普通列表管理模式进行管理。

## 3.4网页内容智能解析器

## 3.4.1网页内容智能解析器功能描述

网页内容智能解析器是对已经获取的网页进行网页内容自动识别和抽取，主要工作包括对网页进行自动分块，智能去噪，抽取网页正文和其它关键信息（标题、文章出处、发布时间、作者等），并将其转换成结构化数据存入到指定数据库，供后续模块调用。

## 3.4.2网页内容智能解析器流程设计

网页内容智能解析器的工作流程如图3-8所示。



**图3-8 网页内容智能解析器工作流程**

其中网页正文抽取采用基于行块分布函数的通用网页正文抽取方法，其工作流程如图3-9所示。



**图3-9 网页正文抽取流程**

## 3.4.3网页内容智能解析器功能结构设计

网页内容智能解析器的系统功能结构如图3-10所示。



**图3-10 网页内容智能解析器的功能结构**

系统主要功能模块包括：

（1）网页预处理

网页预处理模块主要对网页文件进行去噪处理，过滤掉一些广告等垃圾信息。网页预处理的操作主要包括：

编码转换：自动对网页编码进行识别，并按照统一要求自动进行编码转换；

页面清洗：即页面去噪，利用正则表达式过滤掉网页中一些脚本和特殊字符等“垃圾信息”。

（2）网页正文抽取

网页正文抽取模块主要利用基于行块分布函数的通用网页正文抽取技术抽取网页的正文信息。主要操作包括：计算行块字数的分布函数、计算骤升点和骤降点、提取网页正文。该抽取方法主要针对新闻类网页和博客网页，论坛和贴吧类网页采用其它方法。

（3）网页标题抽取

网页标题抽取模块通过扫描HTML源码文件，提取出网页的标题信息。

（4）网页关键信息抽取

对于一些指定网站的网页信息，需提取一些特殊的关键信息，如新浪新闻网页中的“新闻发布时间”、“新闻来源”、“参与人数”等。对于特殊信息的提取，需要调用网页关键信息提取模块进行详细提取，无需提取特殊信息要求的网页可以直接跳过该步骤。关键信息抽取模块采用解析网页DOM树技术进行关键信息抽取。

## 3.5数据库设计

任务监管数据库TaskManagement，包括两张表：

（1）表（TaskInfo）：存放采集任务信息，表结构如表3-11所示

**表3-11 TaskInfo**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **类型（长度）** | **是否允许为空** | **说明** |
| TaskID | 自动增长 | 否 | 主键，任务编号 |
| TaskName | varchar(50) | 是 | 任务名称 |
| TaskDescription | varchar | 是 | 任务描述 |
| GatherThead | int | 是 | 最多采集进程数 |
| InfoDB | varchar(50) | 是 | 索引服务器ID |
| SourceFileSave | bit | 是 | 是否保存网页源文件标志 |
| SourceFileDB | varchar(50) | 是 | 存放网页源文件的服务器IP |
| InfoUpdateServer | varchar(50) | 是 | 去重服务器IP |
| TaskBeginTime | longInt | 是 | 采集间隔时间（单位为分钟） |
| TaskRemark | varchar | 是 | 备注 |
| StartTime | DateTime | 是 | 任务最近采集活动启动时间 |
| StopTime | DateTime | 是 | 任务最近采集活动停止时间 |
| OperateType | varchar(50) | 是 | 启动/停止类型 |
| TotalDataSize | longint | 是 | 累计采集数据量 |
| LastGathDateSize | longint | 是 | 最近活动采集数据量 |
| TaskState | varchar(50) | 否 | 任务状态 |

任务状态：启动、进行中、停止、已停止；

启动/停止类型：用户启动、自动启动、用户停止、任务完成、出错

（2）表（GatherWebsite）：存放定向采集的网络站点信息，表结构如表3-2所示。

**表3-12 GatherWebsite**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **类型（长度）** | **是否允许为空** | **说明** |
| SiteID | 自动增长 | 否 | 主键，网站编号 |
| TaskID | varchar(50) | 否 | 外键，采集任务ID |
| SiteName | Varchar(50) | 是 | 站点名称 |
| SiteType | Varchar(50) | 是 | 站点类别 |
| SiteArea | Varchar(50) | 是 | 站点归属地 |
| SiteWeight | Int | 是 | 站点优先级 |
| SiteEnterURL | Varchar(50) | 否 | 唯一性，站点入口网址 |
| CrawlerDepth | Int | 是 | 采集深度 |
| FilterDomain | Varchar(50) | 是 | 过滤域名 |
| UserName | Varchar(50) | 用户名 | 需登录网站的用户名 |
| Password | Varchar(50) | 用户名 | 需登录网站的密码 |
| Remark | varchar | 是 | 备注 |
|  |  |  | 模板XML |

归属地：省/地市 形式存放，如浙江省/杭州市

（3）表（WebContInfo）：存放结构化网页内容信息的数据库为：WebContInfoDB，其中表（WebContInfo）用于存放定向采集的网页内容的结构化数据，表结构如表3-13所示。

**表3-13 WebContInfo**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **类型（长度）** | **是否允许为空** | **说明** |
| ID | 自动增长 | 否 | 主键，信息编号 |
| TaskID | varchar(50) | 否 | 外键，任务ID |
| SiteID | Varchar(50) | 是 | 外键，站点编号 |
| PageURL | Varchar(50) | 是 | 完整网址 |
| ColectTime | DateTime | 否 | 采集时间 |
| TextTitle | Varchar（50） | 否 | 文章标题 |
| Text | Varchar | 否 | 文章内容 |
| PageTime | DateTime | 是 | 文章发表时间 |
| reproter | Varchar（50） | 是 | 文章来源 |
| reproter | Varchar（50） | 是 | 文章作者 |
| Joiner | int | 是 | 文章参与人数 |
| Commenter | int | 是 | 文章评论人数 |
| EmotionClass | int | 是 | 情感等级 |
| Remark | varchar | 是 | 备注 |
| State | varchar(50) | 是 | 状态 |

# **4. 系统开发与实现实现**

## 4.1 任务监控管理平台

任务监控管理平台需要登录，登录界面如图4-1所示

**4-1任务监控管理平台登录页面**

登录成功后就可以添加相应的任务。添加任务详细信息的页面如图4-2所示。可以具体配置需要采集的站点，多站点采集用逗号隔开，采集站点的深度，需要过滤的域名，需要保留的域名，解析模板的名称。



**4-2任务配置页面**

任务添加完成后将会显示在任务状态页面显示，可以在此页面上对任务进行操作，是否需要启动或是将已启动的任务暂停、停止等操作，如图4-3****

**4-3任务状态页面**

## 4.2 定向采集器

定向采集器按功能大致上可以分为：检查任务状态模块、任务信息初始化模块、多线程集中管理模块以及爬虫线程模块。定向采集模块代码包组织如表4-1所示。

**4-1 定向采集模块的代码包组织**

|  |  |  |  |
| --- | --- | --- | --- |
| 组织方式 | 类名 | 函数名 | 函数功能 |
| **package**  **org.dgaa.crawler**  功能描述：  抓取模块 | **Gather**  功能描述：  抓取模块主类 | **public** **class** Gather **implements** Runnable | 调用函数，取得URL，调用**client**模块打开URL返回网页源码，调用函数获得新URL，调用**check**模块过滤URL后插入待抓队列。 |
| **Client**  功能描述：  调用httpclient获得网页源码 | **public** String HttpClient(urls url) | 获得URL，并同时设置httpclient环境，如：Socket 超时，以及 Socket 缓存大小，设置 user agent等，并同时返回有服务器得到的页面信息，判断页面的正确性。最后返回页面的源码。 |
| **GetURL**  功能描述：  用于获取新url | **Public** ArrayList<String> urlDetector(String htmlDoc) | 用正则表达式抽取页面源码中的新URL |

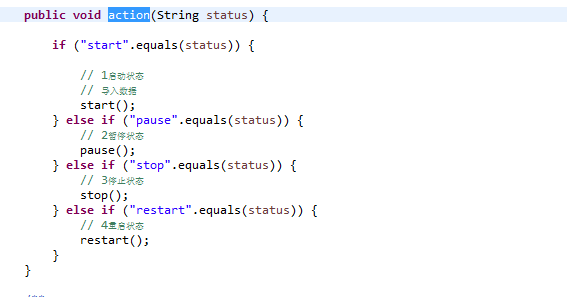
1. 检查任务状态模块：

该模块通过每隔一定的时间轮循控制台数据库中的状态字段，来对爬虫程序作出相应的指令。如图4-4所示



**4-4检查任务状态模块代码**

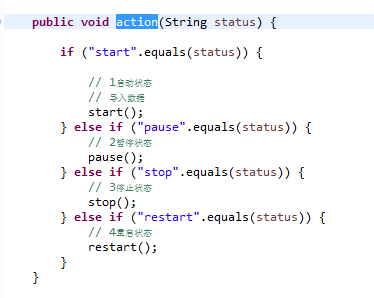
从控制台获取的状态参数，将转化成对应方法的参数，调用不同的方法进行操作。如图4-5所示



**4-5控制台参数管理代码**

1. 任务信息初始化模块

从控制台数据库导入种子队列以及相关的任务信息。如图4-6所示



**4-6 任务信息初始化代码**

1. 线程池调度模块

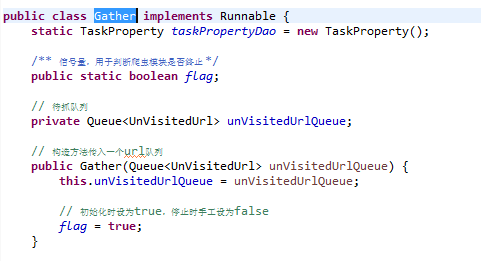
线程池调度模块自身维护了一个用于检测线程池中线程数量的线程，每隔一定的时间，该线程就查询线程池中的活跃线程数，如果小于一定数量，就创建一个爬虫线程，加入到线程池中运行。如图4-7所示



**4-7 线程池调度代码**

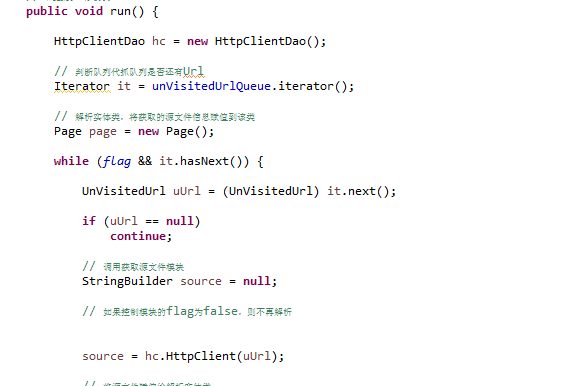
1. 爬虫线程模块

爬虫线程创建时会得到一定数量的待抓url，爬完当前的url，线程自动销毁，这样可以方便的管理整个程序的暂停、停止状态。如图4-8所示



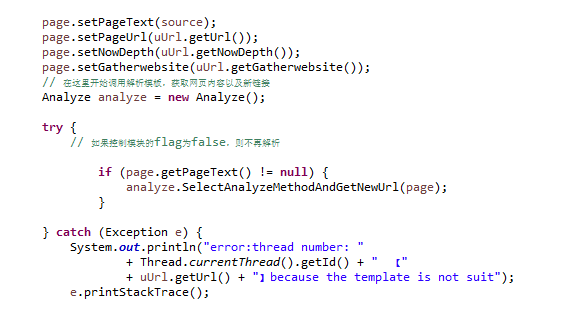
**4-8 爬虫线程代码**

当爬虫运行时，他会从队列中获取队首的url，使用httpclient打开后获得网页的源文件并返回。如图4-9所示



**4-9 网页获取代码**

获得的网页源文件将会封装成一个page类，交给解析模块处理。如图4-10所示



**4-10 网页信息封装代码**

**4.3 网页解析器**

网页解析器的作用就是将从采集器获得的网页源文件进行一次转化，根据用户的需要，将需要的内容转存到指定的数据库。解析器模块包组织见表4-2

**4-2 内容解析模块的代码包组织**

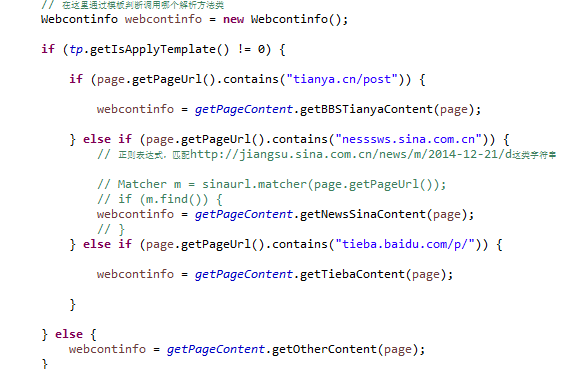
|  |  |  |  |
| --- | --- | --- | --- |
| 组织方式 | 类名 | 函数名 | 函数功能 |
| **package**  **org.dgaa.parser**  功能描述：  解析模块 | **Parser**  功能描述：  解析模块预留类,用于调用其他类 | **public** **void** run() | 调用函数获得URL，调用httpclient打开url获得网页源文件，分别调用**TextExtract和org.dgaa.**  **Parser.moudle**模块得到网页的一般信息和特定信息调用数据格式化模块**org.dgaa.format**，后交给**sql**模块将信息插入数据库。 |
| **Htmlpartser**  功能描述：同抓取模块中的**Client** | **public** String parser(urls url,dispatcher disp,String id) | 同抓取模块中的**Client** |
| **TextExtract**  功能描述：  正文抓取模块 | **public** **static** **void** setthreshold(int value) | 设置正文抽取的阈值 |
| **public static String parse**(String \_html) | 传入该网页的网页源文件 |
| **public static String parse**(String \_html, boolean \_flag) | 传入该网页的网页源文件，传入该网页的网页类型，true为内容页  false为标题页。并调用预处理函数**preProcess()**，调用函数抽取主函数**getText()** |
| **private static void preProcess()** | 正文预处理，通过正则表达式去除网页源码中的所有html标签，js代码 |
| **private static String getText()** | 正文处理主程序，返回抽取结果 |
| **package**  **Org.dgaa.parser.moudle**  功能描述：  网页的特定信息抽取  （分类数量较多，以新浪为例） | **sina**  功能描述：  以新浪网为模版的内容抽取 | **public static void sina**(String str) | 接收网页源文件并调用解析函数sina\_parser(String str) |

当获得网页源文件后，解析模块首先判断是否达到指定的深度，若达到深度，将不解析新的url，直接解析内容。如图4-11所示



**4-11 深度判断代码**

然后解析模块会判断链接的主站，从而引用不同的模板对网页进行进行解析。如图4-12所示



**4-12 调用模板代码**

如果解析的结果是该页面没有正文内容，说明这个页面为导航页或者是广告业，直接舍弃，否则将封装成类存入指定的数据库。如图4-13所示

## 

**4-12 解析代码**

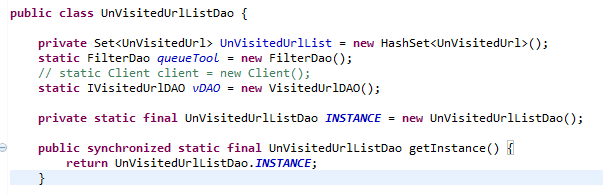
## 4.4 过滤模块

当获得新url后，需要对其进行一次检验，检验其是否已经被抓取过，在数量级小的时候，过滤模块的设计对系统的影响并不明显，但当数量达到百万级的时候，一个合理的队列——遍历设计就显得重要了起来。过滤模块包组织见表4-3

**表4-3 URL合法性监测模块的代码包组织**

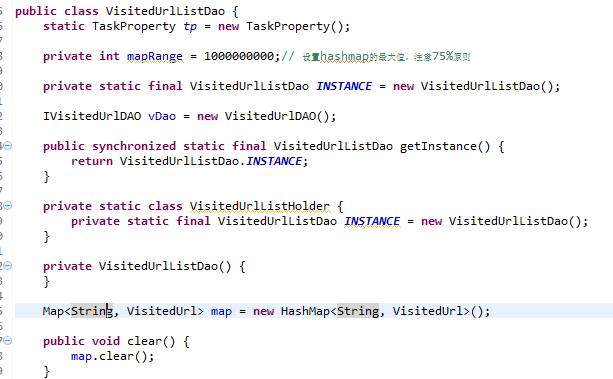
|  |  |  |  |
| --- | --- | --- | --- |
| 组织方式 | 类名 | 函数名 | 函数功能 |
| **package**  **Org.dgaa.check**  功能描述：  用于检查url是否符合条件 | **Check**  功能描述：  同前 | **public** **static** ArrayList<String> check(ArrayList<String> Urls,dispatcher disp,urls url){ | 接受url，以重定向、已抓、待抓的顺序剔除url返回符合条件的url数组。 |
| **rangeHost**  功能描述：  用于判断页面是否在限定范围内 | **public** **static** ArrayList<String> range2(ArrayList<String> analyzedURL, URL rangeURL,**int** deep) **throws** MalformedURLException | 用比对的方法，拿新url和范围url做对比，返回在范围内的url数组。 |

首先，将待抓队列设计成单例，并且用了set——不重复队列，单例是为了防止多线程操作时引起的错误。如图4-13所示



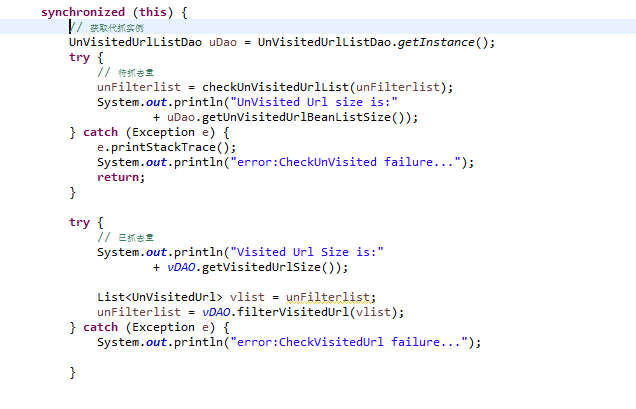
**4-13 待抓队列代码**

已抓队列设计成了遍历效率更高的HashMap，并考虑到了HashMap在存储量为75%时达到一个较优的平衡，并且也将其设计成了单例，理由同上。如图4-14



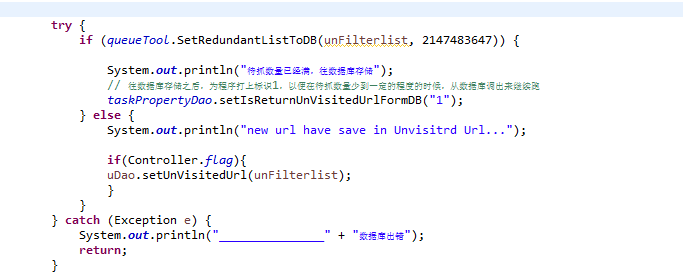
**4-14 已抓队列代码**

过滤时，首先过滤待抓队列，因为其数量相对较少，然后过滤已抓队列。如图4-15



**4-15 待抓、已抓过滤代码**

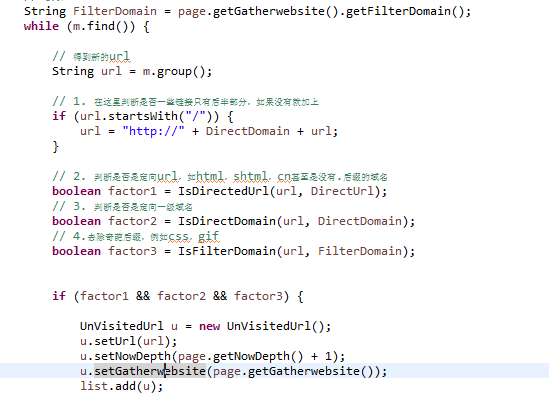
如果两轮过滤都通过，说明该url并未被抓取过，然后判断待抓队列是否过大，如果过大，则将当前url持久化到数据库，等待抓队列数量小于一定范围后再导入。如图4-16





**4-16 url持久化**

新增url还要根据控制台配置的参数去除一些不需要的域名，例如：bbs，news等。如图4-17所示



**4-17 定向代码**

# **5．系统性能测试**

系统测试环境见表5-1.

**5-1 系统测试环境**

|  |  |  |  |
| --- | --- | --- | --- |
| 软硬件环境 | 控制台服务器 | 采集终端 | 数据库服务器 |
| 硬件环境 | CPU 600 MHz 以上，建议 1GHz  内存 256 MB以上  硬盘 200G | CPU 600 MHz 以上,建议1GHz  内存 256 MB以上  硬盘 200G | Cpu 800 MHz,建议1 .6GHz  内存512 MB以上  硬盘 500G |
| 软件环境 | Windows 2003 server  JDK：1.6.0\_45  Apache:2.2.0  Tomcat:7.0.8 | Windows 2003 server  JDK：1.6.0\_45 | Mssqlserver |
| 网络环境 | 10M LAN | 10M LAN | 10M LAN |

## 5.1测试案例设计

为了能够全方位的了解系统的运行情况以及系统的抗压能力，本次测试一共设计了四个实验案例。每一个案例都在同一类网站中具有其代表意义。本次测试将会以12小时为间隔记录数据观察数据的变化情况并随时观察程序输出的日志信息，判断程序是否运行良好。

## 5.3 测试对象

本次测试将会分别以如表5-2中所列的网站为起始地址，分时段记录数据，并将整理好的数据以图表的形式附在测试报告后。

**表5-2 测试网站名称及地址列表**

|  |  |
| --- | --- |
| 网站名称 | 网站地址 |
| 浪新闻 | http://news.sina.com.cn/ |
| 天涯论坛 | http://bbs.tianya.cn/ |
| 百度贴吧 | http://tieba.baidu.com/ |
| 搜狐新闻 | <http://news.sohu.com/> |

本次测试的目的旨在测试软件的性能，在长时间，大规模的情况下程序是否能够高效切稳定运行。测试将从页面采集总体数量、页面总体解析量、单位时间获取页面数、单位时间解析数、以及每个时段的采集效率差等几个方面对软件进行全方面的测试，随后将会附上具体的测试结果报告。

## 5.4 测试结果

测试案例1：天涯社区

①起止时间

开始时间：2014-10-7-21:30

结束时间：2014-10.9-21:30

间隔时间：48小时

②页面相关信息收集

开始时间：2014-10-11 20:31:57.497

结束时间：2014-10-13 20:32:30.223

间隔时间：48小时

总共采集页面数量：1661363

总共解析页面数量：1329090

页面解析率：85.73%

单位时间获取页面数：396.95/分钟

单位时间解析数：340.32/分钟

测试周期页面抓取情况实时反馈如图5-1所示。

**5-1 测试情况反馈图**

③单页面解析效果对比图

原网页如图5-2：



**5-2 原网页**

解析结果（数据库记录）见表5-3：

**5-2 数据库存储记录**

|  |  |  |
| --- | --- | --- |
| url | time | text |
| <http://bbs.tianya.cn/po>  st-feeling-3689741-1.shtml | 2014-10-9-21:42 | “医生请你无论如何都救救我的孩子”一声悲怆却异常有力的声音划破广西贺州市人民医院烧伤整形外科住院部的宁静。 　　6月21日下午5时，一场灾难降临到广西贺州市八步区信都镇西两村一个… |

测试案例2：百度贴吧

①起止时间

开始时间：2014-10-07 20:42:41.143

结束时间：2014-10.9-20:42:41.143

间隔时间：48小时

②页面相关信息收集

总共采集页面数量：314089

总共解析页面数量：138199

页面解析率：44.62%

单位时间获取页面数：571.59/分钟

单位时间解析数：255.04/分钟

测试周期页面抓取情况实时反馈如图5-3

**5-3 测试情况反馈图**

③单页面解析效果对比图

原网页如图5-4：



**5-4 原网页**

解析结果（数据库记录）见表5-3：

**5-3 数据库存储记录**

|  |  |  |
| --- | --- | --- |
| url | time | text |
| http://tieba.baidu.com/p/3339231875 | 2014-10-11 20:31:53.587 | 贴吧熊孩子十一假期过的爽爽的~~~ 十一期间也有很多小伙伴新加入我们贴吧熊孩子大家庭~~~ 为了使贴吧熊孩子大家庭更加的壮大~~~招募给力的八五小伙伴是必须的~~~ 根据规划，本吧准备招募以下方向吧务：… |

测试案例3：新浪

开始时间：2014-10-10 17:18:39.010

结束时间：2014-10.12-17:18:48.033

间隔时间：48小时

②页面相关信息收集

总共采集页面数量：540891

总共解析页面数量：140632

页面解析率：26.83%

单位时间获取页面数：187.81/分钟

单位时间解析数：50.40/分钟

测试周期页面抓取情况实时反馈如图5-5

**5-5 测试情况反馈图**

③单页面解析效果对比图

原网页如图5-6：



**5-6 原网页**

解析结果（数据库记录）见表5-4：

**5-4 数据库存储记录**

|  |  |  |
| --- | --- | --- |
| url | time | text |
| http://news.sina.com.cn/c/2014-10-10/103930968478.shtml | 2014-10-10 17:18:39.273 | 2014年3月25日，第三届核安全峰会在荷兰海牙召开。本次会议围绕“加强核安全，防范核恐怖主义”这一主题，就如何维护全球核安全达成新共识，内容涉及削减高浓缩铀核材料数量、加强放射性材料安保监管、推动国际信息沟通等方面。… |

# **6 结论**

研究这个课题已经大半年的时间了，从最初完全无从着手，到查阅文献，上网搜索，总结出几条可行思路，到发现实现起来困难重重，自己掌握的知识还不足等等。接近半年的时间在查阅文献，找资料中度过，最终完成这项课题，感触良多。

此系统的任务就是能够对指定的互联网站点及信息内容进行自动分类采集，并对采集的网页信息进行智能解析生产结构化数据保存到数据库。最终的毕业设计成果已经达到了最初期望的设计要求，完成了各项需求。

在进行毕业设计的过程中，我充分运用到了本科四年所学到的知识。并且在爬虫这一块也有了一定的研究。通过这次毕设，我的编程思想、编写代码的实操能力、文档编写能力有了很大的提高。

因为时间有限，本毕业设计也不是完成的尽善尽美。爬虫在性能方面一直有瓶颈，如果能够投入跟多的时间，相信整套系统会有更大的改善。