The Research on Ajax Dynamic Parse and Application of Ajax Plugin in Nutch

Abstract

Nutch is an excellent open source web search engine. Its JavaScript plugin has the ability to parse static links but cannot parse dynamic links, so Nutch can't access the content belonging to deep web that is generated by executing script such as JavaScript. To solve this problem, this paper puts forward to crawl dynamic page content by providing JavaScript engine to execute JavaScript on page in the page crawling phase, so as to enable the existing JSParseFilter plugin to pick up as much as possible page links from deep web. Experiments indicate that Nutch's ability to access deep web can be improved obviously by using the method in this paper.

Keywords: search,JavaScript parsing,ajax,deep web,nutch

Introduction

Nutch is a open source search engine based on lucene, which provides a architecture for full-text index and search including crawler,parser,distributed storaging and parallel computing modules,indexer and searcher.Fetures as follows contribute nutch to a best open source search engine:first,making use of lucene efficient indexing;second,providing capability of distributed and parallelly computing through hadoop which implements mapreduce programming model similar to google's and a distributed file system HDFS;third,nutch is a open source project that is very suitable to search engine researcher;last but not the least,nutch supports plugin development model which make it easily extended.Due to these virtues,nutch becomes more and more important in search technology and deep web research.Expanding nutch becomes a usual thing, especially in chinese segment and page sort.However,researchs about dynamic page parsing are few,resulting in that ability of nutch to crawl sites which use lots of javascript and ajax code is nearly absent.Hence,in this paper will introduce rhino based ajax parser into parsing course in nutch so as to get more valid

links.Experiments show that this method effectively increases parsed links in crawl procedure.

1 Nutch And Ajax parser

1.1 Methods To Get Page Links In Nutch

The base principle of current page link parsing method in nutch is that crawl static page by crawler,then extract static links in page by HtmlParser and JSParseFilter.First,Nutch Plugin Chain calls HtmlParser to extract Outlinks in page,including links pointing to outside and inside stations.Then,Plugin Chain call JSParseFilter to extract static links in page,including static links existing in attributes starting with href or on in tags.

1.2 Ajax Parser

The common way to get dynamic page is implemented through putting explorer core in crawler,but in the explorer parsing load lots of content irrelated to the target topic such as layout,compilability,advisement codes,so the time and space efficiency is very low.In order to solve this problem,some one suggests that introduce JavaScript parser into crawl course.The usual event flow is that get page,construct DOM,mark JavaScript,construct host object,execute JavaScript and return dynamic page.Although this way can reduce loading irrelated codes ,still expose some disadvantages:first,loading topic irrelated JavaScript;second,recalling remost host to get external JavaScript is very low efficiency.So this paper introduce a method to get dynamic page content based on rhino.

1.3 System Design of Ajax Parser Based on Rhino

The page to be processed is from crawler in nutch,the whole request flow can be divided into several stages.First,create JavaScript framework and filter library and generate all the host object.Next,generate DOM objects of current page by HTML DOM Parser,marking whether JavaScripts in current page need to be parsed by JavaScript Parser and whether need to be loaded from local JavaScript library by JavaScript framework library manager.Then,run rhino to execute these JavaScript

code segments.Lastly,just return the result page content.The data flow in rhino is showed in picture one,process flow in the whole Ajax Parser is showed in picture two.

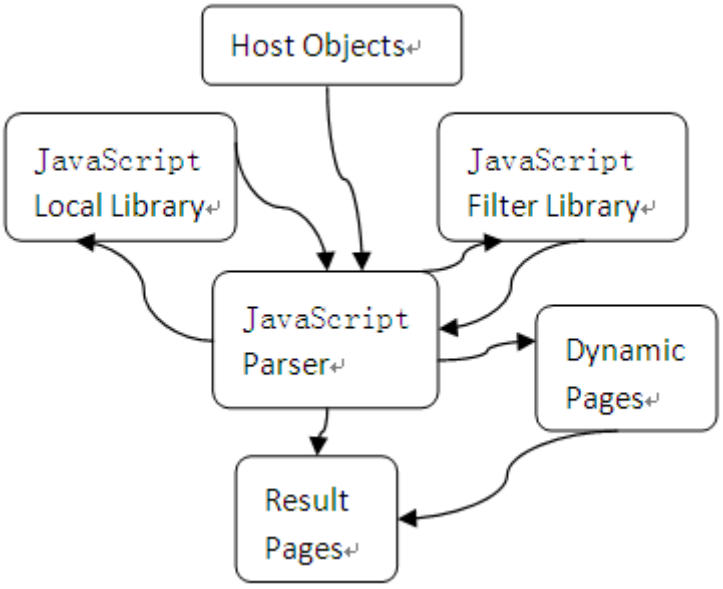

Figure1:Data Flow in Parser

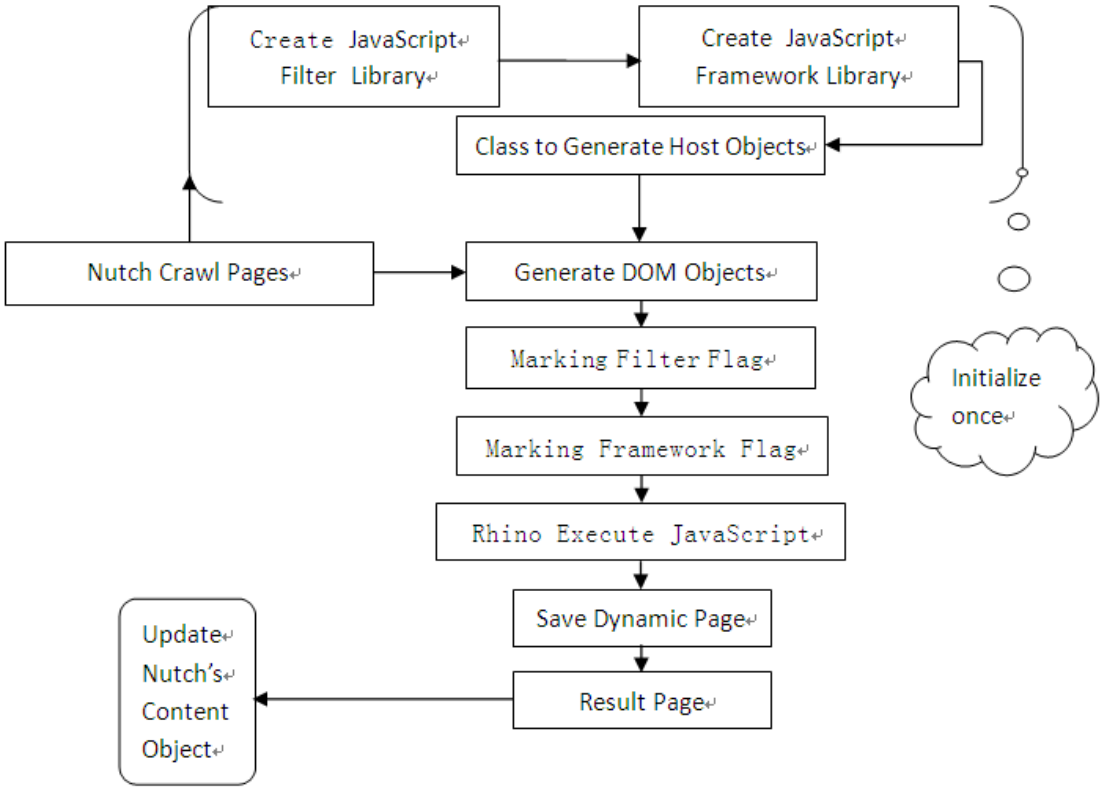

Figure 2:Process Flow of Ajax Parser Based on Rhino

The innovation points in this design are the JavaScript filter and framework library.The filter module consists of topic irrelated JavaScript files in current page and library filter used to check whether exist not needing processed JavaScript files in current page and reduce loading JavaScript files not relevant for target topic so as to

improve parsing efficiency.The framework library is initialized by storing commonly used JavaScript files,subsequently,constantly improving the library based on external JavaScript files requested by remote client.Of course,the library also need a maintenance module to update data manually as needed.Loading JavaScript files should need load from remote host from local framework library will reduce interaction with remote host making for decreasing time for loading external JavaScript files.

Rhino execute JavaScript in current page needing support from all kinds of host objects such as Html DOM,Event Object,Browser Object,CSS Object,XMLHttpRequest Object and so on,at the same time also needing Html Parser to generate page corresponding DOM Object.In the parsing process,rhino takes corresponding actions on the basis of filter flag and framework request flag.

2 The Key Technology of Ajax Parser Based on Rhino

2.1 Construct JavaScript Filter Library

Construct JavaScript filter library irrelevant to target content based on topic,the library mainly comprises two kinds of executable library .One is the JavaScript library file apparently irrelevant to topic,for instance,the JavaScript files used to change page layout.The other one is JavaScript library used to analyse customer satisfication online and sites traffic information,for instance,ForSee Results Survey Code and advertising service showed by JavaScript provided by Baidu Union,Taobao Union and Google AdSence.

For the choice of content,mainly through parsing each page in crawl url set one by one to put those JavaScripts in current page relevant to target topics to JavaScript filter library.

In the selection process,there are a few points to note:

(1) A third JavaScript library useless for a topic maybe is necessary for the other topics.

(2) If the third party JavaScript library includes many JavaScript files,the Complete logic dependency between files should be promised.

(3) Store third party JavaScript library named after the keywords in JavaScript file name.

Moreover,JavaScript filter library needs some management modules:

(1)Add functions for maintaining library information

(2)Add functions for filtering external JavaScript files

When system loads external JavaScript files, search JavaScript filter library based on the keywords in file name.If exists,set filtering flag 1 in corresponding position of DOM,if not,set normally loading flag 0.

2.2 Create JavaScript Framework Library

The library should be initialized with jQuery,Ext,Dojo,Google Web ToolKit,ProtoType,YUI,and then create mapping relations for these libraries so as to search and load rapidly.Subsequently,check whether exist same files based on keywords in JavaScript files from every request.If no such files exist,send a ajax request to get the file and save it to JavaScript library.For the convenience of using,library maintaining modules and judging module are used for marking whether load from the framework library.

A.Library maintaining modules as follows:

(1) adding file to JavaScript framework library needs to validate file's logic completeness.

(2) deleting file from JavaScript framework library needs to validate file's logic completeness.

(3) files included by each framework file set and dependency relationships between files can be configured by xml file.

B.Judging whether load files from JavaScript framework library

Check the filter flag of the JavaScript file to judge whether the file needs to be loaded.There are two situations to process:

(1) If filter flag is true,ignore this request and process subsequent request directly;

(2) If normal loading flag is true,search the JavaScript framework library to find same file based on the keyword of current JavaScript file.Then two situations need to be processed:

   i. If same file is found, set flag attribute value of DOM node corresponding to the file to 2 representing loading from local framework library.

   ii. Or else,process next.

2.3 Generating all kinds of host object

A. Implementing HTML DOM

Implementing corresponding level DOM object based on DOM specification,this implementation should include all DOM objects in current specification and ensure

that all attributes of common object and common attributes of all objects should be implemented so as to relevant objects can be found when JavaScript engine runs.

B. Implements CSS

The only attention in crawling course is target page content,so it is enough to provide basic implementation for CSS used for page layout and presentation.Although,the implementation should include these functions at least,the methods to convert CSS style to text property and text,style set CSS supports,the rules CSS Selectors use and parsing modules CSS StyleSheet use.Notice that,JavaScript engine may throw exceptions in executing course because of missing attributes or methods from corresponding CSS implementation.

C. Implementing EVENT Object

The module should include Event registering and dispatching which are used for triggering events predefined.

D. Implementing BOM Object

This module is mainly used for providing browser objects to JavaScript Parser,including Window,History,Navigator,Screen,Document,Location and so on. Among of which Objects such Window,History,Document,Location should implement all the attributes and methods,or else exception information should be provided.

E. Implementing XMLHttpRequest

Implementation of the object should include open,send,setRequestHeader,getResponseHeader,getResponseHeader,of which send method should process Cookie.

3 Integrate Rhino Based on Ajax into Nutch

3.1 Implementing Ajax Parser Based on Rhino

Create manager class FilterDB and FrameWorkDB for JavaScript filtering library and framework library separately responsible to initialize library and providing related methods talked about in second section aforementioned.Filtering library and framework library both store key/value data,here key storing hash value of file name and value storing file name,so Hadoop filesystem can be used for storing these two libraries in order to improve file search efficiency and time and space usage by using its capability of distributing store and paralleling computing.

Customizing envjs which is a open source browser based on javascript to construct all kinds of host objects.Then,using Rhino parser read file to initialize excuting context javascript will run and access to FilterDB and FrameWorkDB as need.Sequentially,add interface to read page content into Envjs and load page content

into Envjs by Envjs.load().Finally,excute JavaScript codes in current page by Rhino to get parsed page content.

3.2 Integrate Ajax Parser to Nutch

Adding Ajax Parser to Fetcher in nutch and ensure that it is called before HtmlParser and JSParseFilter called ,as Figure 3 depicts,so as to the dynamic page content can be processed by Parser in nutch.However,not all web sites employ dynamic technologies,so it is necessary to adjust page set to be processed by Ajax Parser as need so as to just process pages that include ajax calls.

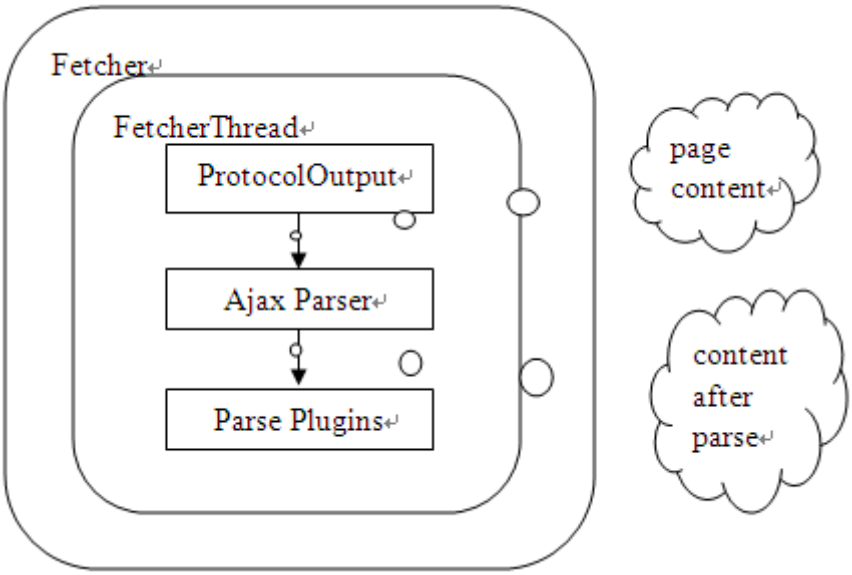

Figure3: The flow Nutch calls Ajax Parser by

4 System Implementation and Result

4.1System Configration

Single pseudo-distributed PC connected with Intel(R) 82579LM Gigabit Ethernet is employed in this experiment,Master Node and Slave Node both run on same pc with DELL i3-2100 CPU@3.10GHz and 4G memory.The pc needs to be equipped with Ubuntu11.04 OS,OpenSSH protocol,JDK1.6,Nutch1.2,Rhino1.7.

4.2 Experiment Result Analysis

In this experiment, site http://www.fda.gov/Safety/Recalls/default.html is selected to test processing efficiency to JavaScript and Ajax in fetching course in nutch.These feaures of selected site which owns a table generated by Ajax request and includes 16

external JavaScript files and some JavaScript code segments,among of 16 external files or segments exist that file requests from local host and file requests from remote host and three files used for investicating customer satisfaction make it very excellent sample to validate system.The validation methods are to check on whether need to filter external JavaScript files and get a better efficiency improvement,whether get a better efficiency improvement by loading files should be loaded from remote host from local host,whether could get right dynamic content parsed by Ajax Parser.Among the course,in order to distinguish out whether need to be filtered we narrowed the scope to food public sentiment,thus three files about Forsee Result Survey become the target to be filtered naturally.

For the sake of watching clearly output from Ajax Parser,classified statistic results in different state are showed in Figure 6 based on whether enable Ajax Parser and its filter library and framework library.The test result mean values were picked up by running 50 times Ajax Parser in nutch in the same input.Because network is unstable when access remote host,the parse time is a interval which also implys the necessity of JavaScript framework library.

```
<!--==== START Dynamic List ====-->
    <noscript>To view the latest recalls, please visit the 2010 Recalls, Market Withdrawals & Safety Alerts pag
    <h3 class="head2">Recently Posted Recalls</h3>
<div id="tabs">
    <ul>
    <li><a id="AllRecallslink" title="Link to All Recalls" href="AJAX/default.htm?Label=All Recalls">All Recall
    <li><a id="Foodlink" title="Link to Food" href="/AJAX/Food/default.htm?Label=Food">Food</a></li>
    <li><a id="Drugslink" title="Link to Drugs" href="/AJAX/Drugs/default.htm?Label=Drugs">Drugs</a></li>
    <li><a  id="AnimalHealthlink"title="Link to Animal Health" href="/AJAX/AnimalHealth/default.htm?Label=Anima
    <li><a  id="Biologicslink" title="Link to Biologics" href="/AJAX/Biologics/default.htm?Label=Biologics">Bio
    <li><a id="MedicalDeviceslink" title="Link to Medical Devices" href="/AJAX/MedicalDevices/default.htm?Label
    </ul>
</div>
<!--==== END Dynamic List ====-->
```

Figure 4: the code in dynamic Table

```
<!--==== START Dynamic List ====-->
<noscript id="" >To view the latest recalls, please visit the 2010 Recalls, Market Withdrawals &amp;amp; Safe
<h3 id=""  class="head2" >Recently Posted Recalls</h3>
<div id="tabs"  class="ui-tabs ui-widget ui-widget-content ui-corner-all" >
    <ul id=""  class="ui-tabs-nav ui-helper-reset ui-helper-clearfix ui-widget-header ui-corner-all" >
        <li id=""  class="ui-state-default ui-corner-top ui-tabs-selected ui-state-active" >
          <a id="AllRecallslink"  title="Link to All Recalls"  href="#Link_to_All_Recalls" >All Recalls</
        <li id=""  class="ui-state-default ui-corner-top" >
          <a id="Foodlink"  title="Link to Food"  href="#Link_to_Food" >Food</a></li>
        <li id=""  class="ui-state-default ui-corner-top" >
          <a id="Drugslink"  title="Link to Drugs"  href="#Link_to_Drugs" >Drugs</a></li>
        <li id=""  class="ui-state-default ui-corner-top" >
          <a id="AnimalHealthlink"  itle="Link to Animal Health"  href="#ui-tabs-1" >Animal Health</a></l
        <li id=""  class="ui-state-default ui-corner-top" >
          <a id="Biologicslink"  title="Link to Biologics"  href="#Link_to_Biologics" >Biologics</a></li>
        <li id=""  class="ui-state-default ui-corner-top" >
          <a id="MedicalDeviceslink"  title="Link to Medical Devices"  href="#Link_to_Medical_Devices" >M
    </ul>
    <table cellpadding="3" border="1" cellspacing="0" id="AllRecalls"
      class="tablesorter" summary="Layout table showing Recalls with six columns: date, brand name, produ
      <thead>
       <tr>
         <th scope="col" title="Click to Sort By Date" align="left" width="15%">Date</th>
         <th scope="col" align="left" width="22%">Brand Name</th>
         <th scope="col" align="left" width="18%">Product Description</th>
         <th scope="col" align="left" width="16%">Reason/ Problem</th>
         <th scope="col" title="Click to Sort By Company" align="left" width="19%">Company</th>
         <th scope="col" align="left" >Details/ Photo</th>
       </tr>
      </thead>
      <tbody>
      <tr >
      <td nowrap>12/10/2011</td>
      <td><a href="/Safety/Recalls/ucm283288.htm">Pacific International Marketing</a>  </td>
```

Figure 5: Output result processed by Ajax Parser

Ajax Parser Using State	JS external Files Num	Loading Num in Parsing	Internal Request Num	External Request Num	Ajax Parsing Time	Result Num	Loading or Filtering State
unused	16	16	14	2	0s	77	no loading
use and not filter	16	16	14	2	21s-3min	117	loading Forsee Result Survey
filter	16	13	11	2	15s-3min	115	no loading framework
load framework	16	13	13	0	9s	115	loading www.google.com/jsapi,addthis_widget.js

Figure 6:Comparison analysis before and after using AjaxParser

From the above results,adding Ajax Parser based on Rhino to Nutch can effectively increase links from page.At the same time,introduces JavaScript filter library and framework library to parser makes remote request local.In turn,avoids nutch performance lessening obviously because of adding dynamic page parsing.

5 Conclusions

Due to only static page content considered in nutch,this paper introduces Ajax Parser to Nutch so as to add dynamic page parsing fuction for nutch.Moreover,because add JavaScript filter library and framework library in dynamic page content parsing,the performance of fetching phase in nutch doesn't drop owing to adding dynamic parsing module.Experiments show that compared with traditional way Ajax Parser based on Rhino has advantages in performance and is capable of getting more links through dynamic parsing in parsing phase in nutch.Of course,the work is also immature,supporting for host object is incomplete which may result in exceptions in parsing pages from various sites, the initialization of JavaScript filter library and framework library is dependent on experience which may be optimized through machine learning algorithms,hoping that these unfinished work can be resolved in future.

References

[1] Rohit Khare, Doug Cutting, Kragen Sitaker, Adam Rifkin, "Nutch: A Flexible and Scalable Open-Source Web Search Engine", WWW 2005, May 10–14, 2005,

Chiba, Japan.

[2] Manuel álvarez1, Juan Raposo1, Alberto Pan1*,Fidel Cacheda1, Fernando Bellas1, V ́ctor Carneiro1, “Crawling the Content Hidden Behind Web Forms”, This research was partially supported by the Spanish Ministry of Education and Science under project TSI2005-07730.

[3] Alexandros Ntoulas,Petros Zerfos,Junghoo Cho , “Downloading Textual Hidden Web Content Through Keyword Queries”, JCDL’05, June 7–11, 2005, Denver, Colorado, USA.

[4] Sriram Raghavan, Hector Garcia-Molina, “Crawling the HiddenWeb”, Proceedings of the 27th VLDB Conference, Roma, Italy, 2001

[5] Luciano Barbosa,Juliana Freire, “An Adaptive Crawler for Locating HiddenWeb Entry Points”, WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.

[6] CaiXiao-yan,KOU Ying-zhan,SHEN Wei,ZHEN Wei,” Rea liza tion of JE Chinese Segmentation on Nutch-0.9”,Science Technology and Engineering,Vol.8 No.17 Sep.2008

[7] LI Cun-he,LU Ke-qiang,”Research of page-ranking modifying method on search engine Nutch”,Computer Engineering and Design, 2009,30 (6) 1343

[8] Fan Xuanmiao,Zheng Ning,Fan Yuan,”DES IGN AND IM PLEM ENTATION OF A CRAWLERMODEL BASED ON AJAX”,Computer Applications and Software,Vol.27,No.1,Jan 2010

[9] LI Shu yu,WU Jian,HU Zheng guo,”Design and Implementat ion of Embedded JavaScript Interpreter”, Application Research Of Computers,2003

[10] PU Dong-bing,YANG Li-ming,ZHOU Yan-jun,CHE Wen-long,MA Zhi-qiang,”Design of JavaScript interpreter on embedded explorer”,Information Techonology,2010.4

[11] JIN Xiao-ou,ZHONG Bao-yan,LI Xiang,” Research and implementation of Interpreting JavaScript Dynamic Web Page Based on Rhino Engine”,Computer Technology and Development,Vol.18,No.2,Feb.2008

[12] WANG Ying,YU Man-quan,LI Sheng-tao,WANG Bin,YU Zhi-hua,” Extracting Dynamic URLs Using JavaScript Engine”,Computer Applications,Vol.24,No.2,Feb.2004