Automatic and Dynamic Updations in Web Collection cycle using Web Crawl System

Mr. F.Destonius Dhiraviam, Ms. R. Vijindra, Ms. J. Hepzibha Elizabeth

Abstract— In Web search engines as well as many more specialized search tools rely on web crawlers to acquire large collections of pages. A Web crawler is a computer program that browses information and means of providing up-to-date data's. Such a web crawler may interact with millions of hosts over a period of weeks or months, and thus issues of robustness, flexibility, and manageability are of major importance. In this paper, we describe an implementation of a web crawler that runs on a network of workstations. It discusses the performance bottlenecks, and describes efficient techniques for achieving high performance. In current trends, web information's is increasing rapidly due to smart devices and advanced technologies. Due to this rapid growth of web information's in network bottleneck problem occurs. The information in the web platforms and social networks is continuously updated in anywhere and anytime from the certain web page. The user has to give URL for retrieving information from particular web page and the fetched information will be stored into the database of the system.

Index Terms— Search Engine, Web Crawl, Bottleneck, Social Networks, Data Base

1 Introduction

The information in the web contents is increasing rapidly based on consumer using current technology platforms. Such as Smart devices, advanced technologies, wireless network are also assisting the flood of information. Development of web data is very much faster than ever we think. They are composed of complex and inter connected non-hierarchical structures. Later they have short creation and destruction cycle, and do not have physical boundaries. Therefore, web data search engine requires the complicated search process and strategy. It is very significant to determine a data collection cycle to increase performance. The short collection cycle is advantageous for frequent update web sites and long collection cycle is the opposite. That is, it will be efficient if we can dynamically determine the collection cycle. In this paper, we are trying to capture some of the basic advanced level of data capture in an automated manner. The initial startup of the project is designing a crawler which will crawl and fetch the entire web contents of a particular page. In addition, the crawler can retrieve the hyperlinks and other information's in a clear segregated manner. After fetching such information's, the data will be arranged in an orderly manner for future usage. Once the website is changed, the images and data will be captured and compared periodically and an automatic fetching mechanism is in-built into our system. Redirecting to multiple sub URL's at the same time in the web page is one of the major advantage of our system.

We define a region in a web document as an HTML fragment that shows information about one or more related items when it is rendered on a web browser. Such items can be data records, e.g., information about products, services, goods, or pieces of news, headers with navigation menus, footers with contact and corporate information, or sidebars with advertisements, to mention a few examples. In the sequel, we make a distinction between individual data records, data regions (which encompass a series of data records). The majority of region extractors focus on data records and data regions.

2 Backround

Internet Archive [6] uses parallel crawler to crawl the websites. Each crawler process is single threaded which takes a list of seed URLs and fetches pages in parallel. Mercator [7, 8] is an extensible web crawler suggested a general architecture of a parallel crawler. In this paper, we suggest dynamic management of web pages collection cycle time depending on the update characteristics of the websites. Frequent updates require shorter collection cycle time for collecting up-to-date web pages. Therefore, we need update characteristics of each websites for considering dynamic changes of update frequency. This can be estimated by probing each web site periodically. In the next section, we propose a system that uses dynamic web collection cycle time.

3 Design And Implementation of Dynamic Web Crawling Scheduler

In this section, we introduce dynamic web crawler design and implementation in detail.

3.1 Design of Dynamic Web Crawling Scheduler

The web data collection cycle is determined by the process of Fig.1 when web data is changed. We first identify whether the web contents are changed or not by periodic website probing. We then calculate web data collection cycle, and store them to database.

We first extract link value from html files and compare it with previous link value. If the values are different, we determine optimal web data collection cycle point. The procedure is

• *Mr. F.Destonius Dhiraviam currently working as a Asstistant Professor/Computer Science and Engineering in V.R.S. College of Engineering and Technology, Arasur, Villupuram. Tamil Nadu. PH-9442472423. E-mail: ddhiraviam@gmail.com*
• *Ms. R. Vijindrais currently currently working as a Asstistant Professor/ Information Technology in Ranippettai Engineering College, Vellore. Tamil Nadu. PH-949488841133. E-mail: vijindrarajenran@gmail.com*
• *Ms. J. Hepzibha Elizabeth currently currently working as a Asstistant Professor/ Information Technology in Mohamed Sathak Engineering College, Ramnad. Tamil Nadu. PH-9751512850. E-mail: Hepzi.it@gmail.com*

as follows.

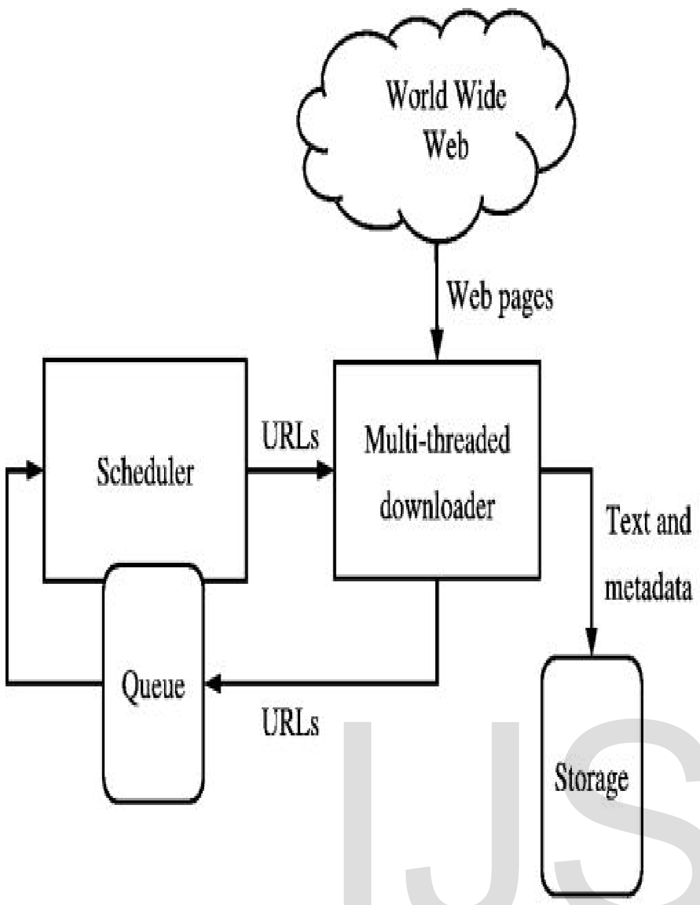


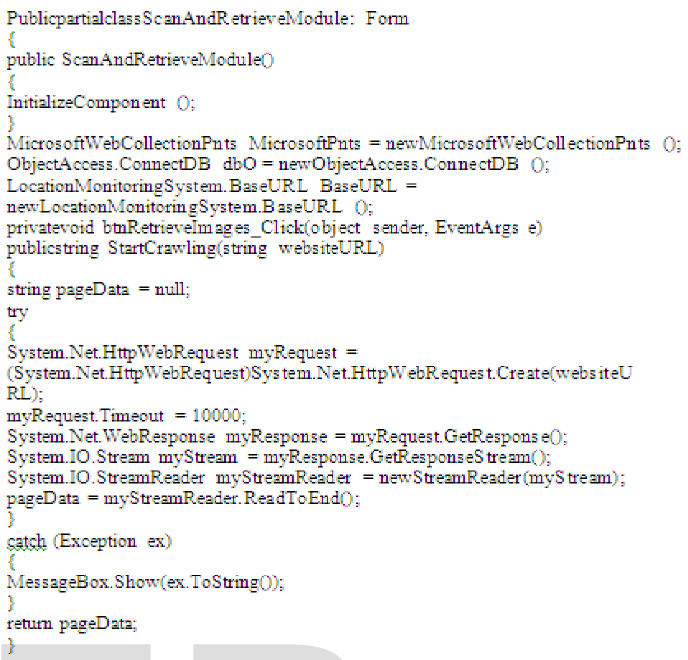
Fig.1 Implementation of Web Crawler

1. A World Wide Web crawler used to crawls the information from the web page.
2. The collected information's will be send to Multi-Threaded downloader for the further process.
3. If the collected information's is text and metadata, it will send to the storage otherwise URLs will be send to Queue.
4. URLs which are present in a queue will be transferred to scheduler, to schedule a information's.
5. From the scheduler the URLs will be again send to step 2.
6. The Procedure from step 1 to 5 will be repeated.

Table I shows the scan and retriving extracting information of the target search web sites. The SQL performs dynamic extraction of target web sites by using two main tables of design view and data sheet view page.

We determine collection cycle with adding half of the middle of the three variables to the biggest variable. If the average collection cycle time is biggest, we calculate collection cycle with multiplying average collection cycle with constant of less than 1. We can reduce data collection cycle with fixing to middle of the previous collection time and current collection time. We multiply the current collection time to constant over 1 when the pages are not changed. We fixed the constant 1.2 as the bigger than 1.2 makes too late delay time.

TABLE I
DYNAMIC EXTRACTION SQL OF TARGET WEB PAGES



```
PublicpartialclassScanAndRetrieveModule:  Form
{
public ScanAndRetrieveModule()
{
InitializeComponent  ();
}
MicrosoftWebCollectionPnts  MicrosoftPnts = newMicrosoftWebCollectionPnts  ();
ObjectAccess.ConnectDB  dbO = newObjectAccess.ConnectDB  ();
LocationMonitoringSystem.BaseURL  BaseURL =
newLocationMonitoringSystem.BaseURL  ();
privatevoid  btnRetrieveImages_Click(object  sender, EventArgs  e)
publicstring  StartCrawling(string  websiteURL)
{
string pageData  = null;
try
{
System.Net.HttpWebRequest  myRequest  =
(System.Net.HttpWebRequest)System.Net.HttpWebRequest.Create(websiteU
RL);
myRequest.Timeout  = 10000;
System.Net.WebResponse  myResponse = myRequest.GetResponse();
System.IO.Stream  myStream  = myResponse.GetResponseStream();
System.IO.StreamReader  myStreamReader  = newStreamReader(myStream);
pageData = myStreamReader.ReadToEnd();
}
catch (Exception  ex)
{
MessageBox.Show(ex.ToString());
}
return pageData;
}
```

3.2 Schedule Module Implementation of the Web Crawler

There are three considerations to implement web crawler applications as follows:

1. How to check the changes of the web pages.
2. How to increase the collection performance on networks.
3. How to keep the copyright of the web contents.

In these terms of network performance, we have to consider the collection cycle. Most of crawlers collect the web contents at regular intervals causing large network traffics. To reduce network traffic, we propose new data collection models by means of calculating optimal data collection cycles according to web sites.

For implementation configuration of the web crawler. We implemented the application with DOT Net, SQL SERVER 2008 and used MySQL DBMS for data management.

Fig. 2 shows the implementation of web crawler schema follows four steps

1. At first, it downloads the web page at start of the queue.
2. Next process is to analyzing the web page contents.
3. After analyzing a page contents from the web page then those contents will be send to add all the links to output
4. Then adding all the links to the further process. When the queue is not empty this cycle is repeated.
5. Repeat process from step 1 to 4

To maintain schedule information for all web pages. *StopUrl* table manages web site information, page table stores web page collection time, average collection time, and *page* update status. Particularly, *sche* table stores data of dynamic web collection cycle, which is used to compare collection cycles of each extraction time.

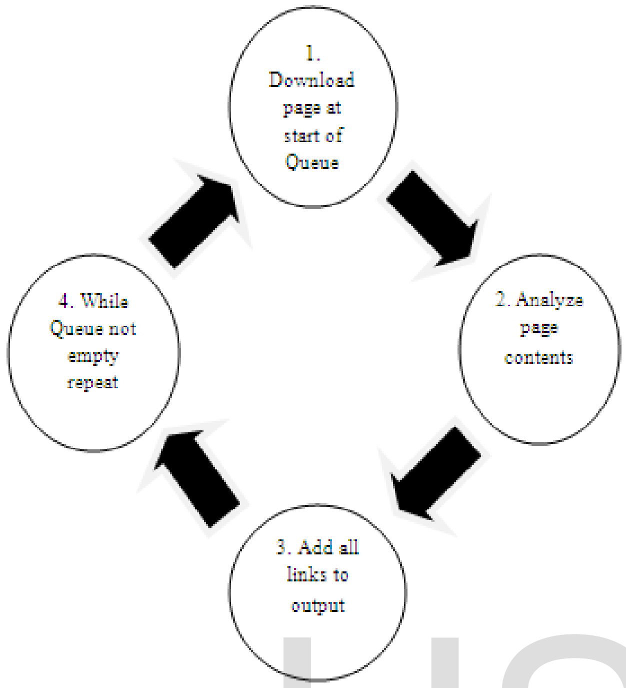

1. Download page at start of Queue
2. Analyze page contents
3. Add all links to output
4. While Queue not empty repeat

Fig.2. Class Diagram for Web Crawler Implementation

Table II provides more detail description for classes in the Web crawler.

TABLE II
DESCRIPTION OF CLASS MODULES

Classes	Function
1. System.Windows.Forms	User Form
2 System.ComponentModel	Scan and retrieve
2.2. Interop.ImageEditor	Retrieve image
3. BRIC.java	BRIC web site content extraction.
4. Interop.ImageEditor	Improve image
5. System.Drawing.Imaging	
6. System.Drawing.Drawing2D	

Fig.2 Database Schema for Web Data Collection Cycles
Fig.3 Class Diagram for Web Crawler Implementation

3.3 Descriptions

In this project web crawling has been done in five main Methods:

Frist Static Web Application mainly focuses on towards retrieving the information's dynamically from the web pages at regular intervals. The large size and the dynamic nature of the Web make it necessary to continually maintain Web based information retrieval systems. But also, it focus the a real time website available in the internet

Second In this Web Crawl Application, we are trying to crawl the web application created in our previous stage. Web crawlers are small programs that exploit the graph structure of the Web to move from one page to another page. In their infancy or normal stage, these programs were also called wanderers, robots, spiders and worms. These robots will try to fetch words that are quite evocative of Web imagery. In this module, the user will be given an option of searching for a particular URL. In that case, the contents in the web page will be fetched automatically with some of the base information from the URL.

Thrid is Web information extration, the individual information of the web page will be extracted for further usage. All the necessary information about the web page will be stored into the database.

- Number of Images
- Size of the web page
- Content information in the particular web page
- Hyperlinks information
- Sub Page information

Fourth is a Sub Hperlinks Crawling method a web crawler application is a system used for the bulk downloading of web pages. Prominently, this is one of the main components of web search engines. The main advantage of the system is to assemble a corpus of web pages, group it and index them, and finally allow the users to issue queries against the index and find the web pages that match their queries. Web Crawler application facilitate the process of following hyperlinks in Web pages and enable the system to automatically download new and updated Web pages.

While some systems rely on crawlers that exhaustively crawl the Web, others incorporate focus within their crawlers to harvest application or topic specific collections. Our application focuses towards creating a sub web engine to crawl individual pages and collect information from those pages.

Fivth is Automated Bot Method Based on the information collected; our system is trying to compare the data with the existing system. If there is a mismatch in the web page size or any other contents. An automatic invocation of our application will happen in turn, the system will try to fetch the web page data instantly. Updated information will be stored in the database and it will be used future data references.

4 Performance Evalution

In this section, Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.Functional testing is centered on the following items: Valid Input is to identify classes of valid input must be accepted. Invalid Input is to identify classes of invalid input must be rejected.

5 Conclusion

We proposed a dynamic web-data crawling method, which includes sensitive checking of web site changes, and dynamic retrieving of web pages from target websites. The method determines web crawling cycle time dynamically based on the 3 parameters: current collection time, collection time for previous update, average collection time. We compared the number of checks and average web crawling cycle for the web sites with network overhead.

The research proved that the dynamic web crawling cycle method can keep up to date data and determine the best data collection cycle corresponding for each different website. Also, controlling data collection cycle during midnights and weekends make the network bottlenecks significantly reduced.

References

[1] K.S. Kim, K. Y. Kim, K. H. Lee, T.K. Kim and W.S. Cho, Design andImplementation of web crawler based on dynamic web collection cycle. In *International Conference on Information Network* (2012).

[2] A. K. Sharma, et al., PARCAHYD: An Architecture of a Parallel Craw ler based on Augmented Hypertext Documents, *International Journal of Advancements in Technology*, Vol 1, No 2 (Oct. 2010).

[3] Junghoo Cho, Parallel Crawlers, In *Proc. WWW2002*, Hawaii, USA (May 2002).

[4] Robert C. Miller and Krishna Bharat, SPHINX: a framework for creating personal, site-specific Web-crawlers, http://www7.scu.edu.au /programme/fullpapers/1875/com1875.htm

[5] Vladislav Shkapenyuk and Torsten Suel, Design and Implementa tion of a High performance Distributed Web Crawler, TR, Department of Computer and Information Science, Polytechnic University, Brooklyn (July 2001).

[6] Mike Burner, Crawling towards Eternity: Building an archive of the World Wide Web, *Web Techniques Magazine*, 2(5) (May 1997).

[7] http://research.compaq.com/SRC/mercator/papers/www/paper.html

[8] Sergey Brin and Lawrence Page, The anatomy of large scale hyper textual web search engine, in Proc. International World Wide Web Conference, volume 30, pp 107-117 (April 1998).

[9] BRIC (Biology Research Information Center), http://bric.postech.ac.kr/

[10] NATE, http://www.nate.com/