Dr. Gulshan Ahuja / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 2, March -April 2013, pp.1107-1110

An Efficient Mechanism for Navigating Web Using Mobile Web Crawler

Dr. Gulshan Ahuja*
*(Department of Computer Science, TIMT, Yamunanagar, India.)

ABSTRACT

With the fast pace growth of World Wide Web and its dynamic nature coupled with presence of large volume of contents, the web crawlers have become an indispensable part of search engines. The growing use of search engines and their dependency in every day life necessitates that the correct and relevant information is presented to users in response to their search queries. Web crawler plays an important role to meet this objective by playing an important part of a search engine. A mobile web crawler is an automated computer program, which transfers itself to web servers in an attempt to download information and contents. Dynamically changing nature of web requires that mobile web crawlers must be able to intelligently decide about new pages and the changes in already crawled pages. This ability of mobile web crawler allows minimizing the consumption of resources. This paper aims to provide an efficient crawling mechanism for implementing a mobile web crawler, which intelligently decides about page changes to reduce overall load on web resources and helps search engines to increase web crawling speed and expand their reach in indexing.

Keywords -World Wide Web, Mobile Web Crawler, Search Engine.

I. INTRODUCTION

The most common and widely used method of finding information on web is by using search engines. The search engines create and maintain a database of indices for processing search related queries. However, to present the search results to users, firstly the pages must be downloaded from various web servers. This task is handled by web crawlers, which continuously traverse the web sites to find and collect information. Search engines for creating indices use the information collected by web crawlers.

Traditional web crawlers crawl web using follow link-download approach, to find newly published pages and changed pages. This approach of following links and downloading of pages has been used by many web crawlers. Researchers have addressed and highlighted various issues related to traditional web crawlers such as scalability, efficiency, high bandwidth consumption and index quality etc. The rapidly growing nature of web from few thousand pages to billions of pages has lead to development of different types of web crawling mechanisms. An alternative approach to traditional web crawling is based on mobile crawlers. A mobile web crawler has the ability to transfer and reside in the memory of a remote server.

Working of many mobile web crawlers on a remote machine puts a high bottleneck on its resources. The high degree of access to a server may further overload a server and sometimes may result in a server crash. Therefore, the mobile web crawlers must be designed in a way that they address not only the scalability issues but also crawl the web in an efficient way. This ensures that a mobile web crawler carefully chooses the pages to visit and download in repeated accesses. An effective crawling strategy requires that the mobile web crawler must crawl the web in a scalable and efficient way, to save bandwidth and maintain the quality of web searches. The most preferred method for efficient crawling is to formulate an ability to efficiently find out the changed pages and revisit pages on selection basis. The crawling strategies must be able to decide about page changes without completely analyzing the page for possible changes. This paper proposes an efficient mechanism, which intelligently decides about the changes in a page using a set of parameters and allows skipping unchanged pages. The rest of this paper is structured as follows. Section II presents some of the research work carried in web crawling. Section III highlights some of the important issues related with working of traditional crawlers. Section IV presents mobile web crawling mechanism. Section V presents the problem statement, which specifies the requirement to develop an efficient mobile web crawling mechanism. Section VI explains the proposed approach for mobile web crawling. Finally, section VII concludes and briefly describes scope for the future work.

II. RELATED WORK

This section presents some of the research work carried in this direction.

Eichmann [1] presented the first web crawler as RBSE spider. This crawler worked with the help of two program snippets named as spider and mite. Cho et al. [2] highlighted an efficient approach based on URL ordering. This paper

addressed the problem of ordering URLs for crawling and defined a number of metrics to evaluate crawlers. The paper used the concept of page ranking and showed it as an important metric. The research was limited to crawling a set of standard pages. Cho et al. [3] presented details about implementation of parallel crawlers, which run in parallel as per the needs and requirements. Chakrabarti et al. [4, 5] proposed focused crawling mechanism to assign priority to URLs for a topic in a specific domain. Shkapenyuk et al. [6], Boldi et al. [7] presented their work for distributed web crawlers, which worked on different machines. Their work highlighted various performance bottlenecks and described how load on the network could be reduced.

Fiedler et al. [8, 9] presented the concept of mobile web crawler, which aimed to reduce the network traffic by minimizing the transfer of data over the network. These mobile crawlers could move from one machine to another machine along with the crawling results. The ability to carry analysis on remote machine resulted in reduction of the number of pages, which were to be transmitted to the search engine. Implementation of security and development of intelligent algorithms remained as issues.

III. Issues with Traditional Crawlers

Web crawlers consume significant resources in their efforts to continuously crawl web servers and retrieve newly added or recently modified pages. This section presents some of the main issues related with traditional crawling mechanisms.

Load Consumption

A traditional crawler attempts to download all pages recursively from a target server. In order to compete with the pace of growth of web, the traditional search engines enhance their crawling activities. This increase in the crawling activities leads to increase in load of network resources and on the remote server. The continuous growth of web further increases the load on overall network. Moreover, traditional search engines crawl the web using HTTP request/response mechanisms. Therefore, for each web page, a separate request message is sent to remote server, which further adds to the load caused by the crawlers.

Local Analysis

The traditional crawlers download all pages to the local site before a search engine issues queries for fetching relevant information and carrying further analysis. All pages transferred to the local site may not be useful and are discarded in the process of indexing. Considering the fact that a large numbers of pages containing irrelevant information are discarded, this results in unnecessary burden on the performance of the search engine and significant wastage in network bandwidth.

Indexation

The search engines need to continuously refresh their indices to keep updated information about changed pages. As per [10] the average rate of change for a page is 75 days, which results in very large amount of data change in a short interval of time. The traditional crawlers can analyze pages only after downloading. This puts heavy burden on search engines to revisit already visited pages and download all those pages before analyzing and starting the process of indexation.

In the next section, we discuss architecture of a mobile web crawler and discuss its working.

IV. Mobile Crawler Mechanism

Mobile crawlers implement smart crawling techniques to optimize their crawling process. Fig. 1.0 presents architecture of a mobile crawler. The main component of search engine, involved in the crawling process, is referred as crawling manager. The crawling manager is responsible for providing the details of web sites, which are targeted by web crawlers and monitors the crawled locations. The mobile crawler starts its operations by receiving a list of target web sites from crawling manager.

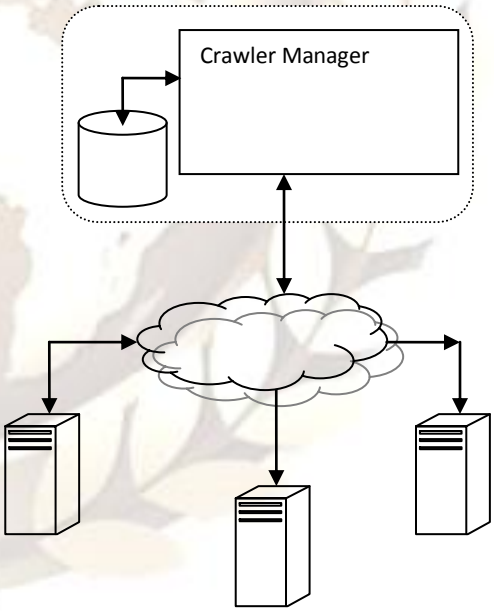

Figure 1.0 Mobile Web Crawler Mechanism

The initial list of URLs is referred as seed URLs. Each mobile crawler is created to include the crawling strategy, which helps it to locate the relevant information. Before the crawling starts, the mobile crawler transfers itself to one of the remote site as specified through seed URLs. The crawling process begins after migrating to the remote site. The mobile crawler implements an intelligent mechanism to minimize the time required to crawl

all pages on a web site in an efficient manner and to reduce the bandwidth consumed during downloading of pages. After crawling entire web site and collecting information about the relevant pages to be downloaded, the mobile crawler returns to its origin and hands over the details to the crawling manager.

The crawling manager retrieved the specified pages recursively once the pages have been downloaded. Thereafter, the search engine starts the process of creating index in its database. The mobile crawler, with to its ability to selectively download pages, considerably reduces the load on the network resources.

V. PROBLEM STATEMENT

As per an estimation [10], over 600 GB of web changes occur every month. A mobile web crawler must revisit the pages, which had been crawled and downloaded earlier, to keep the indexes updated at search engine. As the web pages change in an undefined pattern, the mobile web crawler must be able to decide intelligently about which pages are to revisit and which pages are to skip.

The ability of a mobile web crawler to decide carefully about the pages to be transferred through the network, results in significant reduction of load on the resources of the web server and underlying network. To address the problem as stated above, the next section presents an efficient mechanism for mobile web crawler to revisit the already crawled pages and select only changed pages for downloading to the search engine.

VI. PROPOSED APPROACH

The crawler manager employs a page change matrix (PCM) of records. The PCM contains a record for each URL, which may be crawled by the mobile web crawler. TABLE 1.0 shows the PCM containing a number of parameters.

Table 1.0 Page Change Matrix

URL	DM_P	DM_R	NP_C	PCF
http://www.ez.com	13012013	17012013	4	.13
http://www.cz.com	12022013	22022013	6	.33

The columns in the PCM represent parameters such as DM_P, DM_R, NP_C, and PCF.

DM_P and DM_R fields specify the previous date of modification and the recent modification date of a page. The crawler manager updates these fields, whenever a page change is detected and the page is downloaded from the web server.

NP_C represents the number of times the page belonging to a particular URL has changed.

PCF specifies the page change factor, which is calculated as follows

PCF = t_K / T_M

Where t_K is calculated as the time interval between previous change and recent change in a page. The T_M specifies the total time duration over which a page is monitored for page change.

When a mobile web crawler visits a web server to find out page changes, which have occurred since last crawl, it follows the algorithm as shown in Fig. 1.1. The algorithm starts with the list of URLs to be crawled. The final set of URLs is obtained as final URL list (FUL).

Algorithm: InstCrawl(Input:

URL List // List of URLs to crawl

DM_R //Recent modification date

PCF // Page Change Factor

Output: FUL // Final URL List to download pages)

Step1: While URL list not empty repeat steps 2 to 8

Step 2: Locate last modification date for page from log file

Step 3: If last modification date present do step 4 else go to step 5

Step 4: if DM_R not same as last modification date

Add URL to FUL and go to step 1

else go to step 1

Step 5: Compute PCEV as per (1)

Step 6: if PCEV less than PCF

Skip analyzing page

else

Analyze page for determining page change

Step 7: Obtain result after analysis and add URL to FUL

Figure 1.1 Algorithm for Crawling of Pages

This list corresponds to the links, from where the pages are to be downloaded on the site of search engine for indexing purpose. The page change estimation value (PCEV) is used to determine whether a complete analysis of page is required or not.

The value for PCEV is calculated as follows:-

$$PCEV = e^{-y} r^{y} / x! \quad \ldots\ldots \quad (1)$$

Where x refers to the number of successes and y for a given web page is estimated as

$$y = \sum_{k \in K} t_k / |K|$$

K is the set of all changes occurred in t_k length of time passed between k^{th} and preceding change.

The frequently changing nature of web implies that the mobile web crawler must download pages on requirement basis only. A mobile web crawler decides about the page change by completely analyzing a page. The complete analysis of a page requires computation of a number of factors such as the number of URLs present in a page, number of keywords in a page or the names of URLs etc. and requires considerable time and resources on part of a web crawler.

VII. CONCLUSION

The author of this paper has presented various issues related with traditional crawling mechanisms. An alternative approach to crawl web using mobile web crawler has been proposed. The proposed approach uses the page change matrix, which is used to detect about page changes. The proposed approach allows for intelligent crawling based on page change factor. The computation of page change estimation value, rules out the need to completely analyze a page, for finding a page change. The reduction in the number of times the analysis required for page results in efficient crawling in terms of speed of crawling and conserves useful resources of the remote server and underlying network.

This paper presents mobile crawling approach, which further needs to be evaluated in a real time environment. Future work may be directed towards covering more aspects of crawling such as analysis of page, allocation and consumption or resources. It will be motivating to investigate implication of mobile web crawling on performance of web servers. The future work in mobile web crawling may concentrate on effect of mobile web crawling in terms of a number of factors such as bandwidth consumption, resource utilization, URL ordering etc.

REFERENCES

[1] Eichmann D., "The RBSE spider: balancing effective search against Web load", *Proc. of First World Wide Web Conference*, Switzerland, 1994.

[2] Junghoo Cho., Garcia-Molinga, Lawrence Page, Efficient Crawling through URL Ordering, *Computer networks and ISDN systems, pp. 161-172, 1998.*

[3] Cho. J, Garcia-Molina, H., Parallel Crawlers, *Proc. of 11th International Conference on World Wide Web*, USA, pp. 124-135, 2002.

[4] Soumen Chakrabarti, Martin van den Berg, Byron Dom, Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery, *World Wide Web Conference*, 1999.

[5] S. Chakrabarti, K. Punera, M. Subramanyam, Accelrated Focused Crawling through Online Relevance Feedback, *Proc. of 11th International Conference on World Wide Web*, pp. 148-159, 2002.

[6] Shkapenyuk V., Suel T., Design and Implementation of a High Performance Distributed Web Crawler, *Proc. of 18th International Conference on Data Engineering*, California, IEEE Press, pp. 357-368, 2002.

[7] Boldi P., Codenotti B., Santini M., Vigna S., UbiCrawler : a scalable fully distributed Web crawler, *Software Practice and Experience, vol. 34 issue 8, pp. 711-726, 2004.*

[8] Fiedler J., Hammer J., Using the Web efficiently: Mobile Crawling, *Proc. of 7th International Conference of the Association of Management on Computer Science*, San Diego, pp. 324-329, August 1999.

[9] Fiedler J., Hammer J., Using mobile crawlers to search the Web efficiently, *International Journal of Computer and Information Science, vol. 1, no. 1, pp. 36-58, 2000.*

[10] Brewster Kahle, Archiving the internet, *Scientific American, 1996.*