http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 9 (2): 376-380, 2010
ISSN 1812-5638

Filtering the Web Pages that are not Modified at Remote Site Without Downloading using Mobile Crawlers

[1]S. Bal and [2]R. Nath
[1]Department of Computer Science and Application, Vaish College of Engineering, Rohtak, Haryana, India
[2]Department of Computer Science and Applications, Kurukshetra University, Kurukshetra, Haryana, 136119, India

Abstract: The aim of this study is to develop a crawling technique that reduces load on the network caused by the search engine crawlers. The search engines are used by the users to search the required information on the web. These search engines maintain the index of billions of pages for performing the search efficiently. To maintain the index of these search engines up-to-date, the crawlers of these search engines recursively retrieve the pages that cause 40% of current internet traffic and bandwidth consumption. These crawlers also cause load on the remote server by using its CPU cycles and Memory. The authors address this problem by proposing a novel mobile crawling technique that uses mobile agents to crawl the pages. These mobile crawlers identify the modified pages at the remote site without downloading the pages. Therefore, only those pages are downloaded that are actually modified after the last crawl. The results have shown that the proposed approach is very promising. This approach reduces the internet traffic and load on the remote site i.e., saves CPU cycles of the remote server.

Key words: Search engine, mobile crawler, web sites, agents, web servers

INTRODUCTION

The World Wide Web (WWW) has grown from a few thousand pages in 1993 to more than several billion pages at present. Search engines maintain comprehensive indices of documents available on the web to provide powerful search facilities. Web crawlers are used to recursively traverse and download web pages (Giving GET and POST commands) for search engines to create and maintain the web indices. The needs of maintaining the up-to-date pages in the collection cause a crawler to revisit the websites again and again. Due to this, the resources like CPU Cycles, Disk Space and Network bandwidth become overloaded and sometime a web site may crash due to such overloads on these resources (Cho and Garcia-Molina, 2003; Nath *et al*., 2007).

There is a little coverage in the literature on commercial crawlers. Google project developed by (Brin and Page, 1997) at Stanford University is one source of information. Another source of information about the crawlers is the Harvest project (Bowman *et al*., 1994), which investigates the web crawling in detail. A web crawler consumes a significant amount of network bandwidth and other resources by accessing the web resources at a fast speed. This affects the performance of the web server considerably. Koster *et al*. (1993) has published a set of guidelines for the crawler developers to handle this problem. A significant amount of resources of underlining network are consumed to build a comprehensive full text index of the web. Thus, the crawling activities of a single search engine cause a daily load of 60 GB to the web (Cho *et al*., 1997). About 40% of current internet traffic and bandwidth consumption is due to the web crawlers that retrieve pages for indexing by the different search engines (Yuan and Harms, 2002). The maximum web coverage of any popular search engine is not more than 16% of the current web size (Lawrence and Giles, 1999).

Shkapenyuk and Suel (2002) proposed distributed crawling and Cho and Garcia-Molina (2002) proposed parallel crawling to increase the coverage and to decrease the bandwidth usage but this does not solve the problem.

Fiedler and Hammer (1999) and Hammer and Fiedler (2000) purposed web crawling approach based on mobile crawlers powered by mobile agents. Papapetrou and Samaras (2004) first designed and developed UCYMicra and then IPMicra , a mobile crawler based system. Their migrating crawlers move to the web servers and perform

Corresponding Author: Satinder Bal, Department of Computer Science and Application, Vaish College of Engineering, Rohtak, Haryana, India

the downloading of web documents, processing and extraction of keywords and after compressing, transmit the results back to the central search engine. Further, these migrating crawlers remained in the remote systems and perform constant monitoring of all the web documents assigned to them for changes.

Another mobile crawling approach (Nath and Bal, 2007; Bal and Nath, 2009) was proposed that uses page change probability and other statistics to filter the pages that are not modified after the last crawl. These mobile crawlers have reduced the load on the resources of the remote web server that is considerable but have some limitations.

In this study, the authors present an alternate approach of filtering those web pages that are not modified using mobile crawlers. Our approach retrieves only those web pages from the remote server that are actually modified and perform the filtering of non-modified pages without downloading the pages.

THE ARCHITECTURE OF THE PROPOSED APPROACH

The mobile crawlers are constructed as mobile agents (Fiedler and Hammer, 1999; Papapetrou and Samaras, 2004; Nath and Bal, 2007; Lange and Oshima, 1998) and are dispatched to remote web servers where they extract the size of the web pages for comparison and filtration of non-modified pages and finally, the pages actually modified are compressed and transmitted back to the Search Engine. The major components of our Proposed Approach are (a) Crawler Manager (CM). (b) Statistics Database Module (SDM). (c) Old Database File Module (ODBFM). (d) Comparator Module (CoM). Each Component is shown in Fig. 1 and is briefly discussed below:

Crawler Manager (CM): The CM performs various tasks. The major tasks of CM are mobile crawler generation; delegation of URL's to the mobile crawler for crawling, construction of ODBFM for each mobile crawler, updating the SDM, receiving the crawled pages from the Remote server. The task of decompressing the pages received from the Remote server, in compressed form, is also performed by the CM. After generation and URL delegation to the mobile crawlers, these mobile crawlers are sent to the designated Remote Site to crawl the URL's given in the ODBFM.

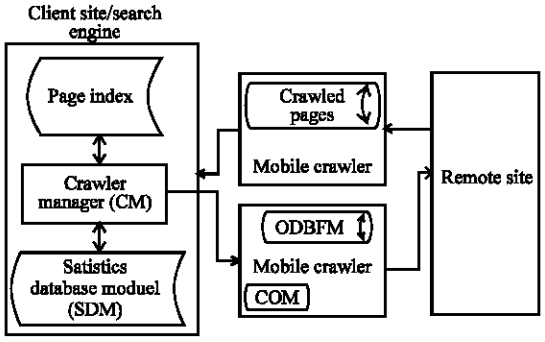

Fig. 1: Architecture of proposed mobile crawling approach

Statistics Database Module (SDM): This module is maintained at the Client Site (Search engine) and contains information about all the pages crawled by the mobile crawlers. This module has two fields: (a) Name of URL-This field stores the names of the pages that are indexed by the Search Engine. (b) Size of the Web page-This stores the size of the web page in bytes.

Old Database File Module (ODBFM): The CM constructs this module one for each mobile crawler. It contains statistics about each URL to be crawled. This statistics is taken from the SDM. This module is sent with the corresponding mobile crawler on to the Remote Site. This module has two fields: (a) Name of URL (b) Size of the Web page (in bytes). This file is deleted after each crawl and made a fresh every time the crawling starts taking updated statistics from SDM.

Comparator Module (CoM): The major function of this module is to compare the size of each URL (HTML page) taken from ODBFM with the size of the corresponding URL retrieved from the remote site by the mobile crawler to filter out the pages that are not modified. A crawled page is modified if its size is increased or decreased in bytes. The page that is modified is sent to the client site (Search engine) for indexing.

Working of proposed approach

The proposed approach works as follows: First time, the Home pages of various sites are downloaded by the mobile crawlers onto the client Site (search engine). These are indexed and their sizes (in bytes) are stored in SDM at the Client Site. From next time onwards, the Crawler Manager (CM) construct the mobile crawlers one for each remote site. This mobile crawler has the ODBFM containing the size of each page allocated to the mobile crawler for crawling. The mobile crawler with the ODBFM moves to the Remote site to crawl the pages allocated to it. At the remote site, the mobile crawler searches the pages using recursive search and then retrieves the size (in bytes) of each page one by one whose URL's are given in the ODBFM without accessing the pages and

passes this information to the CoM. The CoM, in turn, compares the size of each web page with the size of the corresponding page in the ODBFM. If it is matched than the page is not a candidate to be sent to the Client site (search engine) for indexing because the page was not modified since the last crawl (its size is same as the page already indexed by the client site), but if mismatch (size is different from the size of the page currently indexed by the client site) occurs then the page is downloaded from the remote site. After downloading all the modified pages the mobile crawler returns to the search engine after compressing the pages. These modified pages are then updated at the search engine site. The complete working of the proposed system is shown in Fig. 2.

Experimental setup: A virtual environment has been setup to perform the experiments. Two machines named the remote site and the client site (search engine) have Intel processor clocked at 3.06 GHz and has 1 GB of RAM. Both the machines support Java environment and Tahiti server is installed on them to support the Java Aglets. Both the machines have Window XP operating system. These two machines are connected through high-speed LAN. Some sites are selected and home pages of these sites are downloaded and stored on the RS. The mobile Crawlers are built using Java Aglets on Search Engine and URL's are delegated to them for crawling.

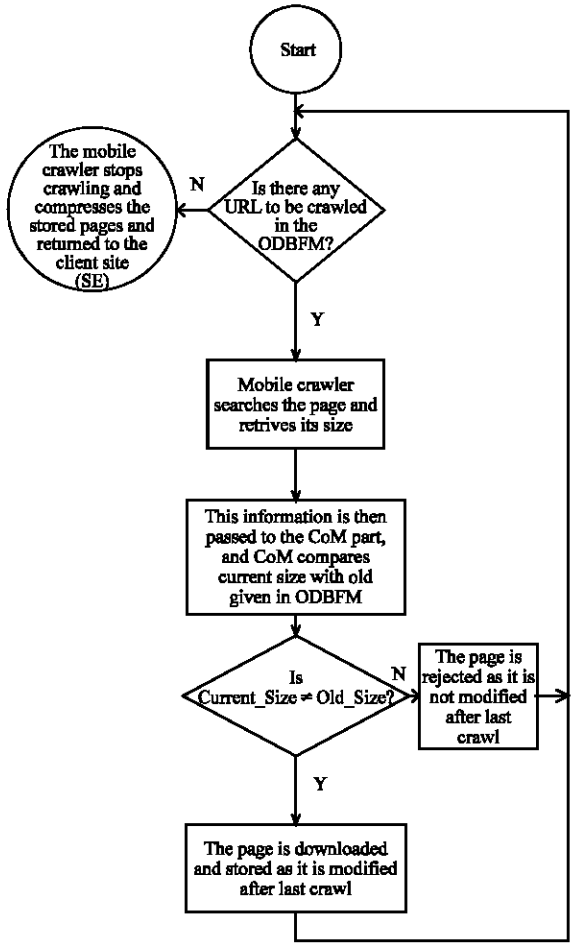

Fig. 2: The working of mobile crawler at remote site

RESULTS

One hundred web sites (a combination of.com,.edu,.gov and.org) are selected and home pages of these web sites are downloaded daily for 30 days and stored on one machine called remote site. The results shows that from the sample pages, on the average, approximately 52% pages change daily that are downloaded by the mobile crawlers as shown in Fig. 3. This means that on the average 48% pages are not retrieved by the mobile crawler and are not sent to the client site .This filtration of non-modified pages by the mobile crawler reduces the internet traffic as less number of pages are sent to the client site as compared to traditional crawler and yet maintaining the up-to-date index.

From results of Fig. 3, the reduction in network traffic is calculated as follows:

Total No. of pages on the Remote Site (TP) = 100

No. of pages that can be down loaded by Traditional Crawler (PTC) = 100

Average No. of pages that are downloaded by the proposed approach (PMC) = 52

Average size of each HTML page = 7 kb

Total average size of all the pages =100×7 = 700 kb

Average network traffic due to mobile crawler (Without Compression) = 52×7 = 364

Reduction in network traffic due to proposed approach (Without compression) = (PTC-PMC)×Average size of each page = (100-52)×7 = 336 kb

Actual network traffic when proposed approach is used (Without compression) = Average network traffic due to mobile crawler + Overhead of using mobile crawlers = 364+30 = 394 KB

Actual reduction in the network traffic due to proposed approach (Without compression) = Total average size of all the pages-Actual network traffic when proposed approach is used = 700-394=306 kb

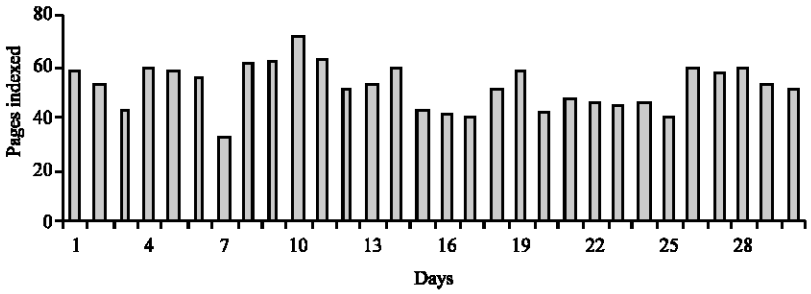

Fig. 3: Pages indexed by the mobile crawler

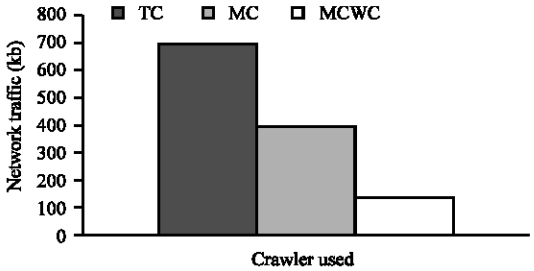

Fig. 4: Comparison various crawling techniques

It is possible to reduce the network traffic further by compressing the pages before sending them to the search engine. It is possible to compress the HTML pages up to 30% of the actual size by using standard compressing software's and more than that if compressing is done using other techniques.

So, average size of the all the pages that are sent to client site by mobile crawler (when compressed) = $\frac{364 \times 30}{100} = 109.2$ kb

The Actual network traffic due to proposed approach (When Compressed) = Average size of the all the pages that are sent to client site by mobile crawler (when compressed) + Overhead of using mobile crawlers =109.2+30 =139 (Approx.) kb

Thus, actual reduction in network traffic due to proposed approach (When compressed) = Total average size of all the pages - The actual network traffic due to proposed approach (When compressed) = 700-139 = 561 (Approx.)

This shows that the proposed approach works efficiently as compared to traditional crawling and helps in reducing the network traffic and hence preserves the bandwidth as shown in Fig. 4.

The same numbers of pages are also not downloaded from the remote site as their size is compared before downloading. This saves CPU cycles of the remote site considerably. The results may vary slightly depending on the pages selected.

CONCLUSION

The results of the proposed approach are very promising. The proposed mobile crawling approach reduces the network load caused by the crawlers considerably by reducing the amount of data transferred over the network as compared to traditional crawler. It also reduces the load (i.e., CPU cycles) on the remote server drastically by not retrieving the pages that are not modified after the last crawl. This reduction in network traffic and CPU cycles used to retrieve the pages is achieved by performing the analysis of the size of the pages at the remote site. Only those pages that are actually modified are accessed and sent to the Client Site. Thus, the mobile crawlers transmit only those pages that are actually modified after the last crawl. The proposed approach performs better than the traditional crawler approach and the previous mobile crawler approach as it preserves bandwidth and also reduces load on the Remote site considerably.

REFERENCES

Bal, S. and R. Nath, 2009. Filtering the non-modified pages at remote site during crawling using mobile crawlers. Proceedings of IEEE international advance computing conference. March 08-09, TIET, Patiala (Punjab).

Bowman, C.M., P. Danzig, D. Hardy, U. Manber and M. Schwartz, 1994. Harvest: A scalable, customizable access system, technical report CU-CS-732-94. Department of computer science, university of Colorado, Boulder.

Brin, S. and L. Page, 1997. The anatomy of a large scale hypertextual web search engine, technical report. Stanford, CA.

Cho, J., H. Garcia-Molina and L. Page, 1997. Efficient Crawling Through URL Ordering. Stanford University Press, Stanford, CA, USA.

Cho, J. and H. Garcia-Molina, 2002. Parallel crawlers. Proceedings of the 11th International Conference on World Wide Web, May 07-11, ACM Press, USA., pp: 124-135.

Cho, J. and H. Garcia-Molina, 2003. Estimating frequency of change. ACM Trans. Internet Technol., 3: 256-290.

Fiedler, J. and J. Hammer, 1999. Using the web efficiently: Mobile crawling. Proceedings of the 7th International Conference of the Association of Management (AoM/IAoM) on Computer Science, Aug. 1999, San Diego, CA., pp: 324-329.

Hammer, J. and J. Fiedler, 2000. Using mobile crawlers to search the web efficiently. Int. J. Comput. Inform. Sci., 1: 36-58.

Koster, M., J. Fletcher and L. McLoughlin, 1993. Guidelines for robot writers. A web document.

Lange, D.B. and M. Oshima, 1998. Programming and Deploying Java Mobile Agents with Aglets. Addison Wesley, New York.

Lawrence, S. and C.L. Giles, 1999. Accessibility of information on the web. Nature, 400: 107-109.

Nath, R. and S. Bal, 2007. Reduction in bandwidth usage for crawling using mobile crawlers. IJCSKE, 1: 51-61.

Nath, R., S. Bal and M. Singh, 2007. Load reducing techniques on the websites and other resources: A comparative study and future research directions. JARCE, 1: 39-49.

Papapetrou, O. and G. Samaras, 2004. Minimizing the network distance in distributed web crawling. Proceedings of CoopIS/DOA/ODBASE, OTM Confederated International Conferences, Oct. 25-29, Agia Napa, Cyprus, pp: 581-596.

Shkapenyuk, V. and T. Suel, 2002. Design and implementation of a high performance distributed web crawler. Proceedings of the 18th International Conference on Data Engineering, (ICDE' 02), San Jose, California, IEEE CS Press, pp: 357-368.

Yuan, X.M. and J. Harms, 2002. An efficient scheme to remove crawler traffic from the internet. Proceedings of the 11th International Conference on Computer Communications and Networks, Oct. 14-16, IEEE CS Press, pp: 90-95.