International Journal of Innovations in Engineering and Technology (IJIET)
http://dx.doi.org/10.21172/ijiet.83.040

Migrating Parallel Web Crawler: A Review

Deepak
Student of M. Tech, Department of Computer Science Engineering
Indus Institute of Engineering & Technology, Jind, Haryana, India

Deepika Mahal
Assistant Professor, Department of Computer Science Engineering
Indus Institute of Engineering & Technology, Jind, Haryana, India

Abstract- **The Internet is a network of networks that consists of millions of private, public, academic, business, and government networks connected by electronic and optical networking technologies. The main services that an Internet provides are infrastructure to support electronic mail and the inter-linked hypertext documents of the World Wide Web (WWW).The World Wide Web, commonly known as the Web, is a system of interlinked hypertext documents contained on the Internet. With a web browser, one can view web pages that may contain text, images, videos, and other multimedia and navigate between them by using hyperlinks. Web search engines employ web crawlers to continuously collect web pages from the web, index and store them in a database. In traditional crawling, the pages from all over the web are brought to the search engine side and then processed, that results in a lot of network traffic. The capabilities of mobile agents have been utilized to design migrating crawler instances called migrants, which move to the information resources for downloading the documents, resulting in reduced load on a single machine. Migrants filter and compress the documents at the remote host itself before transferring them to the search engine repository. This paper describes the basic architecture, and components of Parallel Migrating Crawler. Major issues and challenges in implementing effective Migrating Web crawlers are also identified.**

Keywords – Web Crawlers, Parallel Crawler, Database, Web Wide Web.

I. INTRODUCTION

The World Wide Web is a global, large repository of text documents, images, multimedia and many other items of information, referred to as information resources. Since its inception, World Wide Web has evolved from a simple and static content source into a rich dynamic information delivery channel where new pages are being added, removed, modified and rearranged. It is quite possible that after downloading a particular web page, the local copy of the page residing in the repository of the web pages becomes obsolete compared to the copy on the web. Also these modifications may also take place within an extremely short span of time. Therefore Web Page Change Detection techniques are required to detect these changes and update the database of web pages. The web page change detection system helps the user to detect all such changes effectively. Web page change detection can be utilized in various applications such as web page archiving, temporal querying, indexing, crawling and temporal visualization.

- A crawling module which fetches pages from Web servers known as web crawler.
- Indexing and analysis modules which extract information from the fetched pages and organize the information.
- A front-end user interface and a supporting querying engine which queries the database and presents the results of searches**.**

II. PROPOSED ALGORITHM

Web crawlers are a part of the search engines that fetch pages from the Web and extract information. A simple crawler algorithm is as follows:

Web_Crawler ( )
1. Do Forever

2. Begin

3. Read a URL from the set of seed URL's

4. Determine the IP-address for the Host name

5. Download the Robot.txt file, which carries

download information and also includes the files to be

excluded by the crawler

6. Determine the protocol of underlying Host like

HTTP, FTP, GOPHER

7. Based on this protocol, download the document

8. Check whether the document has already been

downloaded or not

9. If the document is a fresh one,

10. Then

11. store it and extract the links and references to

other sides from that document

12. Else

13. Abandon the document

14. *End*

The Web crawler is given a start-URL and the Crawler follows all links found in that HTML page. This usually leads to more links, which will be followed again, and so on.

III. RELATED WORKS

The literature survey shows that a number of modifications in the basic crawler have been done to improve the crawling speed.

*PARALLEL CRAWLERS[1]***,** A crawler can either be centrally managed or totally distributed. The authors mention that distributed crawlers are advantageous than multithreaded crawlers or standalone crawlers on the counts of scalability, efficiency and throughput. If network dispersion and network load reduction are done, parallel crawlers can yield good results. Their system utilizes memory of the machines and there is no disk access.

PARCAHYD[2] In this, work it has been proposed that if the links contained within a document become available to the crawler before an instance of crawler starts downloading the documents itself, then downloading of its linked documents can be carried out in parallel by other instances of the crawler. Therefore, it was proposed that meta-information in the form Table of Links (TOL) consisting of the links contained in a document be provided and stored external to the document in the form of a file with the same name as document but with different extension. This one time extraction of TOL can be done at the time of creation of the document

As the size of the Web grows, it becomes imperative to parallelize a crawling process. In [3] the author proposes Mercator, which is a scalable, extensible crawler.

The traditional centralized crawling model however suffers from the following limitations:

- The task of processing the crawled data introduces a vast processing bottleneck at the search engine.
- The attempt to download thousands of documents per second creates a network and a DNS lookup bottleneck.

- Documents are usually downloaded by the crawlers in uncompressed form which increases the network bottleneck.
- Traditional centralized crawling cannot effectively catch up with the dynamic web

Due to these problems there is a need to improve the crawling process in such a way that can accommodate the rapid growth of the Web , so MIGRATING CRAWLER was proposed as**,** an alternative approach to Web crawling which is based on *mobile agents*. In these types of crawlers, the mobile agents are transferred to the Web Servers (where data resides) itself, pages are collected and processed at server side only. Processed pages are then returned to the Search Engine itself. This reduces network load and speeds up the indexing phase inside the search engine.

IV. MIGRATING CRAWLERS

The Migrating crawling approach departs from the centralized architecture of traditional search engines by making the data retrieval component, the Web crawler, distributed Migration is being defined in the context of Web crawling as the ability of a crawler to migrate to the data source (e.g., a Web server) before the actual crawling process starts on that Web server. Thus, migrating crawlers are able to move to the resource which needs to be accessed in order to take advantage of local data access. After accessing a resource, migrating crawlers returns their home system, carrying the crawling result in the memory. Migrating crawlers are managed by a crawler manager, which supplies each crawler with a list of target Web sites and monitors the location of each crawler.

Various important component of a migrating parallel web crawler system are central crawler, crawl frontier, local database of each crawl frontier and centralized database. URL input from application is received from central crawler and these URL are forwarded to available migrating crawling process. To increase system performance these crawling process migrate to different machines. Each crawl frontier have local database to locally collect the data. This data is then passed to central database at central crawler.

- *CENTRAL CRAWLER:* Central crawler is the central component of system and started first in system where all other components are started and register with central crawl manager to offer or request services. All URLs to be crawled are present at central crawler. It is only component visible to other components while all crawl frontiers work in independent manner. It is responsibility of central crawler to query DNS resolver for IP addresses of host or servers, after getting the url's of files. Also checks robots.text file in root directory of web servers. Crawl frontiers have to register themselves with central crawler and are logically migrated to different locations. Different url's are assigned to all the crawl frontiers and then after receiving URL they started crawling. In this way the compressed downloaded content from crawl frontiers are passed to central crawler.

- *CRAWLER FRONTIER:* The crawl frontier components begin to crawl received URLS in breadth first manner. Each crawl frontier has their local database where they stored the crawled files related to URLS. The downloaded pages are parsed for hyperlinks and if different URLS are encountered they are added to queue which contains the links to be visited. During crawling there may be two types of links internal links and external links. Internal links are given more priority over external links. Then downloaded files are passed to central database in compressed form.

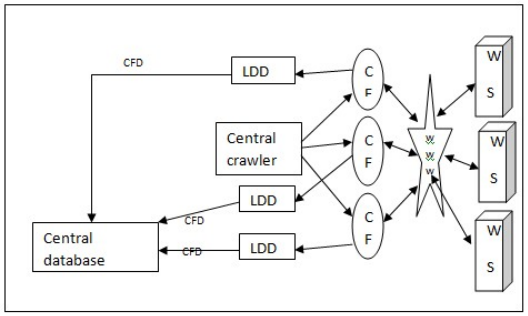

Figure 1:Migrating Parallel Crawler

- *LOCAL DOCUMENT DATABASE*: Some memory space is required by crawl frontier to save the downloaded content. Local database is used for this purpose by crawl frontier and called local document database. It is called the storage area of machine on which crawl frontier is running.

- *CENTRAL DATABASE:* All URLs to be crawled are stored in central database. Also the final compressed documents downloaded by different crawl frontier are stored in central database.

- *WEB SERVER:* A web server is a program that serves the files form web pages to web users. It uses World Wide Web's hypertext transfer protocol (HTTP) and client/server model.

Before the actual crawling begins, the crawler must migrate to a specific remote site using one of the seed URL's as the target address. After the crawler successfully migrated to the remote host, the crawling algorithm is executed. This part of mobile crawling is very similar to traditional crawling techniques since pages are retrieved and analyzed recursively. When the crawler finishes it either returns to the crawler manager (crawler home) or, in case the list of seed URLs is not empty, migrates to the next Web site on the list and continues. Once the mobile crawler has successfully migrated back to its home, all pages retrieved by the crawler are transferred to the search engine via the crawler manager. Once the pages have been downloaded, the search engine can generate the index as before.

V. ADVANTAGES

The advantages of the Migrating crawlers are

- *DATA ACCESS*: By migrating to a remote Web server, mobile crawlers can access Web pages locally with respect to the server. This saves network bandwidth by eliminating request/response messages used for data retrieval.

- *REMOTE PAGE COMPRESSION:* By migrating to a remote Web server, mobile crawlers can compress the content of Web pages before transmitting them over the network. This saves network bandwidth by reducing the size of the retrieved data.

- *REMOTE PAGE FILTERING:* By migrating to a remote Web server, mobile crawlers can reduce the content of Web pages before transmitting them over the network. This saves network bandwidth by discarding irrelevant portions of the retrieved pages.

- *REMOTE PAGE SELECTION*: By migrating to a remote Web server, mobile crawlers can select only the relevant pages before transmitting them over the network. This saves network bandwidth by discarding irrelevant information directly at the data source.

VI. RESEARCH ISSUES

Some of the identified issues are discussed below:

- *HOW SHOULD THE CRAWLER REFRESH PAGES?:* Once the crawler has downloaded a significant number of pages, it has to start revisiting the downloaded pages in order to detect changes and refresh the downloaded collection. Because Web pages are changing at very different rates, the crawler needs to carefully decide what page to revisit and what page to skip, because this decision may significantly impact the "freshness" of the downloaded collection.

- *NETWORK EFFICIENCY IS LOW:* If all the collected documents (relevant or non relevant)will be sent to central crawler then network traffic increases very rapidly. Hence network efficiency decreases.

- *WASTAGE OF NETWORK BANDWIDTH:* If all the documents are send to central crawler then it will use lots of network bandwidth.

- *MORE TIME:* Firstly the old and new fetched pages are converted to tree form and each node is provided with a value. Then both are compared level wise which takes a lot of comparison and lot of time for each page. Due to processing of such a huge amount of data execution time is very high which is not desirable characteristic of any algorithm .

- *MORE SPACE:* Generating trees for each old and new page will also take a lots of space. Because data is massive, to process such a huge amount of data main memory should be large. But main memory is very costlier so it puts a constrain on processing datasets.

VII. CONCLUSION

In this paper a review on an Migrating Crawlers is being presented. Migrating Crawling approach surpasses the centralized architecture of the current Web crawling systems by distributing the data retrieval process across the network. With migrating crawlers it is possible to perform remote operations such as data analysis and data compression at the data source before the data is transmitted over the network. This allows for more intelligent crawling techniques and addresses the needs of applications, which are only interested in certain subsets of the available data. Migrating crawlers can reduce the network load caused by crawlers significantly by reducing the amount of data transferred over the network.

REFERENCES

[1] Junghoo Cho ,Hector Garcia-Molina, “Parallel Crawlers”,In proceedings of WWW2002, Honolulu, hawaii, USA, May 7-11, 2002. ACM 1-58113-449-5/02/005.
[2] A.K.Sharma, J. P. Gupta, D. P. Agarwal,“PARCAHYD: An Architecture of a Parallel Crawler based on Augmented Hypertext Documents”International Journal of Advancements in Technology.
[3] Allen Heydon and Mark Najork, “Mercator: A Scalable, Extensible Web Crawler”, In Journal of WORLD WIDE WEB , Volume 2, Number 4, 219-229, DOI: 10.1023/A:1019213109274.
[4] Ashutosh Dixit,U.C. Pant, “A Novel and Efficient Mechanism for Securing Migrating Crawler Data”.
[5] A.K. Sharma, Ashutosh Dixit, Deepika, and Niraj Singhal, “Security issues in Distributed Crawling based on Mobile Agents”, 5th International IT Conference on Downtrend Challenges in IT, DCIT09, Ludhiana, Punjab, May 2009.
[6] D. Sullivan, “Search Engine Watch,” Mecklermedia, 1998.
[7] S. Brin and L. Page, “The Anatomy of a LargeScale Hypertextual Web Search Engine,” Stanford University, Stanford, CA, Technical Report, 1997.
[8] Abhinna Agarwal, Durgesh Singh, “Design of a Parallel Migrating Web Crawler ” International Journal of Advanced Research in Computer Science and Software Engineering in 2012.
[9] Niraj Singhal,Ashutosh Dixit, “Regulating Frequency of a Migrating Web Crawler based on Users Interest” in 2012.
[10] Md. Faizen Farooqui, “An Extended Model for Effective Migrating Parallel Web Crawling with Domain Specific and Incremental Crawling” International journal on Web Service Computing in 2012.
[11] Joachim Hammer , Jan Fiedler “Using Mobile Crawlers to Search the Web Efficiently” in 2000
[12] Akansha Singh,Krishan Kant Singh, “Faster and Efficient Web Crawling with Parallel Migrating Web Crawler” in 2010.
[13] Md. Abu Kausar “Web Crawler Based on Mobile Agent and Java Aglets” in I.J Information Technology and Computer Science, in 2013.
[14] Md. Faizen Farooqui, “Change Detection in Migrating Parallel Web Crawler: A Neural Network Based Approach” in 2014.