An Effective Parallel Web Crawler based on Mobile Agent and Incremental Crawling

Md. Abu Kausar and V. S. Dhaka
Dept. of Computer & System Sciences, Jaipur National University, Jaipur, India
Email: {kausar4u, vijaypal.dhaka}@gmail.com

Sanjeev Kumar Singh
Dept. of Mathematics, Galgotias University, Gr. Noida, India
Email: sksingh8@gmail.com

Abstract—A huge amount of new information is placed on the Web every day. Large scale search engines frequently update their index gradually and are not capable to present such information in a timely behavior. An incremental crawler downloads customized contents only from the web for a search engine, thereby helps falling the network load. This network load farther will be reduced by using mobile agents. It is reported in the previous literature that the 40% of the current Internet traffic and bandwidth utilization is due to these crawlers. These crawlers also effect load on the remote server by using its CPU cycles and memory, these loads must be taken into account in order to get high performance at a reasonable cost. This paper deal with all those problems by proposing a system based on parallel web crawler using mobile agent. The proposed approach uses mobile agents to crawl the pages. The main advantages of parallel web crawler based on Mobile Agents are that the analysis part of the crawling process is done locally. This drastically reduces network load and traffic which can improve the performance and efficiency of the crawling process.

Index Terms—world wide Web, search engine, mobile crawler, parallel crawler, aglets, Web crawler, network traffic, mobile agent.

I. INTRODUCTION

The current population of the world is about 7.049 billion out of which 2.40 billion people (34.3%) use Internet [3]. From .36 billion in 2000, the number of Internet users has increased to 2.40 billion in 2012 i.e. an increase of 566.4% from 2000 to 2012. In Asia out of 3.92 billion people, 1.076 billion (i.e.27.5%) use Internet, whereas in India out of 1.2 billion, .137 billion (11.4%) use Internet. Same growth rate is expected in near future too. Fig. 1: illustrates Internet Users in the World Regions.

Large size and the absence of centralized control over its contents are two key factors for the success of the web. Unfortunately, same issues are responsible for problem in locating the desired information in time. In-fact quality is distributed very off-centered i.e. interesting pages are sparse in comparison with the rest of the contents. Hence, there is a need of a more effective way of retrieving information from the web. To handle with this complete volume of data available on the web, programs that retrieves websites known as search engines; has been designed. Search engines help the user to find relevant pages from the web based on important words. A crawler downloads the web pages from www to be used by search engine later.

Since the web is dynamic and 53% of its contents change daily [11], to maintain the up to date pages in the group, a crawler needs to revisit the websites many times. Due to more revisit, the property like CPU cycles, disk space, and network bandwidth etc., it will become overloaded and due to this type of overloads sometime a web site may collapse. Study [12] report that the current web crawlers have downloaded and indexed billion of pages and about 41% of current internet traffic and bandwidth spending is due to the web crawlers. Nevertheless, the maximum web scope of any well-
known search engine is not more than 16% of the current web size.

Using mobile agent i.e., mobile crawlers, the method of selection and filtration of web pages can be done at servers rather than search engine side which can reduce network load caused by the web crawlers [9].

II. RELATED WORK

A general search engine has three major parts named as indexer, crawler and query engine. Web crawlers are programs that traverse the web on the search engine’s behalf, and follow links to reach different pages to download them. Beginning with a set of seed URLs, crawler will extract URLs showing in the retrieved pages, and store pages in a warehouse. The downloaded pages are indexed and stored in the search engine database. This continuous updating of database makes a search engine more consistent source of relevant and updated information. Details of crawler are discussed by [17].

The crawler has to deal with two main responsibilities i.e. downloading the new pages and remaining the earlier downloaded pages will fresh. However, superior freshness can only be promised by simply revisiting all the pages more often without placing needless load on the internet. With the available bandwidth which is neither infinite nor free, it will become essential to crawl the web in a way that is not only scalable but also efficient, if several reasonable measure of quality or freshness is to be continued.

III. ISSUES OF EXISTING CRAWLERS

Presently there are two types of crawlers, General Crawler and Focused Crawler [13]. General Crawler is mainly used for search engines, whose goal is to meet common user’s general demand by maximizing the coverage rate of web resources. However, some obvious problems exist in General Crawlers:

1) At the same time of maximizing the coverage rate of web resources, General Crawlers also download a large amount of useless information. Focused Crawling

2) A large amount of web pages are written in JavaScript or Ajax. It’s impossible to extract new URLs by tag matching because these link URLs are generated by JavaScript functions.

3) Most of General Crawlers only support keywords search, but not the attribute search.

Focused Crawler has solved the first problem. It can select the link URLs relevant to some subjects, and filter the useless information. Although Focused Crawler has filtered most of useless link pages, a large amount of useless information still remain.

A web crawler consumes a large amount of network bandwidth and other resources by accessing the web resources at a rapid speed. This affects the performance of the web server significantly. A considerable amount of resources of underlying network are consumed to build a complete full text index of the web pages. Further, to keep the indices of a search engine up-to-date, crawlers continually retrieve the pages at a fast speed. Thus the crawling behavior of a single search engine causes a daily load of 60GB to the web [14].

Parallel and distributed crawling was purposed to enhance the coverage and to reduce the bandwidth usage but this systems distributed and localized the load but does not help a lot in declining the load [13].

IV. MOBILE AGENT BASED CRAWLING

The Web crawling approach proposed in this work departs from the centralized architecture of traditional search engines by making the data retrieval component, the Web crawler, distributed. We describe mobility in the perspective 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 is started on that Web server. Hence, mobile crawlers are capable to go to the resource which needs to be accessed in order to take benefit of local data access. After accessing a resource, mobile crawlers shifted to the next server, carrying the web crawling result in the memory. The role of mobile crawlers and depicts the decentralized data retrieval architecture as established by mobile crawlers given in Fig. 2.

A. Properties of Mobile Agents

Mobile agents have the following unique properties.

Adaptive learning: Mobile agents can learn from past experiences and adjust themselves to the environment. They can supervise traffic in large networks and learn about the problem spots in the network. Based on the experiences of the agent in the network the agent can choose better routes to reach the next host.

Autonomy: Mobile agents can take some decisions on its own. For example, mobile agents are free to choose the next host and when to migrate to the next host. These decisions are crystal clear to the user and the decisions are taken in the significance of the user.

Mobility: Mobile agents have the ability to shift from one host to another host in the network.

The major benefit of the mobile crawling approach is that it permits us to distribute crawling functionality within a distributed system just like the Web. Specifically, we can see the following four advantages:
B. **Localized Data Access**

Due to HTTP request or response model, downloading the contents from a Web server involves major overhead due to request messages which have to be sent for each Web page individually. Using a mobile crawler we can reduce the HTTP overhead by transferring the crawler to the source of the data. The web crawlers then issue all HTTP requests locally with respect to the HTTP server. This approach still needs one HTTP request per web document but there is no need to send out these requests over the network anymore. A mobile crawler therefore saves bandwidth by reducing Web traffic caused by HTTP requests.

C. **Remote Page Selection**

Using mobile crawlers we will distribute the crawling algorithm contained by a system of distributed data sources such as the Web. This will allow us to promote Web crawlers from uncomplicated data retrieval tools to more intelligent components which can use information regarding the data they are believed to retrieve. Crawler mobility permits us to shift the decision whether or not some pages are relevant to the data source itself. Once a mobile crawler has been shifted to the Web server, it can evaluate every Web page prior to sending it back which would involve network resources. By appearing at this called remote page selection from a more abstract position of view, it compares favorably with classical approaches in database systems. If we think the Web as a huge remote database, the job of a crawler is similar to querying this database. The main difference between mobile crawlers and traditional crawler is the way queries are issued. Traditional crawlers implement the data shipping approach of database systems because they download the whole database before they can issue queries to identify the relevant portion. In contrast to this, mobile crawlers implement the query shipping approach of database systems because all the information required to identify the related data portion is moved directly to the data source together with the mobile crawler. After the query executed remotely, only the query result is moved over the network and can be used to set up the desired index without requiring any further analysis.

D. **Remote Page Filtering**

Remote page filtering expands the idea of remote page selection to the contents of a Web page. The goal behind remote page filtering will allow the crawler to manage the granularity of the data it retrieves. With stationary crawlers, the granularity of recovered data is the Web page itself since HTTP permits page level access only. For this reason, stationary crawlers for all the time have to retrieve an entire page before they can extract the appropriate page portion. Depending on the ratio of relevant to irrelevant information, major portion of network bandwidth are exhausted by transmitting ineffective data. A mobile crawler overcomes this difficulty since it can filter out all irrelevant page portions keeping only information which is relevant with respect to the search engine the crawler is working for. Remote page filtering is especially useful for search engines which use a specialized representation for Web pages (e.g., URL, title, modification date, keywords) instead of storing the complete page source code.

E. **Remote Page Selection**

In the case where a crawler must establish a comprehensive full text index of the Web, techniques like remote page selection and filtering are not applicable since every page is considered to be relevant. In order to decrease the amount of data that is to be transmitted back to the crawler controller, we introduce remote page compression as another fundamental feature of mobile crawlers: To decrease the bandwidth required to transfer the crawler along with the data it holds back to the search engine, the mobile crawler reduces its size before transmission. Remote page compression reduces Web traffic for mobile full text crawlers as well as for mobile subject specific crawlers and makes mobile crawling an attractive approach even for traditional search engines which do not benefit from remote page selection and filtering due to their comprehensive full text indexing scheme.

V. **MOBILE AGENT BASED PARALLEL WEB CRAWLER**

The Parallel Crawler based on Mobile Agent includes Crawler Manager, Crawl Frontiers, Centralized Database and home Database of each Crawl Frontier and Central Crawler. It is task of central crawler to getting the URL input from the applications and transfers the URLs to the available mobile crawling process. Crawling process transferred to different machines to increase the system performance. Local repository of each crawl frontier is buffers that locally collect the data. This data is shifted to the central crawler after compression and filtering which decreases the network bandwidth overhead. The central crawler has a centralized database which contains the documents gathered by the crawl frontiers independently. The main benefits of the parallel crawler based on mobile agent are Local Data Access, Remote Page Selection, Remote Page Filtering, Remote Page Compression, Scalability, Network load dispersion, Network load reduction.

VI. **PROPOSED ARCHITECTURE OF PARALLEL CRAWLER BASED ON MOBILE AGENT**

A parallel crawler based on mobile agent consists of several multiple crawling processes which migrate to source of data before the crawling process is actually started on that source of data. A parallel crawler based on mobile agent goes to resources and take the benefits of local data access. Parallel crawler based on mobile agent after accessing a resource transfers to the next host or server or to their home system. All migrating parallel crawling process executes the tasks of single crawler that it downloads pages from the server or host, stores the pages in the local database, extracts all URLs from the downloaded web pages and follows the extracted URLs. When Crawling process executes on the same local area network and communicates through a high speed
interconnection network then it is known as intra site migrating parallel web crawler. When crawling process execute at geographically remote locations connected by the Internet or a WAN then it is known as distributed migrating parallel web crawler. The architecture includes of central coordinator system and crawling process.

A. Central Coordination System

Central coordinator system includes of Central Crawl Manager, Crawling Application, Web Cache, Central Database, URL Dispatcher, URL Distributor. Central crawl manager is the central part of the crawler all other components are started and register with the central crawl manager to request services. It performs as the heart of the system. Central Crawl Manager’s aim is to download web pages in the order specified through the crawling application. The central crawler has a list of web URLs to crawl. After receiving the URLs of files, central crawl manager queries the DNS resolvers for the IP addresses of the servers. The central manager then consideration into robots.txt files in the root directory of the server. The central crawler is a set of available crawling process which has registered themselves with the central crawl manager which logically migrated to different specific locations. The central crawl manager allocate different web URLs to all the crawling process in domain specific manner and it turns the crawling processes begin to crawl the received web URL in breadth first approach and pages are downloaded by multi threaded downloader. It is the responsibility of the crawling application to check for duplicate pages. The crawling application in this paper consider in a breadth first crawl strategy. Breadth first search provides high quality web pages hence improving the quality of downloaded web pages. The crawling process in fact crawls in breadth first manner. The DNS resolver uses the GNU adns asynchronous DNS client library to access a DNS server usually allocated on the same machine. A web cache stores web resources in expectation of future requests. Web cache works with the principle that the very popular resource is possibly to be requested in the future. The benefits of Web Cache are to decrease network bandwidth, less user perceived delay, less load on the servers. Web cache provides fast response time. When the information resides in web cache, the request is fulfilled by a cache, the content has no longer to travel across the Internet. Central database stores the list of URLs received and downloaded pages which are collection of the documents downloaded by crawling process. Web pages will be saved in compressed form. Each document is provided with a unique number called document identifier denoted by doc_id. The documents are stored along with doc_id, length of document and URL. This helps with data consistency and creates data storage much easy and well-organized. Data is saved in the central data base in hierarchical manner.

Each parent node of the tree is the domain and the child node of the parent node is the sub domain. Each node includes of domain name and its equivalent web URLs. The database is updated each time for future use by adding new URLs. Central Database is in touch with the database of the search engine. URL dispatcher reads the URL from the database and retrieves the web URL from the index table maintained in the central database and forwards it for crawling. URL distributor classifies the web URLs on the basis of domains. It then distributes the web URL to the concerned domain for crawling. The URL distributor will balance the load on the web crawling process.

B. Crawling Process

Crawling process consists of Scheduler, New ordered Queues, Site ordering module, URL Queues/ Known URLs, Multithreaded Downloader, URL Collector, Link Extractor, Link analyzer. Scheduler supplies the set of URLs to be downloaded web page based on the customized page rank. The URLs are saved in latest ordered queues. Latest ordered Queues saves the set of URLs based on customized page rank. Site ordering module gives the customized page rank of the web page. Known URLs are the set of already known URLs. They will be treated as seed URLs. Multithreaded downloader takes a URL from URL collector and downloads the related webpage to store it in the local repository. The downloade component fetches files from the web by opening connections to different servers. URL collector maintains the web URL from the downloaded web pages. Link Extractor extracts the URL from the downloaded web pages. Link analyzer verifies the extracted web URLs by the link extractor. If there is any similarity in the URL then such URLs are rejected and not further forwarded for processing. When the web crawling process executes it requires a few memory space to save the downloaded pages. Each web crawling process has its own local repository. The web crawling process saves the downloaded pages in this repository. It is the storage area of the machine on which the crawling process is running. The Ranking Module constantly scans the known URLs and the local database to make the refinement decision. The Ranking Module filters the local repository. The Ranking Module rejects the less important web page from the local repository to make space for the new web page. Locally collected URLs are the set of URLs in the local collection. The Update Module will maintain the local repository fresh. In order to maintain the local repository fresh, the crawler has to select the web page that will increase the freshness most considerably this result is known as update decision.

VII. Conclusion

In this paper we proposed a Model for Effective Parallel Web Crawling based on Mobile Agent and Incremental Crawling. Parallel Web Crawling based on Mobile Agent will yield high quality pages. The crawling process will migrate to host or server to start downloading. Incremental crawling will keep the pages in local database fresh hence increasing the quality of downloaded pages.

The Mobile agent based Parallel Web Crawler can filter out the HTML pages that have not been modified since last crawl. This technique can reduce the usage of

©2013 Engineering and Technology Publishing 89
CPU cycles at the remote site compared to other mobile crawler techniques because the pages, which are not modified and not retrieved, along with this near duplicate detection feature adds more privilege to reduce unwanted downloads that improved performance. The crawler revisit frequency will deduce the page updating activity. Thus, the proposed mobile crawler system will reduce the traffic on the network and saved CPU cycles considerably as compared to Traditional Crawler.

ACKNOWLEDGMENT

The author especially thanks to mention the financial help provided by MANF JRF under University Grant Commission (UGC), New Delhi. No. F1-17.1/2011/MANF-MUS-BIH-3287.

REFERENCES


Md. Abu Kausar received his BCA degree from T. M. Bhagalpur University, Bhagalpur in 2002. Master in Computer Science from G. B. Pant University of Agriculture & Technology, Pantnagar, India in 2006 and MBA (IT) from Symbiosis, Pune, India in 2012. He has received Microsoft Certified Technology Specialist (MCTS). At present, he is pursuing Ph.D. in Computer Science from Jaurpur National University, Jaurpur, India and he is receiving UGC MANF JRF Fellowship during Ph.D Programme. He is having 6 years of experience as a Software Developer. His research interest includes Information Retrieval and Web Crawler.

Dr. V. S. Dhaka is a young and dynamic technocrat with 10 years of intensive experience in industry and academia. He is M.Tech and Ph.D in computer Science from Dr. B R Ambedkar University, Agra. With more than 32 publications in international journals and paper presentations in 27 conferences/seminars, he always strives to achieve academic excellence. He has been awarded by the employers with “Employee of the Quarter Award”, “Mentor of the year award” and with letters of appreciations for his commitment, advocacy and mentor-ship. He has organized several Conferences, Seminars and Workshops.

Dr. Sanjeev Kumar Singh is working as Assistant Professor in Department of Mathematics at Galgotias University, INDIA. He earned his M. Sc. and Ph.D. degrees with major in Mathematics and minor in Computer Science from G.B. Pant University of Agriculture and Technology, Pantnagar, INDIA. Before that he completed B.Sc. (Physics, Mathematics & Computer Science) from Lucknow University, Lucknow. He is having more than nine years of teaching and research experience. Besides organizing three national conferences, he has published several research papers in various International/National journals of repute and several national and international conferences and workshops. His areas of interest include Mathematical Modeling, Differential Geometry, Computational Mathematics, data mining & Information Retrieval.