Rajender Nath, Naresh Kumar / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 5, September- October 2012, pp.266-269

A Novel Architecture for Parallel Domain Focused Crawler

Rajender Nath*, Naresh Kumar**

* Professor, DCSA, Kurukshetra University, Kurukshetra, Haryana, India
**Assistant Professor, MSIT, Janak Puri, New Delhi, India

Abstract

WWW is a collection of hyperlink document available in HTML format [10]. This collection is very huge and thus difficult to refresh quickly because 40% of the web pages changes daily. As the web has dynamic nature, so, to cover more and more pages on the web, the concept of parallel crawlers are used. These crawlers run in parallel and cover the wider area of the web. But the parallel crawlers consume more resources especially bandwidth of the network to keep the repository up to date. So this paper proposes an architecture that uses the concept of mobile crawlers to fetch the web page and download only those pages that are changed since the last crawl. These crawlers run in the specific domain and skip the irrelevant domains.

Index Terms— WWW, Search Engine, URLs, Crawling process, parallel crawler, Internet traffic.

Introduction

WWW [7] is a client server architecture that allows the users to search by using keywords on the search engine interface and in response search engine returns the web pages to the user. Very large number of page are available on the web, therefore search engine require a mechanism called crawler to collects the web pages from the web. A web crawler (also called walker, spider, bot) is a software program that runs repetitively to download the web pages from the web**.** This program takes the URL from the seed queue [1] and fetches the corresponding web page from the web. Millions of web pages are downloaded per day by a crawler to achieve the target of fulfilling the user requirements. But in spite of all this it is not possible to visit the entire web and to download all the requested web pages because according to [13] any search engine can cover only 16% of the entire web. To traverse the Web quickly and entirely is an expensive, unrealistic goal because of the required hardware and network resources [2, 3].

Currently web crawlers have indexed billions of web pages and 40% of Internet traffic and bandwidth utilization is due to web crawlers that download the web pages for different search engines [12].

This paper proposes an approach in which the concepts of mobile agents are used as parallel crawlers based on domain specific web crawlers. These mobile crawlers go to the remote site and takes only those pages that are found modified since the last crawl and send only compressed modified web page to search engine for indexing.

I. RELATED WORK

A domain specific parallel crawler is a program used for searching the information in the specific domain (.org, .com, .edu etc.) only [4]. The main aim of domain specific crawler is to collect all web pages, but selects and retrieves pages only from the specific domain [1, 2, 5].

In [9], last modified date, number of keywords and number of URLs were used to check the change in the page. The purpose of this was to determine page change behavior, load on the network and bandwidth preservation. from the experimental work it was found that 63% of the total pages changed that need to be downloading and indexing. Further the experiment also shows the reduction in downloading up to 37% . It was also shown that proposed scheme reduce the load on the network to one fourth through the use of compression and preserving bandwidth of network.

Focused Crawler introduced by Chakrabarti et al. [6] describe the focused crawler in which web pages were maintained on a specific set of topic hierarchy. The goal of this paper was to selectively look for pages that are relevant to a pre-defined set of topics**.** The proposed system was built from three separate mechanisms namely: - Crawler, classifier and distiller. Classifier tells the page relevance according to the taxonomy. The distiller identifies hyper textual pages that are the access points to many relevant pages within a few links. The author designed the Focused Web Crawler that crawl the web pages and store them locally based on the relevancy and priority criteria to retrieve the web page.

The purposed distributed crawler and parallel crawling in [1] increase the coverage and reduce the bandwidth preservation. But it only distributes and localized the load, not reducing the load in actual.

After studying various papers [9] on the mobile agent these problems are identified by us:

1. The relevancy of any web page can be determined after downloading it on the search

engine end only. So once a page is down loaded then what is the benefit of filtering it because all the network resources are already used while downloading it.

2. The load on the network and bandwidth preservation while downloading is considerably large.

To overcome the above listed problems, this paper proposed an architecture for mobile web crawlers that uses the concept of frequency change estimator and some new component helps in cleaning the page that are not changed at the remote site.

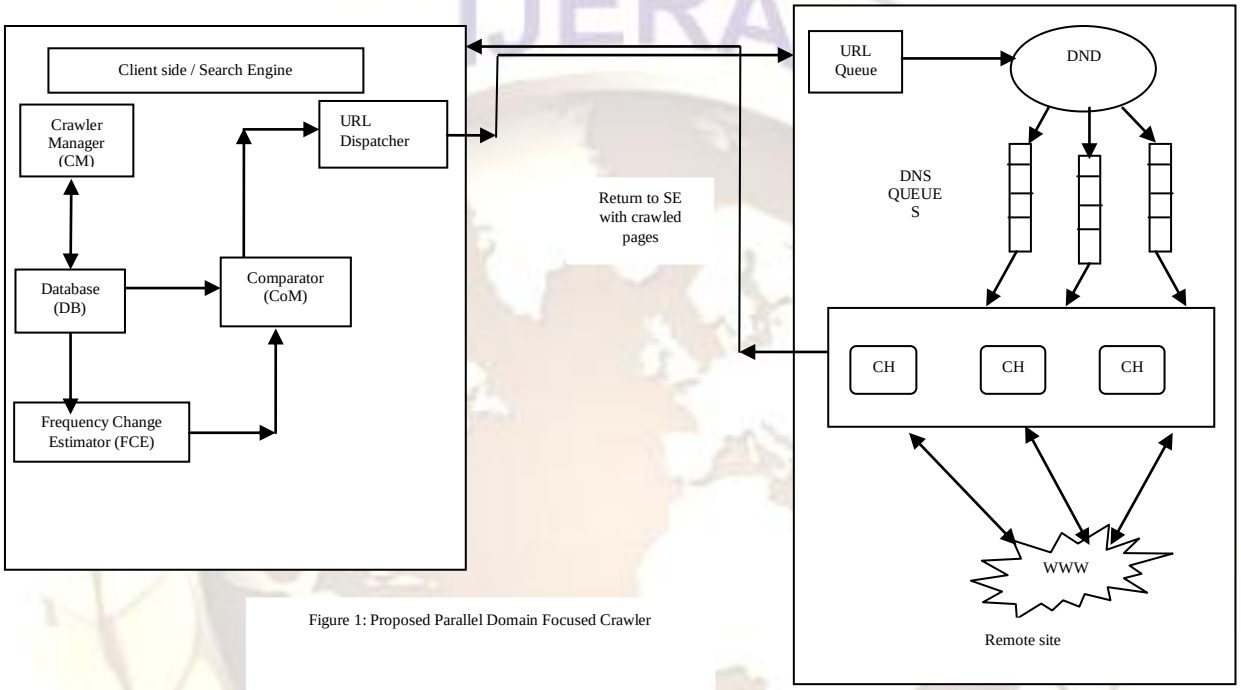

Figure 1: Proposed Parallel Domain Focused Crawler

3. Proposed Work

The proposed Mobile Crawler is shown in Figure 1, in which the mobile crawler are send to the remote site where they check modification in page. If the page is modified then download it otherwise reject the URL. The major components for proposed architecture are listed below:

1. Analyzer Module (AM).
2. Domain Name Director (DND).
3. Crawler Hand (CH).
4. Crawler Manager (CM).
5. Data Base (DB).
6. Frequency change Estimator (FCE).
7. Comparator (CoM).

3.1 Crawler Manager

The main task of CM are mobile crawler generation and assignment of URL's to the mobile crawler for crawling the web pages based on the information taken from the FCE. After receiving the downloaded web pages in compressed form the task of decompressing the web pages is also performed by the CM.

3.2 URL Dispatcher

URL Dispatcher [11] will take the URL from the Comparator module and send them to URL queue for further processing.

3.3 Frequency Change Estimator

This module will identify whether the two version of web pages corresponding to the same URL are same. This module is the part of the client site and remains there. The FCE is used to calculate the probability of the page change by visiting the web page periodically [8]. This module maintains the page change frequency of every page at the SE and also judge that in how many days this page will be modified based on periodical visit of this web page by the web crawler. Whenever the date and month of change for any web page occurs it will send the URL of that web page to URL dispatcher so that this web page can be downloaded to refresh the repository.

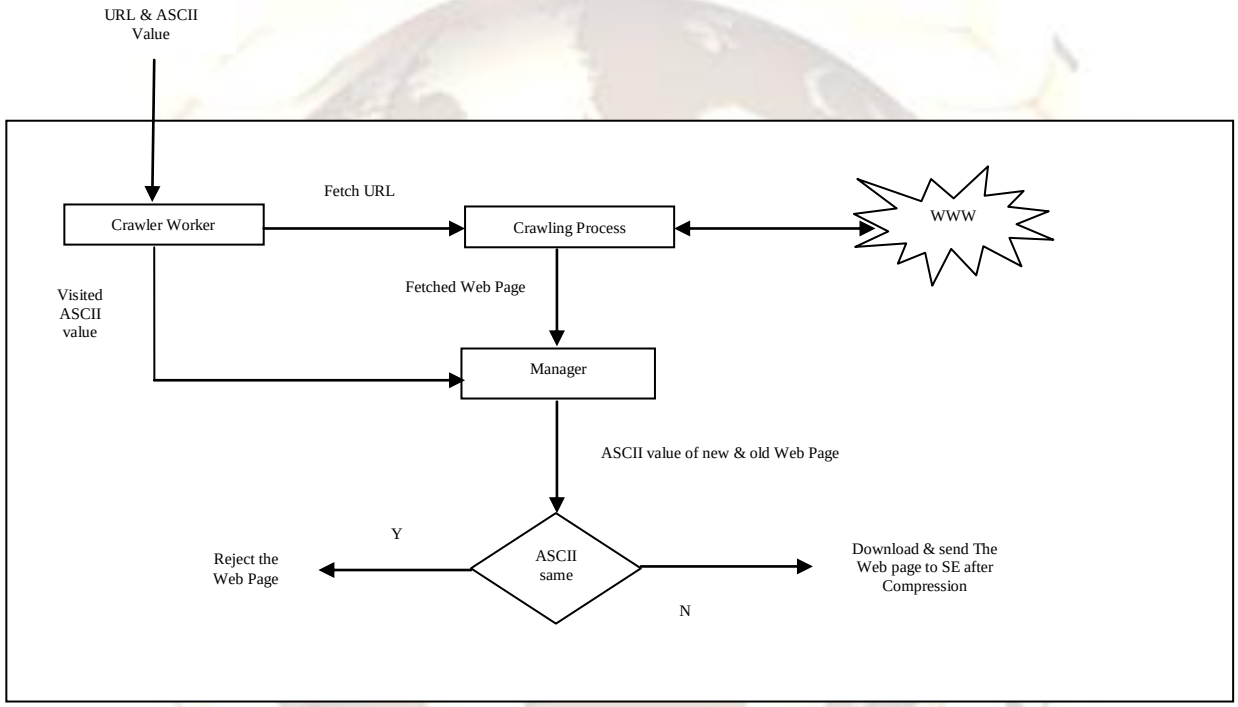

Figure 2: Crawler Hand

3.4 Database (DB)

The data base of any system performs the one of major role. This is maintain at the search engine end and contains the information about all the crawled page crawled by the mobile crawlers. This module has five fields:

1) Name of URL: Name of those URL's that have been visited by the crawler.
2) Parent URL: Parent web page of the current URL.
3) ASCII Count: Total sum of ASCII value of each character of the web page.
4) Last modified date: Date and month on which the page was last modified.
5) Frequency of change: Expected date on which page change will take place.
6) File path: Complete path of the web page at the search engine end.

3.5 Comparator Module

The CoM checks to see if the downloading period of a web page is reached. If yes, then send this URL of Web page to URL dispatcher for further processing otherwise reject this URL. The last modify and frequency of change is taken from the FCE module that was computed when the URL was previously crawled. It must also be noted that ASCII value of the web page is sent with the mobile crawler so that it will help the mobile crawler to decide that the page is actually changed or not.

3.6 Domain Name Director (DND)

The main job of the DND is to take the URL from the URL queue and assign them to the particular domain (.org, .com, .edu etc..). Given a URL to the DNS Queue from where crawler will take the URL and fetch the web page. The number of downloading pages or crawling time will depends upon on the URL itself. The distribution of the load on the Crawler Hand is likely to be different depending on the frequency of demand of the domains.

3.7 Crawler Hand or crawler (CH) - The major module of the entire proposed architecture is crawl hand (as shown in Figure 2). Crawl Hand retrieves the URL from the DNS Queue in FCFS fashion and sends request to web server. The web documents for corresponding URL are fetched and send to the Manager. Manager will calculate the

corresponding ASCII value of that web page. Compare the new and old ASCII value of the corresponding page, if they are same then reject the web page otherwise send this page to SE after compression.

4. Conclusion

The size of the web is exponentially growing and 40% of the web pages are changes daily [9] Crawlers are being used to collect Web data for search engine. This paper has enumerated the major components of the crawler and Parallelized the crawling system that is very vital from the point of view of downloading documents in a small amount of time.

This proposed architecture of domain specific parallel crawler fulfills the following characteristics-Full Distribution, Scalability, Load Balancing and Reliability also because any web page is downloading only when it gets any change. Further more the web page is downloaded only in the compressed form that will reduce the size of the web page.

5. Future Work

To enhance and prove this architecture several interesting issues like page change with frequency, Link Extraction from the downloaded web page etc. are yet need to be identified. This will be produced with implementation results in near future. The results will be compared with the other techniques used by mobile web crawlers in the form of bandwidth preservation, downloading the number of web page, total downloading time that is saved by this proposed work.

References

[1] Junghoo Cho, Hector Garcia-Molina, "Parallel Crawlers " published in *WWW 2002*, May 7–11, 2002, Honolulu, Hawaii, USA. ACM 1-58113-449-5/02/0005.

[2] Bidoki, Yazdani et el, "FICA: A fast intelligent crawling algorithm", Web Intelligence, IEEE/ACM/WIC International conference on Intelligent agent technology, Pages 635-641, 2007.

[3] Cui Xiaoqing Yan Chun," An evolutionary relevance calculation measure in topic crawler " CCCM 2009, ISECS International Colloquium on Computing, Communication, Control, and Management, 267 –270, Aug 2009.

[4] Nidhi Tayagi and Deepti Gupta, "A Novel Architecture for Domain Specific Parallel Crawler", published in Indian Journal of Computer Science and Engineering" Vol 1 No 1 44-53, ISSN: 0976-5166.

[5] *Junghoo Cho, Hector Garcia-Molina, Lawrence Page,* Efficient crawling through URL ordering", 7th International WWW Conference , April 14-18, Brisbane, 1998.

[6] S. Chakrabarti, M. v. d. Berg, and B. Domc, "Focused crawling: a new approach to topic-specific Web resource discovery", published in Computer Networks, 31(11–16):1623–1640. 1999.

[7] Sergey Brin and Lawrence Page," The Anatomy of a Large-Scale Hypertextual Web Search Engine" pulished in the International conference WWW7 , 1998.

[8] Junghoo Cho, "Estimating Frequency of Change" published in ACM Journal Name, Vol. V, No. N, Month 20YY, Pages 1 - 32.

[9] Rajender Nath and Satinder Bal, "A Novel Mobile Crawler System Based on Filtering off Non-Modified Pages for Reducing Load on the Network", published in The International Arab Journal of Information Technology, Vol. 8, No. 3, July 2011.

[10] Naresh Chauhan, A. K. Sharma, " Design of an Agent Based Context Driven Focused Crawler" published in BVICAM'S International Journal of Information Technology, 2008.

[11] Rajender Nath and Naresh Kumar," A Novel Architecture for Parallel Crawler based on Focused Crawling", published in 6th international conference on Intelligent Systems, Sustainable, new and renewable Energy Technology & Nanotechnology, March 16-18, 2012.

[12] Yuan X. and Harms J., "An Efficient Scheme to Remove Crawler Traffic from the Internet," in Proceedings of the 11th International Conferences on Computer Communications and Networks, pp. 90-95, 2002.

[13] Lawrence S. and Giles C., "Accessibility of Information on the Web," Nature, vol. 400, no. 6740, pp. 107-109, 1999.