ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Study of Parallel Crawler and Its Novel Approaches

[I]Komal Agrawal, [II]Anju Gera
[I,II]Dept. of Computer Engineering, BSAITM, Faridabad, Haryana, India

Abstract
World Wide Web is an interlinked collection of billions of documents which are formatted in HTML. With the rapid increase in the size of WWW, the overloaded information produces side effect on web. It has become a difficult task to access all URLs in the web documents and it is also a challenge to handle billions of URLs by single processor. To improve efficiency on web, it is essential to use parallel crawler or to parallelize a crawling process. In this paper, Architecture of parallel crawler is analyzed. This paper focus on the topic how parallel crawler helps to improve efficiency, what are its common challenges and novel architecture to overcome the challenges occur during parallel crawling.

Keywords
WWW, Search Engine, parallel Crawler, Web Crawler, and Domain Specific.

I. Introduction

A Crawler is a program that browses the World Wide Web in systematically manner. World Wide Web is a collection of web pages. These web pages are stored and retrieved by web crawler. These Web pages are retrieved by some specified URLs. A Web Crawler has a seed queue at first, which consist of initial set of URLs.

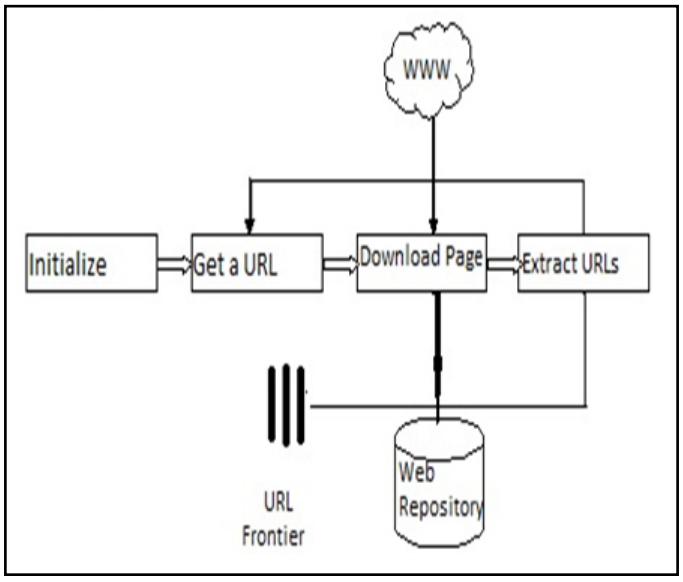

Fig. 1: Flow of a Crawling Process

[4, 5] The working and flow process of a Web Crawler is as follows:

- URL Frontier, Which consist of starting URLs.
- Select a URL from URL Frontier.
- Download web pages consisted by that URL.
- Parse the web page to extract the corresponding URL links in webpage.
- Add the extracted URLs to the Frontier.
- Pick up the next URL to retrieve pages.
- Go to step 3and repeat this crawling process until web crawler decides to stop it.
  In Fig: 1 URL Frontier is collection of starting URLs and Web Repository where downloaded pages are stored.

Different URL consist different web pages. These Web Pages are retrieved and prioritized. Web crawler download web pages by extracting its URL and extract new URL links in web page. These URLs further added to URL Frontier queue. A crawler downloads and stores the web pages by using hyperlinks present in the URL. World Wide Web is growing exponentially because new web pages in a big amount are added to the web every day. [9]According to Google's index size, World Wide Web stands at around 55 billion pages.

It has become more difficult to retrieve the whole portion of Web using a single process. So, to increase download rate, it is imperative to use multiple crawler process instead of single process. In parallel crawler, multiple processes works in parallel. Each crawler process access specific portion of web thus whole portion is retrieved by multiple processes and it improves efficiency and download rate of available data on Web. [10] Primary Goal of Web crawler are Finding new web pages and changes in previous pages after some specific time to make updates on the web.

II. Parallel Crawler

To maximize the download rate and to reduce network traffic it is essential to use multiple crawling processes in parallel i.e. Parallel Crawler. In Parallel Crawler, many of c-proc's run in parallel to crawl the whole web efficiently. C-proc's run at local network having its own URL's queue and database that have collected pages. To achieve desired download rate no. of c-proc's can be increased. Parallel crawler is technique to overcome the shortcomings of single process crawler.

C-proc's crawls the web, downloads and stores the web pages locally, parse them to extract URLs from web pages. These tasks performed by c-proc's may be [2] distributed on the same local network or at geographically distant locations.

Intra-site parallel crawler: Through a high speed interconnect (LAN), when all c-proc's run on same local network is called intra-site parallel crawler. In general architecture of parallel crawler, all c-proc's runs on the same local network.

Distributed crawler: Through Internet (WAN), when c-proc's run at geographically distant locations is called distributed crawler. E.g. one c-proc run in France, crawls all French pages and another c-proc run in U.S, crawls all U.S pages. It reduces the load on the network.

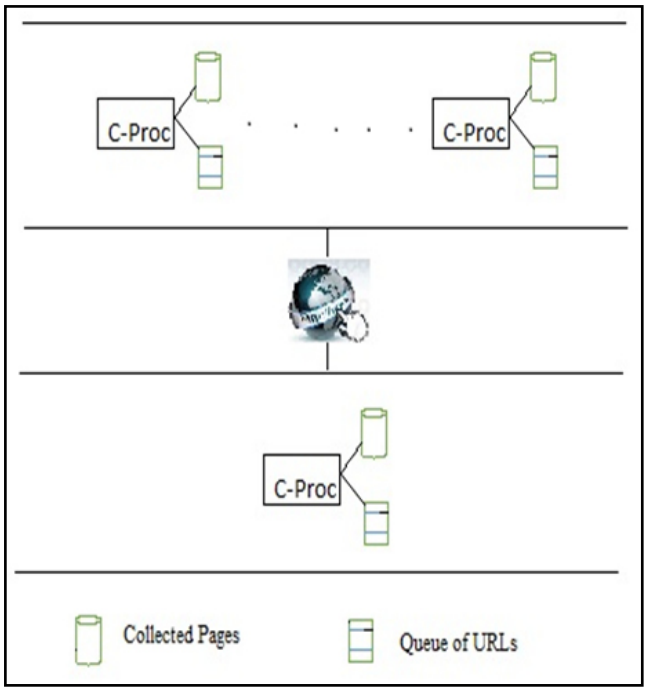

Fig. 2: General Architecture of a Parallel Crawler

There are many important advantages [1, 2] of parallel crawler over single process crawler:

- Download rate: Due to exponential growth of web, it is necessary to use a parallel crawler. Because desired download rate can't be achieved by single process crawler.
- Network Load Reduction: A parallel crawler actually reduces the network load. For Example: Assume that a crawler in Africa retrieve a page from America. It needs to go through the whole network to collect pages. Instead if the area is divided in between crawlers, and allow each crawler to crawl the pages of its area only. It will reduce overall network load because pages will go through only local networks.
- Robust: A parallel crawler is robust by its nature because it handles the situation when additional processes are added to increase download rate.
- Reliability: Reliability matters in each system. In case of parallel crawler, Whole system never gets affected by failure of single crawler worker. Hence it increases reliability.

We have studied its advantages now there are some issues [1] in parallel crawler which makes its working challenging:

- URL Overlap: When the process runs in parallel there may be a situation where same URL is retrieved multiple times by different process. I.e. URL Overlap.
- Content redundancy: Different URL may have same web page so this web page may get retrieved multiple times and will cause of content redundancy. It can be solved by placing a centralized server but this server may act as a single source of failure.
- Communication Overhead: To prevent content redundancy, overlapping of URL and to improve quality of downloaded pages, a conversation can be placed between crawling processes. So that, each crawl process makes aware all process about its downloaded pages. It will remove overlapping because each process knows exactly which pages it needs to download. But these conversations between crawler processes consume time and also enhance network bandwidth.

III. Domain Specific Parallel Crawler

Parallel Web Crawler is an efficient way to crawl the web. In this, multiple single process crawler crawl the web in parallel. It is imperative to implement parallel crawler because of its advantages over single process crawler. As old crawling technique can't produce desired download rate. But there are some issues related to Parallel Web Crawler i.e. URL Overlap, Content duplicity, Communication Overhead. To overcome these issues there is a new approach of parallel crawler i.e. Domain specific parallel crawler [6]. In which, each crawler is assigned to crawl specific domain. It makes crawling more effective in terms of relevancy and load sharing. Each site has a domain name e.g. .com, .edu, .in, .org, .com etc.

Domain specific crawler is a program used to search the information in the specific domain (.org, .com, .edu, etc.) by placing its downloaded web pages in its specific domain. Domain specific parallel crawler is used to collect the web pages, but it retrieves pages only from the specific domain. One crawl process can crawl the data from its specified domain only. This novel approach makes partition of web or distribute web in between some specific domains to reduce load on the network. It reduce redundancy in content of the crawler as crawling process will download page belong to its domain only or will not download page that does not belong to its specific domain. Thus this approach is called domain oriented partitioning approach. A crawl process which is allowed to download web pages from .org domain as its specific domain, it will download pages only related to .org domain and will not download other domain related pages. Hence it reduces URL overlap, content duplicity, network load, and provides scalability and reliability to the whole system. In flow of crawling process there is a seed queue of URLs each URL related to different domains, when process starts URLs retrieved by their crawl process according to their domain name. This approach helps to overcome the issues related to parallel crawler, because we don't need to establish communication between crawl processes after applying this novel approach.

IV. Parallel Domain Focused Crawler

Parallel Domain Focused Crawler downloads only changed pages and ignore those pages that are not changed since the last crawl. This crawler helps to reduce the load on the network. [5] In parallel domain focused crawler, CM (Crawler Manager) generate mobile crawler that is used to crawl the web pages. DND (Domain Name Director) select specific domain for the URLs. So that one process can crawl the pages from its specific domain. FCE (Frequency change estimator) identifies versions of web document so that it can check whether both downloaded web documents are same or not. If two web documents are same then reject the web page otherwise download and send the web page to search engine. AM (Analyzer Module) is used to perform ASCII counts which helps to find out the difference between documents.

ODF (Old Database File) contains statistics about each HTML page to be crawled. ODF is used to send two things i.e. name of URL & ASCII count. CM (Comparator Module) compares between old data file and new data file. LE (Link Extractor) downloads links from web pages. This architecture helps to reduce the load on the network and save memory.

V. Web ParF: Partitioning Framework

Web ParF is web partitioning framework for parallel crawler. It is designed to overcome URL, Content duplication and reduce

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

communication overhead.
URL oriented partitioning used to overcome URL duplication. It divides the web by assigning URLs to partitions but it would result in multiple crawlers because a web page can be referred by different URLs. Content Oriented Partitioning used to overcome content duplication.
[2] Web ParF proposed domain oriented approach in its architecture which overcomes URL and content duplication. This approach makes the system highly scalable as crawling process thread is added to the architecture which breaks domains into sub-domain. It creates domain specific partitions of URL Frontier Queue It provide an order to URLs in the URL pool. It performs this task by using ranker module that assigns a score to each URL and place URL at a specific position. This makes parallel crawler not only scalable but also fault tolerant by balanced distribution of load.

VI. Literature Survey

Web crawler crawls the web and downloads web pages. World Wide Web is big in size as thousands of pages added daily to database. It is important to implement parallel crawler so that it can crawl maximum pages in less time. Jungoo cho et. al. [1] presents whole concept of parallel crawler. This paper shows multiple architecture of parallel crawler. This paper helps to take review about parallel crawler. Issues and challenges of parallel crawler are defined in this paper. In this paper, we can read about the crawling modes. This paper makes a study about architecture of parallel crawler and their performance. This paper addresses shortcomings of parallel crawler. To overcome these issues related to parallel crawler, a new approach is introduced i.e. Domain Specific Parallel Crawler. A new approach is introduced i.e. Domain specific crawler by Nidhi Tyagi et.al. [6]. In which multiple crawling machines are arranged in parallel that download web pages related to different domains. It enhances overall crawling speed and improves scalability, reliability of the system. It divides the web into specific domains based on domain specific crawling in which one crawl process can crawl pages from its domain only. Joo yong lee et. al. [4] presents SCRAWLER i.e. seed by seed parallel web crawler which shows flow process of indexing algorithm. This architecture writes about seed queue which is a collection of seed URLs. In web crawler, it starts with a URL, download its web pages parse it and extract links from web pages and add them to seed queue. It presents various strategies and design choices for a parallel web crawler. It describes crawler's architecture and implementation techniques. This paper writes about seed by seed crawl. Its main components are URL distributor, URL collector, downloader, link extractor, URL analyzer, seed context, web page files and how average no. of pages downloaded per second per thread changes as the number of crawling m/c increase.
More recently Sonali gupta et.al. [2] Proposed a web partitioning framework for parallel crawler. This paper explains how to distribute URLs from seed-URL queue to multiple crawling machines and each crawling machine has its own URL queue of its specific domain. It overcomes communication bandwidth between crawling machine because each will crawl web page of its specific domain without making communication between crawling machines. I this paper, domain oriented partitioning approach is used in which crawling process threads are added by breaking a domain into sub-domains. Rajender nath et. al. [5] proposed a parallel domain focused crawler for reduction in load on the network. It downloads only changed pages and ignores those pages that are not changed. In this paper, author experimentally proves that this implementation reduce load on the network up to 40%. In this crawler, concept of ASCII count is used to update the changed pages. This proposed work results in bandwidth preservation, saving memory and reduction of load on network. A scalable architecture of parallel crawler is proposed by A.K. Sharma et. al. [10] this novel approach aims maximum download rate and minimize overhead by efficiently and effectively crawl the current set of indexed web pages. This paper presents partitioning of system into two major components crawling system and mapping application. Duen Horng Chau et. al. [7] presents centralized queue framework of parallel crawler for online social networks. Crawlers do not get affected by failing of one crawler. In this framework a centralized queue is implemented as database table which is used to coordinate the operation of crawlers to prevent redundant crawling. Divakar Yadav et. al. [8] propose architecture of parallel crawler and web page change detection. For page refreshment, it uses three-step algorithm. This architecture checks whether web page structure or its image has been changed or not if text content is altered then it update it in its database. This proposed architecture maximize the overall performance of crawler, save memory and reduce overhead of bandwidth. This paper propose page update algorithm which address the issue of overlap of web page downloads by client crawlers and update changes or new web pages only. To take a review about web crawler and its working, a review paper on web crawler is written by Md. Abu Kauser et. al. [3]. This paper gives review of working of web crawler and its indexing algorithm by discussing a flow process. This paper discusses three crawling techniques, architecture of parallel crawler etc. Review paper's major objective is to throw light of the working of web crawling body.

VII. Conclusion

Crawlers are used to collect more and more web data for search engine. Every day thousands of web pages added or updated on internet. To crawl more data or to crawl all information available on the internet parallel crawler has been implemented. This paper throw light on the working of single process crawler, architecture of parallel crawler, what challenging issues can be find out while working on parallel crawler. This paper study about how to overcome challenges of parallel crawler and novel approaches which helps to overcome limitations of parallel crawler. This paper also discusses various researches related to parallel web crawler

References

[1] Sonali Gupta, Komal Bhatia and Pikakshi Manchanda, "Web ParF: A web partitioning framework for parallel crawler", International Journal on Computer Science and Engineering, Aug 2013.
[2] Junghoo Cho and Hector Garcia-Molina, "Parallel Crawler" proceedings of www2002, Honolulu, Hawaii, USA, May 7-11, 2002. ACM 1-58113-449-5/02/005.
[3] Md. Abu Kausar, V.S.Dhaka and Sanjeev Kumar Singh, "Web Crawler: A Review", International Journal of Computer Applications" (0975-8887), Feb 2013
[4] Joo Yong Lee, Sang Ho Lee and Yanggon Kim, "Scrawler: A seed-by-seed parallel web crawler" INSTICC Press, (2007).
[5] Rajender Nath and Naresh Kumar, "A novel domain focused crawler for reduction in load on the network", International Journal of Computational Engineering Research Vol. 2 Nov.

2012.

[6] Nidhi Tyagi and Deepti Gupta, "A novel architecture for domain specific parallel crawler", Indian journal of Computer Science and Engineering, June 2010.

[7] Duen Horng Chau, Shashank Pandit, Samuel Wang and Christos Faloustsos, "Parallel Crawling for online social networks", Proceedings of WWW 2007, May 8-12, 2007, Banff, Alberta, Canada ACM 978-1-59593-654-7/07/0005.

[8] Divakar Yadav, AK Sharma and J.P. Gupta, "Parallel crawler architecture and webpage change detection", Wseas Transactions on Computers Vol. 7, July 2008.

[9] www.worldwidewebsize.com.

[10] Shruti Sharma, AK Sharma and J.P. Gupta, "A Novel architecture of a parallel web crawler", International Journal of Computer Applications (0975-8887), January 2011.