Web Crawler Challenges and Their Solutions

Dr. Naresh Kumar[1]
[1]Assistant Professor, CSE Department, MSIT, New Delhi.
Email: narsumsaini@gmail.com

Shivank Awasthi[2], **Devvrat Tyagi**[2]
[2]B.Tech. Student, CSE Department, MSIT, New Delhi
Email: shivankawasthi.cse.msit@gmail.com , tyagidevvrat@hotmail.com

ABSTRACT

In recent times, web has become an indispensable tool in daily life. The enormous growth of web has become a major concern both for end user and search engine itself. Due to this exponential growth, general crawling technique is not able to handle all relevant URLs. So this paper proposes a detailed analysis on different crawling technologies focusing on specific need of end user.

Keywords: Search Engine, Web Crawler, Downloader, Frontier Queue, Indexer, Repository.

1. INTRODUCTION

WWW has now become one of the most important sources of information in present era. Size of the web is increasing exponentially. Currently the size of indexed web repository is around 4.64 billion pages [1]. The information from this repository is retrieve by various search engines. These search engines have automated program called crawlers to do the task of extracting the link from the web page and recursively travelling each page.

Due to increasing size of the internet general crawler works inefficiently as it downloads irrelevant information in bulk. So, it is necessary to make crawler focusing on particular interest of the user.

In this paper, an extensive survey on existing search techniques has been done. This paper has surveyed and analyzed large amount of research work done by different authors on different crawling techniques. Many different models of web crawler such as deep bot [6], internet forum crawler [9], wrapper model of crawler [4], catch crawler [5] etc. has been discussed to focus on specific need of the user. Moreover, some problems and their possible solutions are also discussed in this paper.

2. RELATED WORK

A continuous crawling process was proposed in [2] to reduce the traffic on the network. Obsolete pages available in the repository were further prevented by a three step algorithm where crawling and downloading is done at client side. An algorithm is proposed by author, that converts the image to bitmap first and then stores an 'ivalue' for each image which changes at every page update. Authors have implemented proposed approach on the content of MSNBC web page. The experimental results show the detection of the addition of six words in the page.

To access the hidden web information of the web, focused crawler called "deep bot" is proposed in [6]. It accesses the websites which offer query forms. It executes the fields of the form by supplying query (attributes, values) and by using techniques such as visual distance and text similarity heuristics. Relevance of the form is determined by the inequality $\sum (Ci*Si) > \mu$ where (Si) is specificity index, (Ci) is Confidence of an assignment and (μ) is relevance threshold. To quantify results author uses standard Information retrieval metrics: precision, recall and F-measure. Obtained results are quite promising as all the metrics show high value and some of them even reach 100%.

The page update detection and compression techniques along with crawling methods were proposed in [11]. Authors surveyed to find the most efficient compression utility and then proposed the following algorithm to calculate the load.

$$LOAD = (Pt*Av)\ (1-Cm/100) + Wc$$

Experimental results have shown that the compression rate for data set chosen from the site of British Broadcasting Corporation is about 74.5%.

Authors in [8] address the problem of fetching relevant pages by using a framework LS CRAWLER which checks the relevant keywords in the link & its surrounding link using web ontology language. The relevancy is checked by comparing the semantic similarity in between the taxonomy hierarchy of particular domain and the link. Authors have shown a comparison between the traditional full text indexed search and link semantic search. The achieved Results verify the increased efficiency of proposed work.
A wrapper model of crawler is proposed in [4] to collect the reviews of the product from shopping malls websites. Wrappers sort the review and separate them with the other page contents by going through four analysis steps namely: (1) Category analysis (2) Product list analysis (3) Review list analysis (4) Review extraction. Category DB accurately maintains the position of target data in the form of HTML tag and Store Data collects the extracted review data in review table.

An approach named BINGO was proposed in [10] which detect seed pages from user's bookmarks which serve as the hierarchical ontology and initial training data set for classifier. The training data set gets initialized after every frequent successful crawl by using KLEINBERG'S HITS ALGORITHM and confidence of classifier. Authors implemented BINGO on a medium sized collection of HTML documents. Proposed work highlighted influence of feature selection and periodic retraining.

A CATCH CRAWLER was proposed in [5] which extract useful data from style sheets of web page. The crawler works by searching for the same class in style sheet of web page in which the user is interested in. The author has supported his approach with experimental results that have shown an accuracy of about 90%.

Internet forum crawler based on vertical crawling is proposed in [9] to extract the information from the structured web pages by supplying appropriate templates. It divides the forums into four broad categories based on different processing methods and designed the templates for each category namely pattern1-pattren4 and supplies the template according to the category of the forum.
Another unique model of web crawler in [7] is proposed with embedded private information protection system. Here author uses an Enterprise Management System crawler having Collection Module is use to reach the web server and collects the web pages with web documents (including attached files). These attached files are filtered out by Transformation module which parses their contents and extract their URLs. These web pages are saved as normalized documents (ND) in the form of tag structure. ND are analyzed via Analysis Module. A knowledge summary indexing engine summarizes all web pages and maintains its local index.

3. WEB CRAWLER ARCHITECTURE

General web crawler architecture [3] shown in Fig.1 Web crawlers traverse each URL on WWW and then download web pages for search engine and indexes those web pages for future searching. Crawler needs to revisit the pages to refresh the repository. Seed URLs are needed to begin the crawling process. Links on seed URLs are extracted and traversed recursively. Crawl frontier queue contains the hyperlink to be visited.

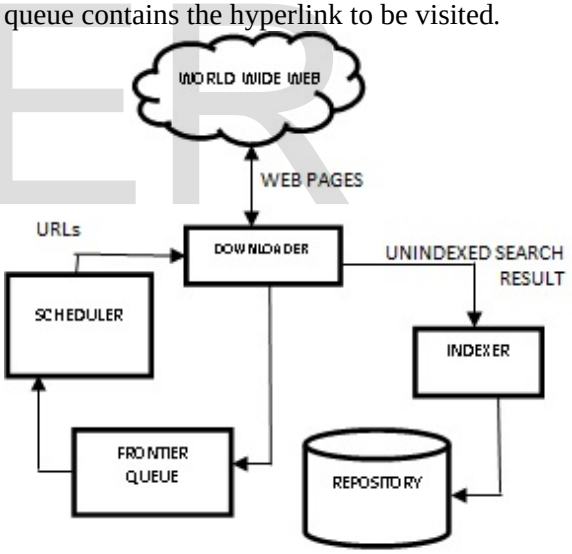

Fig.1:- Web crawler architecture

Functioning of a web crawler:
1) Initialize with seed URL(s).
2) Send it to the Frontier Queue, scheduler schedules the URL to be downloaded and gives it to the downloader.
3) Downloader parses the web pages to find new URL links and places them into the frontier. It then feeds them to the indexer.
4) Indexer indexes the links and settled them in the central repository.

Repeat step (2) while the crawl frontier queue is not empty.

4. CHALLENGES IN CRAWLING TECHNOLOGIES

After surveying these works, here are some of the identified challenges that are still present in these technologies. These are summarized as follows:

- Page updating policy of [2] may lead to re - downloading of pages with every minute change in the structure of page which might not be perceived by user leading to wastage of bandwidth. Image processing takes place with complex calculations leading to delay in processing time and increases load on network due to these complex computations. Failure of server may cause failure of entire system due to absence of distributed server system.
- The HITS algorithm in [10] is obsolete nowadays and hence efficiency of results can be improved by using an efficient algorithm. Authors have not specified any procedure for dealing with pages other than that of HTML type, which may lead to loss of valuable information. Moreover, there is a great possibility that the bookmarks of user may belong to absolutely different domains leading to a state of ambiguity for crawler.

- The domain of approach [5] suggested by authors is limited to highly structured or semi-structured websites. There are chances of huge data loss if data of interest to user, lies in different CSS classes. In case of websites that completely rely on HTML or other popular web technologies, proposed approach may not work.

- The compression utility selected by the authors in [11] is done merely by surveying the compression results obtained from different compression utilities; this utility may become obsolete with time. The processing time required by such types of techniques and utilities may cause delay in the process of searching and querying. Sometimes there is a possibility of data loss during data compression.

- In [8], it is suggested that detection of seeds should be from search engine to get relevant seeds that are specific to domain of query. But this also increases the possibility of adding of non-relevant or sponsored links that are displayed as top results in a search engine but are not of much relevance. These seeds may disturb the crawling process to a large extent.

- Query forms having very few fields (limited to 3) may not reach the threshold value even if correct query is supplied in the field [6]. Sometimes alias for an attribute in the query form may not match with alias defined in domain.

- Wrapper model of crawler [4] will visit many other links regardless of their content in order to collect the review data. Once the reviews for a particular product is collected there is no method suggested to update the list of reviews.

- Crawler proposed in [7] is mainly specific to structured web pages (news and blog sites) it will not work efficiently on non-structured web pages. Experimental results show 100 threads as the threshold value. Performance degrades when thread value exceeds threshold value.

- EMS (Enterprise Management System) crawler [9] is limited to corporate websites only and there is no algorithm suggested to delete or block the exposure of private information gathered from the corporate websites.

5. SOLUTIONS

To address the above said problems, authors also suggest some desirable solutions which are described below:

- As suggested by authors [2] and [11], page update is detected by calculating the sum of ASCII value of all the characters in the web page, but this technique may waste resources in case if very minute changes occur in pages i.e. changing of a punctuation character will

lead the crawler to re- download the same page which may not be of much significance to user. In order to prevent this wastage of resources, a threshold value can be set for the sum of ASCII count that prevents the crawler from downloading the same page for a minute change. Threshold values can be computed simply taking an average of ASCII value of common punctuation symbols; the change in sum of ASCII count less than that of threshold value should simply be discarded.

- An approach proposed in [2] to detect any change in image during page update policy, the proposed approach ends up with complex calculations leading to delay in processing a query. This work suggests an approach for image change detection, the approach is as follows:

  (i) Every image is made up of pixel and every pixel is made up of RGB components i.e. Red, Green, Blue components.
  (ii) Take the sum of RGB value of a pixel and calculate its average, let it be denoted by pavg.
  (iii) Now take the sum of every pixel's pavg and take its average, let it be denoted by iavg.
  (iv) Compare this value with the calculated value at the time of change detection; any change in image will change the value of iavg.

- Amendment should be done in the relevance threshold μ in [6] such that form with very few fields should not get discarded despite of matching the fields correctly with domain defined attribute.

- Page revisit policy should be included in wrapper model of crawler [4] to update the review list of the product such that the user should get the updated reviews

6. CONCLUSION

The crawlers are being used to collect and update the central repository of the search engine. But it fails somewhere to do its task effectively. So in this paper authors try to identify these challenges in the field of parallel crawler, focused crawler, parallel focused crawler which are based on the parameters namely -Distribution, Scalability, Load balancing and reliability. Furthermore, authors also suggest the possible solution to these problems.

REFERENCES

[1] URL: www.worldwidewebsize.com

[2] Divakar Yadav et al., "Architecture for Parallel Crawling and Algorithm for Change Detection in Web Pages", Published in IEEE ,10th International conference on information technology, pp. (258-264),0-7695-3068-0/07.

[3] Dr.Rajendra nath and khyati chopra , "Web Crawlers : Taxonomy, Issues and Challenges, published in International Journal of advanced research in computer science and software engineering Volume 3 ,issue 4 April 2013,ISSN: 2277 128X

[4] Hanhoon Kang et al., "Modeling web crawler wrappers to collect user reviews on shopping mall with various hierarchical tree structures". In IEEE, 2009 International Conference on web information systems and mining,978-0-7695-3817-4/09.

[5] Kwangcheol Shin and Geun Sik Jo, **"**Catch Crawler: Automatic Web Information Extractor using Style sheet", 2008, Published in IEEE International workshop on semantic computing and applications, pp.(99-102),978-0-7695-3317-9/08.

[6] Manuel Alvarez et al., "Deep Bot: A focused Crawler for Accessing Hidden Web Content", Published in DEECS 2007: June 12, 2007, San Diego, California, USA. ACM 978-1-59593-856-5/07/06

[7] Myung sil choi et al., "Private information protection system with web crawler". in IEEE International Conference on wireless and mobile

computing, Networking and communication,978-0-7695-3393-3/08.

[8] M.Yuvarani, et al.: "LSCrawler: A Framework for an Enhanced Focused Web Crawler based on Link Semantics", in Proceedings of the 2006 IEEE/WIC/ACM International Conference On Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06) ,0-7695-2747-7/06.

[9] Qing Gao et al., "A High-Precision Forum Crawler based on Vertical Crawling", in IEEE, Proceedings of IC-NIDC 2009. 978-1-4244-4900-2/09.

[10] Sergej Sizov, et al.,"The BINGO! Focused Crawler: From Bookmarks to Archetypes", in IEEE, Proceedings of the 18th International Conference on Data Engineering (ICDE.02) 1063-6382 / 02.

[11] Sneha Tuteja et al., "Reduction of Load on Network using Compression Technique with Web Crawler", Published in Advances in Computer Science and Information Technology (ACSIT),Print ISSN: 2393-9907; Online ISSN: 2393-9915; Volume 2, Number 3; January-March, 2015 pp. (200-203).