Proceedings of the 12th INDIACom; INDIACom-2018; IEEE Conference ID: 42835
2018 5th International Conference on "Computing for Sustainable Global Development", 14th - 16th March, 2018
Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM), New Delhi (INDIA)

Survey on Crawling Techniques

Naresh Kumar
Associate Professor, Dept. of Computer Science & Engineering
MSIT
New Delhi

Devvrat Tyagi
Student, Dept. of Computer Science & Engineering
MSIT
New Delhi

Shivank Awasthi
Student, Dept. of Computer Science & Engineering
MSIT
New Delhi

***Abstract*--- In present era, web has become one of the basic necessities to mankind. In order to get relevant results for a search query, the need arises of a tool called web crawler. Due to extensive growth in the number of web pages, user experience is deteriorating day by day. The existing crawling technologies are not robust enough to handle such huge data. This paper analyzes different crawling mechanisms and also discusses the problem associated with them. Authors have also provided detailed solutions of problems discussed.**

Keywords—web crawler; focused crawler; parallel crawler

I. INTRODUCTION

We ask that authors followWeb crawler or web spider is a program that collects documents from World Wide Web. It begins the process with seed URLs. It downloads the seed page from WWW and traverses the entire document to extract further links. These links are then stored in the crawl frontier and passed to downloader one at a time to download more pages. These pages are then stored in the repository in indexed fashion. Storing in repository can be in compressed form.

The current size of web is 4.74 billion indexed web pages (http://www.worldwidewebsize.com).The growing size of web with time has degraded user search experience significantly. This has led to origin of two broad methodologies for web crawling: focused crawler [1],[2],[4],[5],[6],[7], [8],[9],[10] and parallel crawler [7],[11],[12],[13],[14]. Authors surveyed these domains of web crawling and crawlers with page revisit policies [15], [16],[17], [18],[19]] to check freshness of web page at regular intervals. This survey was conducted from 2015 to 2016 on papers published during 2002 to 2016 and about 40 works of different categories were surveyed. But due to space restrictions authors have presented their work in tabular form.

This paper is further categorized into seven sections. Section 2 comprises of focused crawling system. Section 3 discusses about parallel crawling system. Section 4 is about page revisit policies. Section 5 consists of proposed solutions. Section 6 and 7 is the conclusion and future work respectively.

II. FOCUSED CRAWLING SYSTEM

Focused crawling systems are those web crawlers that fetch only those pages relevant to their search domain. These types of systems are rapidly gaining popularity with the increase in number of web pages on World Wide Web. Focused crawlers actually check the relevance of web page before crawling it by using any of the possible algorithms (semantic checking, anchor text checking).The architecture and basic working of web crawler is summarized in the next section.

Architecture and Working

The working of focused crawling architecture as shown in fig 1is as follows:

1. Seed URLs are firstly initialized and transferred in the initial set which is then passed to trainer.
2. Trainer then uses topic taxonomy to train the classifier which pages to select and which of them to reject.
3. Downloader fetches URLs from URL queue one at a time, which is then used to download the page from WWW. Downloader then passes a copy of downloaded page to link extractor and a compressed copy to indexer.
4. Link extractor traverses the entire downloaded document and finds all URLs in the document and again passes them to classifier.
5. Indexer indexes all the documents being added to repository to facilitate future searches and stores them in the repository in the compressed form.

Literature Review

Authors have surveyed research papers listed in table 1 on focused crawlers. This table contains the proposed techniques in these works and the problems related to these techniques.

 ISSN 0973-7529; ISBN 978-93-80544-28-1

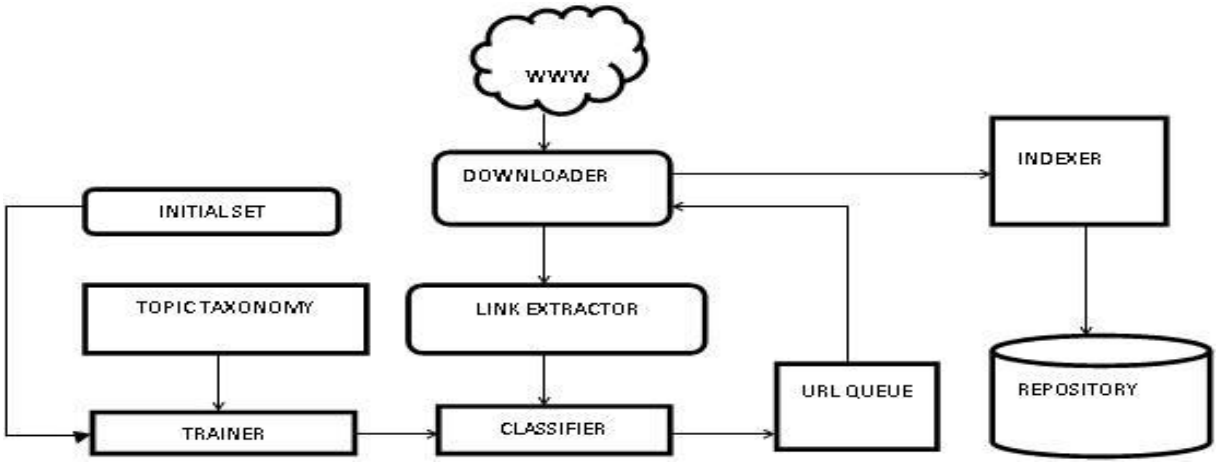

Fig. 1. Focused Crawling System

TABLE I. REVIEW OF FOCUSED CRAWLER LITERATURE

Ref no.	Proposed technique	Problems identified
[1]	This paper predicts the topical focus of an unvisited webpage by using Dewey decimal classification and assigns priority score to the unvisited URLs in that page. Priority score of unvisited URL is (1/level number) based on the similarity of its anchor text compared with the node present at certain level of T-graph. But the text similarity exceeds OSM threshold value of 0.05.	If the word has several meanings then there is no algorithm proposed to determine its exact sense which sometimes might results in false prediction of topic of an unvisited web page.
[2]	Context Driven Focused Crawler with inbuilt GUI in search engine has been proposed. It uses Augmented hypertext documents which supports.TVI extension to store various tags and their content. User can select the topic with related example and its context through category tree. Crawling is assigned to three agents as: 1. User agent 2.Matcher agent 3.Dbase agent with specific crawling responsibilities.	User can ask query only in specific format given in the category tree. Web pages which need human interactions cannot be downloaded.
[3]	Wrapper model is proposed which collect the reviews about the product from shopping malls websites. Process goes through four analysis steps namely: (1) Category analysis (2) Product list analysis (3) Review list analysis (4) Review extraction to sort the review and discard other page contents. Category db maintains the position of target data in the form of HTML tag and Store Data collects the extracted reviews in review table.	All the links extracted are visited regardless of the fact that they contain review data or not as classifier filters review pages after this step. No algorithm proposed to update the collection of reviews.
[4]	Author proposes Internet forum crawler based on vertical crawling technique. Forums have been divided into four broad categories based on different processing methods and accordingly templates have been designed for each category namely pattern1-pattren4. Based on the category of the forum exact pattern template have been supplied to extract the information from these structured web pages.	Crawler works inefficiently on non-structured web pages as it is mainly specific to structured web pages (news and blog sites). Performance degrades when thread value exceeds 100 threads.
[5]	Deep Bot is proposed to access the hidden web content. It accesses the websites which offer query forms. It executes the fields of the form by supplying query (attributes, values) 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.	Query forms having very few fields (limited to 3) may not reach the threshold value even if correct query is supplied in the field. Sometimes alias for an attribute in the query form may not match with alias defined in domain.
[6]	A framework named LS Crawler has been proposed in that checks the relevance of documents by keywords in the link and text surrounding link. OWL is used for checking the document relevance. System begins its working by fetching a relevant ontology from repository for a submitted query.	With seed detection from search engines, some sponsored links may show in leading to irrelevant pages. There is no specified provision to deal with case when repository don't have page for submitted query.
[7]	Use of migrating agents has been proposed to improve crawling results by returning relevant pages to crawler manager for storing it. Relevancy of page is calculated by formula: $PageRelScore = 0.2 * URLText + 0.4 * MetaText + 0.1 * HeadText + 0.3 * BodyText$ Author also has proposed an image based crawler that firstly selects a collection of image to search and then applies image segmentation techniques to it.	Calculating relevance of a page on the basis of user visits cannot be accurate. Proposed approach decreases the relevance of pages containing advertisement images. The formula for checking relevance may also vary from website to website depending on layout.

[8]	A domain specific crawler that groups related web pages together has been proposed. The dataset chosen by authors are pages related to Thailand and the proposed approach contains segment predictor and segment classifier.	Fixing the optimal segment threshold and distance actually stops the crawler to crawl website which may lead to data loss. Harvest rate is quite low while querying for a website segment.
[9]	The proposed crawler tries to fetch relevant pages first. Relevance of pages is calculated by using association metric by text classification algorithms in combination with semantic similarity. Experiments performed shows consistency with proposed algorithm.	Reordering of URL queue repeatedly to fetch important pages increases crawling time.
[10]	A focused crawler on Brazilian web named Co Web has been proposed that aims to crawl .br domain websites with operational and ethical limits. Page freshness is maintained through cache proxy servers. MD5 algorithm is used to improve crawling coverage.	Absence of URL reordering leads to crawling inappropriate pages. Failure of central scheduler leads to failure of system.

III. PARALLEL CRAWLING SYSTEM

As the size of web is increasing rapidly, it becomes imperative to implement a technique which will overcome the failing of conventional crawler. So, parallelization of crawler processes becomes necessary to retrieve maximum content of the web in short interval of time. This type of crawling technique is called parallel crawling.

Parallel crawler has many advantages over conventional crawler technique such as it increases the coverage area of the web, it can disperse the network load by running parallel crawler processes at different geographical locations. Above all it also has certain disadvantages like overlap issues, communication bandwidth overhead etc.

A. Architecture and working

The working of parallel crawler architecture is as follows in fig 2:

1. Process starts when initial set of URLs are placed into URL dispatcher.
2. URL dispatcher sends these URL to central coordinator which downloads pages from these URLs and its link extractor extracts the URLs from these pages.
3. Central coordinator sends distributes these collected URLs to various crawling processes running in parallel on the basis of link prioritizing algorithm.
4. Downloader of each crawler process downloads their link based pages and Link extractor extracts the link of downloaded pages.
5. Downloaded content of each crawl process goes to central repository and extracted links are sent to URL dispatcher to repeat the above process until crawler decides to stop.

B. Literature Review

Author analyzed parallel crawling techniques discussed in various research papers. Review table 2 is shown below which contains the highlighting features, algorithm and some identified problems.

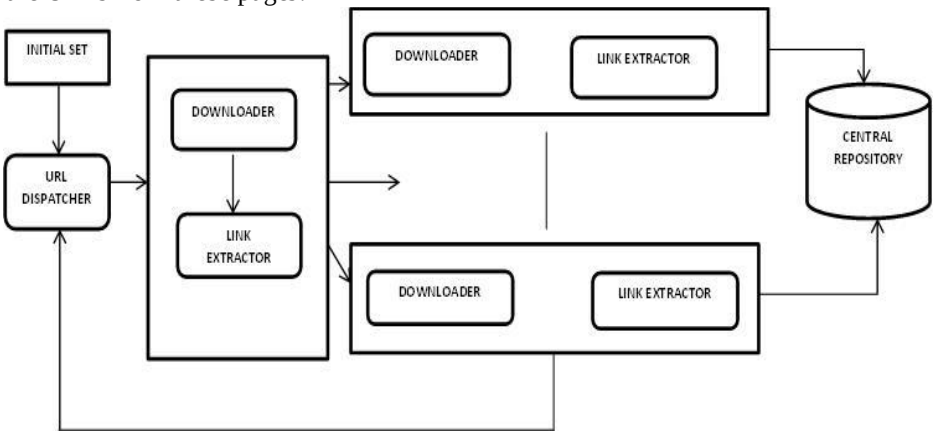

Fig. 2. Parallel Crawling System

TABLE II. REVIEW OF PARALLEL CRAWLER LITERATURE

Ref No.	Proposed Technique	Problems identified
[7]	Focused-based parallel crawler has been proposed. Page importance is calculated as: $I(dj) = \alpha * E * okapi(dj * c*) + \beta * LRdj$ which isbased on clickstream analysis coupled with the technique of Okapi for text analysis where α and β are normalization factors, E is the emphasis factor and $LR_{dj}$is log ranking based on click stream analysis. In order to minimize communication overhead, role of central coordinator is assigned to coordinator section present in each parallel agent.	Notification overhead is computed as: $n * m * (m-1) * L$ where m is the number of parallel agents and L is the number of frontier divisions. So, increasing the number of parallel agents abnormally increases the notification.
[11]	Mobile phone crawler has been discussed having Mobile content filter used for distinguish the mobile content on the certain basis. History database manages the history of the crawled pages while Common boards are used to increase the multiplicity by establishing single http session at a time. To optimize intrusive collection, a minimum server access interval has been set. Actual collecting speed comes out to be 5000 pages per hour.	There are chances of duplication of URLs while extraction process because sometimes two different web pages access by different common board in charge can have same forward link.
[12]	An algorithm for parallel crawling and a three-step algorithm for checking freshness of page have been proposed. Freshness of page is checked for a change in an image or a change in text. The algorithm proposed, sends URL to client for crawling to avoid additional load on server. Proposed work is supported by experimental work that detects minute changes in the text.	Minute and insignificant changes may lead to re-downloading of page which wastes the bandwidth of network. Detection of change in image leads to highly complex calculations.
[13]	A distributed parallel crawling approach has been proposed by authors. This approach tries to eliminate the problem of failure of center controller. Approach was tested in a cluster environment with one as master node and others as compute node which showed an average crawling rate of 25 pages per second.	Use of database for URL is not possible if access is not provided. Load calculation formula is not justified by any arguments.
[14]	An application named CRAYSE has been proposed that performs searching and crawling in parallel, solving the problem of high storage requirement to store crawled and downloaded pages. Text searching is achieved through KMP algorithm. Files in pdf format are first downloaded and then searched.	Proposed approach increases load on network as crawling will happen continuously.

IV. PARALLEL CRAWLING SYSTEM

Web crawler downloads bulk of pages in short interval of time and store in its database. After some span of time database contents gets outdated because most of the pages gets changed or updated. So, web crawler revisits these URLs in order to keep the freshness of database as high as possible.

LiteratureReview

After analyzing various papers on revisit policy, author reviewed various methodologies and algorithm implemented in these papers. Some flaws have also been detected which help in improving and making these technologies better in future. Authors have provided a review of these papers in table 3.

TABLE III. REVIEW OF PAGE REVISIT CRAWLER LITERATURE

Ref. No.	Proposed Technique	Problems identified
[15]	Parallel domain focused crawler is proposed in this paper which work on the concepts of mobile agents. Crawler is send to remote site where they checked for modification in the page. Crawler hand compares the ASCII count for that page, if the value is same as previously stored in database it rejects the page otherwise it sends the page to search engine after compression.	Web pages having high multimedia content contain maximum information in the form of images. So, any change in the images cannot be detected by comparing ASCII count of that page.
[16]	Revisit policy based upon the frequency updation is proposed. Refresh rate is computed by – $tn + 1 = tn + \Delta t$ where : t_n is present refresh time of that site,t_{n+1} is adjusted refresh time and Δt is change in the refresh time calculated dynamically. URLs are distributed among several UrlQ(1…n) queues refreshing time wise but they are stored page rank wise. URL from each UrlQ(1…n) is selected at each n^2 *20μ sec. Crawl manager reads URL from these UrlQ(n) .These URLs are assigned to crawl workers for downloading and later it updates the database.	There are high chances of bandwidth wastage as URLs which are visited very frequently might not have high page rank relative to those URLs in another URL pool. This also results in increase of web traffic.

[17]	Algorithm for the page revisit policy has been proposed to keep the database update. Page score of URL decides which URL have to visit first and it is calculated as: Page score=Page rank/Age where Page rank is popularity of that URL based on link analysis and age of page is: F(ag)=Present date time-last modified date time. Pareto's principle of 80-20 rule is applied in which 20% pages with high page score are kept in B collection and rest 80 in A collection. Thirty websites has been analyzed and observed that sites in B collection have high updation rate, they changes in 2-3 days. So, crawler revisits sites in B collection with higher priority.	When number of websites increases then the difference between the page score of bottom URLs in B collection to the top URLS in A collection becomes insignificant So it will become hard for crawler to revisit any URL on the basis of Pareto's principle of 80-20 division criteria.
[18]	The proposed work in suggests some changes in the existing algorithms to update the crawled pages. Author proposes to incorporate compression techniques to store pages in repository in compressed form. Author has surveyed to figure out the utility with maximum compression ratio and also proposed an algorithm to calculate load. Compression utility used has given an efficiency of 74.5%	Use of compression techniques may lead to loss of sensitive data. Compression also increases the actual crawling time. Continuous up gradation of technology is must with the advent of new compression techniques.
[19]	The paper proposes a page update policy to solve the problem of energy consumption and carbon footprints of servers by reducing the number of requests to web servers. An optimal policy has been devised here based on instantaneous value of page staleness and greenness indicators.	Proposed approach leads to complex calculations that may lead to delay in crawling and increased load on network. The approach proposed here is applicable to general crawlers that crawl millions of pages and not of significance to focused, topical crawler.

V. SOLUTIONS

Authors have proposed solutions to the above discussed problems which provide some new research directions.

A. *Focused Crawler*

- Algorithm for page revisit should be implemented in wrapper model [3] such that whenever the new reviews are entered the user should get the updated reviews.
- Amendment should be done in the relevance threshold μ in [5] such that form with very few fields should not get discarded despite of matching the fields correctly with domain defined attribute.
- The approach proposed in [6] can be improved by also considering the case when repository does not have a page for a user query. Such queries must be stored in a separate module and must be checked periodically, so that any similar query in future is not unanswered. Sponsored Links must also be filtered by a module by checking every result from search engine with respect to user query.
- PageRelScore proposed in [7] should be modified such that major portion of relevance is from the text body. Meta text is a voluntary tag and web pages may have relevant meta text but body text might not be that relevant to the user.
- Absence of URL reordering or inefficient URL reordering with respect to time are found in [9],[10]. This can be enhanced by using some ordering metric that is efficient in time. Moreover, the data structure queue that is being used for reordering can also be replaced by other data structure (eg. Hash maps, Linked list) that supports random access.

B. *Parallel crawler*

- ASCII count in [12] should be replaced by mean of ASCII value of all characters to nearest integer. This method will also overlook some of the very minute changes in text that would otherwise be wasting the system's bandwidth. The technique proposed for freshness of image can be simplified by the average of RGB values of each pixel and then finally computing mean of all pixels RGB mean.
- Crawled and downloaded pages in [14] can be compressed before storing them in the repository. Moreover, crawling must be scheduled periodically so that a tradeoff exists between the freshness of repository and bandwidth consumption of system. PDF files must be downloaded by client's bandwidth to prevent the load on the system.

C. *Page Revisit Policy*

- Categorization of URLs in [16] on the basis of frequency of revisiting should be implemented by combining refresh rate and Page rank of all URLs together such that the page with high frequency must not visited repeatedly as it may have less page rank relative to those URLs which are present in low frequency UrlQ pool.
- Compression utility in [18] must be applied only to images, videos as no loss of sensitive data is feasible. Moreover, the technique used for data compression must be efficient enough such that it is not the bottleneck of system performance.
- Instead of reducing requests to web servers in [19], number of web servers should be increased such that the load is balanced such that no web server consumes more energy than the threshold. Moreover, calculations must be done prior to crawling so that faster results are ensured.

VI. CONCLUSION

Different Web Crawlers with different approaches are used to face challenges like reduction of load on network, bandwidth preservation, and page change detection etc. have been studied and presented in this paper. The authors have also explained the problems identified in these papers. This may use the concept of priority queue and hashing etc. The

expected outcome of chosen approach may result in increase in crawling efficiency and efficient utilization of bandwidth. Several other factors may be considered before finalizing and implementing the chosen approach.

Proposed crawling approaches are not enough to resolve the drawbacks as identified earlier in this paper. Hence, authors propose to work in two different domains i.e. priority resolution and page revisit policy.

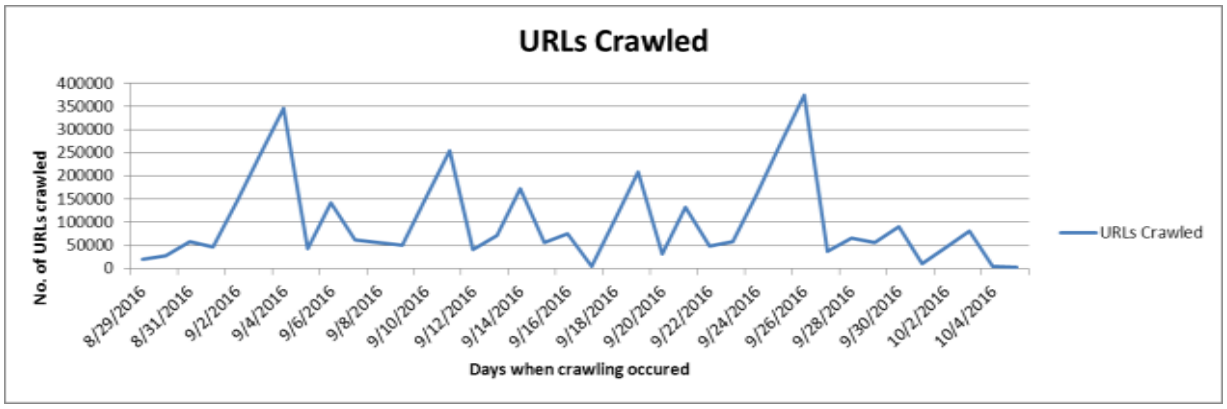

Fig. 3. Performance of Efficient Crawler

- An updated page revisit policy is proposed that checks the freshness of a web page by detecting any change in text of web page, images of web page and style of web page. This three-layered approach gave a significant improvement of 84% in maintaining the freshness of repository.
- Web crawler named Efficient Crawler is proposed which associates a priority with each web page before downloading. This improves the system's efficiency as important pages are downloaded first. The approach replaces traditional URL queue with a linked list, this improves overall efficiency significantly as reordering of queue is prevented. Fig 3. depicts performance of Efficient Crawler in graphical from. It shows number of URLs crawled in a single day. This approach gave an improvement of 70.04% with respect to [10]. But the implementation is still under process to make the system robust enough for large seed pages. Authors are currently working on these approaches to make further improvements.

REFERENCES

[1] http://www.worldwidewebsize.com date:11/11/2016

[2] A. Sefyi, A. Patel, &J.C. Junior, (2015),"Empirical evaluation of link and content-based Focused Treasure Crawler". Computer Standards & Interfaces, 44(2016) 54-62. http://dx.doi.org/10.1016/j.csi.2015.09.007

[3] N. Chauhan, &A.K. Sharma, (2009),"Design of an Agent Based Context Driven Focused Crawler". BVICAM's International Journal of Information Technology, 1(1), ISSN 0973-5658.

[4] H. Kang, S.J. Yoo&D. Han (2009),"Modeling Web Crawler Wrappers to Collect User Reviews on Shopping Mall with various Hierarchical TreeStrucure",International Conference on Web Information Systems and Mining, IEEE, 978-0-7695-3817-4/09.

[5] Q. Gao, B. Xiao, Z. Lin, X. Chen &B. Zhou (2009),"A High-Precision Forum Crawler based on Vertical Crawling", IC-NIDC, IEEE,978-1-4244-4900-2/09.

[6] M. Alvarez, J. Raposo, A. Pan, F. Cacheda, F. Bellas, &V. Carneiro(2007),"DeepBot: A Focused Crawler for accessing hidden web content" DEECS 2007:June 12, 2007. San Diego, California, USA, ACM 978-1-59593-856-5/07/06.

[7] M. Yuvarani, N. Iyengar, S.N., Kannan (2006),"LS Crawler: A framework for an enhanced focused web crawler based on link semantics", IEEE/WIC/ACM International Conference on Web Intelligence, IEEE, 0-7695-2747-7/06.

[8] A. Tailang, (2016),"An effective approach for document crawling with usage pattern and image based crawling", International Journal of Computer Applications Technology and Research, pp. 49 - 55, ISSN:- 2319–8656.

[9] A. Rungsawang, T.Suebchua, B.Manaskasemsak, (2014),"Thai related Foreign language – Specific website segment crawler", 28th International Conference on Advanced Information Networking and Applications Workshops, IEEE, 978-1-4799-2652-7/14.

[10] S. Ganesh, M.Jayaraj, V. Kalyan, S. Murthy, G.Aghila, (2004): "Ontology-based Web Crawler. International Conference on Information Technology: Coding and Computing", IEEE, 0-7695-2108-8/04.

[11] A.S. da Silva, E.A.Veloso, P.B.Golgher, B.R.Neto, A.H.F. Laender, N.Ziviani, :"CoWeb- A Crawler for the Brazilian Web", Federal University of Minas Gerais , 312270-970, Belo Horizonte MG, Brazil.

[12] F.A. Abkenari, &A. Selamat, (2010),"A Clickstream-based Focused Trend Parallel Web Crawler",International Journal of Computer Applications (0975-8887), Vol 9(5).

[13] H. Takeno, M. Muto, N. Fujimoto &K. Hagihara (2006),"Developing a Web Crawler for Massive Mobile Search Service",7th International Conference on Mobile Data Management (MDM'06), IEEE, 0-7695-2526-1/06.

[14] D. Yadav, A.K. Sharma, J.P.Gupta ,N. Garg &A. Mahajan, (2007),"Architecture for Parallel Crawling and Algorithm for Change Detection in web pages",10th International Conference on Information Technology, IEEE, 0-7695-3068-0/07.

[15] M. Wu, J. Lai, (2010),"The research and implementation of parallel web crawler in cluster", 2010 International Conference on Computational and Information Sciences, IEEE, 978-0-7695-4270-6/10.

[16] V. Radhakrishnan, Y. Farook, S.Selvakumar(2010),"CRAYSE : Design and Implementation of Efficient Text Search Algorithm in a

Web Crawler", ACM SIGSOFT Software Engineering Notes, Page (1-8), DOI:10.1145/1811226.1811236.

[17] R. Nath &N. Kumar (2012),"A Novel Architecture for Parallel Domain Focused Crawler",.International Journal of Engineering Research and Applications (IJERA), 2(5), 266-269,ISSN: 2248-96.

[18] N. Singhal, A.Dixit&A.K. Sharma(2010),"Design of priority based frequency regulated Incremental Crawler",International Journal of Computer Applications(0975-8887), 1(1).

[19] V. Sharma, M. Kumar &R. Vig (2012),"A Hybrid Revisit Policy for Web Search",Journal of Advances In Information Technology, 3(1), DOI: 10.4304/jait.3.1.36-47.

[20] S. Tuteja, N. Kumar, R. Nath(2015),"Reduction of Load on Network using Compression Technique with Web Crawler",Advances in Computer Science and Information Technology (ACSIT), Print ISSN: 2393-9907; Online ISSN: 2393-9915, pp.200-203.

[21] V.Hatzi, B.B.Cambazoglu, L.Koutsopoulos, (2016),"Optimal Web Page Download Scheduling Policies for Green Web Crawling", IEEE Journal on Selected Areas in Communications , DOI: 10.1109/JSAC.2016.2520246.