Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 69 (2014) 492 – 499

Procedia Engineering

www.elsevier.com/locate/procedia

24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013

A Fast Distributed Focused-Web Crawling

Harry T Yani Achsan[a], Wahyu Catur Wibowo[b]*

[a]*Paramadina University, Jl. Gatot Subroto Kav. 97, Jakarta 12790, Indonesia*
[b]*University of Indonesia, Depok 16424, Indonesia*

Abstract

Mining data from a web database becomes more challenging in recent years due to the exploding size of data, the rising of dynamic web, and the increasing performance of web security. Mining data from a web database differs from mining data from web sites because it is intended to collect specific data from a single web site. Collecting a very large data in a limited time tends to be detected as a cyber attack and will be banned from connecting into the web server. To avoid the problem, this paper proposes a crawling method to mine web database faster and cheaper than conventional web crawlers. The method used is to run hundreds of threads from a single web crawler in a single computer and to distribute the threads into hundreds or thousands publicly available proxy servers. This web crawler strategy highly increases the speed of mining and is more secure than using single thread of web crawler.

Selection and peer-review under responsibility of DAAAM International Vienna

Keywords: focused web crawler; proxy; multi thread; web database

1. Introduction

Every search engine in the internet must have at least one web crawler. The other names of web crawler are web spider and bot. The main task of a web crawler is to crawl every web pages fed to it. It will retrieve the contents of web pages crawled, and parse it to get the data and hyperlinks. It then continues to crawl the found hyperlinks. The parser sends data to the indexer and saves it into the database. A search into a search engine actually does not search into the real web site, but it searches the search engine database. The output of search engines is a list of snapshots of web pages with its hyperlinks. The user should open web pages he needs by clicking the hyperlinks.

* Corresponding author. Tel.: +62 818 0854 0094; fax: +62 21 799 3375.
E-mail address: harry.achsan@paramadina.ac.id

1877-7058
Selection and peer-review under responsibility of DAAAM International Vienna
doi:10.1016/j.proeng.2014.03.017

Since the output of search engines is a list of snapshot of web pages, it is impossible to find any specific structured data using search engines. For example, we cannot gather all conversation made by someone in a forum, or friends of friend data in a social network. Using general search engines like Bing, Hotbot, Google, or Yahoo, we cannot collect all book data with Library of Congress control number between 89211901 and 89211999.We have to use a special web crawler to collect specific and structured data called Focused Web Crawler.

Focused web crawler, sometimes called vertical or specific web crawler, is a tool for mining specific data from web databases. The data mined are structured or semi-structured because it is retrieved from database in web sites such as databases from social networks, forums, blogs, online libraries, online stores, or any web sites that use database to display their information. If we can collect the data from a web site, then we can retrieve the information and discover the knowledge contained in it.

Collecting specific data from a web site is trickier than gathering contents of web pages, because there is sometimes no hyperlinks to/among each entity that makes regular crawler unable find it. For instance, if we need data of all books about automation in Library of Congress web site (http://www.loc.gov), we usually do it using search engines by supplying a key words "site:loc.gov automation book". Google and Yahoo gave no matched result in the first 300 of its results. On the other hand, the search facility in the home page of Library of Congress web site gave 806 matched results, but we have to retrieve41 web pages of its result set to get the data. Focused web crawler also gave 806 results with 100% matched and automatically put the data into its database.

A focused web crawler has to run carefully. Some web sites have a monitoring tool to watch every visitor. The tool can differentiate between human and bot by using special algorithms and can monitor data transfer rate for each visitor. It can also ban user's IP address if the user violate any of web site restrictions. Web site restriction can be found in the file robots.txt in the root directory of web site. It consists of all directories and web pages forbidden to be crawled. If the monitoring tool detects any bot or crawler opening any of restricted web pages, it will ban the crawler IP address from accessing its web site. If the crawler originates from a university network, the consequence is fatal, no one can access the web site anymore.

The non existence of robots.txt file does not mean that there is no restriction. The monitoring tool can still watch the visitor behavior. It will check every visitor data transfer rate and compare it to the maximum data transfer rate allowed. It also monitors the duration a visitor interacts with the web site continuously, and check the size of the data transferred. All visitor behavior will be compared to fair usage. Any breach to fair usage will have consequences. These restrictions make focused web crawler do their job very slowly. It can take weeks or months to collect tens or hundreds of thousands entities.

The aim of this paper is to develop algorithms for fast focused web crawler that can run safely. It will be achieved by using multi-threaded programming and distributed access via proxy servers. This paper will also show how to retrieve pairs of IP address and port of public proxy servers and how to crawl nicely.

2. Related works

Focused web crawler plays an important role in information society. It is used to crawl social networks [1], to crawl forums [2], to crawl web pages in specific language [3], to browse offline, to mirror web site [5], to generate web site map [6], and to develop Business Intelligence [7]. Many people need it, and some people give the software for free [8-9] and free to try [4-6]. Unfortunately, not all of focused web crawler's behaviour is in accordance with netiquette [10], resulting in many web sites implement user agent monitoring tool [11] to reject impolite crawler. To honour netiquette, focused web crawlers have to be improved.

Improvement of focused web crawler has been done by improving its strategies. Several strategies used by focused web crawler in the last decade has been reviewed and compared [12]. In the recent years, some researchers optimized the precision of focused web crawling results by implementing Bayesian classification [13-16], ontology [17], similarity [18], relevant topic [19], and Genetic Algorithm [34]. The more precision of a crawler makes it less web page visited, less data transfer rate, and more polite. Since many of web pages implement dynamic content, the content which are displayed different from the HTML source code, a number of improvements has been developed [13, 20, 21, 35-37] to overcome the problem.

One of the most important problems to overcome is to increase the speed of crawling politely. A multi-threaded crawler is proposed [22], which can speed up crawling, however it is detected as an impolite crawler if used to

collect data from a web site. Other researchers developed distributed crawler [23-28]. There are two types of distributed crawler, there are crawler distributed in a Local Area Network (LAN) and crawler distributed in Wide Area Network (WAN). Since LAN only uses one global IP address, a crawler distributed in a LAN tend to be detected as impolite crawler. On the other hand, implementing crawler distributed in WAN is costly.

To overcome the above problem, this paper proposes a focused web crawler which implement multi-threading in programming and implement distributed system in WAN. To lower the development cost, the system will use publicly available proxy server.

3. Architecture

Mining big data from a web site is very risky, because it should be fast enough to save time. But fast crawler tends to be banned, as mentioned above. The concept proposed is to develop a distributed focused web crawler using publicly available proxy servers, as shown in Fig. 1. To make it cost effective, the crawler should implement multi-thread programming which uses only one computer to run many crawlers. The other benefits of implementing multi-threaded crawler are centralized controller and easier to maintain.

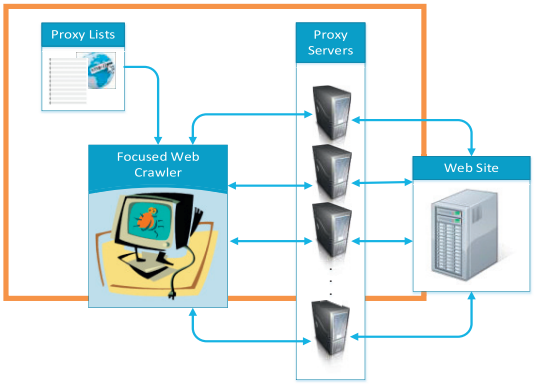

Fig. 1. Architecture of distributed focused web crawler.

3.1. Publicly available proxy servers

There are thousands of publicly available proxy servers on the internet and there are many lists available. List of public proxy servers can be sought using search engine with keywords: public proxy list. Some proxy lists which contain list of hundreds to thousands of proxy servers that can be retrieved page by page could be found in [29-31]. For simplicity, a table of more than five thousand proxy server addresses could be downloaded all at once from authors' cloud storage at [32], which is publicly available.

By the ability of hiding its information, proxy servers can be divided into three categories: Transparent, Anonymous, and Elite. A web crawler can use any of those categories of proxy server. If a crawler uses transparent proxy to access the destination web site the web server knows crawler's IP address and knows that the crawler using a proxy. An anonymous proxy server could hide crawler's IP address, but the web server still knows that the crawler using a proxy. A web crawler should use an elite proxy to hide its IP address and to avoid being detected using a proxy.

A web crawler should check the availability of a proxy server before using it. Checking the availability of 5,043 proxy servers downloaded from [32] gave different results. On 5 September 2013 it showed that 1,931 (38.23%) of proxy servers were available, but on 7 September 2013 only 1,576(31.26%) of proxy servers were available. In most cases, the proxy servers were turned off making them unavailable. Table 1 is a sample of the result of checking of

5,043 proxy servers, which shows 88.92% of proxy servers were not turned on, the message returned was "Unable to connect to the remote server", and 4.83% were very slow.

Table 1. A sample of list of errors of unavailable proxy servers.

Error Messages	# Proxy Servers	%
Unable to connect to the remote server	2,745	88.92
The operation has timed out	149	4.83
The remote server returned an error: (403) Forbidden.	63	2.04
The remote server returned an error: (404) Not Found.	39	1.26
The underlying connection was closed: The connection was closed unexpectedly.	21	0.68
The remote server returned an error: (504) Gateway Timeout.	16	0.52
The remote server returned an error: (407) Proxy Authentication Required.	9	0.29
The remote server returned an error: (500) Internal Server Error.	9	0.29
The underlying connection was closed: An unexpected error occurred on a receive.	9	0.29
The remote server returned an error: (503) Server Unavailable.	8	0.26
The request was aborted: The operation has timed out.	6	0.19
The server committed a protocol violation. Section=ResponseStatusLine	5	0.16
The remote server returned an error: (502) Bad Gateway.	4	0.13
The remote server returned an error: (400) Bad Request.	2	0.06
The remote server returned an error: (501) Not Implemented.	1	0.03
Too many automatic redirections were attempted.	1	0.03
Total	3,087	100.00

To check the availability of proxy servers, the crawler has to memorize all of proxy servers' address. Use each proxy server to crawl its destination web site. Record the message returned and the time span to retrieve a page. Put the list of available proxy servers into a file. Sort it by the time span to enable the crawler choose only the first N fastest proxy servers' address.

3.2. Distributed Focused Web Crawler

The distributed focused web crawler developed in this research is intended to crawl a single web server and to collect a specific data from a web database. In this research, we try to collect book classification/metadata from http://classify.oclc.org web database. The database contains more than 91 million book metadata[33]. Users can retrieve books metadata manually by filling in any key word to the search form provided by its web page. But a web crawler has two options to search any book metadata, those using web service or filling any book's data like ISBN, OCLC number, UPC, ISSN, book author, book title, or its subject heading.

This kind of web crawler can be built using any 4th Generation Languages like Java, C#, Perl, or Python. All these programming languages support multi thread. Multi thread is a concept in software engineering that enables a software to run many processes at once. A multi thread web crawler is a software that can download many web pages at the same time. One benefit of multi thread web crawler is that it can download web pages much faster than single thread web crawler.

3.3. Web Server

To optimize the crawling, targeted web server should be analyzed. The first thing to check is the availability of web services provided by the web server. Collecting data using web services is better than crawling web sites because web services are intended to provide data access for software like web crawler. Some benefits of crawling by using web services are that it is fast and structured. Collecting data via web service is faster than crawling the web pages because the data size from web service is smaller than from web page, which does not HTML presentation markups. The book entitled "Douris and the painters of Greek vases" has file size 7,202 bytes in HTML

format downloaded from [33] using a web browser. The same information in XML format is only 4,461 bytes retrieved using web service. XML is a structured file format that makes it easier and faster to be parsed to put into a crawler's database than parsing HTML files.

If the targeted web server provides no web service, we have to analyze the structure of web site manually. Analyzing the structure of web site manually can filter the hyperlinks to be crawled effectively, and can give understanding to the user how the page displayed. Filtering hyperlinks can avoid crawling to unneeded web pages. Knowing the pattern of displaying web page can give the shortest way to retrieve the data needed. For instance, a crawler can retrieve books metadata on web page http://www.lookupbyisbn.comby inserting a book's ISBN (International Standard Book Number) into its form and follow the Search button. By analyzing the web site, we know that the pattern to display books metadata based on its ISBN is just http://www.lookupbyisbn.com/Search/Book/<ISBN>/<page#>, then to retrieve books with ISBN 978-1433805615 can be done by a crawler by feed it a URL http://www.lookupbyisbn.com/Search/Book/978-1433805615/1.

3.4. Netiquette

Netiquette is etiquette in using the Internet. There is also an etiquette to be adhered for web crawlers. The most important one is the robots.txt, a file that resides on a web site root directory containing allowed and disallowed paths to crawl. We can display Google's robots.txt file contents by browsing http://www.google.com/robots.txt. The first two lines of Google's robots.txt are: "User-agent: *", and "Disallow: /search", it means any agent should not crawl Google's web pages which URL address contains /search. Another netiquette to be considered is crawling speed. Crawling speed to a web site should be limited, for example 10 pages/minute and stopped one hour for every six hours. This netiquette should be obeyed to prevent slowing down of the web server connection speed.

4. Experiments

The experiments was intended to collect books metadata from OCLC's web site [33]. OCLC provides two ways to collect book metadata: using web interfaces and using web services. Common users can collect book metadata by filling the web form with key words manually, but web crawler programmers prefer to use web services by putting a parameter to a specific URL.

In our experiments, we developed the web crawler using C#, an object oriented programming language. C# provides libraries to access data via computer networks and to develop multi threads, i.e. System.Net and System.Threading. The crawler only uses WebClient and WebProxy classes from System.Net library to access web resources and to utilize proxy server. To download a book metadata the crawler has to run a command `wc.DownloadString("http://classify.oclc.org/classify2/Classify?isbn=<ISBN>&summary=false")`, where `wc` is an object of WebClient and `<ISBN>` should be replaced with ISBN of a book. Assigning a proxy server could be done by command `wc.Proxy = new WebProxy(<string proxy address:port>)`.

The web crawler is tested in 7 conditions: a) with no proxy which is connected directly to OCLS web server, b) using 5 proxy servers, c) using 10 proxy servers, d) using 50 proxy servers,e) using 100 proxy servers, f) using 500 proxy servers, and g) using 1000 proxy servers. Each test condition ran for 3 hours to see the differences and to seek the optimum condition.

The proxy servers used in this experiment was tested and filtered, and only available proxy servers were used, and all unavailable proxy servers were eliminated. The speed of proxy server in downloading a book metadata varies, the fastest was 0.210012 seconds, the slowest was 324.2295448 seconds or more than 5 minutes, and the average was 11.0371659 seconds.

5. Results

Testing was conducted several times using different number of proxy servers. The number of thread is equal to the number of proxy used. Fig. 2 shows that increasing number of proxy servers used in crawling tends to increase the download speed. An exception occurred for 100, 500 and 1000 number of proxies, the download speeds shows a little different. Using a higher computer specification and a higher internet connection speed, the download speed for 500 proxies and 1000 proxies will be higher than using 100 proxies. The crawling speed also depends on thread maximum speed allowed. If the maximum thread speed allowed is over 100 pages/minute, the download speed can reach more than three thousand pages/minute.

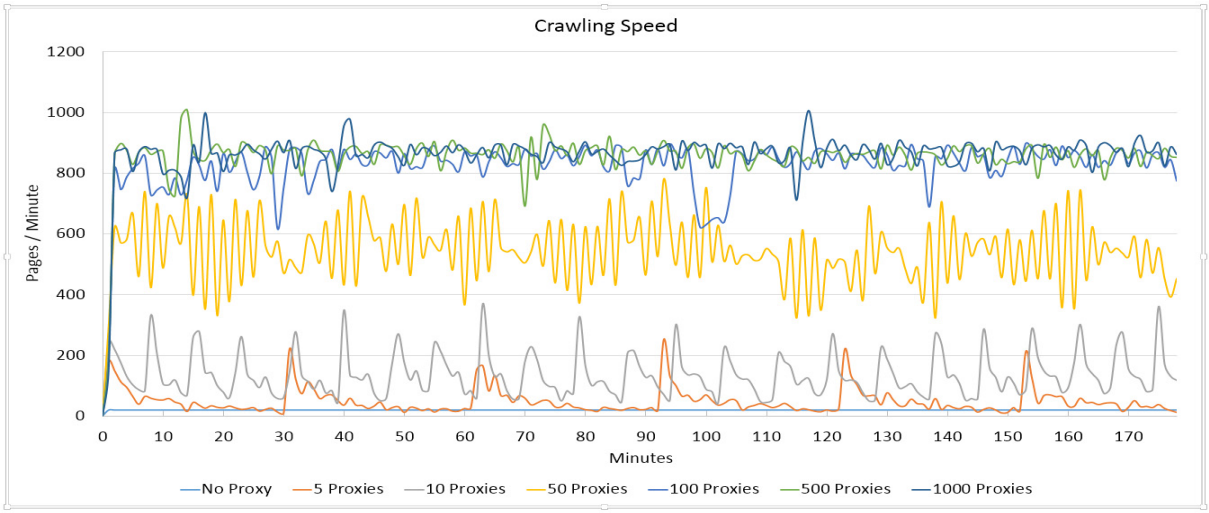

Fig. 2. Crawling speeds of the focused web crawler using different number of proxy servers.

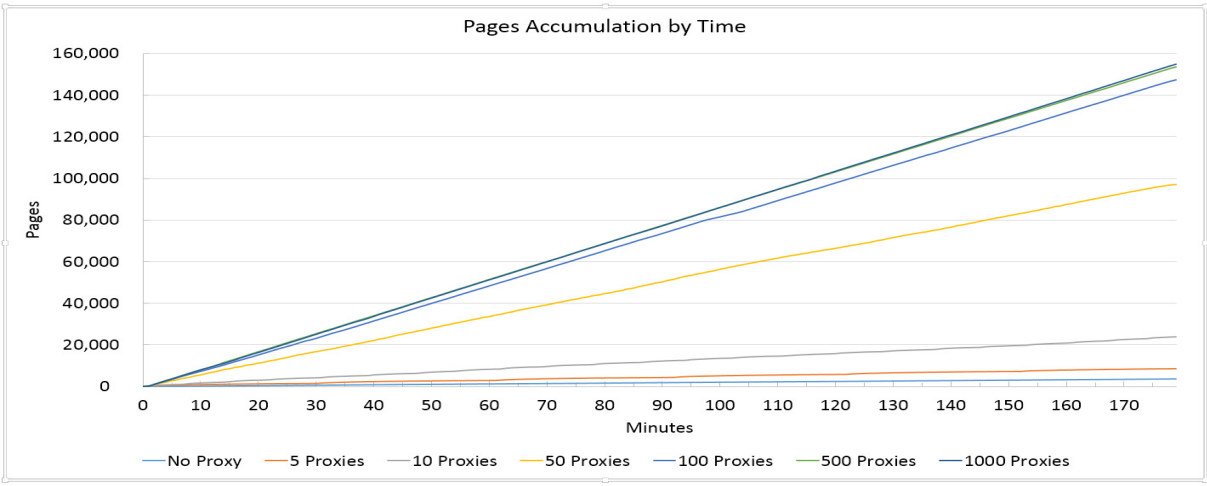

Fig. 3. Web pages accumulation by time for different number of proxy servers.

The downloading speed fluctuates by time, because the list of available proxy server is sorted by proxy speed. Fig 3 shows that using 5 proxies, the download speed is about 180 pages/minute at the first minute. This is caused by performance of several high speed proxies. The speed decreased to almost zero at minute 30th. It is caused by low speed proxy servers. The speed rose to more than 200 pages/minute on minute 32nd. This means that the crawler reuses the faster proxy servers.

Although the download speed fluctuates by time, their average speed is steady as shown in Fig. 3. The download speed on the figure is the slope line. The higher angle of the slope means the higher download speed. The figure also shows that the optimal number of threads is 100. Running more than 500 threads will not increase the speed, but tends to reduce the computer performance. Running 100 threads of a web crawler is much faster than running a single thread (no-proxy) web crawler, see Table 2.

Table 2. Accumulative number of web pages downloaded in 180 minutes.

Number of Threads	Pages Downloaded
1	3,600
5	8,456
10	24,059
50	97,235
100	147,527
500	153,709
1,000	154,833

The download speed of the crawler decreases by time if we run it for days or weeks. It is likely that some of the available proxy servers become unavailable anymore.

6. Conclusion

Multi thread and distributed web crawler is faster in collection data than a direct (no proxy) single thread web crawler. Multi thread web crawler should use the optimal number of threads to maximize the download speed. Too many threads tend to reduce computer performance, but too few threads will reduce the speed of collection. Developing a multi thread web crawler distributed to publicly available proxy servers is easier and cheaper than developing distributed web crawler into many computers controlled by one computer.

References

[1] Z. Xiao, B. Liu, H. Hu, and T. Zhang, “Design and Implementation of Facebook Crawler Based on Interaction Simulation,” 2012, pp. 1109–1112.
[2] Q. Gao, B. Xiao, Z. Lin, X. Chen, and B. Zhou, “A high-precision forum crawler based on vertical crawling,” in *Network Infrastructure and Digital Content, 2009. IC-NIDC 2009. IEEE International Conference on*, 2009, pp. 362–367.
[3] P. Tadapak, T. Suebchua, and A. Rungsawang, “A Machine Learning Based Language Specific Web Site Crawler,” 2010, pp. 155–161.
[4] Tenson Software Corporation, “Website Ripper Copier, Download Accelerator Manager (DAM) - Home.” [Online]. Available: http://www.tensons.com/. [Accessed: 01-Sep-2013]
[5] DB4ALL, “Webminer 2.0 overview,” 2010. [Online]. Available: http://www.db4all.com/webminer/. [Accessed: 01-Sep-2013]
[6] QiBing Software Co.,Ltd., “DSL Speed sitemaps generator - free,” 2012. [Online]. Available: http://www.dsl-speed.org/sitemaps-generator.htm. [Accessed: 01-Sep-2013]
[7] G. Pant and F. Menczer, “Topical crawling for business intelligence,” in *Research and Advanced Technology for Digital Libraries*, Springer, 2003, pp. 233–244.
[8] Craigd, “Populating a Search Engine with a C# Spider - CodeProject,” 2004. [Online]. Available: http://www.codeproject.com/Articles/7605/Populating-a-Search-Engine-with-a-C-Spider. [Accessed: 25-Oct-2012]
[9] Anonimous, “OpenWebSpider | Free software downloads at SourceForge.net.” [Online]. Available: http://sourceforge.net/projects/openwebspider/. [Accessed: 01-Sep-2013]
[10] Wikipedia, “Etiquette (technology) - Wikipedia, the free encyclopedia,” 2013. [Online]. Available: http://en.wikipedia.org/wiki/Etiquette_(technology). [Accessed: 01-Sep-2013]
[11] J. Yuanshu, T. Wenzhong, and G. Liyong, “Offensive and defensive strategy of web crawler.”
[12] I. Avraam and I. Anagnostopoulos, “A Comparison over Focused Web Crawling Strategies,” 2011, pp. 245–249.
[13] W. Ma, X. Chen, and W. Shang, “Advanced Deep Web Crawler Based on Dom,” 2012, pp. 605–609.

[14]M. S. Safran, A. Althagafi, and Dunren Che, "Improving Relevance Prediction for Focused Web Crawlers," 2012, pp. 161–166
[15] D. Taylan, M. Poyraz, S. Akyokus, and M. C. Ganiz, "Intelligent focused crawler: learning which links to crawl," in *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on*, 2011, pp. 504–508
[16] D. Chen, F. Liying, Y. Jianzhuo, and B. Shi, "Semantic focused crawler based on Q-learning and Bayes classifier," in *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, 2010, vol. 8, pp. 420–423
[17] C.-C. Wu, J. Zhao, and H. Ma, "The research of ontology-based focused crawler," in *System of Systems Engineering (SoSE), 2012 7th International Conference on*, 2012, pp. 736–738
[18] Q. Dong, "Search-Engine-Oriented Theme Crawler Design," 2010, pp. 303–306
[19] L. Wei-jiang, R. Hua-suo, Z. Tie-jun, and Z. Wen-mao, "A New Algorithm of Topical Crawler," 2009, pp. 443–446
[20] G. C. Paul Suganthan, "AJAX Crawler," in *Data Science & Engineering (ICDSE), 2012 International Conference on*, 2012, pp. 27–30
[21] X. Zhang and H. Wang, "AJAX Crawling Scheme Based on Document Object Model," 2012, pp. 1198–1201.
[22] M. Ke, P. Z. Zhang, and G. W. Chen, "The Crawler of Specific Resources Recognition Based on Multi-thread," 2012, pp. 569–572
[23] V. Shkapenyuk and T. Suel, "Design and implementation of a high-performance distributed web crawler," in *Data Engineering, 2002. Proceedings. 18th International Conference on*, 2002, pp. 357–368
[24] B. Zhou, B. Xiao, Z. Lin, and C. Zhang, "A distributed vertical crawler using crawling-period based strategy," in *Future Computer and Communication (ICFCC), 2010 2nd International Conference on*, 2010, vol. 1, pp. V1–306
[25] Qingdao ke ji da xue, "A dynamic URL assignment method for parallel web crawler," in *2009 Second International Workshop on Computer Science and Engineering proceedings: WCSE 2009 : 28-30 October 2009, Qingdao, China*, 2009
[26] A. Tripathy and P. K. Patra, "A Web Mining Architectural Model of Distributed Crawler for Internet Searches Using PageRank Algorithm," 2008, pp. 513–518
[27] A. Selamat and F. Ahmadi-Abkenari, "Application of clickstream analysis as web page importance metric in parallel crawlers," in *Information Technology (ITSim), 2010 International Symposium in*, 2010, vol. 1, pp. 1–6.
[28] M. Wu and J. Lai, "The Research and Implementation of Parallel Web Crawler in Cluster," 2010, pp. 704–708.
[29] "Free Proxy List - MultiProxy," 2009. [Online]. Available: http://multiproxy.org/txt_all/proxy.txt. [Accessed: 04-Sep-2013]
[30] Spys, "Free proxy list. Public proxy servers list.," 2013. [Online]. Available: http://spys.ru/en/free-proxy-list/. [Accessed: 04-Sep-2013]
[31] Free Proxy Lists, "Free Proxy Lists - HTTP Proxy Servers (IP Address, Port)," 2013. [Online]. Available: http://www.freeproxylists.net/. [Accessed: 04-Sep-2013]
[32] H. T. Y. Achsan, "A List of Publicly Available Proxy Servers," 2013. [Online]. Available: https://googledrive.com/host/0B343pgVVzFE4NUpheTA2SmJKaE0/Proxy_GOOD.txt. [Accessed: 07-Sep-2013]
[33] OCLC, "OCLC Classify -- an Experimental Classification Service," Apr-2013. [Online]. Available: http://classify.oclc.org/classify2/. [Accessed: 09-Sep-2013]
[34] B. W. Yohanes, Handoko and H. K. Wardana, "Focused Crawler Optimization Using Genetic Algorithm," *Telkomnika,* vol. 9, no. 3, pp. 403-410, 2011
[35] Anuradha and B. Ahuja, "Hidden Web Extractor Dynamic Way to Uncover The Deep Web," *International Journal on Computer Science and Engineering (IJCSE),* vol. 4, no. 6, pp. 1137-1145, 2012
[36] Anuradha and A. Sharma, "Hidden Web Data Extraction Using Dynamic Rule Generation," *International Journal on Computer Science and Engineering (IJCSE),* vol. 3, no. 8, pp. 3047-3058, 2011
[37] M. Kumar and R. Vig, "Learning Capable Focused Crawler for Information Technology Domain," *International Journal of Computer Applications,* vol. 43, no. 23, 2012