ISSN No. 0976-5697

Volume 8, No. 3, March – April 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Web Crawler : Review of Different Types of Web Crawler, Its Issues, Applications and Research Opportunities

Keyur Desai
Student, Department of Computer Engineering
Institute of Technology, NU.
Ahmedabad, India.

Virala Devulapalli
Student, Department of Computer Engineering
Institute of Technology, NU.
Ahmedabad, India.

Prof. Smita Agrawal
Asst. Prof., Department of Computer Engineering
Institute of Technology, NU.
Ahmedabad, India.

Prof. Preeti Kathiria
Asst. Prof., Department of Computer Engineering
Institute of Technology, NU.
Ahmedabad, India.

Dr. Atul Patel
Professor and Dean of CMPICA
CMPICA, CHARUSAT University,
Changa, India

Abstract: Today's search engines are equipped with dedicated agents known as "web crawlers" keen to crawling large web contents online which are analyzed and indexed and make the content available to users. Crawlers act together with thousands of web servers over periods expanding from weeks to several years. These crawlers visits several thousands of pages every second, includes a high-performance fault manager, are platform independent or dependent and are able to get used to a wide range of configurations without including additional hardware. This paper is focused on prerequisites of crawler, process of crawling and different types of crawlers. This paper give review about some potential issues related to crawler, applications and research area of web crawler.

Keywords: search engine, web crawler, www, Indexing, website analysis,

I. INTRODUCTION

Crawling refers to collecting of web pages that follows hyperlinks starting from a small set of web pages for further processing. There are quite good number of challenges for a crawler to crawl a handful of websites. These challenges include: make sure about the politeness of the web servers, URL normalization, detection of duplication, spider traps avoidance, preserve a line/queue of pages that are not retrieved, a storehouse(repository) of crawled pages is maintained, re crawling, and so on. The crawling problem is considered under subject area web mining. Web Mining is defined as "extracting knowledge from web" [9, 14]. This definition generates many questions in our mind. Why web mining? Which types of information or knowledge are mined? How does it perform? How are heterogeneous web data transformed in furnished data? All these questions generate a lot of issues in our mind that justify the requirement of an effective and robust mechanism for web crawler. Data mining commonly defined the processes to find pattern and knowledge form different data sources like databases, text files, images, videos, Web etc. But in all cases data are predefined and static and in case of web data mining, data are dynamic in nature. Due to dynamic nature of web data, extracting knowledge from web data is different from traditional data mining. That why we can say that web mining is an extended version of data mining.

II. LITERATURE SURVEY

The web and the web crawlers almost entered the world at the same time. The first crawler was created by Matthew Gray known as "wanderer". Many researches about web crawling was shown at the first to Worldwide Web Conferences. But at that time, the web was so small than what it is today that the problems regarding scaling were not addressed. The famous search engines use the crawlers that scale up to a considerable portions of the web. But, as there were competitions among search engines, the designs of these crawlers [4] were not described publicly.
There are 2 prominent exceptions:
1) Internet Archive Crawler
2) Google crawler
The Google crawler was developed at Stanford University. URL server process used to read URLs out from a file and used to forward them to multiple crawlers. Each crawler process is executed on distinct machine which was single threaded and used asynchronous input output to get data from up to 300 web servers in parallel at the same time [6]. The crawlers used to send the downloaded pages to a single store server process which is used to compress the pages and then they were stored to a disk. After this step the pages were then read back from disk by an indexer process, which pulled out links from HTML pages and saved them to a distinct disk file. A URL resolver process is used to read a link file and delinked the URLs contained, and kept the fixed URLs to the disk file which the URL server read. But 3-4 crawler machines were used because of the whole system consisted of 4-8 machines [4].
The Stanford web based project has applied a high performance distributed crawler that was able to download 50-100 documents/second.

Whereas the Internet Archive used multiple crawlers which crawled the entire WWW. Each and every of the crawlers was given 64 sites for crawling, and none of the site was given to greater than one crawler. Every single threaded[4] executed a

directory of all the seed URLs for its given sites from the disk to the site queues and later used asynchronous input output to fetch pages in parallel from these queues. If a page was retrieved/downloaded once then the crawler used to get the links consisted in it. Eventually, a batch process has combined these logged cross site URLs into site seed sets sifting out the replicas in the process[6].

III. ABOUT WEB CRAWLER

Web crawlers are programs that do automatic search on the Worldwide Web 'interpreting' the entire of the content they meet on the web.
They are also called as:

- Bots
- Robots
- Spiders
- Wanderers
- Worm
- User agents

A search engine consists of 3 parts in 1) Crawler: Traverses the web that collects the information and save them into a large repository after being compressed. It goes after hyperlinks from corner to corner on the web that collects the information from HTML web pages. 2) Indexing: Prepares the index of the local database. The indexer obtains the web pages fetched by the crawlers and store them into a highly well-organized index. 3) Query processor: Handles the user queries by searching through the index.

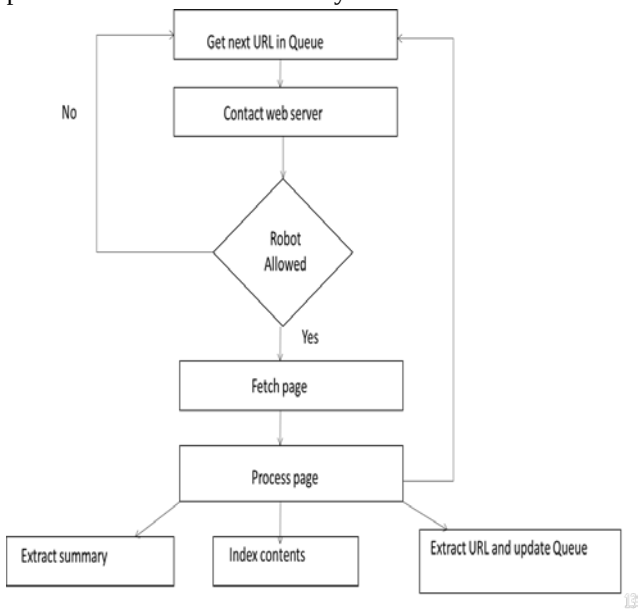
IV. PRE-REQUISITES OF A CRAWLER

Robustness: This is the ability of a web crawler to handle spider traps which generate web pages that deceive crawlers so that they get stuck by retrieving a countless number of pages in a specific domain. So the crawlers must be designed to be durable enough to such spider traps. It may also happen that not all of the spider traps are harmful, some are the side-effects of an erroneous website development.
Politeness: The speed at which the crawlers visit the website. The crawler must only retrieve one page at a time from the same site and should wait for some time between 2 successive downloads [1]. Also crawlers must only crawl the websites that are allowed to crawl. The consent to crawl a website is given by robots.txt file.
Distribution: The crawling should be distributed within several machines. This is to acquire advantages in deployment, programming, and debugging.
Scalability: The architecture of the crawler should allow balancing of the crawl rate by inserting up further machines and bandwidth.
Efficiency: This is the smart use of the memory, bandwidth and processor. E.g. as few idle processes as possible.
Quality: The web crawler should detect the most useful pages, to be indexed first. Simply the pages of high quality or highest page rank should be crawled first.
Freshness: The web crawler should constantly crawl the web. The database should consist of latest pages.
Extensibility: The web crawler supports new formats of data. E.g.XML-based formats, new protocols like FTP, etc.

V. PROCESS OF CRAWLING

A crawler is initiated by putting in the first set of URLs, in a queue, where each and every URL is to be fetched are kept and ranked. Starting from this queue, a crawler receives a URL by using some methods, downloads the page, retrieves URLs from the page that is downloaded, and deposits the new URLs in to the queue. This process is reiterated. Accumulated web pages are afterwards used for some other purposes such as a Web cache or a Web search engine.
Detailed process would be:

1. The search engine will follow the links to find other pages that will consist of the links that will point to other pages that'll and so on. First/Starting page is the seed.
2. If you start crawling in your private/personal website and if it doesn't comprise of any link, or if it only links to a page that is without links, your search engine will believe the entire internet only has only 1 or 2 pages.
3. If you want to have a good search engine it must have indexed as many pages as possible, so the seed page is very significant. Example: http://www.google.com could be a good initial point.
4. A bad seed must be a website with none of the links. That would mean that we do not crawl any site except for an initial seed. A good seed may be massive lists of websites and seed the crawler with a group of pages.
5. More than one URL inside the seed list becomes problematic. So it can have only one seed set.



[Figure-1: crawling process [3]]

The crawling process is elaborated in the Figure-1 as:

1. Check for the subsequent page to download:- the system keeps trail of pages to download in a queue.
2. Check to see if the page is "allowed" to be downloaded(robots.txt)
3. Download the whole page.
4. Extract all links from the page and add those to the queue mentioned above to be downloaded later [3].

5. Extract all words, save them to a database associated with this page, and save the order of the words so that people can search for phrases, not just keywords [3].
6. Non-compulsorily it filters for things like adult substance, language form for the page, etc.
7. The review of the page is stored and the last processed date for the page is revised so that the system knows when it should re-check the page in the future.

VI. SOME POTENTIAL ISSUES

Dealing with the form unless it is in standard format:

If the code is not properly written in a suitable form, then our parser will not be able to easily extract information from the webpage having that form [2]. Because of that the survival of the form may not be identified.

Managing the forms:

The post method's constraints are sent in the body of the HTTP request and its URL is just simply making it hard for us to deal with it (e.g., http://ask.com/find). Whereas the Get method adds its constraints to the action in the URLs in a format of dynamic URLs that are frequently clearly visible (e.g. http://ask.com/find?src=cdd&ts=go).

Forms that cover across several pages:

Here the same form is extended over multiple continuous pages [2].

Forms with JavaScript embedded:

Usually input fields of this type of form have a lot of restriction such as the type of input, the format, the length, the syntax [2].

Forms with personal information:

Constraints such as username, password, E-mail will not be considered for privacy raison [2].

VII. TYPES OF WEB CRAWLERS

1. Breadth first crawler:

Begins with a bunch of pages and then searches the remaining pages by following links in the breadth first manner[4][9]. The frontier is employed as a First in First out (FIFO) queue. The process of crawling in breadth first fashion is:

- All the given seeds are kept in the queue.
- Lists of visited nodes are made which are at first empty.
- If the queue isn't vacant:-
  - the initial node from the queue is discarded
  - that node is appended to the list of visited nodes
  - for each of the edge beginning at the node:-
    - Does nothing with that edge if the node at the last is in the queue already or occurs on the list of visited nodes.
    - Else, the node is appended at the last of the edge up to the end of the queue.

In reality web pages are not navigated strictly in breadth first manner but it may use a variety of policies[4]. For example it may crawl most significant pages first.

2. Incremental web crawler:

Incremental crawler [10] updates current group of downloaded pages as an alternative of restarting the crawl from the beginning each time. This engrosses a way of ascertainment if a page has changed from the last time it was crawled. A crawler will persistently crawl the entire web supported by some group of crawling cycles. An adaptive model is used which uses data from preceding cycles to determine which pages should be tested for updates resulting in huge freshness and achievement of low peak load[7]. So, if the crawler can approximate how frequently pages changes, then the incremental web crawler may re-visit only the pages that have been changed rather reviving the whole collection together.

3. Focused web crawler:

Focused crawler [11] is such web crawler that downloads the web pages which are correlated with each other. It collects documents which are definite and correlated to the given topic. Therefore, a focused crawler looks for, obtains, indexes and keep up pages on definite set of topics which represents a relevantly narrow segment of the web. It is also called as Topic Crawler for the reason that is its way of working. The focused crawler ascertains the Relevancy and Way Forward. It ascertains to how much extent the given page is related to the particular topic and how to proceed forward [2]. The advantages of focused web crawler is that it is economically feasible in terms of hardware and network resources also it can reduce the amount of network traffic and downloads.

4. Hidden web crawler:

Data on the web is situated in the file/database and can be fetched by giving queries or by filling up the forms on the web. Current day crawlers crawl only publicly index-able web (PIW) [12] that means set of pages which are accessible by following hyperlinks, overlooking search pages and forms which require authorization or earlier registration. Recent study is estimated that the hidden web's size is 500 times more than the size that is of the publicly index-able web (PIW).These web crawlers are usually implemented by central bureau, trademarks offices, etc.

5. Parallel web crawler:

Because the dimension of the web grows, it becomes more complicated to retrieve the whole or a significant portion of the web using a single process. Therefore, lots of search engines frequently execute multiple processes in parallel [5] to do the above task, so that download speed is taken full advantage. Multiple crawlers are frequently executed in parallel which are therefore known as Parallel crawlers [7][12]. Consists of crawling processes called as C-procs. Problems with parallel crawling include:

- Overlap: It is possible that more than one c-proc might download the same page numerous times when a multiple number of c-procs are running is known as overlap.
- Quality: More significant pages are downloaded first, however this could be a difficult task for parallel web crawlers.
- Bandwidth of communication: Communication can develop importantly if the number of c-procs augment and must be reduced to preserve the effectiveness of the crawl.

6. Distributed web crawler:

This crawler [13] sprints on group of workstations. In distributed web crawler, a Uniform Resource Locator (URL) server assigns peculiar URLs to multiple crawlers, which downloads web pages in parallel, the crawlers will afterwards

send the downloaded pages to an innermost indexer on which links are retrieved and sent by means of the URL server up to the crawlers. This distributed character of crawling process decreases the hardware requirements and increases the largely download speed and dependability. First distributed crawler was developed which uses client server model that is centralized. But with this model there were problems like obstruction, expensive administration and being a single point of failure. But when it is developed to be fully distributed crawler where there is no central controller because of which problems like scalability, increased autonomy of nodes and failure resilience have been overcome. Although, fault tolerance is considered, then if an error occurs, the whole system will be difficult to maintain, and may be very difficult to ignore the loss of information.

VIII. ADVANTAGES AND DISADVANTAGES OF WEB CRAWLER

Advantages: The web crawlers offer huge databases of websites for searching. The whole text of individual web pages is often searchable. Good for searching very vague terms or phrases.

Disadvantages: There is no need for humans to dig out the difficulties, such as duplicates. The large of database may lead to huge numbers of search results. The search commands could be frequently be compound and baffling.

IX. APPLICATION AND RESEARCH AREA

General web search:
A web crawler for general search of web engine should carefully balance between coverage and quality. Coverage means that it must scan pages that allow it to create index which may be used to answer many difficult queries. Quality means that the pages must be of a high quality.
Vertical crawling:
Vertical crawling for data collection: It can be used to collect data from different sources. Most common type of this crawler is shopbot, that is intended to download information from on-line shopping lists and provides an interface for evaluating prices. News crawler to gather news items from sources that are pre-defined.Vertical crawling of specific formats: This search also includes segmentation by a data format. There are several machines that crawl image, audio, or videos.
Web characterization: It is the requirement for building effective web search engines. This is a difficult issue the even if the web contains a finite quantity of information, it may contain infinite pages. Page centered characterization measures page sizes, technologies, and other properties. While for link centered characterization the choice of starting URLs for performing the crawl is difficult.
Mirroring: It is the act of keeping a partial or complete copy of a web site. The aim of mirroring is to distribute server load and provide quicker access to users in distinct network locations. The copies are called mirrors. Web archiving does the act of keeping a huge set of pages without removing the outdated copies, i.e., the entire history of the each page is kept.
Web site analysis: A web crawler can be used to analyze a website and it can make changes on its own according to criteria that are predefined. The majority of the common form of this is link validation which automatically scans the web pages from broken links to inexistent pages. Another application is the web directory that automatically finds the web sites that are not available and to test them the attention of the directory's editors is called.
Semantic Based Crawling: The Automated ontologies formed in the paper[15] are based on lexical relations of a word i.e synonyms, concepts of hypernymy and hyponymy, meaning of the words. To get various, accurate details of important terms and the related terms these created ontologies are used and semantic based web crawling can be easily done.

X. REFERENCES

[1] Castillo, Carlos. "Effective web crawling." Acm sigir forum. Vol. 39. No. 1. Acm, 2005.

[2] Mahmud, Hasan, Moumie Soulemane, and Mohammad Rafiuzzaman. "A framework for dynamic indexing from hidden web." IJCSI (2011).

[3] Khurana, Dhiraj, and Satish Kumar. "Web Crawler Web Crawler: A Review."

[4] More effective, efficient, and scalable web crawler system architecture, N. A. El-Ramly; H. M. Harb; M. Amin; A. M. Tolba, Electrical, Electronic and Computer Engineering, 2004. ICEEC '04. 2004 International Conference on Year: 2004 Pages: 120-123, DOI:10.1109/ICEEC.2004.1374396 IEEE Conference Publications

[5] Prof. Smita Agrawal and etl. Performance evaluation of counting words from files using OpenMP, DOI: 10.090592/IJCSC.2016.008

[6] Liu F., Ma F., Ye Y., Li M., Yu J. (2005) IglooG: A Distributed Web Crawler Based on Grid Service. In: Zhang Y., Tanaka K., Yu J.X., Wang S., Li M. (eds) Web Technologies Research and Development - APWeb 2005. APWeb 2005. Lecture Notes in Computer Science, vol 3399. Springer, Berlin, Heidelberg

[7] Pavalam S. M., S. V. Kasmir Raja, Jawahar M., and Felix K. Akorli, Web Crawler in Mobile Systems, International Journal of Machine Learning and Computing, Vol. 2, No. 4, August 2012

[8] Hafri, Younès, and Chabane Djeraba. "High performance crawling system." Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval. ACM, 2004.

[9] Bing Liu, "Web Content Mining" the 14th international world wide web conference (www 2005) China .japan.

[10] Jumgoo Cho and Hector Garcia-Molina, "The evolution of the Web and implications for an incremental crawler", Prc. Of VLDB Conf, 2000.

[11] Paolo Boldi_ Bruno Codenotti† Massimo Santini‡ Sebastiano Vigna," UbiCrawler: A Scalable Fully Distributed Web Crawler".

[12] A.K. Sharma, J.P. Gupta, D. P. Agarwal, "Augmented Hypertext Documents suitable for parallel crawlers", communicated to 21st IASTED International Multi-conference _ Applied Informatics AI-2003, Feb 10- 13,2003, Austria.

[13] Vladislav Shkapenyuk and Torsten Suel, "Design and Implementation of a High performance Distributed Web Crawler", Technical Report, Department of Computer and Information Science, Polytechnic University, Brooklyn, July2001.

[14] Jai Prakash, Bankim Patel, Atul Patel,"Web Mining: Opinion and Feedback Analysis for Educational Institutions", 2013, IJCA, Volume 84 – No 6, December 2013

[15] Preeti Kathiria ,Sukraat Ahluwalia, "A Naïve Method for Ontology Constructions" 2016, International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI), Vol.5, No.1, February 2016