ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 10, October 2013

WEB CRAWLER

Raja Iswary[1] Keshab Nath[2]

Department of Information Technology, Assam University Silchar, India[1,2]

Abstract: The web data has exploded to a considerable amount with the invention of Internet Technology. The web being very vast covering billions of websites has been monitored by a tool or a program called ‘Crawler’. The main goal of this paper is to focus on a range of issues that are generic to crawling and the various mechanisms to build the crawler for the proper implementations of web search engine. This paper represents the significant algorithms that are the building blocks of the crawler for searching and outlines some of the fundamental challenges that are taken up with respect to crawling.

Keywords— Web Crawler, Robot Elimination protocol, Design Issues, Policies and Algorithm, Crawl Techniques.

I. INTRODUCTION

The web being one of the services communicated via the internet is a collection of interconnected documents and other resources which are linked by hyperlinks and URLs. The user of the web has all the privileges to have an access over it through the search engine. Thus, in order to have an access over the web a collection of web pages are required and an index needs to be build based on the collected resources. The above activities are carried out by Web Crawler, a tool for exploring the subsets of the web. In the first section the web crawler has been described and discussed as how it works. Web crawler is the central part of the search engine which browses through the hyperlinks and stores the visited links for the future use.
The next section 2 covers the working of web crawler and its various features that a web crawler must and should be built with along with its policies and algorithm in the next section 3. The design issues are also being discussed as the most prolific part of the crawler. Section 4 covers the various techniques for crawling.

II. WEB CRAWLER

A web crawler (also known as a robot or a spider) is a system, a program that traverses the web for the purpose of bulk downloading of web pages in an automated manner. Web crawlers are prominently one of the main components of web search engines that assemble a corpus of web pages or creates a copy of all the visited pages, index them, and allow users to issue queries against the index, provide fast searches and find the web pages that match the queries. Interacting with hundreds of thousands of web servers and name servers, crawling is considered as the most fragile application since it is beyond the control of the system.
Crawler follows very simple steps yet very effective work in maintenance, checking of the downloaded links and also the validation of HTML codes as follows:
It starts with the list of URL’s to visit, called seeds and downloads the web page.

It visits the downloaded page and parses through it and finally retrieves, identifies all the links and adds them to list of URLs called the crawl frontier.

It visits the downloaded page and parses through it and finally retrieves, identifies all the links and adds them to list of URLs called the crawl frontier.

For each of the retrieved hyperlinks, repeat the above process.
URLs added to the frontier are visited based on some Policies (discussed in section B). [1]

A. FEATURES A CRAWLER MUST/SHOULD BE BUILT UPON

Basically a crawler has the following features which are taken care of while downloading web pages.

- Freshness: It means that the downloaded copies of web pages are up-to-date. It is defined as a degree to which the acquired snap-shots of the pages are up-to date.
- Quality: A high quality portion of the web pages are aimed to be retrieved along with broad coverage.
- Coverage: The fraction of the desired pages that are successfully downloaded by the crawler.
- Scalable: Scaling up the crawl rate by adding extra machines and bandwidth must be supported by the crawler architecture.
- Robustness: There are spiders traps that are created by the servers with a motive to mislead the crawlers and make them struck somewhere. In such a case the crawler must be designed to be strong to such traps.
- Politeness: The politeness policy must be taken care of while crawling such as not creating web server overloaded by requesting more web pages, privacy aspect is also an issue i.e. they may access parts of websites that were not meant to be public. [3]

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 10, October 2013

B. CRAWLER POLICIES

A Web crawler has various tasks and goals that must be handled carefully in spite of various contradictions amongst them. Also the various resources that are available must be used by web crawlers efficiently, also including network bandwidth which must exhibit a high degree of parallelism without affecting the web server by overloading. So policies are followed to maintain a crawling decorum.

a) a selection policy that states which pages to download.

b) a re-visit policy that states when to check for changes to the pages.

c) a politeness policy that states how to avoid overloading Web sites.

d) a parallelization policy that states how to coordinate distributed Web crawlers.

C. WEB CRAWLER ARCHITECTURE

The web crawler architecture below shows how it crawls through a whole site on the Inter-/Intranet.

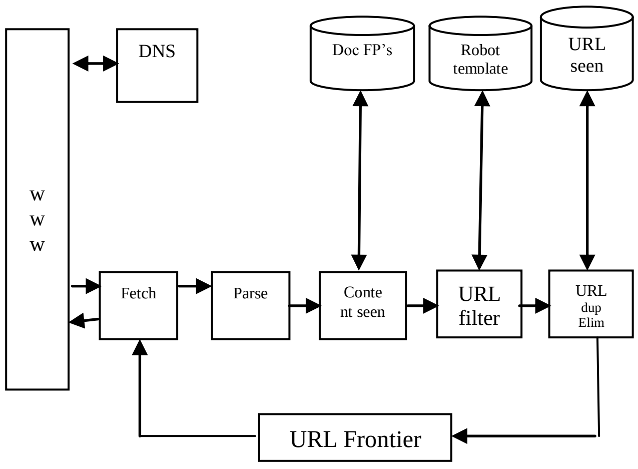

Fig.1: Basic web crawler Architecture [3]

The above architecture outlines some of the modules that depict a very useful working scenario of the crawler. The URL frontiers, contains URLs yet to be fetched soon. A DNS resolution module converts the URLs to its equivalent IP address. A fetch module to retrieve the web pages associated with its URL. A Parsing module is used for extraction of text and set of links from a retrieved web page. Finally, a duplicate elimination module checks to see whether there are any duplicate URL in the frontier.

A crawler fetches web pages associated with an URL by taking an URL from the frontier using http protocol. Once fetched, the web pages are stored temporarily, thereafter the page is parsed and the text as well as the links in it is being extracted.

After the URLs being parsed, the very next task is to check whether the web page with same content has being seen at another URL.

The *URL filter* has another promising task which filters out the URLs that it should be excluded from the frontier based on several tests. For instance it may exclude (say all .com URLs). The *Robot Exclusion protocol* is used by most of the host by placing certain portions of the websites off-limits to crawling.

This protocol works by placing a robot.txt file at the root of the URL.

The final process is to check for any duplicate URLs, if the URL is already present in the frontier i.e. already crawled then we do not add it to the frontier. [3]

D. DESIGN ISSUES

Various Data structures are required while considering design issues such as one data structure for maintaining the set of URLs that have been discovered (whether downloaded or not), and a second data structure for maintaining the set of URLs that are yet to be downloaded. The first the second data structure (usually called the frontier) must support adding URLs, and selecting a URL to fetch next, while the second data structure (sometimes called the "URL-seen test" or the "duplicated URL eliminator") must support set addition and set membership testing. [2]

1. FRONTIER DATA STRUCTURE

The Frontier Data Structure works based on the First-in-first-out (FIFO) technique which results in a Breadth-first search of the web graph. A Frontier that works as FIFO queue has a disadvantage that contains long runs of URLs on the same server resulting in overloading web server which is an act of "impolite". This is because most of the web pages are relative to each other which refer to another page on the same server. Thus in order to obey the politeness policy the crawler assigns priorities to web pages in the frontier as a disk-based priority queue ordered by its usefulness. [2]

2. URL SEEN TEST

The crawler keeps track of the set of URLs that are previously downloaded and added to the frontier. The main objective of this data structure is to avoid adding multiple same URLs to the frontier. While workings with the simple Batch crawling in which the pages are downloaded only once, this data structures needs to support insertion and set membership testing and in case of continuous crawling where pages are re-downloaded, it must support deletion as well. [2]

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 10, October 2013

III. ALGORITHM

The basic web crawler algorithms are modified in many variations that give search engines different levels of coverage. Billions of websites are crawled and indexed using algorithms depending on a number of factors.

a) Breadth-First search Algorithm: This Algorithm starts its crawling process right from the root node and sweeps down searching for the related neighbouring node at the same level. While crawling, if at the very first level itself finds the relevant node or its objective then a success occurs and terminates else goes on finding its objective down the very next level.

Breadth first is well suited for situations where the objective is found on the shallower parts in a deeper tree. [4]

b) Depth-First search: A technique which systematically traverses through the search by starting at the root node and traverses beneath down the child nodes is a powerful Depth-First search. While visiting each child nodes the objective is searched and so on the process continues if not found. If there is more than one child, then which node to visit depends upon the priority (i.e. left most child) and traverses deep until no more child is available. A backtracking method is used for traversing to the next unvisited node and then continues in a similar manner. [4]

c) PageRank Algorithm: In a PageRank, each of the pages on the web has its own measure which is independent of any informational needs. This algorithm ranks the web pages according to their importance or relevance. The page rank of a given page is calculated as: [3]

$$PR(A)= (1-d) + d(PR(T1)/C(T1) +...+ PR(Tn)/C(Tn))$$

Where

$PR(A) \rightarrow$ Page Rank of a website,

$d \rightarrow$ damping factor,

$T1.....Tn \rightarrow$ Link

IV. CRAWLING TECHNIQUES

The crawling method used by various search engines in order to download pages that have already been downloaded and those that are yet to be downloaded relies greatly on various techniques such as follows:

A. FOCUSED CRAWLER

Focused crawling was coined by [5] as an efficient resource discovery system. It has three components – crawler, classifier and distiller. The Focused crawling is a technique which is specifically designed to gather web pages on a specific topic by carefully prioritizing the crawl frontier (for example: To crawl pages from the .com domain or crawl pages with larger Page Rank), whereas a general crawler gathers as many pages as it can from a particular set of URLs.

Chakrabarti et al. coined the term focused crawler and used a text classifier [5]

The pages that are relevant to a pre-defined set of topics are sought out by using exemplary documents. The focus crawler sets its crawl boundary to find the links that are relevant and thus avoiding crawling irrelevant regions of the web which ultimately leads to significant savings in network resources and results in more up-to-date data.

An approach taken by Pinkerton in the early development of crawler has used an anchor text of links to predict the probability an unvisited page will be relevant before downloading the page. Different techniques for searching are used by focused crawlers such as Best – Fit search strategy and some focused crawler prospers with page rank technique for giving out the most important page

The crawling efficiency is significantly influenced by seed selection for focused crawlers. The richness of links in the specific topic being searched are greatly depended by the focused crawler as performance aspect.

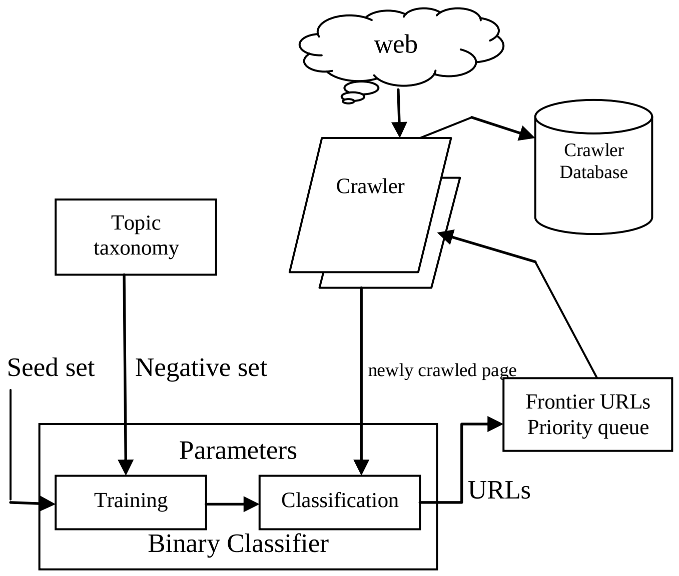

Fig.3: Focused Crawler Architecture [5]

B. DISTRIBUTED CRAWLER

In distributed web crawler a URL server distributes individual URLs to multiple crawlers thereby downloading the web pages in parallel and then sends the downloaded pages to a central indexer on which links are extracted and sent via the URL server to the crawlers.

The crawling over network of workstations in parallel requires a reasonable amount of time to complete the crawl process which ultimately leads to reduction of hardware requirements, increases the reliability and download speed. Thus, the distribution of process to multiple processes makes the system scalable. It basically uses Page rank algorithm for its increased efficiency and quality search. [3][5]

C. INCREMENTAL CRAWLER

An Incremental crawler continuously crawl the entire web and updates the existing downloaded pages instead of restarting. A data from previous cycles is taken to decide which pages should be checked for updates resulting in high freshness and low peak load is achieved.
Continuous crawling requires the crawler to revisit the same resources at certain intervals which means that some intelligent control, reschedule resources retains detailed state information i.e. a record of each resource's history is used in turn to determine its ordering in a priority queue of resources waiting to be fetched. Here, an incremental strategy is called for rather than the snapshot strategy used in broad and focused crawls.

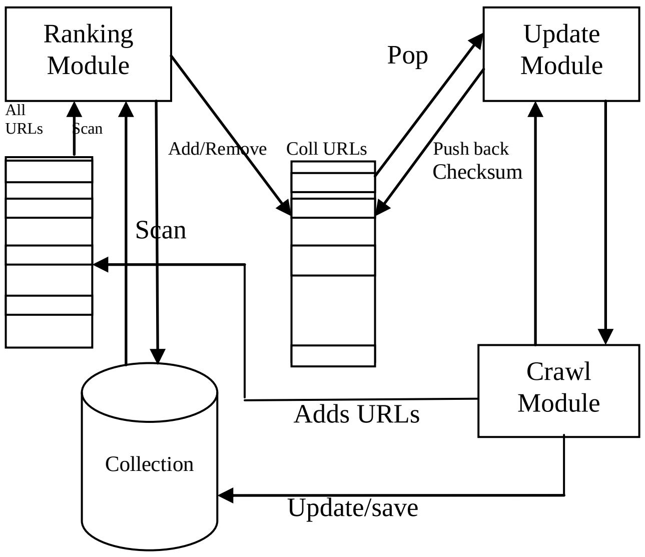

Fig.2: Incremental Crawler Architecture [6]

D. HIDDEN WEB CRAWLER

Hidden web is the term used to describe the information available on the web that is hidden behind the search query interfaces that act as entrance to backend databases. Many web pages cannot be crawled without filling up the forms as such they are inherently hidden behind search forms and are termed as “Deep web”. A study describes that only publicly indexable web (PIW) i.e. set of pages which are accessible by following the hyperlinks. A further study by Raghavan and Garcia-Molina observed that crawling of “Deep web or Hidden web” has revealed that the hidden web pages are created dynamically by firing user query, so they cannot be indexed by traditional search engines, cannot be reached by following links, but only by querying forms. [6].

V. CONCLUSION

In this paper we have studied how to build an effective web crawler. The study carried out based on crawl ordering reveals that the incremental crawler performs better and is more powerful because it allows re-visitation of pages at different rates. Crawling at other environment, such as peer-to-peer has been a future issue to be dealt with.

REFERENCES

[1] Monica Peshave and Kamyar Dezhgosha, ”How Search Engines Work and A web crawler Application”.
[2] Christopher Olston and Marc Najork, “Web Crawling”, Foundations and Trends in Information Retrieval, Vol 4.No 3(2010) C.
[3] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze, ”Introduction to Information Retrieval” Cambridge University Press, 2008, Ch 20, pg.405-416.
[4] Pavalam S M, S V Kashmir Raja, Felix K Akorli and Jawahar . ”A Survey of Web Crawler Algorithms”, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 6, No 1, November 2011 ISSN (Online): 1694-0814
.[5] S. Chakrabarti, M. van den Berg, and B. Dom. Focused Crawling: A New Approach for Topic-Specific Resource Discovery. WWW Conference, 1999.
[6] Dhiraj Khurana, Satish Kumar, “Web Crawler: A Review” , IJCSMS International Journal of Computer Science & Management Studies, Vol. 12, Issue 01, January 2012
ISSN (Online): 2231 –5268”.

BIOGRAPHIES

Raja Iswary Pursuing Master of Technology in Information Technology at Assam University, Silchar India. He has received his Bachelor of Engineering in Computer Science from Anna University, Chennai.

Keshab Nath Pursuing Master of Technology in Information Technology at Assam University, Silchar India. He has received his Bachelor of Technology in Information Technology from North Eastern Hill University,Shillong.