ISSN:0975-9646

A Comparative Study on Web Crawling for searching Hidden Web

Beena Mahar[#], C K Jha[*]

[#]*Research Scholar, Department of Computer science*
Banasthali University, Rajasthan- India
[*]*Associate Professor (HOD), Department of Computer science*
Banasthali University, Rajasthan -India

Abstract- **A web crawler is a software program that browses the web in a very systematic manner. Crawlers are used to create a replica of all the visited web pages that are processed by a search engine that will index the downloaded the pages that help in quick searchers. This is used by the search engine and other users to ensure that their database is up to date. A large number of HTML pages via web pages are continually being added every day and information is constantly changing. There are some web pages which are not directly located by the search engines because today in almost all search engines searchable databases are not properly index able or qyeryable. So they appear hidden to the average internet user. These pages are referred to as the Hidden Web or the Deep Web. In world wild web the huge amount of information is available only through surface web. The deep web is the largest growing area of now days of information on the internet.**

This paper briefly studies the concepts of web crawler, their type, and architecture for searching the hidden web documents. The various category of web crawler with working is also taken for the study and provide some future directions for research on web crawling for searching hidden web.

Keywords- web crawler, hidden web, Architecture, Traditional web crawler, types

I. INTRODUCTION

Today the major source of information is the WWW. Using this source of information is shared among various communities. This information is available in the form of text, audio, video and other multimedia forms.

Crawling is the process of exploring web applications automatically. Web crawlers have a long and interesting history. The web crawler aims at discovering the web pages of a web application by navigating through various applications. As the huge amount of information on the web has been increasing rapidly, web users increased rely on search engines to find desired information or data. In order to search engines to learn about the new data as it become available, the web crawler has to constantly crawl and update the search engine database.

Web crawling is the process where we gather web pages from the www, in order to index and support a search engine. The main objective of web crawling is to easy, quickly and efficiently gather as many various useful web pages as possible and together with the link structure that interconnects them.

A web crawler is a software program which is automatically traverses the World Wide Web by downloading the web documents and following links from one web page to other web page. It is a web tool for the search engines and other information seekers to gather data for indexing and to enable them to keep their database up to date. Generally all search engines use web crawlers to keep fresh copies of data from database. Google is one of the example of web crawler which is run on a distributed network of thousands of low cost computers and carry out fast parallel crawling processing with the returns results with in the fraction of seconds[1]. In this paper we analyses the concepts of web crawler with the help of their different types of working. This paper does comparative study of various crawling strategies that are used for downloading the HTML web pages from the WWW. Especially this paper is focus on the hidden web and all the related aspects for crawling the searching hidden web documents.

II. FEATURES OF WEB CRAWLER

A web crawler should have the following features:

Distribution: A web crawler should have the ability to execute in a multi machines.

Robust: A web crawler should have the ability to handle dynamic HTML web pages.

Politeness: A web crawler to make such kind of policies about the frequency of robot to visits.

Scalable: A web crawler should have scaling by adding more machines and extending more bandwidth.

Efficiency: A web crawler should have more efficient to make clever use of processor, storage, memory and bandwidth.

Extensible: A web crawler should be extensible to cope up with a new data structure like XML/EXML, new protocols etc.

Quality: A web crawler should be identify the most useful and meaning full web pages and make the indexed for those kinds of web pages.

Freshness: a web crawler should be ensuring that the search engines index contains a fresh current page of each indexed web page. Its mean that a crawler should continuously crawl the web pages.

III. ARCH ITECTURE OF WEB CRAWLER

For any web crawler, a web page means to identify URLs (Uniform Resource Locators). These URLs are points to different network resources. Basically a web crawler is a program that store and download web pages through URL for a web search engine. Web crawlers are an important

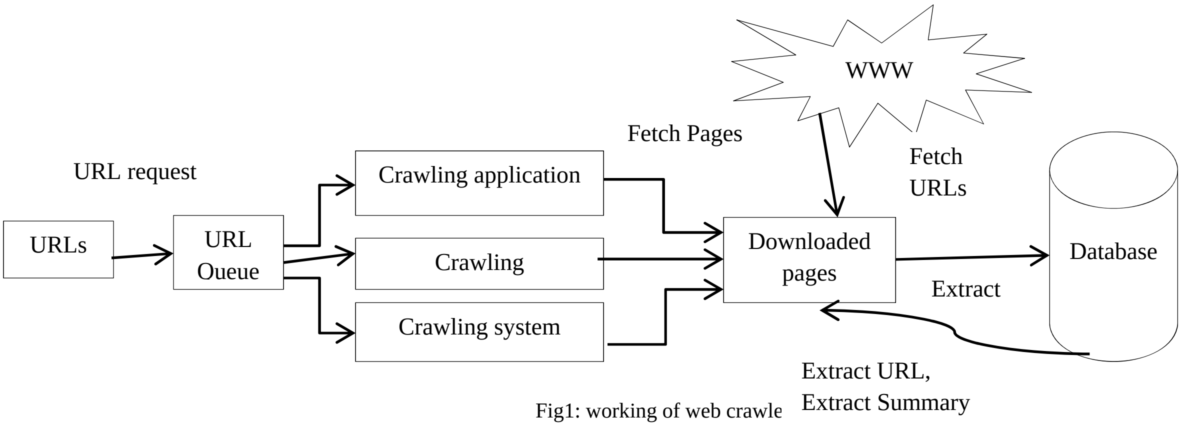

Fig1: working of web crawle

Type	Input	Application
Traditional web crawler	Set of seed URLs	Nods are pages with different URL and a directed edge exist from page p1 to page p2 if there is a hyperlink in page p1 that points to page p2.
Deep web crawler	Set of seed URLs, User and Domain specific data	Nods are pages and a directed edge exists between page p1 to page p2 if submitted a form in page p1 gets the user to page p2.
RIA web crawler	Starting page	Nodes are DOM states of the application and a directed edge exist from DOM d1 to DOM d2 if there is client side java script event, directed by the web crawler that if triggered on d1 changes the DOM state to d2.
Unified model web crawler	Seed of URL	Nod is calculated based on DOM and the URL. An edge are transmission between two state triggered through client side events. Redirecting the browser is a special client side event.

Table1: Different types of Web Crawler

component of web search engine, where they are used to collect the corpus of web pages indexed by the search engine.

A web crawler starts with the initial set of URLs request. In a URL queue where all URLs to be retrieved are kept and prioritized. From this queue the crawler gets a URL, download the pages, extract any URLs in the downloaded page, and put the new URL in the queue. This process is repeated. With the one or more seed URLs, web crawler download the web pages associated with these URLs, extracts any hyperlinks contained in them and continuously to download the web pages identified by these hyperlinks. Figure 1 describes the general process how web crawler works:

A Web crawler starts with a list of seed URLs which are passed to the URLs Queue through URL request. A group of Crawlers listening on the queue then get a subset of the URLs to crawl them. Depending on the requirement the return subsets, such that all URLs in a subset are from the same domain, from distinct domains or are quit random. Each crawler will then fetch the web pages with the help of page downloader. After the downloading the pages it pass it to the extractor which would extract the required data and out links (hyperlinks). The data can be batched to the database and the extract out links (hyperlinks), URL and summary pushed in to the queue.

IV. Categories Of Web Crawlers

A web crawler is divided into five main categories. Table1, explain the different types of web crawler with the working of each crawler.

A. Traditional - web crawler

In the traditional web crawler all the content of a web application is reachable through URLs. Traditional approach for web based application are based on the multi-page interface in which the web application consist of number of multiple unique pages in which each page having a unique URL called URL seed.

Figure 2 show the architecture of a traditional web crawler. The working of this web crawler starts with the Frontier module where this module gets as a input URLs from a set of seed URLs. The seed URLs are passed to the next module which is fetcher. The fetcher module retrieves the contents of the web pages which is associated with the unique URLs form the www. These contents are passed to the URL Extractor. The URL extractor parses the HTML pages and extracts new links from them. New HTML links are passed to the HTML-page filler module and the database. Database interacts with the data files and stores the new links. HTML-page filler filters URL that is not interesting to the web crawler. After that the URL passed to the URL-seen module. The working of this module is to finds the new URLs that are not retrieved and passed them to the Fetcher for retrieval. This looping is continues until all the reachable related links are visited.

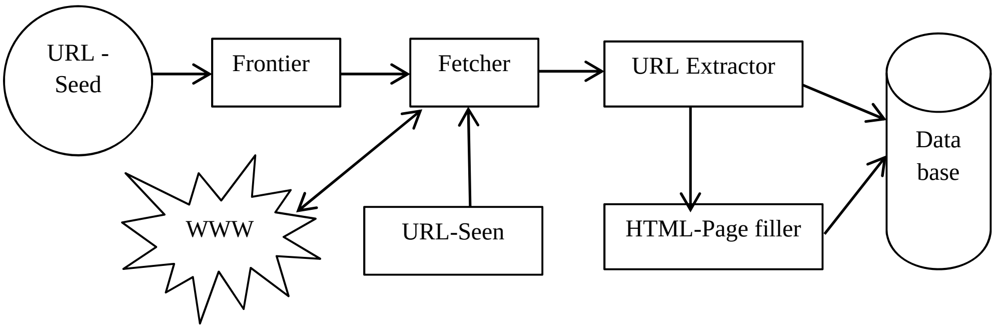

Fig2: Architecture of a Traditional web crawler

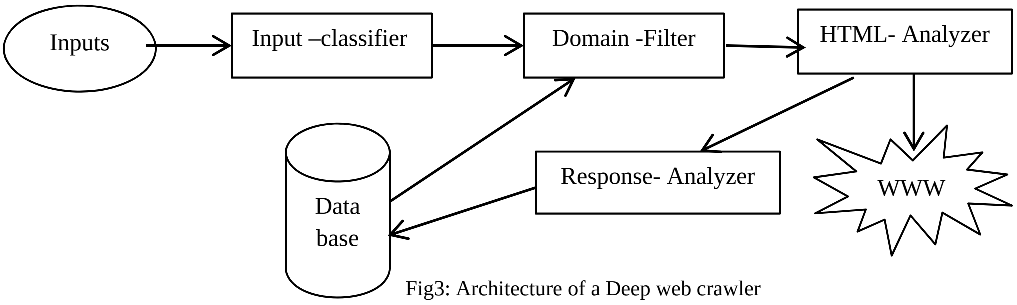

Fig3: Architecture of a Deep web crawler

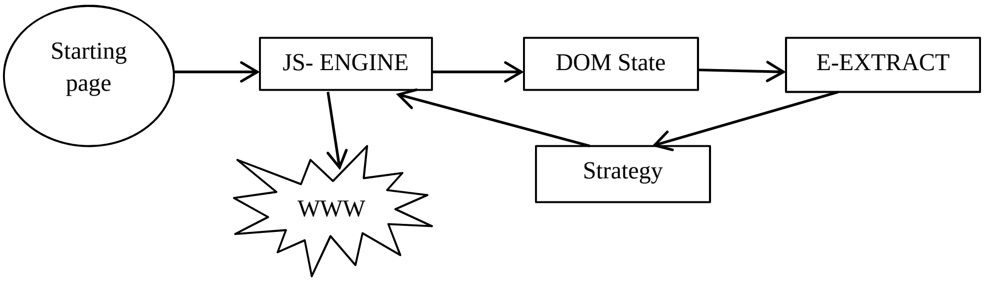

Fig4: Architecture of a RIA web crawler

B. Deep- web crawler

In web, millions of HTML pages daily crawl and index by searching through hyperlinks. Yet a large amount of data is hidden behind web queries and therefore, extensive research has been conducted towards finding and analyzing the hidden-web behind web forms [2]. The information of web content is behind web forms and the client side scripting is referred to as the hidden web, which is estimated to consist of many millions of web pages.

Figure 3 show the architecture of a Deep web crawler. Deep web crawler consists of three main inputs which are set of seed URLs, domain data and the user specific. Input-classifier gets as input set of seed URLs, domain data and the user specific. Input-classifier then chooses the html elements to interact with the various inputs. After that domain-filter uses these data to fill up the html forms and passes the results to the html-analyzer. Html-analyzer submits the form to the web server and fetch the newly web formed page. After that response analyzer parses the page and based on the result, updates, edit the database and this process is continues.

C. RIA- web crawler

RIA crawling is differ from other crawling category specially is differs from the traditional web crawling. For example Crawljax, a RIA crawler that see the user interface into account and used the changes made to the user interface to direct the crawling category. The aim of the Crawljax is to crawl and take a static snapshot of Ajax state for indexing and testing. [3][4]. In the RIA web crawler nodes are DOM states of the application and a directed edge exist from DOM d1 to DOM d2 if there is client side java script event, directed by the web crawler that if triggered on d1 changes the DOM state to d2

Figure 4 show the architecture of a RIA web crawler. Java-script engine starts with a web browser and runs a JS engine. After that it retrieves the starting web page associated with a URL seed and loads it in the web browser. The constructed DOM is passed to the DOM state and to determine if this is the first time the SOM state is seen. After that the DOM state is passed to the e-extractor to extract the JS events from it. When JS event is passed to the strategy module then this module is decides which event is to execute. The chosen event is passed to the java –script engine for the further execution. This process is continues until all the reachable DOM states are seen.

D. Unified model - web crawler

In unified model web crawler nod is calculated based on DOM and the URL. An edge are transmission between two state triggered through client side events. In this crawler redirecting the browser is a special client side event. For this UML using three main models which are user model, object model and the dynamic model. The user model consists of use case diagram, object model represent by the class diagram and dynamic model is represent by the sequence diagram

.

V. Type Of Web Crawler

There are various types of web crawlers in the web depending upon how the web pages are crawled and how each successive web page are retrieved for accessing to the next web page. These are:

A. Breadth first web crawler

In the breadth first web crawler firstly crawl most important pages. This start with a small set of web pages and then explores other pages by following links in the breadth first faction. [5][15]. Implementation of this faction is through breadth first algorithm, which is work on a level by level. The algorithm starts with the seed URL and searches the all the neighbors URL at the same level. If the desired URL is found then the searching is stop and if it is not then searching continues to the next level. This process is repeats until the goal is reached [13][14]. This algorithm is very easy and simple as compare to other.

B. Focused web crawler

Focused web crawlers are tries to download the web pages that are related to each other. It determines how far the given web page is relevant to the specific topic and how they are to proceed further. It collects documents which are specific and relevant to the given topics. That's why focused web crawler also called the topic crawler. It can reduce the amount of network traffic and downloads [6] but due to this search exposure is very large. [7] [8]

C. Incremental web crawler

In incremental web crawler updates an existing set of downloaded pages instead of restarting the crawl from scratch each time whether a page has changed since the last time it was crawled. It solved the problem of the freshness of the page. It incrementally refreshes the existing collection of web pages by visiting them frequently; based upon the estimate as to how often pages change. Due to incremental web crawler only the valuable data is provided by the user, so that network bandwidth is saved and the all the data enrichment is achieved. [9] [10]

D. Distributed crawler

Distributed web crawler based on distributed computing technique where a URL server distributes individual URLs to multiple crawlers in which download web pages in parallel, the crawlers then send the downloaded pages to a central indexer on which hyperlinks are extracted and send back to the URL server to the crawlers. It basically used page rank algorithm for its increased efficiency and quality search [15][16]. Uses of the distributed web crawler it reduce the hardware requirements and increase the total download speed and reliability along with.

E. Parallel crawler

Many search engines run with multiple process in parallel to download the web pages, so that the rate of the downloading the web pages is high. Generally this process of crawling is called parallel web crawler where multiple crawlers are often run in parallel network of workstations. [11]. It depends on the page freshness and page selection [12].

F. Hidden web crawler

In web, millions of HTML pages daily crawl and index by searching through hyperlinks. Yet a large amount of data is hidden behind web queries and therefore, extensive research has been conducted towards finding and analyzing the hidden-web behind web forms [2]. The information of web content is behind web forms and the client side scripting is referred to as the hidden web, which is estimated to consist of many millions of web pages. These web pages are available in a very huge amount of data where a number of distinct domains have become available.

In the world wild web the huge amount of information is available only through surface web. But there are some web pages which are not directly located by the search engines because today in almost all search engines searchable databases are not properly index able or qyeryable. So they appear hidden to the average internet user. These pages are referred to as the Hidden Web or the Deep Web. The deep web is the largest growing area of now days of information on the internet. Total quality content of the deep web is at least 1000-2000 times greater than that of the surface web

VI. Conclusion

The Hidden Web is important because it retrieves high-quality information. Therefore there is a need to implement an indexing technique to be more efficient to index the high quality data. Many research focus on the crawling and indexing algorithms for client side as well as server side DOM state changes, some are the hidden web behind forms, text and search query with their own advantages and disadvantage. For this we will try to design a better, secure framework as compare to others for crawling the hidden web, based on breadth-first and depth-first crawling approach which make more distributed data-structure

In this paper we have studied different crawling technologies where how to crawl searching a hidden web documents with different ways. We explain the working of each web crawler to make quite effective in practice, efficient and higher coverage. We have tried to make our working design as simple as possible. Compared to other crawling technology the focused and hidden web crawling technology is designed for advanced web users on the particular topics. Our future work will be including a complete implementation, analysis and evaluation of this approach.

References

[1] Swati Mali, Dr. B.B Meshram, *"Implementation of multiuser personal web crawler"*, In CSI 6th Int. Conf. on SE(CONSEG), IEEE Conf. Publication, 2012.

[2] S. Raghavan and H. Garcia-Moline, *"Crawling the hidden web"*, In Proc. of the Conf. on Very Large Data Bases, pages 129-138, 2001.

[3] A. Mesbah, E. Bozdag and A.V. Deursen, *"Crawling AJAX by inferring user interface state changes"*, In the Proc. Of 8th Int. Conf. on Web Engineering (ICWE), Washington DC,USA, IEEsE-CSI , pages 122-134, 2008.

[4] A. Mesbah, A. Van Deursen and S. Lenselink, *"Crawling Ajax based web applications through dynamic analysis of user interface state changes"*, In ACM Transaction on the web- TWEB, volume 6, Issue 1,page 3, 2012.

[5] Bing liu, *"Web Content Mining"*, In the 14th Int. WWW Conf. on China, Japan, 2005.

[6] Debashis Hati, Biswajit Sahoo and Amritesh Kumar, *”Adaptive Foousect crawling based on link analysis”*, In the 2ns Int. Conf. on Education Technology and Computer(ICETC), 2010.

[7] Pavalam S.M, S.V. KumarRaja, M. Jawhar and Felix K. Akorli, *“Web crawler in mobile systems”*, In the International Journal of Machine Learning and Computing, volume 2, Issue 4, 2012.

[8] Manas Kanti Dey, Debaker Shamanta, Hasan Md. Suhag Chowdhury and Khandakar Entenam Unayes Ahmed, *“Focused Web crawling: A framework for crawling of country based financial data”*, In the IEEE Conf. of Information and Financial Engineering(ICIFE), 2010.

[9] A.K. Sharma and Ashutosh Dixit, *”Self adjusting refresh time based architecture for incremental web crawler ”*, In the Int. Journal of International Journal of computer science and network security, volume 8, Issue 12,page 349-354, 2008.

[10] Niraj Singhal, A.K. Sharma and Dr. Ashutosh Dixit, *“Design of a Priority based frequency regulated incremental crawler”*. In the Int. Journal of International Journal of computer applications(0975-8887), volume 1, Issue 1,page 42-47, 2010.

[11] Shruti Sharma, A. K. Sharma and J. P. Gupta,*”A noval architecture of Parallel web crawler”*, In the Int. Journal of International Journal of computer applications(0975-8887), volume 14, Issue 4, 2011.

[12] AH Chung Tsol, Daniele Forsali, Marco Gori, Markus Hagenbuchner and Franco Scarselli, *“A simple focused crawler”*, In the Proceeding of 12th Int. WWW Conf., pages 1,2003.

[13] Junghoo cho and Hector Garcia Molina, *“Effective page refresh policies of web crawling”*, In the ACM Transactions on Database Systems, 2003.

[14] Steven s Skiena,*”The Algorithm design”*, Manual 2nd edition , Springer, Verlag London , 2008.

[15] Rahul Kumar, Anurag Jain and Chetan Agarwl,*”Survey of web crawling algorithms”*, In the Int. Journal of Advances in Visions Computing(AVC), volume 1, Issue 2/3, 2014.

[16] Aviral Nigam, NIT-Calicut, *“Web Crawling Algorithms”*, In the International Journal of Computer Science and Artificial Intelligence, volume 4, Issue 3,Pages 63-67, 2014.