WEB CRAWLER FOR MINING WEB DATA

S.AMUDHA, B.SC., M.SC., M.PHIL.,

Assistant Professor, VLB Janakiammal College of Arts and Science,

Tamilnadu, India

amudhajaya@gmail.com

ABSTRACT

Web crawler is a computer program that browses the World Wide Web in a methodical, automated manner or in an orderly fashion. Web crawlers are full text search engines which assist users in navigating the web. Web crawling is an important method for collecting data on, and keeping up with, the rapidly expanding Internet. Users can find their resources by using different hypertext links. A vast number of web pages are continually being added every day, and information is constantly changing. Search engines are used to extract valuable Information from the internet. Web crawlers are the principal part of search engine, is a computer program or software that browses the World Wide Web in a methodical, automated manner or in an orderly fashion. This Paper is an overview of various types of Web Crawlers and the policies like selection, revisit, politeness, and parallelization.

Key Words: **Web Crawler, World Wide Web, Search Engine, Hyperlink, Uniform Resource Locator.**

1. INTRODUCTION

A Web crawler starts with a list of URLs to visit, called the seeds. As the crawler visits these URLs, it identifies all the hyperlinks in the page and adds them to the list of URLs to visit, called the crawl frontier. URLs from the frontier are recursively visited according to a set of policies. If the crawler is performing archiving of websites it copies and saves the information. The archives are regularly stored in such a way they can be viewed, read and navigated as they were on the live web, but are preserved as 'snapshots'.

The large volume implies the crawler can only download a limited number of the Web pages within a given time, so it needs to prioritize to download. The high rate of change can imply the pages might have already been updated or even deleted. The number of possible URLs crawled being generated by server-side software has also made it difficult for web crawlers to avoid retrieving duplicate content. Endless combinations of HTTP GET (URL-based) parameters exist, of which only a small selection will actually return unique content. For example, a simple online photo gallery may offer three options to users, as specified through HTTP GET parameters in the URL. If there exist four ways to sort images, three choices of thumbnail size, two file formats, and an option to disable user-provided content, then the same set of content can be accessed with 48 different URLs, all of which may be linked on the site. This mathematical combination creates a problem for crawlers, as they must sort through endless combinations of relatively minor scripted changes in order to retrieve unique content. The World Wide Web has grown from a few thousand pages in 1993 to more than two billion pages at present. The contributing factor to this explosive growth is the widespread use of microcomputer, increased case of use in computer packages and most importantly tremendous opportunities that the web offers to business .New tools and techniques are crucial for intelligently searching for useful information on the web [10].

Web crawling is an important method for collecting data and keeping up to date with the rapidly expanding Internet. A web crawler is a program, which automatically traverses the web by downloading documents and following links from page to page. It is a tool for the search engines and other information seekers to gather data for indexing and to enable them to keep their databases up to date. All search engines internally use web crawlers to keep the copies of data a fresh. Search engine is divided into different modules. Among those modules crawler module is the module on which search engine relies the most because it helps to provide the best possible results to the search engine. Crawlers are small programs that 'browse' the web

on the search engine's behalf, similarly to how a human user would follow links to reach different pages. Google crawlers run on a distributed network of thousands of low-cost computers and can therefore carry out fast parallel processing. This is why and how Google returns results within fraction of seconds. Web crawlers-also known as robots, spiders, worms, walkers, and wanderers- are almost as old as the web itself. The first crawler, Matthew Gray's Wandered, was written in the spring of 1993, roughly coinciding with the first release of NCSA mosaic [11].

Web crawler, renamed Robots, Spiders and Wanderers appeared almost simultaneously with network. The first web crawler was Wanderer developed by Matthew Gray in 1993. However at that time information scale on the Internet was much smaller. No papers investigated about the technology for dealing with enormous web information which is encountered at present. In the back-end of each search engine, different web crawlers are working. For the reason of competition, the design of those web crawlers is not open [3].

2. WEB CRAWLER

A crawler is a program that downloads and stores Web pages, often for a Web search engine. Roughly, a crawler starts off by placing an initial set of URLs, in a queue, where all URLs to be retrieved are kept and prioritized. From this queue, the crawler gets a URL (in some order), downloads the page, extracts any URLs in the downloaded page, and puts the new URLs in the queue. This process is repeated. Collected pages are later used for other applications, such as a Web search engine or a Web cache [2].

2.1 HISTORY OF WEB CRAWLER

The first Internet "search engine", a tool called "Archie" shortened from "Archives", was developed in 1990 and downloaded the directory listings from specified public anonymous FTP (File Transfer Protocol) sites into local files, around once a month. In 1991, "Gopher" was created, that indexed plain text documents. "Jughead" and "Veronica" programs are helpful to explore the said Gopher indexes. The introduction of the World Wide Web in 1991 has numerous of these Gopher sites changed to web sites that were properly linked by HTML links. In the year 1993, the "World Wide Web Wanderer" was formed the first crawler. Although this crawler was initially used to measure the size of the Web, it was later used to retrieve URLs that were then stored in a database called Wandex, the first web search engine. Another early search engine, "Aliweb" (Archie Like Indexing for the Web) allowed users to submit the URL of a manually constructed index of their site. The index contained a list of URLs and a list of user wrote keywords and descriptions. The network overhead of crawlers initially caused much controversy, but this issue was resolved in 1994 with the introduction of the Robots Exclusion Standard, which allowed web site administrators to block crawlers from retrieving part or all of their sites. Also, in the year 1994, "WebCrawler" was launched the first "full text" crawler and search engine.

The "WebCrawler" permitted the users to explore the web content of documents rather than the keywords and descriptors written by the web administrators, reducing the possibility of confusing results and allowing better search capabilities. Around this time, commercial search engines began to appear with being launched from 1994 to 1997. Also introduced in 1994 was Yahoo!, a directory of web sites that was manually maintained, though later incorporating a search engine. During these early years Yahoo! and Altavista maintained the largest market share. In 1998 Google was launched, quickly capturing the market. Unlike many of the search engines at the time, Google had a simple uncluttered interface, unbiased search results that were reasonably relevant, and a lower number of spam results. These last two qualities were due to Google's use of the PageRank algorithm and the use of anchor term weighting[5].

2.2 WEB CRAWLER WORK

A Search Engine Spider (also known as a crawler, Robot, Search Bot or simply a Bot) is a program that most search engines use to find what's new on the Internet. Google's web crawler is known as GoogleBot. There are many types of web spiders in use, but for now, we're only interested in the Bots that actually "crawls" the web and collects documents to build a searchable index for the different search engines. The program starts at a website and follows every hyperlink on each page. So they can

say that everything on the web will eventually be found and spidered, as the so called "spider" crawls from one website to another. Search engines may run thousands of instances of their web crawling programs simultaneously, on multiple servers.

A web crawler visits one of your pages, it loads the site's content into a database. Once a page has been fetched, the text of your page is loaded into the search engine's index, which is a massive database of words, and where they occur on different web pages. All of this may sound too technical for most people, but it's important to understand the basics of how a Web Crawler works. So, there are basically three steps that are involved in the web crawling procedure. First, the search bot starts by crawling pages of your site. Then it continues indexing the words and content of the site, and finally it visit links (web page addresses or URLs) that are found in your site. When the spider doesn't find a page, it will eventually be deleted from the index. However, some of the spiders will check again for a second time to verify that the page really is offline. The first thing a spider is supposed to do when it visits your website is look for a file called "robots.txt". This file contains instructions for the spider on which parts of the website to index, and which parts to ignore. The only way to control what a spider sees on your site is by using a robots.txt file. All spiders are supposed to follow some rules, and the major search engines do follow these rules for the most part. Fortunately, the major search engines like Google or Bing are finally working together on standards [7].

The general process that a crawler takes is as follows:-

- It checks for the next page to download – the system keeps track of pages to be downloaded in a queue.
- Checks to see if the page is allowed to be downloaded
- Checking a robots exclusion file and also reading the header of the page to see if any exclusion instructions were provided do this. Some people don't want their pages archived by search engines.
- Download the whole page.

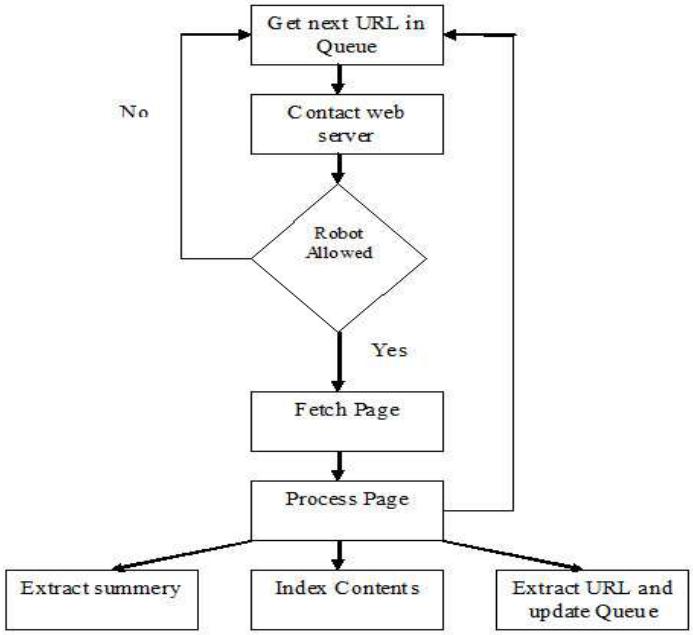

Fig. 1.1 How Crawler Works

- Extract all links from the page (additional web site and page addresses) and add those to the queue mentioned above to be downloaded later.

- 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.
- Optionally filter for things like adult content, language type for the page, etc.
- Save the summary of the page and update the last processed date for the page so that the system knows when it should re-check the page at a later stage.[6]

2.3 POLICIES OF WEB CRAWLER

The behavior of a Web crawler is the outcome of a combination of policies:

- A selection policy which states the pages to download,
- A re-visit policy which states when to check for changes to the pages,
- A politeness policy that states how to avoid overloading Web sites.
- A parallelization policy that states how to coordinate distributed web crawlers.

SELECTION POLICY

Large search engines cover only a portion of the publicly available part. As a crawler always downloads just a fraction of the Web pages, it is highly desirable that the downloaded fraction contains the most relevant pages and not just a random sample of the Web. This requires a metric of importance for prioritizing Web pages. The importance of a page is a function of its intrinsic quality, its popularity in terms of links or visits, and even of its URL. Therefore good selection policy is very important .some of common selection policies are :

Restricting followed links: A crawler may only want to seek out HTML pages and avoid all other MIME types. In order to request only HTML resources, a crawler may make an HTTP HEAD request to determine a Web resource's MIME type before requesting the entire resource with a GET request. To avoid making numerous HEAD requests, a crawler may examine the URL and only request a resource if the URL ends with certain characters such as .html, .htm, .asp, .aspx, .php, .jsp, .jspx or a slash. This strategy may cause numerous HTML Web resources to be unintentionally skipped.

URL normalization: Crawlers usually perform some type of URL normalization in order to avoid crawling the same resource more than once. The term URL normalization, also called URL canonicalization, refers to the process of modifying and standardizing a URL in a consistent manner. There are several types of normalization that may be performed including conversion of URLs to lowercase, removal of "." and ".." segments, and adding trailing slashes to the non-empty path component.

Path-ascending crawling: Some crawlers intend to download as many resources as possible from a particular web site. So path-ascending crawler was introduced that would ascend to every path in each URL that it intends to crawl.For example, when given a seed URL of http://llama.org/hamster/monkey/page.html, it will attempt to crawl /hamster/monkey/, /hamster/, and /. Cothey found that a path-ascending crawler was very effective in finding isolated resources, or resources for which no inbound link would have been found in regular crawling.

Focused crawling: The page for a crawler can also be expressed as a function of the similarity of a page to a given query. Web crawlers that attempt to download pages that are similar to each other are called **focused crawler** or **topical crawlers.** The main problem in focused crawling is that in the context of a Web crawler, we would like to be able to predict the similarity of the text of a given page to the query before actually downloading the page. A possible predictor is the anchor text of links.

Academic-focused crawler: An example of the focused crawlers is academic crawlers, which crawls free-access academic related documents search engine. Other academic search engines are Google Scholar and Microsoft Academic Search etc. Because most academic papers are published in PDF formats, such kind of crawler is particularly interested in crawling PDF, PostScript files, Microsoft Word including their zipped formats. Because of this, general open source crawlers, such as Heritrix, must be customized to filter out other MIME types, or a middleware is used to extract these documents out and import them to the focused crawl database and repository.

RE-VISIT POLICY

The Web has a very dynamic nature, and crawling a fraction of the Web can take weeks or months. By the time a Web crawler has finished its crawl, many events could have happened, including creations, updates, and deletions.
From the search engine's point of view, there is a cost associated with not detecting an event, and thus having an outdated copy of a resource. The most-used cost functions are freshness and age.

Freshness: This is a binary measure that indicates whether the local copy is accurate or not. The freshness of a page p in the repository at time t is defined as:

$$F_p(t) = \begin{cases} 1 & \text{if } p \text{ is equal to the local copy at time } t \\ 0 & \text{otherwise} \end{cases}$$

Age: This is a measure that indicates how outdated the local copy is. The age of a page p in the repository, at time t is defined as:

$$A_p(t) = \begin{cases} 0 & \text{if } p \text{ is not modified at time } t \\ t - \text{modification time of } p & \text{otherwise} \end{cases}$$

Two simple re-visiting policies were studied by Cho and Garcia-Molina:

- Uniform policy: This involves re-visiting all pages in the collection with the same frequency, regardless of their rates of change.
- Proportional policy: This involves re-visiting more often the pages that change more frequently. The visiting frequency is directly proportional to the (estimated) change frequency.

POLITENESS POLICY

Crawlers can retrieve data much quicker and in greater depth than human searchers, so they can have a crippling impact on the performance of a site. Needless to say, if a single crawler is performing multiple requests per second and/or downloading large files, a server would have a hard time keeping up with requests from multiple crawlers.

The use of Web crawlers is useful for a number of tasks, but comes with a price for the general community. The costs of using Web crawlers include:

- Network resources, as crawlers require considerable bandwidth and operate with a high degree of parallelism during a long period of time.
- Server overload, especially if the frequency of accesses to a given server is too high.
- Poorly written crawlers, which can crash servers or routers, or which download pages they cannot handle.
- Personal crawlers that, if deployed by too many users, can disrupt networks and Web servers.

PARALLELIZATION POLICY

A parallel crawler is a crawler that runs multiple processes in parallel. The goal is to maximize the download rate while minimizing the overhead from parallelization and to avoid repeated downloads of the same page. To avoid downloading the same page more than once, the crawling system requires a policy for assigning the new URLs discovered during the crawling process, as the same URL can be found by two different crawling processes.

3. ARCHITECTURE OF WEB CRAWLER

The generalized architecture of web crawler has three main components: a **frontier** which stores the list of URL's to visit, **Page Downloader** which download pages from WWW and **Web Repository** receives web pages from a crawler and stores it in the database. Here the basic processes are briefly outline.

Crawler frontier: - It contains the list of unvisited URLs. The list is set with seed URLs which may be delivered by a user or another program. Simply it's just the collection of URLs. The working of the crawler starts with the seed URL. The crawler retrieves a URL from the frontier which contains the list of unvisited URLs. The page corresponding to the URL is fetched from the Web, and the unvisited URLs from the page are added to the frontier. The cycle of fetching and extracting the URL continues until the frontier is empty or some other condition causes it to stop. The extracting of URLs from the frontier based on some prioritization scheme.

Page downloader: - The main work of the page downloader is to download the page from the internet corresponding to the URLs which is retrieved from the crawler frontier. For that, the page downloader requires a HTTP client for sending the HTTP request and to read the response. There should be timeout period needs to set by the client in order to ensure that it will not take unnecessary time to read large files or wait for response from slow server. In the actual implementation, the HTTP client is restricted to only download the first 10KB of a page.

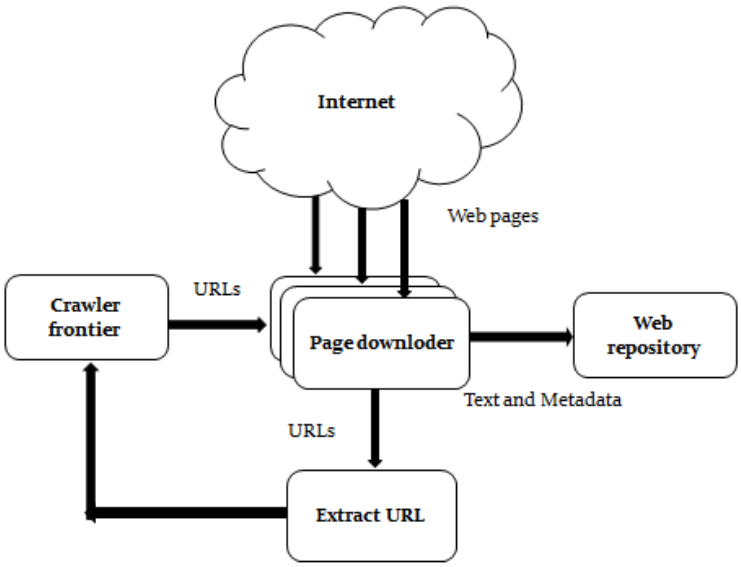

Fig 1.2 Architecture of Web Crawler

Web repository: - It use to stores and manages a large pool of data **"objects,"** in case of crawler the object is web pages. The repository stores only standard HTML pages. All other media and document types are ignored by the crawler. It is theoretically not that different from other systems that store data objects, such as file systems, database management systems, or information retrieval systems. However, a web repository does not need to provide a lot of the functionality like other systems, such as transactions, or a general directory naming structure [12]. It stores the crawled pages as distinct files. And the storage manager stores the up-to-date version of every page retrieved by the crawler.

4. TYPES OF WEB CRAWLER

Different strategies are being employed in web crawling. These are as follows.

4.1 GENERAL-PURPOSE WEB CRAWLER

General-purpose web crawlers collect and process the entire contents of the Web in a centralized location, so that it can be indexed in advance to be able to respond to many user queries. In the early stage when the Web is still not very large, simple or random crawling method was enough to index the whole web. However, after the Web has grown very large, a crawler can have large coverage but rarely refresh its crawls, or a crawler can have good coverage and fast refresh rates but not have good ranking functions or support advanced query capabilities that need more processing power. Therefore, more advance crawling methodologies are needed due to the limited resources like time and network and width [9].

4.2 ADAPTIVE CRAWLER

Adaptive crawler **is** classified as an incremental type of crawler which will continually crawl the entire web, based on some set of crawling cycles. The adaptive model used would use data from previous cycles to decide which pages should be checked for updates. Adaptive Crawling can also be viewed as an extension of focused crawling technology. It has the basic concept of doing focus crawling with additional adaptive crawling ability. Since the web is changing dynamically, adaptive crawler is designed to crawl the web more dynamically, by additionally taking into consideration more important parameters such as freshness or up to date-ness, whether pages are obsolete, the way pages change, when pages will change, how often pages change and etc. These parameters will be added into the optimization model for controlling the crawling strategy, and contribute to defining the discrete time period and crawling cycle. Therefore, it is expected that more cycles the adaptive crawler goes in operation, more reliable and refined will the output results [9].

4.3 BREADTH FIRST CRAWLER:

It starts with a small set of pages and then explores other pages by following links in the breadth-first fashion. Actually web pages are not traversed strictly in breadth first fashion but may use a variety of policies. For example it may crawl most important pages first [6].

4.4 INCREMENTAL WEB CRAWLERS:

An incremental crawler , is one, which updates an existing set of downloaded pages instead of restarting the crawl from scratch each time. This involves some way of determining whether a page has changed since the last time it was crawled. A crawler, which will continually crawl the entire web, based on some set of crawling cycles. An adaptive model is used, which uses data from previous cycles to decide which pages should be checked for updates, thus high freshness and results in low peak load is achieved[6].

4.5 HIDDEN WEB CRAWLERS:

A lot of data on the web actually resides in the database and it can only be retrieved by posting appropriate queries or by filling out forms on the web. Recently interest has been focused on access of this kind of data called “deep web” or “hidden web”. Current day crawlers’ crawl only publicly indexable web (PIW) i.e., set of pages which are accessible by following hyperlinks ignoring search pages and forms which require authorization or prior registration. In reality they may ignore huge amount of high quality data, which is hidden behind search forms [6].

4.6 PARALLEL CRAWLERS:

As the size of the Web grows, it becomes more difficult to retrieve the whole or a significant portion of the Web using a single process. Therefore, many search engines often run multiple processes in parallel to perform the above task, so that download rate is maximized. This type of crawler is known as a parallel crawler [6].

4.7 DISTRIBUTED WEB CRAWLER:

This crawler runs on network of workstations. Indexing the web is a very challenging task due to growing and dynamic nature of the web. As the size of web is growing it becomes mandatory to parallelize the process of crawling to finish the crawling process in a reasonable amount of time. A single crawling process even with multithreading will be insufficient for the situation. In that case the process needs to be distributed to multiple processes to make the process scalable. It scales up to several hundred pages per second. The rate at which size of web is growing it is imperative to parallelize the process of crawling. In distributed web crawler a URL server distributes individual URLs to multiple crawlers, which download web pages in parallel. The crawlers then send the downloaded pages to a central indexer on which links are extracted and sent via the URL server to the crawlers. This distributed nature of crawling process reduces the hardware requirements and increases the overall download speed and reliability. FAST Crawler is a distributed crawler, used by Fast Search & Transfer [6].

5. ARCHITECTURE FOR DATA EXTRACTION

There are mainly five components of the Proposed Architecture of Data Extraction as shown below:

QUERY INTERFACE:- A Query Interface is designed as an entrance to the Project which is used by the user to enter the query. It is the user interface where the user has to input the query to get the required data. This interface contains a Text Box where the user has to place the query.

RESULT PAGE:- After the query has been processed by the query processor and the Attribute-value based Searcher, the required data is returned to the user in the form of result page. QUERY PROCESSOR:- The user queries are processed by the Query Processor to fetch the desired data and return it back to the user with desired results. It has the following two components:

EXTRACT QUERY STRING:- The Extract Query String module of the Query Processor extracts the Query String from the Query Interface and passes it to the Attribute-value based Tokenization module.

ATTRIBUTE-VALUEBASED TOKENIZATION:- It tokenizes the Query string into a number of tokens. It passes the various tokens to the Attribute-value based Searcher for searching.

ATTRIBUTE-VALUE BASED SEARCHER:- Attribute-value based Searcher matches the tokens respective to the domain with the various attribute and their corresponding values stored in the Data Repository to retrieve the postings lists which are then intersected to return the result page to the user.

DEEP WEB DATABASE:- The Data Repository stores the various postings lists of the attribute values along with the attributes and their respective domain. The Search interface allows a user to search some set of items without altering them. The user enters a query by typing to get the data of interest. Result Page is a page containing data of interest. This Project implements attribute-value based domain-specific searching technique for hidden web.

OFFLINE SEARCH:-In this project user can search either online or offline. For online searching process is described above. In offline searching, user can enter the query in textboxes and then search in the local database for the result. During online searching when result page is displayed then that result is also saved in the local database that is already designed for saving the retrieved data. Database is designed first according to the retrieved attributes. Data extracted and inserted under the respective column names in the table and values under each column will be stored as row [4].

CONCLUSION

The indication of different crawling technologies has been offered in this paper. When only information about a predefined topic set is required, "focused crawling" technology is being used. This paper describes about Different types of Web crawler and the policies used in the web crawlers. Different data structures used in web crawler and working is studied. A web crawler is a way for the search engines and other users to regularly ensure that their databases are up to date. Web crawlers are a central part of search engines, and details on their algorithms and architecture are kept as business secrets. New Modification and extension of the techniques in Web crawling should be next topics in this area of research.

REFERENCES

1. Bhavin M. Jasani, C. K. Kumbharana," Analyzing Different Web Crawling Methods", International Journal of Computer Applications (0975 –8887) Volume 107 –No 5, December2014.
2. Dhiraj Khurana1, Satish Kumar2, "Web Crawler: A Review", IJCSMS International Journal of Computer Science & Management Studies, Vol. 12, Issue 01, January 2012 ISSN (Online): 2231 –5268.
3. Hongyan Zhao," Research on Detection Algorithm of WEB Crawler", International Journal of Security and Its Applications Vol.9, No.10 (2015), pp.137-146.
4. Komal Taneja, Puneet Rani," Hidden Web Data Extractor", International Journal of Computer Science information and Engg., Technologies ISSN 2277-4408 || 01072014-001.
5. Md. Abu Kausar,V. S. Dhaka,Sanjeev Kumar Singh, "Web Crawler: A Review", International Journal of Computer Applications (0975 –8887) Volume 63–No.2, February 2013.
6. Mini Singh Ahuja, Dr Jatinder Singh, Bal Varnica, "Web Crawler: Extracting the Web Data", International Journal of Computer Trends and Technology (IJCTT) – volume 13 number 3 – Jul 2014.
7. Mridul B. Sahu, Prof. Samiksha Bharne, "A Survey On Various Kinds Of Web Crawlers And Intelligent Crawler", International Journal of Scientific Engineering and Applied Science (IJSEAS) – Volume-2, Issue-3, March 2016 ISSN: 2395-3470.
8. Nikita Suryavanshi; Deeksha Singh & Sunakashi,"Offline/Online Semantic Web Crawler" International Journal of Research p-ISSN: 2348-6848 e-ISSN: 2348-795X Volume 03 Issue 09 May 2016.
9. S.S. Dhenakaran and K. Thirugnana Sambanthan," WEB CRAWLER - AN OVERVIEW", International Journal of Computer Science and Communication Vol. 2, No. 1, January-June 2011, pp. 265-267.
10. Shalini Sharma," Web Crawler", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 4, April 2014 ISSN: 2277 128X.
11. Trupti V. Udapure1, Ravindra D. Kale2, Rajesh C. Dharmik3," Study of Web Crawler and its Different Types", OSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 1, Ver. VI (Feb. 2014), PP 01-05.

BIOGRAPHY

Amudha.s was born in Tamilnadu, India, in 1985. I received the B.Sc degree in Computer Science from Bharathiar University of sri Ramallinga Sowdambigai College, India, in 2003, and the M.Sc degree in Computer Science from Bharathiar University, India in 2008. M.Phil degree in Computer Science from Bharathiar University of VLB Janakiammal College of arts and Science, India in 2011 respectively. In 2008 joined the Department of Computer Applications, VLB Janakiammal College of Arts and Science as an Assistant Professor.