*IOSR Journal of Computer Engineering (IOSR-JCE)*
*e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 2, Ver. VIII (Mar-Apr. 2014), PP 79-82*
*www.iosrjournals.org*

# Crawler with Search Engine based Simple Web Application System for Forum Mining

M.Maheswari[1], N.Tharminie[2]
[1]*(M.E. CSE, Magna College of Engineering, Chennai, INDIA)*
[2]*(HOD of CSE, Magna College of Engineering, Chennai, INDIA)*

***Abstract :*** *Now-a-days the growth of online users increased infinitely depending upon the information in web sources. Web mining is an important term to manage the data from web which has different categorization as structure, content, usage. In this paper web site fetcher that is the crawler and search engine design are developed using .NET based application. The URL, content those type of crawling approches clearly explained by that system in a varient way. The designed crawler performs two functions, URL Crawling (structure mining) by page classification and Content Crawling (content mining) by Pattern clustering. This type of Crawler design is supported for providing efficient way to retrieve the forum data to small scale search engine as possible.*

***Keywords :*** *Content mining, Structure mining, World Wide Web, Web Forum Crawler, Web Search Engine.*

## I. Introduction

Data Mining is the process of extracting patterns from data. It is currently used in a wide range of profiling practices, such as marketing, surveillance, fraud detection, and scientific discovery. Web mining is a technique that applies data mining techniques to analyze different sources of data in the web. The World Wide Web has becoming one of the most valuable resources for information retrievals and knowledge discoveries. Web mining technologies are the right solutions for knowledge discovery on the Web. The knowledge extracted from the Web can be used to raise the performances for Web information retrievals, question and answering, and Web based data warehousing. Web mining is moving the World Wide Web toward a more useful environment in which users can quickly and easily find the information they need. It includes the discovery and analysis of data, documents, and multimedia from the World Wide Web. Web mining uses document, hyperlink structure, and usage statistics to assist users in meeting their information needs.

The Web itself and search engines contain relationship information about documents. Web mining is the discovery of these relationships and is accomplished within three sometimes overlapping areas. Content mining first, Search engines define content by keywords. Finding contents keywords and finding the relationship between web pages content and a users query content is content mining. Hyperlinks provide information about other documents on the web thought to be important to another document. These links add depth to the document, providing the multi-dimensionality that characterizes the web. Mining this link structure is the second area of web mining. Finally there is a relationship to other to other documents on the web that are identified by previous searches. These relationships are recorded in logs of searches and accesses. Mining these logs is the third area of web mining i.e. Web Content Mining describes the discovery of useful information from the web contents, data and documents to content publish on Internet, usually as HTML. Web Structure Mining operates on the webs hyperlink structure. Web Usage Mining is the automatic discovery of user interactions with a web server.

A forum consists of a tree like directory structure. The top end is "Categories". A forum can be divided into categories for the relevant discussions. Under the categories are sub-forums and these sub-forums can further have more sub forums. Web search engines and some other sites use Web crawling or spidering software to update their web content or indexes of others sites web content. Crawlers can validate hyperlinks and HTML code. Web crawlers can copy all the forum pages they visit for later processing by a search engine that indexes the downloaded pages so that users can search them much more quickly.

Based on the above mentioned details in this paper the design of crawler starts off by placing an initial set of URLs, so in a queue where all URLs to be retrieved are kept and prioritized. From this queue, the crawler gets a URL in some order and downloads the page, extracts any URLs in the downloaded page, and puts the new URLs in the queue. This process is repeated until the crawler decides to stop. Collected pages are later used for other applications, such as a Web search engine or a Web cache.

## II. Related Work

Brin.S and Page.L [2], Web Search Engine it is a large-scale search engine which makes heavy use of the structure present in hypertext, for example is Google. Google is designed to crawl and index the web efficiently and produce much more satisfying search results than all. It answers tens of millions of queries every day. This paper provides an in-depth description of large-scale web search engine. It makes heavy use of the structure present in hypertext. But it doesn't deal how to effectively deal with uncontrolled hypertext collections, where anyone can publish anything they want. The technical challenges involved with using the additional information present in hypertext to produce better search results.

J. Zhang, M.S. Ackerman, and L. Adamic [5], expertise location was identified and automatically analyze who knows what. In this method follows the ranking based algorithm it provide accurate result for only large scale data. Previesely which has content based algorithm comparison to the different organizations which extract the content information from that email community. The issue under that type is performance wise low result for small scale data.

Vidal Caj.R, Yang. J.M, Lai.W, Wang.Y, and Zhang.L [3], iRobot has an intelligence to understand the content and the structure of a forum site, and then decide how to choose traversal paths among different kinds of pages. Furthermore, it also achieves the following advantages: Significantly decreases the duplicate and invalid pages, Saves substantial network bandwidth and storage as it only fetches informative pages from a forum site, It provides a great help for further indexing and data mining, Effectiveness: it intelligently skip most invalid and duplicate pages, while keep informative and unique ones, Efficiency: iRobot only need a few pages to rebuild the sitemap. It is also have some disadvantages such as, it follow a spanning tree traversal, so it didn't allow more than one path from starting page to ending page. It doesn't deal how to design a repository for forum archiving.

U. Schonfeld and N. Shivakumar [4], Sitemap construction was done to evaluate various web sites. Compare the crawling of classic discovery with sitemap protocol and it makes the hidden data to visible. This kind of protocol does provide them with hints that help Web crawlers do a better job of crawl the site. The experimental result shown only for large scale data. The drawback of this process is no use of crawler algorithm in that.

H.S. Koppula, K.P. Leela, A. Agarwal, K.P. Chitrapura, S. Garg, and A. Sasturkar [7], Web page de-duplication process was considered in this paper. Mining the crawl logs to represent the data as original. In that use of decision tree algorithm and some mapping data result as experimental. Different way of tokenizing algorithm is used for rule generation. But the major issue in that in without fetching the content it checks the data as valuable or not. It was somewhat difficult to all kind of pages.

- From this survey result each paper has some demerits no one method is common to both large and small scale data evaluation.
- Some of the approaches are working like a black box.
- In what way means those specified algorithms and approaches are not show their internal operations as clear as follows;

There is no multithreaded downloader design available to download and stores Web pages. As the size of the web grows, it becomes more difficult to retrieve the whole or significant portion of the web information using a single process. Downloading rate is minimized and downloading time is maximized due the above reason. Storage of static pages is not usually seen in any of the search engines. Error pages when encountered are not stored separately. Page content crawling is not considered. In some cases when searching the web it retrieves all the contents. It doesn't provide an optimization result based on the user Search which also provide some duplicated data and not care about the time.

## III. Proposed Method And System Architecture

All the drawbacks of existing systems are overcome in proposed system of this paper. In this, the presents of Parallel Crawler approach to improve the crawler performance that means Multi- threaded Downloader Supported. Crawler with multi-threaded downloader is responsible for starting threads and obtaining the information about the website being fetched. Multiple processes are run in parallel to perform the above task, so that download rate is maximized and downloading time is minimized. URL Type is recognized easily by the algorithm specified path. Algorithm Specific Path (EIT-> URL Crawling for Index, Page flipping, Thread Detection). URL crawling by EIT path Specification means, DOM tree structure. i.e. Anchor text length and their depth value detection (link, hyperlink analysis). So the Page Structure not affected. Static pages are stored in user desired folder to make the local search engine. Any difficulties encountered can be viewed separately in the Error's view. Search Content gives the results faster.

### 3.1 System Overview

The crawler working function is explained from the different kind of layers in the system architecture.

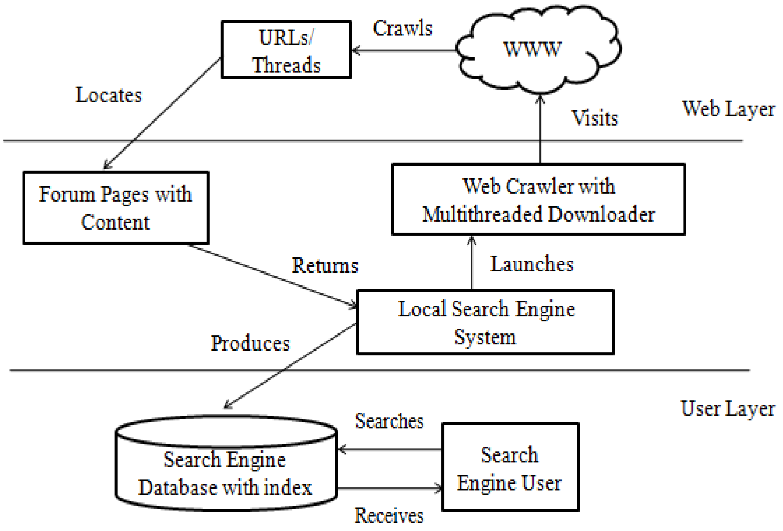


Fig.1: Web crawler with multithreaded downloader

The top level layer is for retrieving details of web sources such as WWW or Internet. Which has URL and their data i.e. URI of web pages. The middle layer provides the interaction with both user and web. Download all needed details to user by that crawler launched by the system server. The bottom layer performs the database functions to user as well as server. The user interacts with database to search and retrieve the information and the search engine update their database index based on the crawled details.

### 3.2 Algorithm Summary

The crawler performs the function of URL and content crawling. In URL crawling the URL's are maintained as index on the database. Before that using page classification approach the collected web pages are classified by algorithm specified path. The algorithm follows entry page- index-page – thread page type of implicit paths to navigate the pages. For retrieving effective and efficient search result at the search time the crawler design is must be considered by this specified path flow. From that initially classify the index page and their sub pages i.e. page flipping URL's by some URL detection algorithms. Find the thread page from the anchor text length and value. Then in each part avoid the duplicated data based on condition list to restrict file type as MIME. After that store those URL's in allotted folder to give the details for local server database. Content crawling which is the major work of the paper, URL relevant page is downloading to refer their content search from the web source using pattern clustering approach. Grouping the URL's of similar sites by alignment algorithm and thread URL has the thread i.e. user Posted Content from that stored pages.

### 3.3 Module Specification

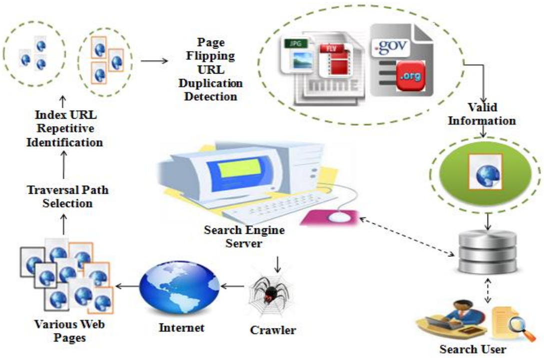
This project has the modules details as follows,

- Simple Crawler design for URL Detection URL Discovery
- Crawled Page Storage
- Local Search Engine System

Initially a simple crawler design for both index and page flipping URL detection and make them as training data. When the entry URL entered which will check with previous data and produce the resultant URL that is stored in the database index. If the user search data is similar to that database detail retrieve the needed content by the URL and HTML based search engines.

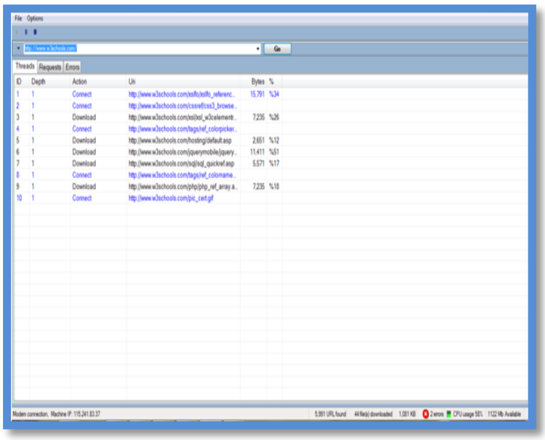
## IV. Expected Results

This resultant diagram gives the overview of the final working system input, output process and their flow. The dotted circles represent URL's and the straight lines for representing internal process of the system. Finally the external function flow shown by the dotted lines between search user, database and search engine server. The search engine user gives their needed data as thread the resultant URL be the type of thread URL.



INTERNET <→ CRAWLER < → SERVER< → DB < → USER

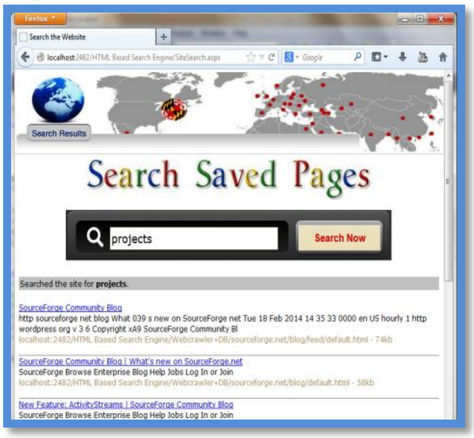
Fig. 2: Input and Output Flow



| ID | Depth | Action | Uri | Bytes | % |
|---|---|---|---|---|---|
| 1 | 1 | Connect | http://www.w3schools.com/xslfo/xslfo_referenc... | 15,791 | %34 |
| 2 | 1 | Connect | http://www.w3schools.com/cssref/css3_browse... | | |
| 3 | 1 | Download | http://www.w3schools.com/xsl/xsl_w3element... | 7,235 | %26 |
| 4 | 1 | Connect | http://www.w3schools.com/tags/ref_colorpicker... | | |
| 5 | 1 | Download | http://www.w3schools.com/hosting/default.asp | 2,651 | %12 |
| 6 | 1 | Download | http://www.w3schools.com/jquerymobile/jquery... | 11,411 | %51 |
| 7 | 1 | Download | http://www.w3schools.com/sql/sql_quickref.asp | 5,571 | %17 |
| 8 | 1 | Connect | http://www.w3schools.com/tags/ref_colorname... | | |
| 9 | 1 | Download | http://www.w3schools.com/php/php_ref_array.a... | 7,235 | %18 |
| 10 | 1 | Connect | http://www.w3schools.com/pic_cert.gif | | |

Modem connection, Machine IP: 115.241.83.37    5,991 URL found    44 file(s) downloaded    1,081 KB    2 errors    CPU usage 58%    1122 Mb Available

Fig. 3: URL Crawling



Searched the site for **projects**.

SourceForge Community Blog
http sourceforge net blog What 039 s new on SourceForge net Tue 18 Feb 2014 14 35 33 0000 en US hourly 1 http wordpress org v 3 6 Copyright xA9 SourceForge Community Bl
localhost:2482/HTML Based Search Engine/Webcrawler+DB/sourceforge.net/blog/feed/default.html - 74kb

SourceForge Community Blog | What's new on SourceForge.net
SourceForge Browse Enterprise Blog Help Jobs Log In or Join
localhost:2482/HTML Based Search Engine/Webcrawler+DB/sourceforge.net/blog/default.html - 58kb

New Feature: ActivityStreams | SourceForge Community Blog
SourceForge Browse Enterprise Blog Help Jobs Log In or Join

Fig. 4: Content Crawling

User entered data as a content that automatically forms like query string behind the URL of a site. Check with DB index. Which have the URL's in a Queue. Content Related URL search and download page. That page having the thread (user content details). Those indexing of URL and Downloading page process done by crawler.

## V. Conclusion & Future Scope

In this work no need to consider page score and weights for analyzing the web pages. By using some data mining approaches with web sources the crawler function was discussed and developed. The simple design of back end component to the search engine developed in the first part of module implementation. This is the process of URL crawling. The search engine module and their function are discussed. Their detailed functional part shown from the resultant diagram. This is the process of content crawling. Use this type of backend design (crawler) to local search engine for one particular Organization like College which has the group of institutions in a real world. In what way means Download and maintain their web pages using this local server. Filter some kind of Links which is unwanted while surfing their request based on some more combination of algorithms and techniques with this.

## References

[1] Jingtian Jiang, Xinying Song, Nenghai Yu, Member, IEEE, and Chin-Yew Lin, Member, IEEE "FoCUS: Learning to Crawl Web Forums" *IEEE Transactions On Knowledge And Data Engineering*, vol. 25, no. 6, june 2013

[2] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine." *Computer Networks and ISDN Systems*, vol. 30, nos. 1-7, pp. 107-117, 1998.

[3] R. Cai, J.-M. Yang, W. Lai, Y. Wang, and L. Zhang, "iRobot: An Intelligent Crawler for Web Forums," *Proc. 17th Int'l Conf. World Wide Web*, pp. 447-456, 2008.

[4] U. Schonfeld and N. Shivakumar, "Sitemaps: Above and Beyond the Crawl of Duty," *Proc. 18th Int'l Conf. World Wide Web*, pp. 991-1000, 2009.

[5] J. Zhang, M.S. Ackerman, and L. Adamic, "Expertise Networks in Online Communities: Structure and Algorithms*," Proc. 16th Int'l Conf. World Wide Web*, pp. 221-230, 2007.

[6] G.S. Manku, A. Jain, and A.D. Sarma, "Detecting Near-Duplicates for Web Crawling," *Proc. 16th Int'l Conf. World Wide Web*, pp. 141-150, 2007.

[7] H.S. Koppula, K.P. Leela, A. Agarwal, K.P. Chitrapura, S. Garg, and A. Sasturkar, "Learning URL Patterns for Webpage De-Duplication," *Proc. Third ACM Conf. Web Search and Data Mining*,pp. 381-390, 2010.

[8] X.Y. Song, J. Liu, Y.B. Cao, and C.-Y. Lin, "Automatic Extraction of Web Data Records Containing User-Generated Content," *Proc. 19th Int'l Conf. Information and Knowledge Management*, pp. 39-48, 2010.

[9] InternetForumhttp://en.wikipedia.org/wiki/Internet_forum, 2012, Web crawler http:// *en.wikipedia.org/wiki/web_crawler*