Crahid: A New Technique for Web Crawling In Multimedia Web Sites

[1,]Dr. Ammar Falih Mahdi , [2,]Rana Khudhair Abbas Ahmed
[1,]*Rafidain University college./Software Engineering Department*
[2,] *Rafidain University College/ Computer Technical Engineering Department*

ABSTRACT
With the quick growth and the huge amount of data that propagate in the web, we spend a lot of our time in finding the exact required information from the huge information retrieved from the web crawlers of search engines. Therefore, a special software is required to collect and find the exact required information and save our time and effort in finding what is good from the huge amount of retrieved data from the web. In our research, we proposed a software called "CRAHID" which uses a new technique to crawl (image, sound, text and video) depending on an information hiding technique for describing media. We supposed a new media format which describes media using hidden information to maintain time and effort for finding the exact information retrieved from the crawler.

KEYWORDS: *Computer, Crawling, Information hiding, Multimedia, Search , Technique, Web.*

I. INTRODUCTION

A crawler is a computer program which traverses a network and tries to capture the content of the documents within. In general the "crawled" network is the World Wide Web, and the documents are web pages. Due to the fact that the structure of the WWW is unknown, therefore crawlers implement different strategies to traverse it. These strategies calculate which links the crawler should follow and which not. This way the crawler only crawl a relative small subset of the whole network. Policies discussed later make sure that the relevant pages are contained within this crawled subset. Crawls the whole WWW is not possible due to its size and growth [1].

II. REQUIREMENTS FOR A CRAWLER [1]

[1] **Flexibility:** The design of the crawler should make it possible to use the same parts in different scenarios. For example change the crawler application to implement a different strategy with keeping the crawler system the same.

[2] **Robustness:** Due to that fact that the crawler downloads documents from a large variety of servers, it has to be able to deal with strange server behavior, badly formed HTML and other glitches that can occur on a network that is open to so many contributors. Also because a one iteration of the crawling can takes weeks, or is a continuous process the crawler should be able to handle crashes and network failures.

[3] **Etiquette and Speed Control**: Following standard conventions like the robots.txt and robot specific meta-data is very important for any serious crawler.

[4] **Manageability / Reconfigurability:** The administrator of a crawler should be able to change settings and add elements to a blacklist and monitor the crawler state and memory usage during the crawling process. It should also be possible make changes or adaption to the software after a crash or restart to fix bugs or make improvements based on knowledge gathered during the crawl.

III. WEB CRAWLER STRATEGIES

There are different strategies used in Web crawling, we explain some of these strategies in the section below:

3.1 Breadth First Search Algorithm:

This algorithm aims in the uniform search across the neighbor nodes. It starts at the root node and searches the all the neighbor nodes at the same level. If the objective is reached, then it is reported as success and the search is terminated. If it is not, it proceeds down to the next level weeping the search across the neighbor nodes at that level and so on until the objective is reached. When all the nodes are searched, but the objective is not met then it is reported as failure. Breadth first is well suited for situations where the objective is found on the shallower parts in a deeper tree. It will not perform so well when the branches are so many in a

game tree, especially like chess game and also when all the path leads to the same objective with the same length of the path.

3.2 Depth First Search Algorithm

This is the powerful technique that systematic traverse through the search by starting at the root node and traverse deeper through the child node. If there are more than one child, then priority is given to the left most child and traverse deep until no more child is available. It is backtracked to the next unvisited node and then continues in a similar manner. This algorithm makes sure that all the edges are visited once breadth. It is well suited for search problems, but when the branches are large then this algorithm takes might end up in an infinite loop.

IV. WEB CRAWLER ARCHITECTURE

Web crawlers are a central part of search engines, and details on their algorithms and architecture are kept as business secrets. When crawler designs are published, there is often an important lack of detail that prevents other from reproducing the work. There are also emerging concerns about "search engine spamming", which prevent major search engines from publishing their ranking algorithms [3]. The typical high-level architecture of Web crawlers is shown in Figure (1).

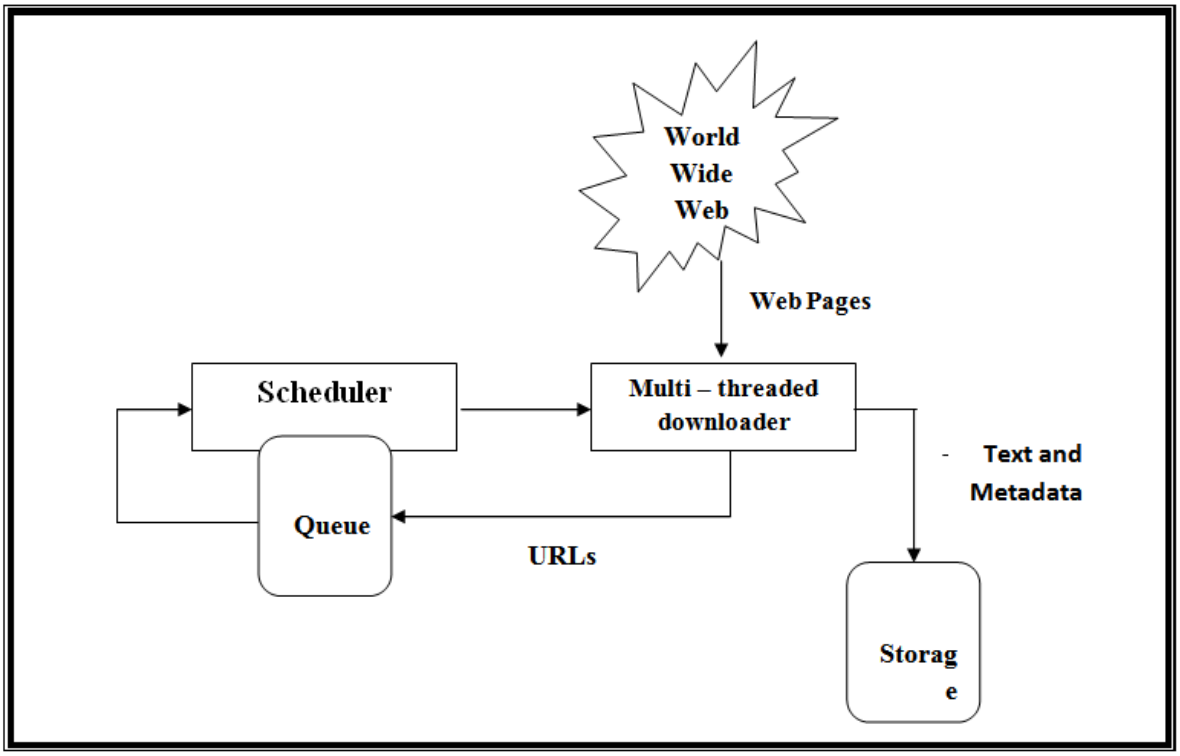


Figure (1): Typical high-level architecture of a Web crawler [4].

While it is fairly easy to build a slow crawler that downloads a few pages per second for a short period of time, building a high-performance system that can download hundreds of millions of pages over several weeks presents a number of challenges in system designed, I/O and network efficiency, and robustness and manageability [4]. Modern crawlers require a flexible robust and effective architecture to accomplish their task of crawling millions of web pages and download hundreds of gigabyte of data in a reasonable amount of time. Depending on the crawlers strategies the requirements may differ. A breadth-first crawler will need to store which URLs have already been crawled, whereas a link analysis crawler may need to store the graph of the crawled network. These requirements must reflect in the architecture of the individual crawler. But on the other hand there are some tasks every crawler will have to perform, allowing for a more general formulation of crawlers architecture [1]. The design of a crawler can be separated into two mayor components, the crawling application and the crawling system. The crawling application decides on what URLs to crawl and analyses the downloaded data supplied by the crawling system.The crawling system is responsible for the actual task of downloading data, taking care of crawler policies e.g. crawler.txt and other crawler relevant meta-data. The crawling application is responsible of implementing the crawling strategy.

With this basic concept the same crawling system can be used to implement a wide variety of crawling strategies by only changing the crawler application [1]. The crawler system seems very primitive, but when taking in account that thousands of pages have to be downloaded every second, this task becomes in fact very complex. In a real-world environment the crawler system will probably be distributed over many servers, and even geographical over different parts of the world to increase download peed. In many cases even the whole crawling application will be distributed over a many servers spread over different locations [1].

V. COMPONENTS OF A WEB CRAWLER

a. **Multi-threaded Downloader**:-It downloads documents in parallel by various parallel running threads.
b. **Scheduler**:-Selects the next URL to be downloaded.
c. **URL Queue**:-A queue having all URL of page.
d. **Site-ordering Module**:-It score the site based on various factors and order them based on the score.
e. **New ordered queue**: - URL s sorted based on their score.
f. **World Wide Web**: - Collection of interlinked documents.
g. **Storage**:-to save the downloaded documents.

VI. INFORMATION HIDING IN MULTIMEDIA FILES

In General, there are techniques for hiding information in multimedia files, and usually this information is very important to be secret. With this technique we can retrieve the information from the files without changing the size of file before and after hiding the information in it. There are many algorithms for hiding information in multimedia files. One of these algorithms is LSB "Least significant Bit" algorithm which hides the information bits in LSB bits of bytes in a media file.Because of wasting time and effort in finding the exact information that satisfies the user's needs from the retrieved information of the search engine. It is necessary for the search engines to accommodate text description for all hypermedia files (text, sound, ..., etc.). In our proposed web crawler software called CRAHID, we supposed using a new format for saving media file which uses information hiding to describe the media file. This means, when we talk about sound file called "aa.mp3", to know what is the content, of this file, we need to extract the hidden information from it by using LSB algorithm. The extracted information is processed by the populated keywords in the search engine database.

VII. PROPOSED "CRAHID" SOFTWARE

In our research, we proposed a new web crawling software called CRAHID which uses a new technique to crawl not only HTML, Document, PDF and text file but also sound, image and video depending on an information hiding technique. The indexing operation in web site search engine needs text to extract keywords with their properties from the web page in order to help the searching operation, but what about image, sound and videos. Usually, crawling operation reads multimedia file name and some small information related to this file in order to help the indexing operation. But this information does not describe the multimedia file in details, therefore in our CRAHID crawling technique, the crawler reads hidden information in the multimedia file which describes the media in details. Then saves this information as a text for indexing to maintain time and effort to find the exact information retrieved from the web crawlers. Our CRAHID crawler software is described in the following sections.

7.1 Proposed CRAHID Crawling Flowchart

Our proposed CRAHID crawler software can be described using the following flowchart as shown in figure (2). From the flowchart, crawler reads input from URL list (queue) then if there is a URL, it starts processing by detecting the type of URL source file. This operation is done by using text processing for reading file extension and MIME type information from the page. By this information, the crawler decides the strategy of crawling this URL and saves extracted text information with this URL in a database for the indexing operation. In this case of an HTML file, the crawler extracts all hyperlink URLs and adds it to the queue of URL. Then removes all unwanted HTML tags and saves the raw text in the database with its URL. In the case of a PDF file, the crawler extracts all texts from the PDF file and saves it in the database with its URL. In the case of document file, the crawler reads and extracts the text from the file and saves it in the database. Also this procedure executes in the case of a text file. In the case of image file and the remaining types, the CRAHID reads the status byte in an image file (the byte, tells the CRAHID crawler about information hidden in the image or not). If there's information, the CRAHID crawler reads this information using (LSB) Least Significant Bit and saves the resulting text in a database with the URL. The information hidden in a multimedia file can be saved as (Arabic or English) language. By the way, multimedia file searching can be in Arabic, English or any other language.

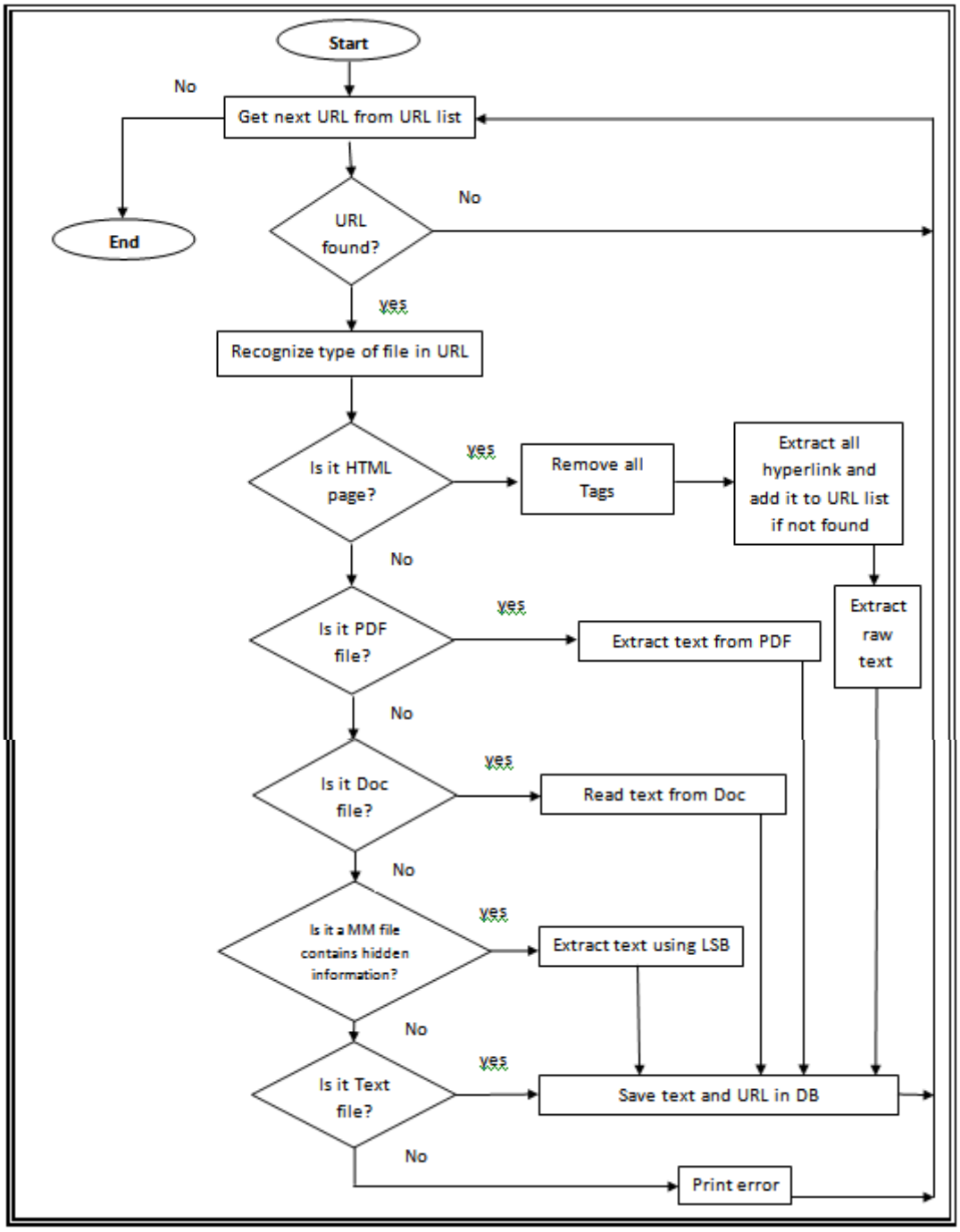


Figure (2): The CRAHID flowchart

7.2 The CRAHID Algorithm

In this section, we explain the algorithm used in our CRAHID software for crawling multimedia files using a proposed algorithm depending on information hiding technique LSB to extract information that describes the multimedia file. This algorithm is shown in figure (3) below.

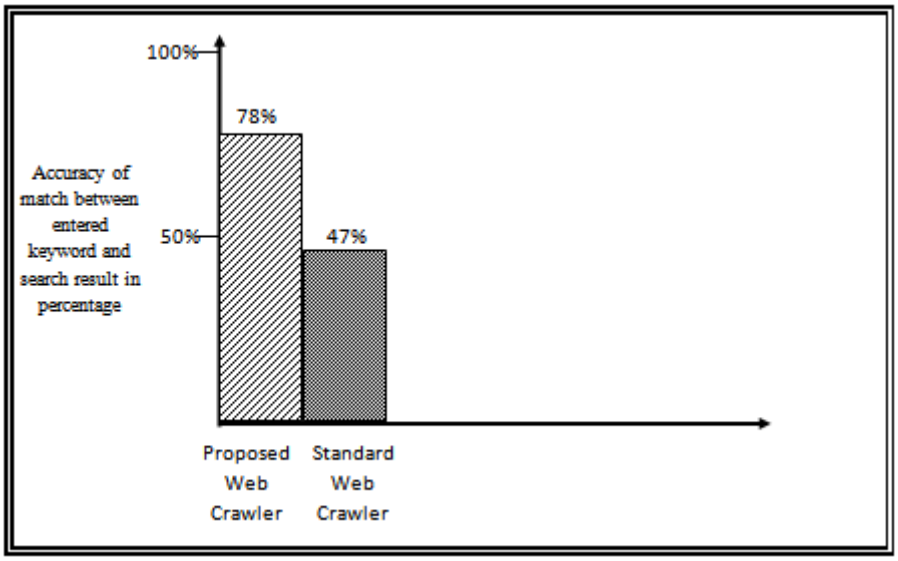
Algorithm: Multimedia file crawler
Input: URL list.
Output: Database table contains two fields (URL, text extracted from it).
Process:

1. Begin
2. While there is a URL in URL list
3. Begin
4. Download file from Internet
5. Recognize type of file from MIME information in URL
6. If the file is text file then extract text and save it with URL in Database
7. Else if the file is a Document file then extract text and save it with URL in Database
8. Else if the file is PDF file then extract text and save it with URL in Database
9. Else if the file is a multimedia file test it if contains hidden information or not.
10. If contains hidden information then
11. Begin
12. Read this information using LSB algorithm.
13. Save text and URL in database
14. End
15. Else print error message.
16. End of while
17. End of algorithm

Figure (3): The CRAHID new crawling algorithm.

VIII. RESULTS

We tested our proposed CRAHID web crawler software in an Intranet with about 100 sample of URLs of web sites containing different types of files (HTML, Doc., PDF, Text and Multimedia). We got the following results. The results are shown in figures (4) and (5) below.



IX. CONCLUSIONS

From this paper, one can conclude the following:

[1] The proposed Web crawler is the first crawler that crawls multimedia files depending on an information hiding technique.

[2] The text hidden in a multimedia file is very useful to describe this file content. This means that the searching operation is to find the exact file when searching the hidden text information in it.

[3] The process of extracting information is done separately from the searching operation. This job is executed in a scheduler way.

[4] The information hidden in a multimedia file can be saved as (Arabic or English) language. By the way, multimedia file searching can be in Arabic, English or any other language.
[5] This crawling technique is useful not only for the web site search engine, but also for any program that searches for local files in a PC computer.
[6] We tested our CRAHID new crawler software in an intranet with a sample of 100 web sites containing different types of media. We found that it is more accurate than the normal crawler and we got an accurate crawling result (information) from the new crawling process.

FUTURE WORK

We suggest to use another information hiding technique to crawl the exact information from search engines and compare its results in maintaining time and effort with our CRAHID software.

REFERENCES

[1] Christine Pichler, Thomas Holzmann, Benedict Wright, "Crawler Approaches and Technology", Information Search and Retrieval, WS 2011 LV-Nr.: 506.418, Group 02.
[2] Pavalam S M, S V Kashmir Raja, Felix K Akorli and Jawahar M , "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 , www.IJCSI.org.
[3] Carlos Castillo, " Effective Web Crawling", Submitted to the University of Chile in fulfillment of the thesis requirement to obtain the degree of Ph.D. in Computer Science , 2004.
[4] Vladislav Shkapenyuk and Torsten Suel. "Design and implementation of a high-performance distributed web crawler", In Proceedings of the 18th International Conference on Data Engineering (ICDE), pages 357 – 368, San Jose, California, February 2002. IEEE CS Press.
[5] Sandhya, M. Q. Rafiq, "Performance Evaluation of Web Crawler ", Proceedings published by International Journal of Computer applications (IJCA), International Conference on Emerging Technology Trends (ICETT) 2011.