International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

Enhance the Security and Improve the Performance of Web Crawlers using Web Crawling Algorithms

Vandana Shrivastava, Harvir Singh, Arvind K. Sharma

Abstract*: *World Wide Web (WWW) is a massive resource of information that contains data in zeta bytes. Large number of users surfing Internet for various purposes need a quick response from web. Search engines are preferable, easily available and reliable source to retrieve data very rapidly with minimum resources. Search engine uses Web Crawler to search abundant data from web. It is a program that automatically passes through hypertext structure of the web and access webpage and hyperlink within it. Web crawling is a technique that focuses on obtaining relevant and fruitful links. To make the traversing accurate and productive, a properly designed web crawling algorithm is required. Web crawler and crawling techniques are kept secret as it is a hidden formula for the success of any search engine. In this paper, we have presented various web crawling algorithms along with their pros and cons to be utilized towards web search. Also we have proposed a novel web crawling algorithm to enhance the security and improving the performance of web crawler using single threaded and multi-threaded crawling algorithms.

Keywords*: *Crawling Algorithms, Search Engine, Uniform Resource Locator, Web Crawler, World Wide Web

I. INTRODUCTION

Internet is the widest network spreads all around the world with complex structure. It has open architecture that facilitates interoperability and globalization of various services [1]. The most viable part of the Internet is World Wide Web (WWW) discovered by Tim Berner Lee. It is a huge collection of websites and its remarkable feature is hypertext. WWW is a service to hook up documents in websites by hyperlinks. Hypertext is a way of instantaneous cross linking or cross referencing to some other text, file and website. Organization of hypertext link is a powerful new source of evidence for web semantics [35]. It provides easiest way to connect to some other text or file without knowing its physical address and quick accessing help users to access thousands of millions of web pages in a single click.

World Wide Web is the widest source of information contains text, audio, video and other multimedia form of data that do not follow similar format. Web search is generating voluminous traffic and search engines have to handle the large sized web pages. Information on the web is scattered on large number of web pages stored on multiple servers [2]. Out of total data available, only 2-4% of data found is structured, 16% data is semi-structured and more than 80% data is unmanaged. Accessing such unmanaged data in minimum time, using least number of resources is the most challenging task for the search engine [36]. A Web crawler or Web robot is an essential component that automatically traverses the web and searches links related to the user's query [38]. It is a resource discovery and retrieval tool for search engines [3,32]. Web Crawlers use various methods to traverse data from distributed and diversified websites. Web Crawling is the technique to explore web applications automatically, browse hyperlinks, create index and store them for future use. Traffic on the web is increasing tremendously that makes traversing task difficult [4]. Efficiency of search engine depends on web crawler and crawling technique used by them, thousands of web pages are indexed in every second and facilitates users to search and traverse accordingly [30]. Basic working of all crawlers is approximately same but different crawling mechanism makes them different from each other [39].

The remainder of paper is organized into following sections- Section I discusses about the search engine and web crawler, Section II covers a background and related work, Section III focuses on different web crawling algorithms and their comparative analysis, Section IV shows a proposed a novel web crawling algorithm. Conclusion is given in the Section V while the references are mentioned at the last.

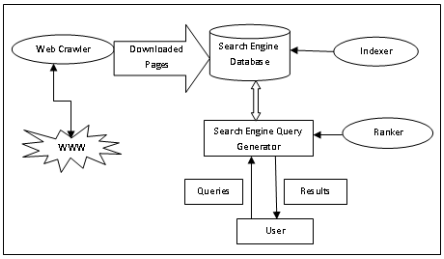

Figure 1 Architecture of Search Engine

***Search Engine*-** Search Engine is the most important tool used to locate data and information on gigantic web [5]. Its database is comprised of millions of web pages which are indexed for fast searching. Results displayed by search engine depend on various crawling algorithms. The basic tool of search engine is web robot or crawler that automatically traverses web pages and make index of visited content and hyperlinks. The architecture of search engine is shown in figure 1.

Manuscript published on November 30, 2019.
* Correspondence Author
Vandana Shrivastava*,Research Scholar, School of Computer and System Sciences, Jaipur National Univetsiry, Jaipur, India

The crawled page is processed and indexed using following steps-

- Exclude all the stop word
- Record the word that has frequently occurred in web page
- Record hyperlinks from the visited page
- Record information regarding media and images

The traversed and collected data is used to create ranks of the page by the ranker. Ranking method is used by the search engine to decide which page should be shown where and in what order. Once the crawled data is processed, it is arranged in different files and stored on various systems and web database repository for future use. Web crawler is core component of search engine that access web, indexed pages, keep them up-to-date and preserve the route to other web pages [25].

Web Crawler is important because-

- It keeps copy of popular web sites
- It maintains index of contents
- It checks sites to see if their structure or content is changed or not
- It explores the copyright or patent infringement

II. BACKGROUND AND PREVIOUS WORK

In 1990 when search engine was developed, it revolutionized the Internet surfers as it was a convenient method for searching required web data easily. Internet is a cheaper and efficient resources used by diversified services [31]. A Lot of search is conducted to improve, enhance the capacity and performance of search engine. Actual working tool that acts as backbone of search engine is the Web crawler. The Design, working and methodology of web crawlers are kept secret by most of search engines as behind their success, lots of research is conducted. Search engine and web crawlers are the topic of recent area of research and a lot of experiments and innovations are conducted for the same. A large number of research papers are studied that described the pros and cons of various web crawling algorithms.

Authors Bra , Houben, Kornatzky & Post [6] suggested a navigational algorithm "The Fish Search" based on heuristics that perform partial search in distributed hypertext . Basic assumption behind this method is that pertinent links have relevant connecting nodes and links.

Hersovici, Jacovi, Maarek, Pelleg, Shtalhaim, & Ur [7] introduced Shark search Algorithm which was a refined and aggressive version of Fish Search Algorithm. To evaluate relevance of document, a "similarity engine" is used. This engine is built by Vector Space Model that evaluates two documents dynamically and gives a "Fuzzy score" whose value lies between 0 and 1. Value 1 is assigned if it's a perfect match with given query, 0 otherwise.

Brin, & Page [8] devised an algorithm with name Page Rank Algorithm. Page-rank and anchor text of document both are used for ranking purpose.

Abiteboul, Preda & Cobena [9] proposed a new crawling algorithm named OPIC (Online Page Importance Calculation). This algorithm is modified version of Page Rank algorithm.

Hypertext Induced Topic Search (HITS) was developed by Kleinberg [10] that primarily works on relevance of the topic provided in the form of query by users. Relevancies of the traversed pages are considered for result set only.

Extended Hypertext algorithm is an extended version of HITS algorithm suggested by Miller, Rae, Schaefer,Ward, LoFaro, and Farahat [11]. They pointed some noticeable limitations of the HITS algorithm. The HITS algorithm does not always behave as expected and produce unclear results while Extended HITS method gives some improvements over the previous one.

Chakrabarti, Berg & Dom [12] discovered a new crawling method known as Focused Crawling. They explained that web crawler should selectively choose only those pages that are relevant against the user's query instead of accessing irrelevant hypertext. Topic oriented searching is the key of this crawler.

Yuan, MacGregor & Harms [13] suggested active index system to reduce traffic from Internet and improve performance of crawler.

Bidoki,Yazdani & Ghodsnia [14] proposed an intelligent novel crawling algorithm called FICA (Fast Intelligent Crawling Algorithm) based on Reinforcement Learning. The distance among pages is calculated and the page with lowest distance would get high priority for crawling. It finds hot pages at a rapid rate and is a fast and easy method having low complexity.

Derhami & Bidoki [15] proposed an extended version of FICA named as FICA+ algorithm. It is based on the relationship between background knowledge of visited web pages and web structure environment. A benefit of FICA+ is that it don't need to save the complete matrix of web graph but a vector of web graph is enough to hold distance of web pages. It gives improved performance and finds hot pages faster.

Mishra, Jain & Sachan [16] advocated that keeping traffic low at crawler and find updates at a faster rate is a prime need. Authors used dynamic web pages to inform crawler about changes (made in web pages) and newly added web pages, so traffic can be reduced by accessing only updated pages of website in next visit rather than downloading all pages.

Bhushan, Gupta & Gupta [17] suggested that traffic from web crawler can be reduced by using sitemap of website.

Saranya, Zoraida & Paul [18] advocated a new web crawling algorithm named Competent Crawling Algorithm to improve efficiency of retrieved information. This algorithm exploits the features of page ranks and best search method.

Pavani & Sajeev[19] explained a novel approach to access hidden and relevant pages using rank and semantic similarity information. Most of the crawlers used Page Rank Algorithms where pages having more links got higher rank while pages which got lower rank could not get a higher rank. Authors consider moderate ranked pages and semantic similarity to access and rank the pages.

Ye, Jing, Huang & Chen [20] in their research paper "The Research of a Lightweight Distributed Crawling System" developed a platform for distributed crawling system. Web Crawler can be make more efficient if required web pages are available on clusters on distributed environment instead of single computer environment.

III. WEB CRAWLING ALGORITHMS

World Wide Web is largest source of getting information easily and searching for data against user's queries is the major task all around the world.

Internet users use search engine to explore various subject matter of different topics. Web has a graph like structure (shown in fig. 3) and crawling is a technique to search data from giant web at the minimum cost. Crawling mainly consists of two activities-

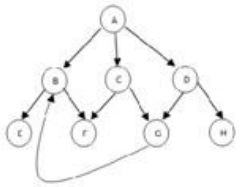

Figure 2 Web Graph

a) **Crawling**-It is the process to access the World Wide Web to find documents, construct the index, rank the pages and collect the data from centralized or distributed location.

b) **Serving**-This is the process to receive the queries from users and use index to find the relevant pages to serve the query.

Crawlers can be of different types based on the functionality and crawling algorithm used by them. Customary web crawler searches web pages randomly; focused crawlers are topic oriented and incremental crawler work on keeping the web fresh [39]. The web pages collected during crawling is actually a part or subset of web and is far smaller than the total dimension of the web. Intelligent design of the web crawler ensures that small and manageable collection of web is representative of the full web. Web crawling is a difficult task because of big size of web, fast rate of change, addition and deletion of web pages and dynamic page generation rate. The large volume implies that the crawler can only download a fraction of the web pages within a given time, so there is a need to download the pages on the basis of priority. The number of pages generated by server side software has also made it difficult for web crawlers to avoid retrieving duplicate contents and thus it becomes harder to schedule unvisited web pages.

Search engine used crawling algorithms to search and locate data. Algorithms discussed in following sections can be categorized as follows-

A) Algorithms Based on Heuristic, VSM Method

1. Fish Search Algorithm-This is a dynamic algorithm with heuristic nature. The basic search method is to find a good starting point and get optimal path order to retrieve the nodes. Initially list is empty and has a start or current node which is parsed to check that it contains keywords or search string out going links, number of links. Seed URL and search query is provided as an input and a priority list of next URL is created dynamically. The search agent explores the web and analyzed be a scoring component to check whether content is relevant or not. A heuristic decides in which direction the exploration would be continued. The fetched document is searched for links and it is assumed that if parent is relevant its child would also be relevant. Children are assigned with some predefined value; otherwise the depth of children is set less than the depth of parent node. When it reaches zero that direction is not explored more and none is inserted in list. In Fish Search method Ordering of URL is quite important to search relevant results at quicker rate. Searching mechanism moves around pages where relevant contents and links are found and will not redirect where probability to find relevant contents is least. Searching in least probable area is wastage of time and resources [6].

This algorithm is very simple to implement and dynamic in nature but key problem is very low differentiation of priority of pages in URL list. When multiple documents having same priority, crawler devotes more and more time to documents situated at header of the list due to limitation of time. The scoring capability of fish search is not very efficient. Child nodes are cut down randomly if scored with less value and it may lose some valuable information.

2. Shark Search Algorithm- This is an improvement over fish search algorithm. A similarity engine based on vector space model is used to evaluate the page relevance instead binary function. It returns a fuzzy score of 0 if no relevance and 1 if perfect match is found. Child node inherits fuzzy score from their parents and their score can be improved using link contained in that page. First set of children is selected on the basis of most promising anchor and anchor context information. Heuristics used in shark search method is proven to be very effective for scoring the documents on priority list [7].

This algorithm has many improvements over fish search method as it can analyze document relevance more precisely with finer estimation and relevance of neighbouring pages can also be more accurately fetched. It can crawl website more deeply in those areas where chances of finding the relevant information are more and saves precious communication time. Child node gets an inherited score from parent node and node with higher score is fetched on prior basis thus preventing from accessing irrelevant nodes. This algorithm discovers more relevant information in same time than fish search.

3. A* Search Algorithm-It is a dynamic and heuristic algorithm that combines features of lowest-cost-first and best-first searches. It considers cost of page and heuristic information to choose which path is to be followed for searching to get optimal outcome [21]. Function f is used for A* searching-

f(n) =g(n)+h(n), here n is the next node on path, g(n) is total cost of path from start node n and h(n) is a problem specific heuristic function that is used to calculate path with minimum cost from n to targeted node or link. Searching terminates when there is no fruitful path for further traversing. It calculates the relevancy of each link and the difference between expected relevancy of the required web-page and the current link. The sum of these two values serves as the measure for selecting the best path. During crawling the frontier list, A* calculates an estimate of total path cost starting from initial node to the target node restricting to a particular path only and terminates when a required node is selected from extension.

4. Best First Search Algorithm-This heuristic based searching algorithm uses some scoring function for evaluation of page relevance. Relevancy Calculation is done for links associated with that page and the page that scores highest relevance value is obtained from the frontier list. With this search most relevant pages can be obtained and it simply prevents irrelevant pages from getting downloaded as well as indexed [37]. It ensures that every time best available link is accessed [21].

B) Algorithms Based on Ranking Methods –

5. Page Rank Algorithm- Page Rank Algorithm works on the significance and importance of the webpage by calculating inlinks (backlinks) or citations. Link structure of network of web pages are used to assign weights to each page and it is used for ranking purpose [33]. It defines the popularity of the website and is used to measure the rank in absence of any query. The number of input links present within the traversed page is used to know the concern degree of page to others. The total weighted summation of input links describes the page rank of a webpage. Page rank is calculated by the following formula-

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

Where

PgRk(A) is PageRank of Page A

d is damping factor its value set between 0 and 1

T1, T2… Tn number of links which links to page A

Webpage having high input links is considered to be more important than other pages. Interest degree of the page is proportional to the input links, means as the number of input links increases, interest rate increases proportionally. To calculate page rank for a page, all inbound and outbound links are calculated and sum of all values provides rank of that page. When a web page receives links from another important web page, its rank is automatically updated with high ranking. After calculating page rank, index is generated to show the relevance of website with search words [8].

6. Batch Page Rank Algorithm-In this method, when N pages are downloaded, an estimated page rank is assigned to the downloaded pages. This estimation is done on the basis of pages already seen so far from downloaded pages. Next lot of N pages which are selected to be downloaded is the pages which have peak approximation Page Rank [22].

7. Partial-page rank is a little bit different from batch page rank method. Here the sum of page rank of website pointing to new the downloaded pages is taken into consideration to assign a temporary page rank to fresh downloaded pages. Then this page rank is divided by the total number of outbound links of those pages and finally the actual page rank of new pages is calculated. [23].

8. Online Page Importance Calculation (OPIC) Algorithm- Abiteboul, Preda & Cobena proposed a new crawling algorithm named OPIC (Online Page Importance Calculation). This algorithm is modified version of Page Rank algorithm. Some cash value is initially dispensed to each page. When a page is crawled, it indicates some other pages linked to it, this cash value is equally distributed to associated pages and recorded in page history (credit history) and is used to calculate importance of page. Here it is assumed that flow of cash through a page is proportional to its importance. Page Importance Architecture allows providing resources among discovered and refreshed pages. When new page is to be discovered some cash value is received from virtual page and distributes to the unread pages. To increase the importance of small edge pointing pages, system points some virtual pages so that it can accumulate more cash. When pages get refreshed its opposite is followed. Similar methods can be used to spotlight on a subset of interesting pages [9].

C) Algorithms Based on Hub and Authority-

9. HITS Algorithm- Hypertext Induced Topic Search algorithm works on the relevancy of topic provided in the form of query. Hub Value and Authority are the parameters used for computation of page relevance. Traversed pages which give upright information for a given topic is called Authority pages. Web pages which provide links to authority page are termed as Hub. An authority value is computed as summation of scaled hub values pointed to a page and hub values is the summation of scaled authority values of page it points. Here relevance is considered for result set only. After traversing weights are assigned to each hub and score is calculated accordingly.

It is basically a "Link Analysis Algorithm" that is used to rate the web pages, here hub is a page that represents many other useful pages and hub value represent its expended root set into a base set as shown in figure 3; authority is a value that is used to estimate the links through which the pages is referenced, it shows the contents of the page.

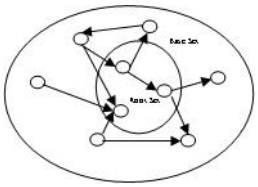

Figure 3 HITS Crawling

Initially sets of most relevant page against given query is searched. This set of pages is called base or root. Root contains hyperlink of other pages with related contents and itself referenced by some other pages [10].

It creates a sub-graph of web pages. In series of iteration during execution Authority and Hub updated on regular basis.

10. HITS Algorithm with Exponentiated Input - HITS algorithm suffers from some drawbacks. First difficulty is that authority vector is not exclusive as it depends on seed page. Second limitation is that when some nodes are inserted in the middle of the graph, it is assigned with zero authority and is addressed as nil weighting or non-uniqueness. In Modified HITS method, links are provided weight as per the number of times it is followed by a user. It yields better results as Weighted Input provides more acceptable results and determine best authority and hub in a website [11].

11. SALSA Algorithm-Lempel & Moran (2000) proposed a new algorithm to find hub and authority named Stochastic Approach for Link Structure Analysis (SALSA). Authority chain and Hub chain are two Markov chains derived by link structure of web graph and random walk is examined on both by SALSA method. Authority (hub) Markov chain performs random walk on the most regularly visited sites and group of authority (hub) is formed. This algorithm also does ranking of web pages and is proved to be computationally lighter than reinforcement policy. SALSA is less susceptible to TKC or Tightly Knit Community effect (it is a small set of sites that scores high in link-analysis method even though not very much relevant to topic in case of multi-topic group or may have just one aspect of a searched topic in case of single topic collection) [24].

D) Algorithms Based on Reinforcement Learning

12. FICA Algorithm-FICA algorithm was proposed by Bidoki, Yazdani & Ghodsnia based on reinforcement learning [14]. FICA is a fast and proficient algorithm having low complexity.

When a web-page is crawled, its priority is defined by logarithmic distance from root or starting URL. A crawled page p with distance dp, the distance of each of its child nodes, q is computed as dq = Log(O(p)) + γ*dp where O(p) shows out-degree of p and γ is the discount factor of the crawler agent. A visible benefit of this method is that it tries to model a user surfing the web in ad hoc manner. Initially user click on page very randomly but as time passes user can get more knowledge from environment and manages its click by surroundings and current status of visited pages. Page importance or priority is calculated during crawling process online by DistanceRank ranking method. It finds hot pages faster and also models user's random actions during web surfing. There is no need to store graph matrix, only a vector is sufficient to store the distance of each page. Authors proved that FICA has higher output than OPIC and Page Rank Method. An important feature of FICA is its adaptability towards the web graph; it can dynamically adjust itself as web graph changes. It is easy to implement having low cost and requires fewer resources.

13. **FICA+ Algorithm-**It is an extended version of FICA algorithm. When pages are accessed during crawling process, their selection is based on background knowledge of accessed pages and web environment. This algorithm uses relation between background knowledge of explored page and web constitution surroundings. Main feature of this method is that is does not require to save the whole matrix of web graph instead only a vector of web graph is sufficient to save the distance of web pages [15].

E) Algorithms Based on Topic or Domain

14. Focused Crawling-Also known as Topic Oriented or Domain Specific Crawling [12]. It is the process to fetch pages and collect documents which are specific, similar and relevant to the given topic (query) and ignores irrelevant pages. It is a technique to categorize pages according to topic and determines how far the given page is relevant to the particular topic and how to proceed forward. During searching crawler use contents of already downloaded or visited pages and assume the likeness between query and unaccessed pages. URL, anchor text etc are some useful resources which help to envisage the similarity of unvisited pages of the website. Focused crawler learns specialization from examples. Relevance and popularity rating mechanism are used to explore the web [34].

15. OTIE Algorithm- OTIE (On-Line Topical Importance Estimation) algorithm was proposed by Wang , Guan, Chen, Bu, Ang and Lin. This algorithm is a topic oriented crawling which primarily used Link based and Content based criteria to assess the advantage of fetching un-crawled URL. Initially a cash value is assigned to the pages and distributed to associated pages. This algorithm proved to faster, require less storage and does not require complex data structure. OTIE require lesser time to download pages and yields high quality topical pages efficiently [28].

(F) Algorithms based on Reducing Traffic on Crawler-

16. Active Index System-An efficient scheme to remove crawler traffic from the Internet was suggested by Yuan, MacGregor & Harms [13]. They proposed an active index system to gateway router to access web. It reduced crawler traffic, save bandwidth and server resources, improves performance of web accessing, avoids delay, improve user's response ratio and reduced investment on servers. In normal crawling method some pages never visited by crawler, as chance to fetch a page is dependent on the link with other pages but if an active indexing is done, chance of accessing unvisited pages becomes more. Active index provides index data to router for each passing page including dynamic pages and can pull more pages. Opportunity to access more pages can be induced regardless of its link relation to other pages. It increases probability to index dynamic pages. Apart from that, new network protocols and applications can be easily deployed.

17. Use of Dynamic Page- Mishra, Jain & Sachan advocated that keeping traffic low at crawler and find updates at a faster rate is a prime need during web surfing. Authors used dynamic web pages to inform crawler about changes (made in web pages) and newly added web pages, so traffic can be reduced by accessing only updated pages of website in next visit rather than downloading all pages. Crawlers simply query to server about changes in web pages which reduces resource requirement. Rate of crawling and downloading is compared with general crawling rate and it is observed that instead of all pages of site, only those pages were actually downloaded which were modified after last visit. This scheme was approx 4.83 times faster than simple crawling [16].

18. Use of Sitemap-Bhushan, B., Gupta, M., & Gupta, G. (2012) suggested that traffic from web crawler can be reduced by using sitemap of website. The extended sitemap used different data structure named Anchor Text, URL and Last_Updated. Crawler sends HTTP request and XTEND parameter to the server. When a website is visited first time by the crawler, whole website is downloaded and index is maintained by crawler with date and time of visit. If some changes are made within website (any webpage), then crawler search for Last_ Updated data structure in sitemap and only modified pages get downloaded instead of whole website, in this way crawler maintains freshness of web data, reduces the traffic from website and also decreases burden of web crawler to crawl individual page of website. Crawler also maintains a table of visited web pages with an entry named Last_Visited field which is used to find when webpage is visited by crawler. It is an intelligent method to maintain the freshness of website in lesser time and make web repository updated, it helps crawler to cover more of web contents and consume lesser bandwidth. It reduces web traffic from crawler but as far as space is concerned it requires more memory to hold some additional data structure and of course sitemap must be updated after each change in webpage to inform crawler to revisit the page otherwise it would be unaware of changes and users would probably get old data [17].

19. By HTTP Get Request and Dynamic Web Page- This is a query based method to reduce the web robot traffic using HTTP Get Request. Crawler can easily ask search engine about any updation using request URL and get updated information from servers. When a website is visited and downloaded by the crawler (especially dynamic website) then crawler download only updated pages of that website rather than downloading the whole website again.

This query based approach ensures that all updates are being informed to the web crawler to prevent it traversing and downloading whole site again and again that means it reduces number of pages to be visit by the crawler for updation [16].

(F) Other Methods-

20. Path-Ascending Crawling Algorithm- This algorithm traverses each path of given URL from home to last file which means rather than from following the links, it crawls the URL path. It helps the crawler to get more information about the URL. The segment of the URL which remains unaccessed during general link traversing, path ascending is a good choice to follow the complete path. For example a path with seed URL of http://abcd.org/abc/contact.html is provided and it will crawl /abcd.org/, /abc/ and /contact.html as shown in figure 4. The visible benefit of this method is that it not only effectively finds the hidden resources but also digs out those resources which has no inbound links and remains unaccessed most of the time [40].

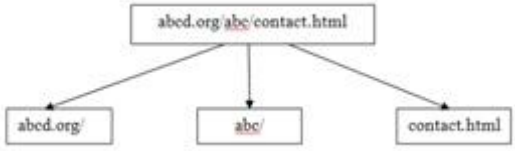

Figure 4 Path Ascending Crawling

21. Competent Crawling Algorithm- Saranya, Zoraida & Paul in 2015 advocated a new web crawling algorithm named Competent Crawling Algorithm to improve efficiency of retrieved information. This algorithm exploits the features of page rank and best search method. In page rank method pages that points (in-links) similar pages get high score and high priority while in best search method precision, recall, accuracy, F-score and popularity are considered during selection of URL. Here URL is selected on the basis of lexical similarity between keywords and source page of URL. Dequeuing of repeated pages is done at a faster rate to save time. Dynamic Hash Tables are used so that scalability can be implemented and system can be kept more reliable against crashes and failure [18].

22.-Priority Queue Based Estimation of Importance of Web Pages for Web Crawlers –A new web crawling algorithm was suggested by Baker and Akcayol in 2017. Seed Frontier is a priority queue and when a webpage is crawled, the URL divided into intra domain and inter domain link. Intra domain link indicated links within pages and inter link domain pointed links to other pages and websites. This algorithm focused more on inter-domain links to avoid link-loops inside a domain and is capable to discover new links from different sites. Weight is given to the hyperlinks by algorithm as per links indicated by them and is stored in the priority queue. This method proved to be more efficient to crawl unreached pages and eliminate duplicate pages as well.

23. Lightweight Distributed Crawling System-Feng Ye, Zongfei Jing, Qian Huang and Yong Chen in research paper "The Research of a Lightweight Distributed Crawling System" (2018) developed a platform for distributed crawling system. Web Crawler can be make more efficient if required web pages are available on clusters on distributed environment instead of single computer environment.

The key mechanism of platform is arranged as follows –

a) Design of Distributed Framework-For design of distributed network, open source software like Zookeeper, Mesos etc used.

b) Smart Spider Component- The crawling task is handled by intelligent crawler. It gets task from core database, initialize settings, generates crawler file and performs its assigned task using Docker.

c) Deduplication of URLs-When crawler crawls a webpage, it is possible to find same URL following different link so deduplication detection must be done to avoid loops. URLs that are not crawled yet only be considered during crawling process and this objective was achieved using scrappy framework.

d) Distributed Crawling Management-It consists of two modules-

i) **Monitor Module**-This module provides web interface.

ii) **Daemon Module**-It works with monitor module and manages task scheduling, crawling, resources and other tasks. Their experiment showed that light weighted distributed crawler consumes less memory, has lower CPU utilization than standalone crawler, higher execution efficiency and faster growth rate in memory usage. It has also been observed that if simple crawler deployed on clusters, they use resources in more efficient way.

24. Breadth First Search-This web crawling algorithm traverses nodes on equal level first. This method makes exclusive search around the adjacent nodes or hyperlinks. It starts search from root node and then traverses all the neighboring nodes at same level as shown in figure 5. Here frontier is used as First in First out (FIFO) queue and crawling is accomplished by traversing links in order by which they occurred in frontier list. If required URL is found then searching ends but if not, then all nodes of the next level are scanned to find the desired node. This process continues until all nodes are traversed or the goal is achieved [37].

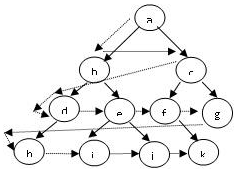

Figure 5 Breadth First Search

After retrieving all nodes of the URLs, if required URL is not found, a failure report is generated. If numbers of pages are more, useful pages must be selected from frontier to make traversing more efficient. This algorithm is useful where required pages lie on surface rather than deeper part of the search graph. This method gives highest quality pages in early stage of crawling but as crawling precedes the quality of pages declines gradually [29].

25. Depth First Search-It is a recursive and more systematic web crawling algorithm that is used to traverse the graph from root or core and move in depth through child URLs as shown in figure 6. Starting from seed, accessing continues deeper in current path. Traversing on same path is continued until all unvisited nodes are traversed. First link of the website is accessed and then link on every page is navigated till end [37]. Backtracking is the prime feature of this algorithm which means that if there is no node to traverse further in the current path, it moves back and finds other nodes to traverse.

This algorithm ensures that all links or edges are accessed i.e. there will be no node left unvisited.

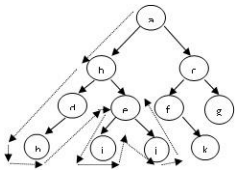

Figure 6 Depth First Search

26. Genetic Algorithm- This algorithm is based on biological evolution and is beneficial when user wants to access large data in minimum time. It provides multimedia results in more efficient manner. Generally other methods start searching from a single point while genetic method operates on a big database and ensures that the search would not concentrate in a localized area hence reduces the chance to trap in a single still point.

27. Naive Bayes Classification Algorithm-It is an algorithm based on probabilistic erudition and categorization method. Basic assumption behind this algorithm is that one feature is sovereign to another and provides much better results than other methods. Web crawler based on Naïve Bayes method is capable to exploit many relevant pages. This method uses four different attributes namely anchor text, lexical database (WordNet), parent pages of URL and surrounding pages of URL for searching and categorization. The lexical database is primarily used to discover and put in new related keywords to improve prediction correctness.

❖ COMPARATIVE ANALYSIS OF WEB CRAWLING ALGORITHMS

Here we focus on comparison of various well known crawling algorithms. A comparative analysis of crawling algorithms is summarized in the table-I below.

Table-I: Comparative Analysis of Web Crawling Algorithms

Algorithm	Merits	Demerits
Fish Search	♦ Searching always starts from relevant pages and moves around them ♦ Used Binary Scoring for URL sorting ♦ It uses regular expression for searching	♦ Sometimes it does not find sufficient search in limited time ♦ If many pages have same priority the crawler may inspect only those pages which are at the header position ♦ keeping some important pages unaccessed as they are at deeper side of list
Shark Search	♦ Performance is better than Fish Search ♦ uses fuzzy logic for distributing score among URL ♦ Use Vector Space Model for searching ♦ Better estimation of neighboring page relevance	♦ Complexity is more than companion algorithm
Best First Search	♦ It ensures to choose the best of all unvisited node of graph and not restricted in only small subset of nearest neighbors ♦ It explores those links on priority basis which have best score ♦ Link evaluation is done before downloading the content	**Many times** it covers more distance than consideration
Depth First Search	♦ Better than Breadth First Search as no node remains unvisited ♦ Well suited when target is situated at deeper of graph ♦ It assures that all node are accessed during crawling	♦ If branches are more than due to recursive nature it may ends in infinite loop
A * Algorithm	♦ Relevancy calculation is done on the basis of difference between expected relevancy and current link relevancy ♦ Give best results in lesser time ♦ Fairly constant in traversing time	♦ Traversing is limited in only a restricted path

algorithm	Merits	Demerits
FICA	♦ Fast and proficient method based on reinforcement learning ♦ Capable to adjust itself if web graph changes ♦ Search hot page faster ♦ Requires low cost and less resources to implement	♦ FICA only depends on the out degree of the accessed web page. ♦ At initial stage of web crawling process, the crawler has very few knowledge about the web environment hence wrong choices in decision can be taken.
FICA+	♦ Shortest Response Time • Topic relevance is prime factor for searching in this algorithm ♦ Gives Relevant results	♦ Crawler visits only those pages which have similar text, it may left some important pages unvisited ♦ Contents of unvisited pages are predicted on the basis of URL and anchor text similarity without downloading and checking those pages.
Crawling through URL Ordering	♦ If only fraction of webpage is to be download it is quite useful method ♦ If need to traverse the webpage for detecting changes to be made or not, it is a faster method ♦ It is very effective method to find isolated resources which have very less or no inbound links.	♦ If clusters are present in the website its performance degraded.
HTTP Get Request and Dynamic Web page	♦ Only updated pages are downloaded after last visit than downloading whole website again ♦ Maintains freshness of webpage ♦ Reduces web traffic	♦ If updation are not properly managed by sitemap or server, site remains unvisited in next crawling and user may get results without updation
Naïve Bayes Classification	♦ Four different attributes anchor text, lexical database, parent pages of URL and surrounding pages of URL are used during crawling ♦ Search more relevant pages ♦ Improve predication accuracy	♦ Its simple logic is not applicable in realistic environment
Genetic Algorithm	♦ Gives fastest results in minimum time	♦ Sometimes it might not find the most relevant and most favorable solution

III. PROPOSED METHODOLOGY

In this section we proposed a novel web searching algorithm to improve quality of web search. Here basically three learning stages named Content Based Learning, Link Based Learning and Sibling Based Learning are considered.

The DMOZ URL data set undergoes pre-processing where the raw data are inspected and checked administered to make it ready for further processing. Data set contain the different topics like arts, computer science, health, shopping, business etc. After this, the frontier is initialized with the seed URLs. Crawler crawls the pages of the website and the contents of pages are then examined and stored in the index. The content of the page is in HTML format or in text format. The Single threaded and Multithreaded crawling algorithm arranges the content according to the requirements. After arranged by the algorithm, the result is prioritized using hierarchical clustering. In clustering step, relevant pages are grouped together. The Crawler performance is typically measured by the percentage of downloaded pages that are relevant to the topic. The performance matrices will be harvest rate and crawling time of proposed work and compare the results with existing traditional crawling algorithms for improving the performance of web crawlers using Single threaded and multi-threaded crawling algorithms.

- **Flow Chart of Proposed Methodology**

Flow chart shown depicts the working of algorithm.

- The DMOZ URL data set undergoes pre-processing where the raw data are administered to make it ready for further processing. Data set contain various topics like arts, computer science, health, shopping, business etc.
- After this, the frontier between data is initialized with the help of seed URLs.
- The contents of pages are examined and stored in the index. The content of the page is in HTML format or in text format.
- The Single threaded and multithreaded crawling algorithm arranges the content according to the requirements.
- After arranged by the algorithm we prioritized the result using hierarchical clustering. In clustering, we group the relevant pages. The Crawler performance is typically measured by the percentage of downloaded pages that are relevant to the topic.
- The performance matrices will be harvest rate and crawling time of proposed work and compare the results with existing traditional crawling algorithms for improving the performance of web crawlers using Single threaded and multi-threaded crawling algorithms multi-threaded crawling algorithms

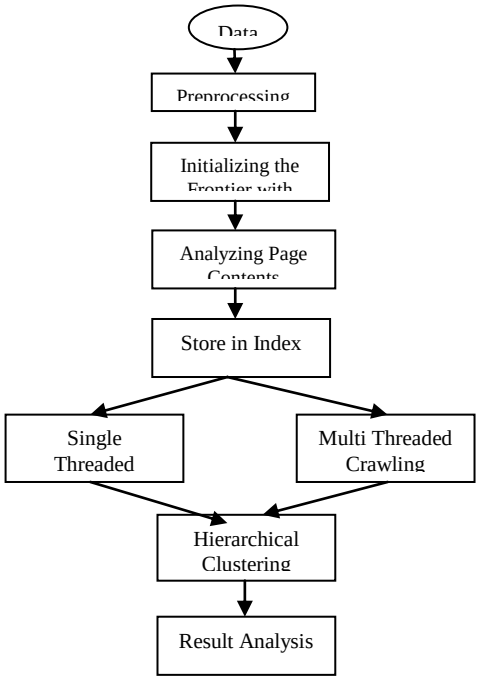

Figure 7-Flow Diagram of Proposed Methodology

❖ **Proposed Algorithm**

The pseudo code of our proposed single and multithreaded web crawling and indexing is as follows:

Import: Queue (Qu), List(Li), Hash(Ha)
Input: DMOZ URL List (Sturl), Number of Clusters (Cl)
Output: Relevant Link (R_L), Execution Time (T_E), Harvest Time (T_H)
Function crawl(St_{url})
crawled ← new Li
Qu ← new Queue
visited ← new Ha
Ts ← getTime (Tg)
St_{url} ← normalize_url(St_{url})
Qu.push(Sturl)
while (not Qu.empty?)
url ← Qu.pop()
page ← fetch(url)
visited[url] ← true
p← visited[url]
for asset in page.assets
data ← fetch(asset)
write_file(data)
simi ← asset
End for
for link in page.links
link ← normalize_url(link)
Qu.push(link) if not visited[link]
Compute priority = simi + priority / 2
End for
Retrieve Cd← crawled.append(url, page.assets)
End While
Compute T_E ← TEX - Ts
Compute T_H
return Cd and Td
return T_E, T_H
End Function
Func Ind (Cd)
Initialize C_l
Assign n ← Cd
for i = n-1 to 1
Computer or[i] is Cd mod C_l
C_l = C_l - or[i]
if C_l < 0 then s ← C_l + v
t ← t / v
endfor
or[n] ← Cl
return or[n]
End Ind

// Hierarchical clustering
Input: Number of Clusters C_l, ,or[n]
Output :Act
Function HClust (C_l , or[n])
Begin
Initialize T← 0
Act← , or[n]% C_l
Nro_C ← N % C_l
CI ← N+1 % Nro_C
Repeat
t← t+1
New_C ← Ci U Cj % Nro_C
Nro_C ← Nro_C – 1
CCI ← New_G
CI ← CI +1
Act ← (Act-1 - { Ci,Cj}) U { Ccr}
Until Nro_C ← K
End HClust

IV. CONCLUSION

The main aim of web crawling algorithm is to retrieve the crawled web pages in the URL. The web crawler has capability to visit web page on Internet and index both current and new web pages. It transmits HTTP requests from one web page host to other. After providing a better HTTP connection, it initializes particular associated web page that has been visited. Each hyperlink of web page has been extracted. In order to identify new URL, it is essential for user to parse several web pages for indexing process. It is a simple automated program where Internet pages can be crawled that is deployed to retrieve information from web. This paper depicts the proposed novel web crawling algorithm using single-multiple threaded crawling and hierarchical clustering for crawling web pages. Depending upon this novel algorithm, the web pages in URL has been deliberated as seed frontier. This proposed method will be developed in Java programming by adding crawled web pages into frontier then linked pages as a future work.

REFERENCES

1. Sadiku, Musa, Sudershan R. Nelatury. “Future Internet Research”. International Journal of Advances in scientific Research and Engineering 3.2(2017)
2. Jaiganesh, S., P. Babu, and K. Nimmati Satheesh. "Comparative study of various web search algorithms for the improvement of web crawler." Int. J. Eng. Res. Technol.(IJERT) 2.4 (2013).
3. Najork, Marc. "Web crawler architecture." Encyclopedia of Database Systems (2009): 3462-3465.
4. Castillo, Carlos, et al. "Scheduling algorithms for Web crawling." WebMedia and LA-Web, 2004. Proceedings. IEEE, 2004.
5.] Kassim, Junaidah Mohamed, and Mahathir Rahmany. "Introduction to semantic search engine." 2009 International Conference on Electrical Engineering and Informatics. Vol. 2. IEEE, 2009.

6. De Bra, Paul, et al. "Information retrieval in distributed hypertexts." Intelligent Multimedia Information Retrieval Systems andManagement-Volume 1.1994.
7. Hersovici, Michael, et al. "The shark-search algorithm. An application: tailored Web site mapping." Computer Networks and ISDN Systems 30.1-7 (1998): 317- 326.
8. Brin, Sergey, and Lawrence Page. "The anatomy of a large-scale hypertextual web search engine." Computer networks and ISDN systems 30.1-7 (1998): 107-117.
9. Abiteboul, Serge, Mihai Preda, and Gregory Cobena. "Adaptive on-line page importance computation." Proceedings of the 12th international conference on World Wide Web. ACM, 2003.
10. Kleinberg, Jon M. "Authoritative sources in a hyperlinked environment." Journal of the ACM (JACM) 46.5 (1999): 604-632.
11. Miller, Joel C., et al. "Modifications of Kleinberg's HITS algorithm using matrix exponentiation and web log records. " Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2001.
12. Chakrabarti, Soumen, Martin Van den Berg, and Byron Dom. "Focused crawling: a new approach to topic-specific Web resource discovery." Computer networks 31.11-16 (1999): 1623-1640.
13. Yuan, Xiaoqin, Mike H. MacGregor, and Janelle Harms. "An efficient scheme to remove crawler traffic from the internet." Proceedings. Eleventh International Conference on Computer Communications and Networks. IEEE, 2002.
14. Bidoki, Ali Mohammad Zareh, Nasser Yazdani, and Pedram Ghodsnia. "FICA: A novel intelligent crawling algorithm based on reinforcement learning." Web Intelligence and Agent Systems: An International Journal 7.4 (2009): 363-373.
15. Golshani, Mohammad Amin, Vali Derhami, and AliMohammad ZarehBidoki. "A novel crawling algorithm for web pages." Asia Information Retrieval Symposium. Springer, Berlin, Heidelberg, 2011.
16. Mishra, Shekhar, Anurag Jain, and A. K. Sachan. "A query based approach to reduce the web crawler traffic using http get request and dynamic web page." International Journal of Computer Applications 14.3 (2011): 8-14.
17. Bhushan, Bharat, Meenakshi Gupta, and Garima Gupta. "Increasing the efficiency of crawler using customized sitemap." International Journal of Computing and Business Research 3.2 (2012).
18. Saranya, S., B. S. E. Zoraida, and P. Victor Paul. "A Study on Competent Crawling Algorithm (CCA) for Web Search to Enhance Efficiency of Information Retrieval." Artificial Intelligence and Evolutionary Algorithms in Engineering Systems. Springer, New Delhi, 2015. 9-16.
19. Pavani, Kolli, and G. P. Sajeev. "A novel web crawling method for vertical search engines." 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2017
20. Ye, Feng, et al. "The Research of a Lightweight Distributed Crawling System." 2018 IEEE 16th International Conference on Software Engineering Research, Management and Applications (SERA). IEEE, 2018.
21. Nigam, Aviral. "Web Crawling Algorithms." International Journal of Computer Science and Artificial Intelligence 4.3 (2014): 63.
22. Cho, Junghoo, Hector Garcia-Molina, and Lawrence Page. "Efficient crawling through URL ordering." Computer Networks and ISDN Systems 30.1-7 (1998): 161-172.
23. Boldi, Paolo, Massimo Santini, and Sebastiano Vigna. "Do your worst to make the best: Paradoxical effects in pagerank incremental computations." International Workshop on Algorithms and Models for the Web-Graph. Springer, Berlin, Heidelberg, 2004.
24. Lempel, Ronny, and Shlomo Moran. "The stochastic approach for link-structure analysis (SALSA) and the TKC effect." Computer Networks 33.1-6 (2000): 387-401.
25. Cao, Fengyun, Dongming Jiang, and Jaswinder Pal Singh. Scheduling Web Crawl for Better Performance and Quality. Technical Report, TR-682-03, 2003.
26. Wang, Can, et al. "On-line topical importance estimation: an effective focused crawling algorithm combining link and content analysis." Journal of Zhejiang University-Science A10.8 (2009): 1114-1124.
27. Baker and Akcayol "Priority Queue Based Estimation of Importance of Web Pages for Web Crawlers" International Journal of Computer Electrical Engineering 9.1(2017):330-342
28. Wang, Can, et al. "On-line topical importance estimation: an effective focused crawling algorithm combining link and content analysis." Journal of Zhejiang University-Science A10.8 (2009): 1114-1124.
29. Najork, Marc, and Janet L. Wiener. "Breadth-first crawling yields high-quality pages." Proceedings of the 10th international conference on World Wide Web. ACM, 2001.
30. Pavalam, S. M., et al. "A survey of web crawler algorithms." International Journal of Computer Science Issues (IJCSI) 8.6 (2011): 309.
31. Stevanovic, Dusan, Aijun An, and Natalija Vlajic. "Feature evaluation for web crawler detection with data mining techniques." Expert Systems with Applications 39.10 (2012): 8707-8717.
32. Guo, Weigang, Yong Zhong, and Jianqin Xie. "A web crawler detection algorithm based on web page member list." 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics. Vol. 1. IEEE, 2012.
33. Farahat, Ayman, et al. "Authority rankings from HITS, PageRank, and SALSA: Existence, uniqueness, and effect of initialization." SIAM Journal on Scientific Computing 27.4 (2006): 1181-1201.
34. Jamali, Mohsen, et al. "A method for focused crawling using combination of link structure and content similarity." Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence. IEEE Computer Society, 2006.
35. Menczer, Filippo, Gautam Pant, and Padmini Srinivasan. "Topical web crawlers: Evaluating adaptive algorithms." ACM Transactions on Internet Technology (TOIT) 4.4 (2004): 378-419.
36. Wang, Wenxian, et al. "A focused crawler based on naive bayes classifier." 2010 Third International Symposium on Intelligent Information Technology and Security Informatics. IEEE, 2010
37. Filipowski, Krzysztof. "Comparison of Scheduling Algorithms for Domain Specific Web Crawler." 2014 European Network Intelligence Conference. IEEE, 2014.
38. Cao, Fengyun, Dongming Jiang, and Jaswinder Pal Singh. Scheduling Web Crawl for Better Performance and Quality. Technical Report, TR-682-03, 2003.
39. Shrivastava, Vandana " A Methodical Study of Web Crawler" International Journal of Engineering and Applications 8.11 (part-I) 2018: 01.
40. Janbandhu, Dahiwale,Raghuwanshi "Analysis of Web Crawling Algorithms" International Journal on Recent and Innovation Trends in Computing and Communication 2.3,2014:488

AUTHOR'S PROFILE

Vandana Shrivastava is a Ph.D. scholar in the School of Computer & Systems Sciences at Jaipur National University, Jaipur (India) and doing research on improving performance of Web Crawler. She is currently working as Assistant Professor in the Department of Computer Science at S.S. Jain Subodh P. G. College Jaipur and has more than 18 years of teaching experience. Her research interest includes Data Mining and Web Crawling.

Prof. (Dr). Harvir Singh received his Ph.D. degree in 2006 and is working as a Director in School of Engineering & Technology, Jaipur National University, Jaipur. He has more than 24 years of experience and has supervised 12 Ph.D scholars and currently supervising 10 scholars. His publication includes more than 30 papers in national and international journals.

Dr. Arvind K. Sharma holds Ph.D degree in Computer Science and M.Tech degree from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, India. He has more than 18 years of work experience. He has supervised 5 Ph.D scholars and currently supervising 4 scholars. He has published more than 120 research papers, including 80 papers in refereed Journals and 40 papers in National and International Conferences.