Research Article

Design and Implementation of Distributed Web Crawler for Drug Website Search using Hefty based Enhanced Bandwidth Algorithms

Saran Raj S[a],AarifAhamed S[b], R. Rajmohan[c] and K.S.Guruprakash[d]
[a,b] Vel Tech Rangarajan Dr Sagunthala R & D Institute of Science and Technology
[c]IFET College of Engineering, Villupuram, Tamil Nadu, India
[d]Assistant Professor, Department of Computer Science and Engineering,
K.Ramakrishnan College of Engineering, Trichy, Tamil Nadu, India.

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstract: The development of an expert system based search tool needs a superficial structure to satisfy the requirements of current web scale. Search engines and the internet crawler are used to mine the required information from the web and surf the internet in an efficient fashion. Distributed crawler is one of the types of a web crawler, which is a dispersed computation method. In this paper, we design and implement the concept of Efficient Distributed Web Crawler using enhanced bandwidth and hefty algorithms. Mostly Web Crawler doesn't have any distributed cluster performance system and any implemented algorithm. In this paper, a novel Hefty Algorithm and enhanced bandwidth algorithm are combined for a better-distributed crawling system. The hefty algorithm is implemented to provide strong and efficient surfing results while applying on the drug web search. We also concentrate on the efficiency of the proposed distributed web crawler by implementing Enhanced Bandwidth algorithm.

Keywords: Distributed crawler, Page surfs, Bandwidth

1. Introduction

Internet Page swarmer is a meta- hunt engine, which combines the top search results from the represented search engines. The search of users may also involve the audio, video, news, and more. A lot of trustworthy web pages, in precise, examine contraptions, use spider process to provide the summary of website content. The website content includes the Uniform Resource Locator (URL) and the text-based summary. The Web Crawlers castoff towards generation of a list of URL of altogether the pages to provide information for future exploration which reduces the amount of consuming the time and resource. Web Crawlers is used for the maintenance of task on Web sites, such as examination of associations and authenticating the HTML enigma. It collects the particular material as of in webs, such as garnering electronic message addresses.

The everyday Web Crawler procedure part takes no updation, since its first release in 1993. Practically every crawler goes along with specific modification on the elementary web page inspection procedure. A Crawler is a massive search engine with the following concerns. Major, it should need an upright swarming technique i.e., which file will be downloaded next. Next, it should need an exceedingly improved framework engineering which transfers an enormous page count. The Web crawlers always need to pact with the disputes of expandability as the WWW inflates. The speed of traversing the web by the Crawler is restricted by the quantity of aspects, together with the bandwidth and the latency of the grid. Determining once a web folio is profitable to revolution, will benefit to minimize the quantity of unwanted balloting thru a crawler. The polling will exploit a minimal amount of resource, the more that can be assigned to the errand of finding new data. At last, crawlers will be depending after speaking with others and being occurrences of themselves (in the parallel sense). This emerges the requirement for self-governing agreeable sharing web crawlers - crawlers that can settle on choices all alone, and speak with others when the need arises.

Strewn web crawling is a scattered computing method in web where countless crawlers are at work to scatter the web crawling process, which establishes the maximum exposure to the web. A median proxy manages the exchanging information and harmonization of the nodules, geologically scattered as World Wide Web. Google's ranking algorithm is generally used to improve the competence and eminence search from the web. This Page ranking for a web sometime have a low rank, i.e., 2 or 3 or sometimes 0. The high ranking page will have number of traffic and the overall performance is not good enough. And also the surfing results occur only at the spell of indexes but never at query execution.

Other than the page rank algorithm, there are many distributed crawling algorithm which crawls for the web pages from the web server in an analogous harmonized manner. The harmonization procedure improves productivity by circumventing recurrent swarming on similar documents. Without harmonization the crawler will individually reconnoiter its peculiar queue without considering the others crawlers queue. Since this crawling without coordination will not be infrequent. So the crawling algorithm should have the coordination technique guided by link analysis metrics, re vigorous concerning the underlying seed reports, i.e., the bunch of records is gotten from various seeds are generally covering. This heartiness makes the parallelization impact as, without coordination, various crawlers will creeps similar archives.

The section I involves the system architecture, section II includes the proposed model of the system, section III includes the performance analysis.

2. Related Work

In paper [8], the distributed computing highlights and the MapReduce framework method are presented in the Web swarming. The Crawler creeps the web by utilizing appropriated operators and every specialist stores its own finding on a Microsoft Azure cloud. But it consumes more time to access the unstructured data.

In another paper, Topic Sensitive Page Ranking algorithm [9] has been used, the search process carried out in context by emphasizing the confrontations in a webs. At the time of query execution, these significance scores are consolidated dependent to search parameters. This uses three methodologies for scoring the pages they are query sensitive scoring, Query Time Importance Score and topic sensitive scoring. The Google’s Rank algorithm trajectories engendered are not exaggeratedly profound to certain adoptions thru by specific ODP correctors.

In paper [6], commonly used link analysis algorithm such as HITS and Google PageRank was discussed. The drawback of page rank algorithm is query independent and fallouts are sorted conferring to prominence of web-pages. The subjective page rank algorithm is also query independent and consumes more indexing time. The HITS has the topic drift and efficiency problem. In paper [7], propose a technique to figure the pertinence of a page to a looked through question put together not just with respect to the data contained in its literary substance yet in addition by registering the significance of the connected links. The suggested calculation speaks to utilize vector space model for calculating weight parameters for each page links. This is time-consuming process to generate the weights of the documents.

In paper [11], cast the picture positioning issue into the assignment of recognizing "expert" hubs on an induced visual likeness diagram and proposed Visual Rank to dissect the visual connection structures among pictures. The pictures observed to be "specialists" are picked as those that answer the picture questions well. Our exploratory outcomes show huge improvement, as far as client fulfillment and significance, in contrast with the latest searching techniques; however this requires more resource space to compute the large scale computing process.

In paper [10], JaytrilokChoudhary proposed the priority-based semantic web crawler. The semantic score is calculated for the given URL, which is downloaded from the web page. The semantic score is calculated from the anchor text of the unvisited URL. By means of the semantic score of the documents, the webs are get prioritized. The semantic score is computed by using the topic ontology also. The scoring is only focused on the semantic not on the similarity of the content of the web page and also not having the experimental outcomes. In paper [12], a parallel migrating crawler is dividing the crawler work to numerous search threads thus improvising the capacity of downloads. The suggested structure was tested and the outcomes were recorded. It has the central database communicates with the search engine. The principal catalog stocks the slant of links expected as of the apps and stockpiles the ultimately transferred bumf which is moved down by numerous crawlers. Since this use the same old mapper system to record the appeal and answer amongst the examine appliance and the crawler.

3. Existing Work

From the related work, we have identified some of the problems in the distributed web crawling.

- In web crawling the network traffic is the big issue while fetching vast amount of web pages.
- Load balancing is the main issue while fetching vast amount of web pages. However there are many proposed algorithm to improve the scalability, crawling speed, URL prioritization only some of them have been implemented.
- The bandwidth utilization is an important issue for the distributed crawling.
- So far, deep websites are utilized to search for drug details. Alphabay, Valhalla, Hansa, Zocalo Marketplace are the currently available deep websites.

4. Design Objectives

- Selecting URL and Content: A web crawling is desirable to have some mechanism to rank URL/domains to provide proper seed and content selection.
- Freshness: There are some websites which are to be frequently updated leads to updating in the crawling content also. It should maintain a strategy to determine the schedule for each URL to be downloaded.
- Deep Crawling: The process of crawling the web till the lower level to identify the unvisited websites and list out those in the URL list.
- Focussed Crawling: The focused web crawling is the estimation of the unvisited websites that are near similar information before downloading the internet contents.
- Challengers: Every web crawler has to face some irrelevant process such as crawling traps, spam sites, cloaked content. The crawling trap is the group of web sites that creates uncountable sets of URLs for the crawler to find. The spam sites are the malicious web sites which exploit user’s resource and bandwidth. Cloaked content is the process of serving a web page for search engine and providing entirely different one to user.
- Politeness: The site owner bans the misbehaving crawlers. To avoid this condition, tracing the robots.txt on every site to follow their terms and condition.

5. Proposed Framework Design

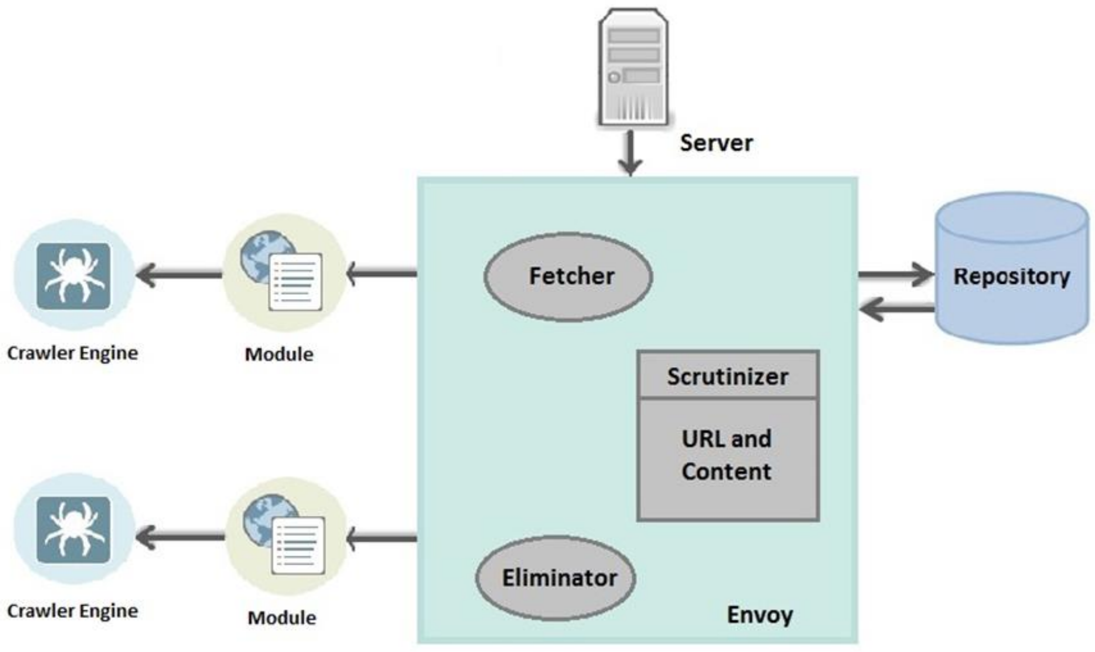


Fig 1: Proposed Model

The system architecture includes the Crawler engine, Module, Envoy, server, and repository. The crawler engine has the automated script to search the web where it simultaneously the number of search engine for searching the web (such as google, yahoo, bing). The crawler engine searches the web to collect a cluster of URL typically denoted as seed URL document. The additionally the crawlers validate the HTML code, hyperlinks, and web scraping. At starting the keyword is entered into the web crawler engine, and it sends the request to the webserver. The web server is the computer system which serves the user request.

The Envoy is a crawling agent which consists of the following phases such as fetcher, Scrutinizer, and Eliminator. The fetcher fetches the URL from the web server with the mostly related terms. The Scrutinizer scrutinize the URL and then content to prioritize it in the more relevance URL to the keyword. The eliminator will remove the visited URL request from the list and replace it with the unvisited URL. Then the URL list is extracted and denoted it as the modules and displayed in the web crawling engine. The repository stores the URL lists that are crawled from the webserver.

In our design, the hefty algorithm and bandwidth algorithm are combined to give an efficient crawling strategy. Initially, the user mentions the keyword in the crawler engine; the list of URL is extracted from the webserver. The collected URL is partitioned into separate sets, here noted as URL modules. Each URL module is handled by a separate web crawler. Each Web Crawler parses and logs into the URL that is present inside the module of it; the remaining URL are sent to the corresponding module. Each Crawler has prior knowledge of the search table relating to the URL modules (maps IP address) by identifying the crawler threads.

The first URL in the module is surfed, and then the next URL will be in a queue for the next visit. Here the Envoy is an agent where the modules have been processed. The identified URL has been analyzed by comparing it with the other URL in the module. If same visited URL found then the URL has been removed from the module. A novel link will be moved into the module. After the link has been surfed the content inside in it was scrutinized. If it has similar content (which is already surfed), then that URL has also been replaced with a new one.

The crawler is made out of a lot of fetchers, present inside the agent. Every module is kept running as a different string and the pool of fetcher strings are made upon beginning up. A fetcher holds up until the Envoy actuates it for recovering a given URL. At that point it plays out a HTTP solicitation to the fitting Web server and, subsequent to having moved the record; it comes back to a hold up state. The Envoy executes the crawler control approach by bolstering suitably the fetchers. The administrator chooses the URLs to be recovered during their needs and it initiates the ideal number of fetchers to abuse the doled out transmission capacity. The duplicate detector is used to identify the duplicate URL or similar URL to avoid the excessive usage of memory for the repository. This will improve search engine quality.

Distributed Crawling-Hefty Algorithm (DC-HA):

The distributed crawling algorithm is categories in three stages, such as the starting stage, formation stage, and relocation stage. In the starting stage, the keyword is entered which extract a seed set of documents for various crawlers. In the formation stage, the various URL is organized by using their priority. The similar

URL and the visited URL are identified to filter out. The estimation of all related information is used to find which documents to fetch next. This process carried out while bringing the current document itself.

The duplication in URL can be easily identified because it will be completely identical, whereas the near-duplicate have some structural difference. Some of the structural changes are date change, small edits, metadata and other changes. The similarity score estimates the near duplication. The similarity between the URL is computed to find the near duplicates between the web pages. The similarity score estimates the similarity degree between the web pages. The higher computation score indicates a higher percentage of similarity. So, the higher computation value indicates the web page have more near duplicates. The computation value estimates all the analogous pages whose likenesses exceeds the given threshold. The similarity value is between the two internets contents are projected as follows:

The specific keywords in a given web document is represented as

S1={(Xa1,Yb1),(Xa2,Yb2),……,(Xan,Ybn)}

and the computational count of each above keywords are represented as

S2={(Xa1,Yb1),(Xa2,Yb2),……,(Xan,Ybn)}

At first, the similarity between the web pages is measured by using the keyword for the first web page S1. It is measured as the set difference between the occurrence counts of both the keywords. If the keyword presence in the document is zero means then the computation value is null.

$$Yb_v[S_1]=\frac{1}{N1}\sum_{i=1}^{N1} Abs[S1(Yb,Xai)-S2(Yb,Xai)] \rightarrow [1]$$

ifMi not belongs to S2, then count=0

Where N1= | S¬1|.

Next the outstanding keywords (RK) will be assumed and the likeness is measured for those keywords in a new page P2.

R=S2-S1

$$Yb_v[S_2]=\frac{1}{N2}\sum_{i=1}^{N2} Abs\mathrm{R}[Yb,Xai] \rightarrow [2]$$

Where N2=|R|.

Progressively, the concluding0020likeness score is calculated as follows:

$$Yb_V[Xa_i]= Yb_V[S_1]+ Yb_V[S_2] \rightarrow [3]$$

The similarity between the two pages is obtained by the computing value BV[Mi]. The near duplicates of the web pages are obtained by using the data present in the repository. The web page with the highest computation value is deleted from the Module i.e., URL list

```
Procedure 1: DC-HA
Input: Enter the Query
Output: List of URL links L
1.  Start
a. Enter the Query (say Q).
b. Obtain URL list Li.
c. Visit the first URL L1   //by First In First Out.
d. Scrutinize T(Li)
e. Compare the URL L1 with the next URL L2
f. Calculate the similar computation value
1.  If computation score is higher
a.  Remove URL
b.  Add Lnew=Lj+1
2.  else
a.  Repeat T(Li)
3.  End if
a.  Scrutinize C(Li)
b.  Remove Lj+1
c.   Add Lnew=Lj+1
4.  else
a.  Repeat T(Li)
5.  End if
    End
```

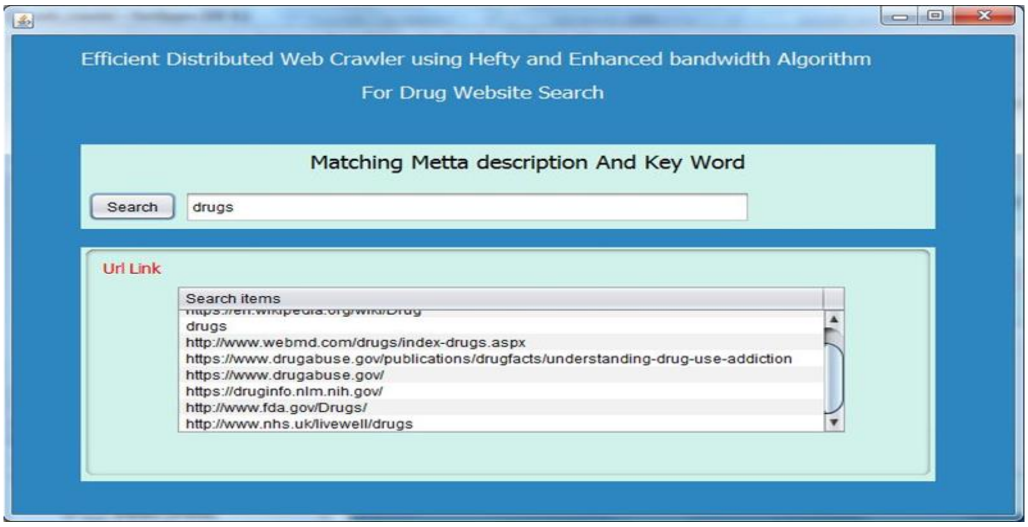
Bandwidth Algorithm (BW-A):

For choosing the next link L, the Envoy guesstimates the bandwidth BW(L) where a catcher is predicted to devour by recovering the file (f). After downloading the data, the Envoy can measure the actual rate of transfer as E(L) and also used to calculate the server speed untill the end of the process. The Envoy calculates the predicted overall transfer rate of the swarmer DCBW as l∈T. BW(F),in which F denotes files grouped by the catcher. The swarmer latency is recalculated as ABW=ABW−BW(Lc) and the envoy examines the link queue for a page Lf such that it satisfies the following threshold condition: ABW + BW(Lf) ≤ N.

```
Procedure 2: BW-A
Assume the variable f is set to zero.
Identify the page Lf at distance e in the link list
If ABW + BW(Lf) ≤ N, allocate Uf to catcher and apprise ABW = ABW + BW(Lf)
Set f++
Echo Lf identification until CBW ≤ N or f=maximum of e
```

6. Experimental Work

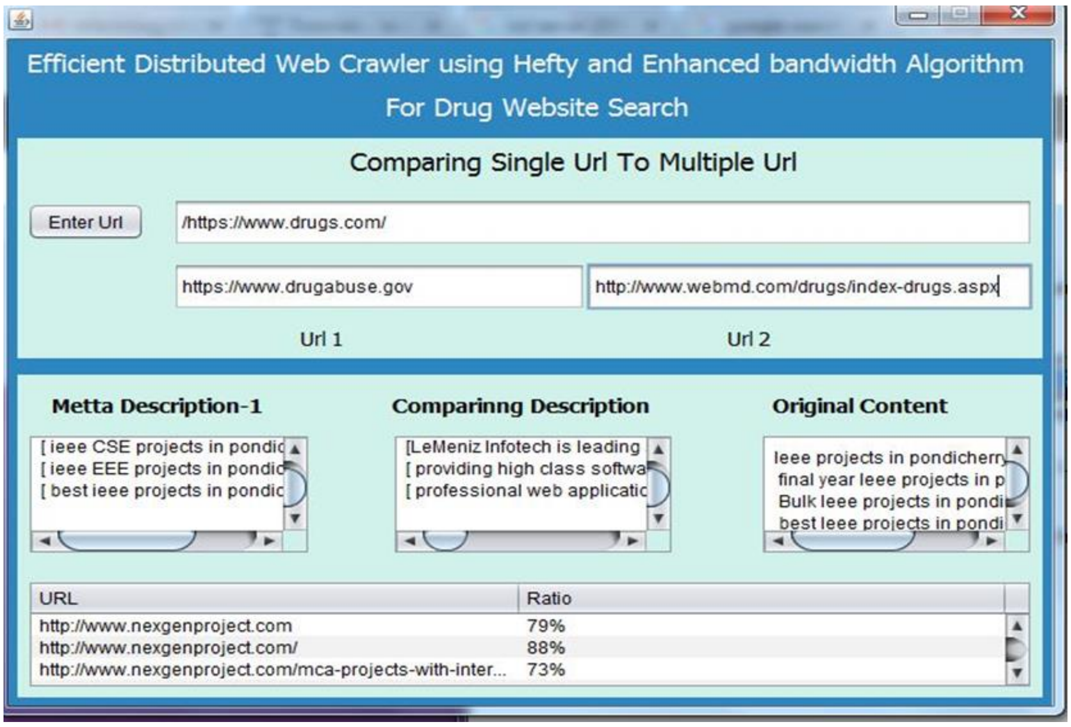
Apache NetBeans IDE 8.2 has been used to experiment with the entire system. The web crawler is implemented using the single method to measure the performance of pages crawled and the bandwidth of the node. After several testing and debugging, these data are obtained. The web crawler (single node) has 2GB RAM, Core2Duo 2.8 GHz Processor and the bandwidth limit was set to 1.5 Mbps for crawling the data. Initially, for the given keyword 40 URL's are crawled and denoted as the seed URL. The measurement of bandwidth utilization is recorded for 1000 minutes.


Efficient Distributed Web Crawler using Hefty and Enhanced bandwidth Algorithm
For Drug Website Search
Matching Metta description And Key Word
Search
Url Link
Search items

drugs

https://en.wikipedia.org/wiki/Drug
drugs
http://www.webmd.com/drugs/index-drugs.aspx
https://www.drugabuse.gov/publications/drugfacts/understanding-drug-use-addiction
https://www.drugabuse.gov/
https://druginfo.nlm.nih.gov/
http://www.fda.gov/Drugs/
http://www.nhs.uk/livewell/drugs

Fig 2: Represents the initial searching process carried out using the DC-HA


Efficient Distributed Web Crawler using Hefty and Enhanced bandwidth Algorithm
For Drug Website Search
Comparing Single Url To Multiple Url
Enter Url
Url 1
Url 2
Metta Description-1
Comparinng Description
Original Content

/https://www.drugs.com/
https://www.drugabuse.gov
http://www.webmd.com/drugs/index-drugs.aspx

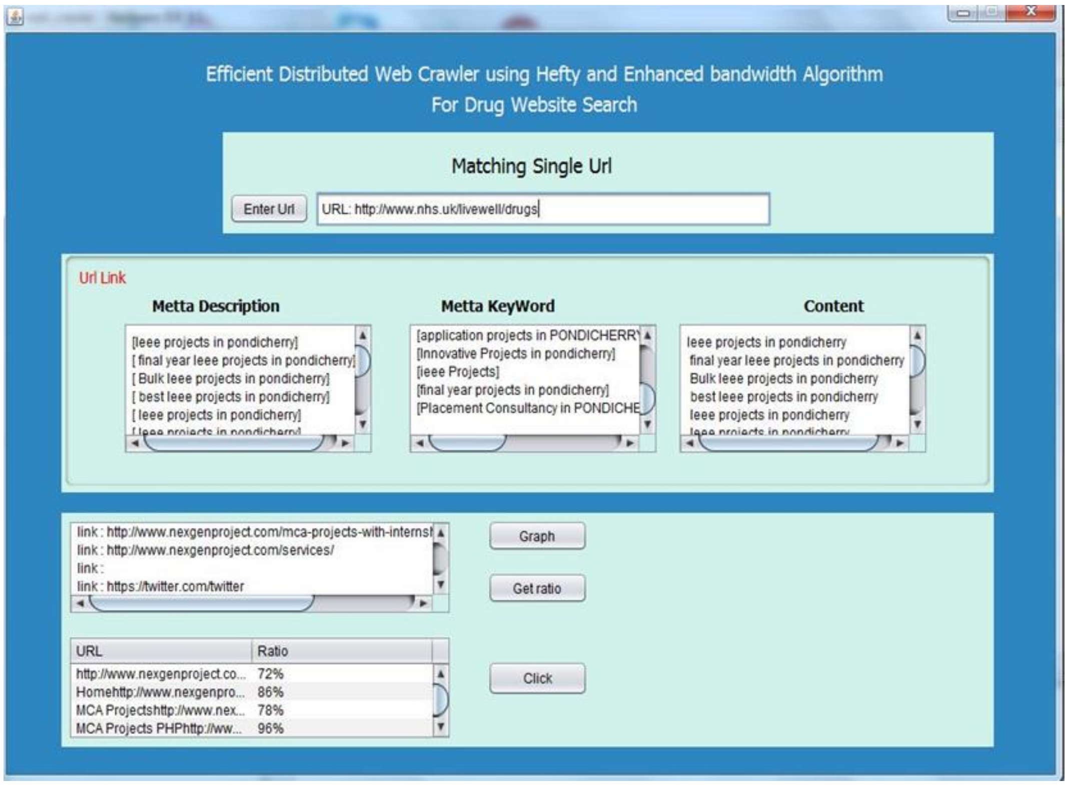
[ieee CSE projects in pondic
[ieee EEE projects in pondic
[best ieee projects in pondic

[LeMeniz Infotech is leading
[providing high class softwa
[professional web applicatic

Ieee projects in pondicherr
final year Ieee projects in p
Bulk Ieee projects in pondi
best Ieee projects in pondi

URL	Ratio
http://www.nexgenproject.com	79%
http://www.nexgenproject.com/	88%
http://www.nexgenproject.com/mca-projects-with-inter...	73%

Fig 3: Comparing the first URL with the next URL



Efficient Distributed Web Crawler using Hefty and Enhanced bandwidth Algorithm
For Drug Website Search

Matching Single Url

URL: http://www.nhs.uk/livewell/drugs

Url Link

Metta Description

[leee projects in pondicherry]
[final year leee projects in pondicherry]
[Bulk leee projects in pondicherry]
[best leee projects in pondicherry]
[leee projects in pondicherry]

Metta KeyWord

[application projects in PONDICHERRY
[Innovative Projects in pondicherry]
[ieee Projects]
[final year projects in pondicherry]
[Placement Consultancy in PONDICHE

Content

leee projects in pondicherry
final year leee projects in pondicherry
Bulk leee projects in pondicherry
best leee projects in pondicherry
leee projects in pondicherry

link : http://www.nexgenproject.com/mca-projects-with-internst
link : http://www.nexgenproject.com/services/
link :
link : https://twitter.com/twitter

URL	Ratio
http://www.nexgenproject.co...	72%
Homehttp://www.nexgenpro...	86%
MCA Projectshttp://www.nex...	78%
MCA Projects PHPhttp://ww...	96%

Fig 4: Comparing the initial URL content with the upcoming URL

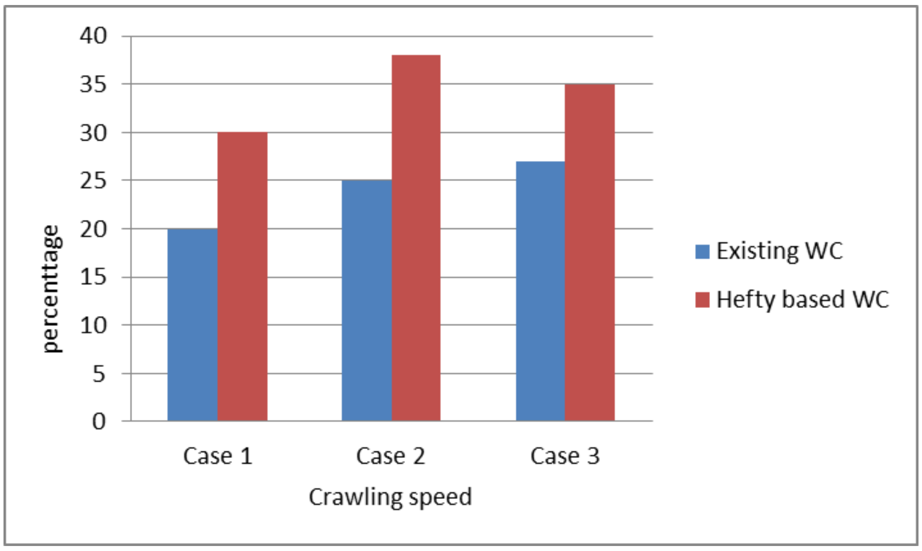


Fig 5. Crawling speed tested in three tries by comparing the existing crawling technique and the Hefty based web crawling.

7. Conclusion

Generally, web crawlers contend with one another depending on the size and money of the fundamental database, notwithstanding the quality and reaction time of their level classification. Distributed web engine architecture was introduced, aiming at crawling web pages and providing index service. The proposed work fluctuates hefty model and bandwidth procedures are patronized together to provide efficient results. The experimental work shows the improved number of web pages downloaded and operates within the limited bandwidth using drug websites.

Reference

S. Saranya, P. Victor Paul and Zoraida, “A Study on Competent Crawling Algorithm (CCA) for Web Search to Enhance Efficiency of Information Retrieval”, Advances in Intelligent Systems and Computing book series, AISC, volume 325, pp 9-16, November 2014.

Zhixing GAO, Kunhui LIN, “Design and Implementation of an Efficient Distributed Web Crawler with Scalable Architecture”, Journal of Computational Information Systems, pp -1817-1823, 2009.

Mini Singh Ahuja, Dr Jatinder Singh, BalVarnica, “Web Crawler: Extracting the Web Data”, International Journal of Computer Trends and Technology (IJCTT) – volume 13 number 3 – Jul 2014.

Sam Marsden “How Search Engines Work?” DeepCrwal, May 2018.

Qi Lin ; Yitong Liu ; Yun Shen ; HuiShen ; Lin Sang ; Dacheng Yang, “Bandwidth estimation of rate adaption algorithm in DASH” IEEE Globecom Workshops (GC Wkshps), 19 March 2015.

R Jain et al. “Page Ranking Algorithms for Web Mining”, IJCA, Vol 13, No.5, 2011.

Saxena, P. & Gupta, Jai & Gupta, Namita. (2010). Web Page Ranking Based on Text Content of Linked Pages. International Journal of Computer Theory and Engineering. 42-51. 10.7763/IJCTE.2010.V2.115.

Bahrami, Mehdi &Singhal, Mukesh&Zhuang, Zixuan. (2015). A cloud-based web crawler architecture. 2015 18th International Conference on Intelligence in Next Generation Networks, ICIN 2015. 216-223. 10.1109/ICIN.2015.7073834.

Haveliwala, T.H.. (2003). Topic-sensitive PageRank: A context-sensitive ranking algorithm for Web search. Knowledge and Data Engineering, IEEE Transactions on. 15. 784- 796. 10.1109/TKDE.2003.1208999.

Choudhary, Jaytrilok& Roy, Devshri. (2013). Priority based Semantic Web Crawler. International Journal of Computer Applications. 81. 10-13. 10.5120/14197-2372.

Shoaib, Mohammad &Maurya, Ashish Kumar. (2014). URL Ordering based Performance Evaluation of Web Crawler. 10.1109/ICAETR.2014.7012962.

Akansha, Singh & Singh, Krishna Kant. (2010). Faster and Efficient Web Crawling with Parallel Migrating Web Crawler. International Journal of Computer Science.