THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE

Enhancing the Efficiency of Smart Crawler

Shreya S. Tandur
Student, Computer Science and Engineering, Shree Devi Institute of Technology, Mangaluru, India
Shrikanth N. G.
Assistant Professor, Computer Science and Engineering,
Shree Devi Institute of Technology, Mangaluru, India

Abstract:
Now days, as the data in the internet is increasing and we are dumping the information in the web, some of the major problems heads towards us is that managing the big data, performance of the web, including these one more causes a serious aspect that is accessing the deep web services with high performance and the performance of the web can be measured by wspt tool (web server performance testing tool). The main purpose of wspt is to measure the performance of server side application. So to overcome the problem of accessing the deep web services hence proposed a system called as smart crawler and WPC, FSC, FCC and also the major constraint here is that maintaining the big data of the hidden web services is again a challenging problem because sometimes in deep web services the user requires only some highly ranked or visited sites but the existing smart crawler lists all the sites unrelated of ranks by this the user will be uncomfortable with it, so hence proposed the different framework that that can overcome this problem that contains the two kinds of database 1) temporary – contains only the highly ranked sites 2) permanent – contains all the related links of the search string given by the user.

Keywords: *WPC, FSC, FCC and smart crawler.*

1. Introduction

The name crawler means it's a program which visits websites and reads the content of it or may be any information which intern helps in creating the entries for the search engine index. The other names for this program is, "spider" or "bot". these crawlers are mainly used to visit the websites that have submitted by their owners in case of new or any updated documents. Here whole site can be indexed and selectively visited. It acquired the name crawler as it crawl's through the whole website and index the pages.

Web crawler has its own database than the search engines; it provides the best searches by crawling through out the web. The working of the simple web crawler is as follows, firstly it starts with the group of url's its also known as seeds. When the crawler starts visiting these url's it starts collecting the hyperlinks and their url's called as "crawl frontier", these url's are recursively visited according to some policies and archiving of url's is done, these archived url's are stored in the database in the form of snapshots but one disadvantage of this web crawler is that it can only download the limited url's so it needs to prioritize its downloads and here comes a problem that to avoid accessing the duplicate content by the crawlers because, sometimes for example an online photo gallery may offer a three different options so in that with the help of "HTTP Get" request which are unique, but in actual we have four options and the same options will be referred through many different url's this overcomes by the new proposed methodology called "smart crawler".

As the data is present in the large amount in web we need to just query the data either by post query or pre query here pre query means it just analyse the database of the web and the variation of structure and content of forms and post query means here we get the results by submitting query to the forms.

Here the efficiency of smart crawler is increased due to adding of temporary storage data which fetches only some highly rated websites if the user is satisfied with it then it's a good thing, the time is saved and also the efficiency is increased. If the user want search still more links of deep websites, then the data is fetched from the main database.

We usually have crawling policies as stated in the above paragraph in web crawlers that is, one is Selection policy, revisit policy, politeness policy, parallelization policy. In selection policy we need to search the pages and index them usually the search engines can index them but usually the search engine is indexable up to only 40% to 70% so here comes the most important thing is prioritizing the so many theories have been proposed by many scientists like vertical search engines restricted to top domains, breadth first search, back link, page ranking etc.

To overcome the above suspects, they proposed a system called smart crawler in which the search engines can index to the deep web services due to efficient algorithms it includes two main concepts known as site locating and in-site exploring, through these concepts

we are able to find the deep web services which are located in the web and also to improve the accuracy they include working of FSC(form structure classifier), WPC(web page classifier), FCC(form content classifier).
Here finding the deep web services which are hidden deeply in the web is quite challenging task there are many algorithms for searching the deep web services efficiently they are reverse searching technique, onsite learning, in site exploring and incremental site prioritizing and decision tree algorithm given by various scientists.
Here I am comparing the efficiency in terms of percentage of data in searchable form classification, form structure classification and all searchable forms.

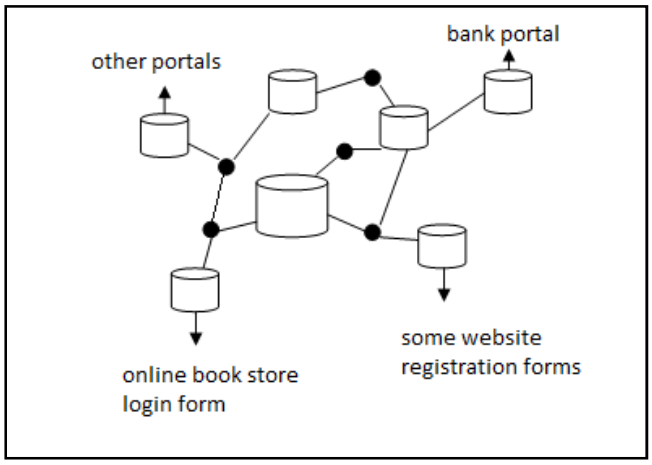

Figure 1: showing deep web databases connecting different websites

2. Related Work

2.1. Finding the Source of Contents in Deep Web
The study of recent papers has shown that the rate of harvesting the deep web is low as in because now at present environment people expect themselves and the technology that they use should be updated so some of them don't bother about the outdated data or technology that is most beneficial than the existing one so here generic crawlers are mainly meant for searching such deep web services many pervious papers show regarding the deep web samples [1], [2], [3]. Even if there are many methodologies to find the deep web services they are not content specific they just go on searching all random searchable forms. Earlier there was a concept called Meta Querier [4] which discovers the query interfaces automatically here it finds the root pages by IP based sampling but one IP address can have many virtual address so it fails to find many sites in the web thus this is the big disadvantage of MetaQuerier, to overcome this again they suggested stratified random sampling which combines both pre query and post query approaches and this was implemented in Russian search engine called Yandex

2.2. Selection of Relevant Links or the Data Source
There are many hidden web directories [6], [7], [8] as some crawler doesn't search specifically the focused crawler came to an existence that searches only for the relevant links or the URL's so that we get more information and its also said as best focused crawler because it crawls 100,000 movie related pages and harvests 94 movie search forms, some ofThe focused crawlers are ACHE and FFC [9], [10]here to improve the efficiency of crawler they used adaptive link learner and also automatic feature selection. And the smart crawler is the domain specific crawler which searches for the relevant documents and also categorizes the forms which are searchable and also provides ranking to it and here the WPC, FSC are also come under a domain specific methods or searches.

3. Proposed Work
The architecture of the proposed system is,

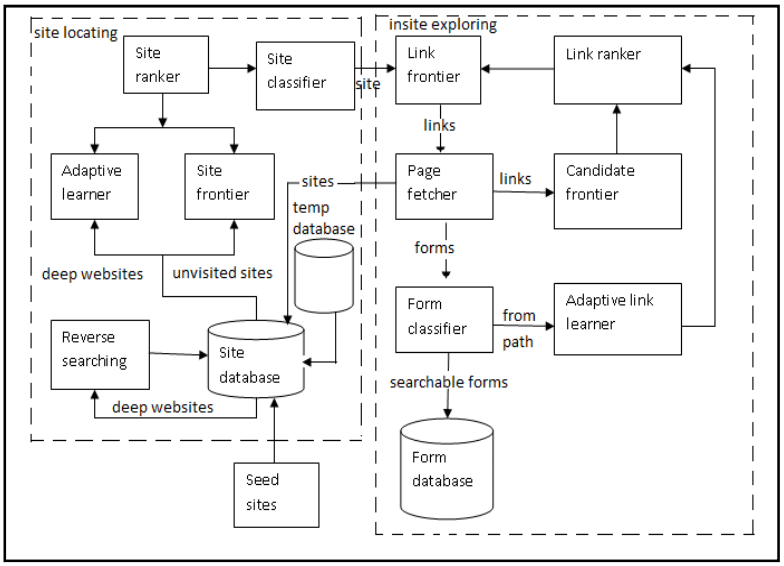

Figure 2: graphical representation of the result of smart crawler in different domains.

The output of efficient smart crawler is,

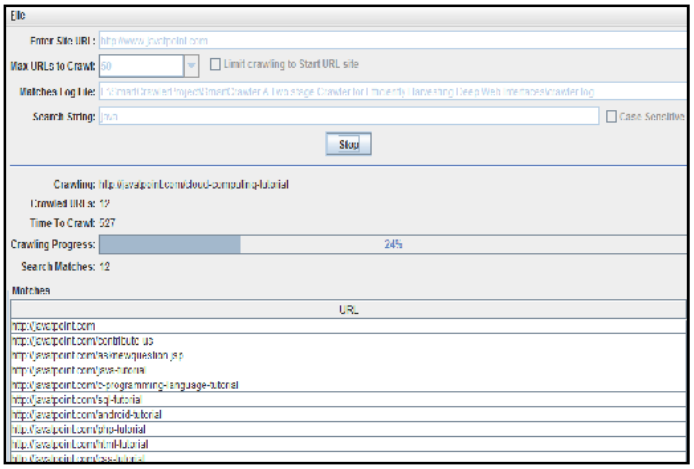

Figure 3: intermediate execution of efficient smart crawler

The result of the prequery and postquery methods are given by Ying Wang, Huilai Li, Wanli Zuo, Fengling He, Xin Wang, and Kerui Chen, is, Here increasing the performance of crawler by introducing a new architecture that contains 2 kinds of database as shown in the Figure 2, one is permanent database and another is temporary database in temporary database it contains the highly visited or highly ranked sites and in permanent database it contains all the links related to the search keyword specified by the user.

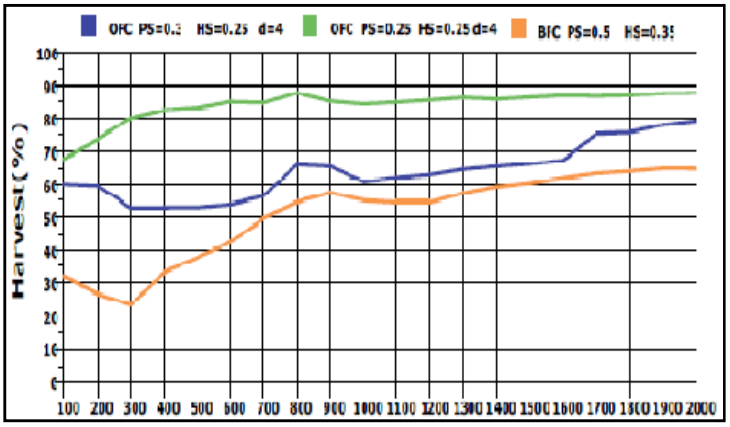

Figure 4: graphical representation of pre query and post query outcome.

The final result is got by the efficient smart crawler the output is shown as,

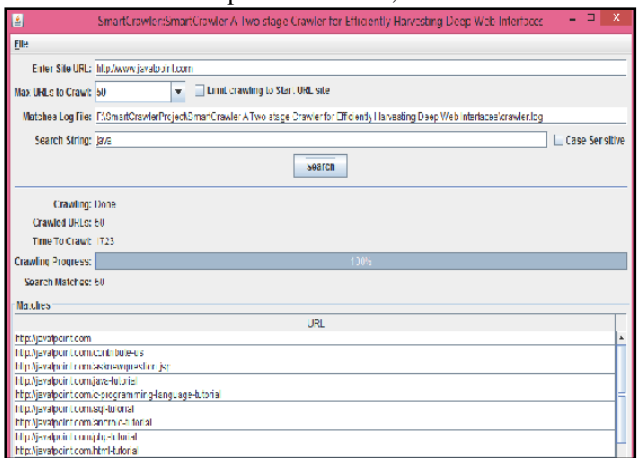

Figure 5: complete execution of the smart crawler

The output table of pre query and post query is given in the graphical representation it says as, the PS value of 0.25 is greater than 0.3 so the 0.25 PS (page similarity threshold) can miss some of the domains to search.

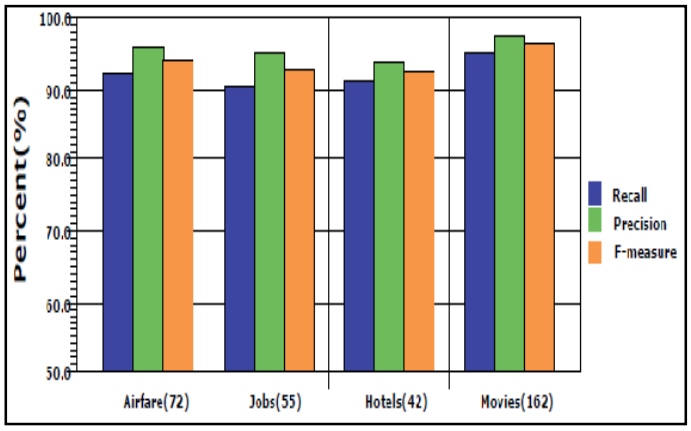

Figure 6

4. Conclusion

This paper says that the efficiency of the smart crawler can be increased or enhanced by using the 2 kinds of database one is temporary and another is permanent so by using these again the user can excess the data still more efficiently even when the information is being deposited from many years and also it avoids the random search of the sites which is not needed by the user.

5. References

i. Cheng Sheng, Nan Zhang, Yufei Tao, and Xin Jin. Optimal algorithms for crawling a hidden database in the web. Proceedingsof the VLDB Endowment, 5(11):1112–1123, 2012.

ii. Panagiotis G Ipeirotis and Luis Gravano. Distributed search over the hidden web: Hierarchical database sampling and selection. In Proceedings of the 28th international conference on Very Large Data Bases, pages 394–405. VLDB Endowment, 2002.

iii. Nilesh Dalvi, Ravi Kumar, Ashwin Machanavajjhala, and Vibhor Rastogi. Sampling hidden objects using nearest-neighbor oracles. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1325– 1333. ACM, 2011.

iv. Kevin Chen-Chuan Chang, Bin He, and Zhen Zhang. Toward large scale integration: Building a metaquerier over databases on the web. In CIDR, pages 44–55, 2005.

v. Shestakov Denis. On building a search interface discovery system. In Proceedings of the 2nd international conference onResource discovery, pages 81–93, Lyon France, 2010. Springer.

vi. Brightplanet's searchable database dirctory. http://www. completeplanet.com/, 2001.

vii. Clusty's searchable database dirctory. http://www.clusty.com/, 2009.

viii. Infomine. UC Riverside library. http://lib-www.ucr.edu/,2014.

ix. Luciano Barbosa and Juliana Freire. An adaptive crawlerfor locating hidden-web entry points. In Proceedings of the16th international conference on World Wide Web, pages 441–450.ACM, 2007.

x. Luciano Barbosa and Juliana Freire. Searching for hidden-web databases. In WebDB, pages 1–6, 2005.