International Journal of Engineering & Technology, 7 (2.7) (2018) 372-375

SPC

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

Enhancing the performance of search engines based heap based data file and hash based indexing file

Dr. Jkr Sastry [1] *, Chandu Sai Chittibomma [1], Thulasi Manohara Reddy Alla [2]

[1] Dept. of E.C.M Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India – 522502
[2] U.G Students, Dept. of E.C.S.E Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India
**Corresponding author E-mail: drsastry@kluniversity.in*

Abstract

WEB clients use the WEB for searching the content that they are looking for through inputting keywords or snippets as input to the search engines. Search Engines follows a process to collect the content and provide the same as output in terms of URL links. Sometimes enormous time is taken to fetch the content fetched especially when it goes into number of display pages. Locating the content among the number of pages of URLS displayed is complex. Proper indexing method will help in reducing the number of display pages and enhances the seed of processing and result into reducing the size of index space.
In this paper a non-clustered indexing method based on hash based indexing and when the data is stored as a heap file is presented that helps the entire search process quite fast requiring very less storage area.

Keywords: *File Organization; Heap Files; Indexing; Search Engine; Information Retrieval; Snippets; Search Speeds.*

1. Introduction

Information decimations is being done using WEB. Information is made available to users in just few seconds through search engines. Many search engines have been introduced quite recently and all of these almost behave in similar fashion.
The search engines follow a process, starting from accepting snippets from end user and then looking for the URLs at which the content has the snippet words. A crawl is a moving agent that is made to visit the WEB sites based on the Meta directory main-tained for each of the WEB site. The Crawler when finds an URL having the desired content writes the URLs to a database as it moves.
The databases of URLS are indexed based on the snippet words using which the URLs are retrieved. The URLs in the indexed database are then ranked based on some criteria as recommended by a ranking algorithm. The URLS as per the ranking in the as-cending order displayed for the end user. It becomes the responsi-bility of the end user to visit the URLs reported to find where the content for which they are looking is resident. A kind of mini search is required within the list of URLs displayed. At times the users land up to read the content to find if it is of use to them.
The search process described above consumes lots of time when the content retrieved by the search engine is enormous and runs into thousands of pages. If the number of pages of URLs retrieved runs into beyond the cache size the search fails.
The URLS fetched by the crawler are stored in a database using one of the file structure usually the Inverted tree structure and a clustered index based on a key developed on snippet strings. This kind of file organization and indexing leads to enormous data storage also takes enormous time in retrieving the content from the disk for the purposes of ranking.
In this paper a heap based file structure with a hash index is used to achieve less storage requirement for storing the retrieved URLs and for loading the indexes.

2. Related work

Google Scholar is one of the search engines that is quite fre-quently used for providing search on publications. The algorithm being used by them has not been published. A reverse engineering technique [1] has been found to determine the kind of indexing and ranking process used by Google to develop Google scholar
search engine. It has been found that Citation count of the papers published is being used for ranking the publications. Highly cites papers are more often found when a search is made. It has been found that the snippet words influence more in the search compared to number of citations.
With the advent of internet more and more information is being hosted on the WEB. While some information hosted on the WEB is useful and correct, some information is useless. Sometimes incorrect information is being hosted on the WEB. No control as such exists in this case. Search engines generally fetch the infor-mation required and also not required as the search is generally undertaken using snippet words. It has been found that considering user behavior profile / behavior as a part of search process will fetch exact information required by the users. Many ideas have been floated [2] that can be considered for building the user be-havior within the search process
The traditional algorithms that have been in existence for web searching and caching have not been quite effective especially when the speed of information addition to the WEB is quite rapid and high. Clicking through data analysis which is an inverted file replacement algorithm has been presented [3] which do efficient web caching. A new cache policy has been used which is based on poison arrival model. It has been found that the retrieved information organised as inverted file which enhance the speed of searching quite rapidly.

Mining algorithms are being used for web searching. A deep extraction tools have been presented that uses clustering technique for web searching [4]. The presentation by the authors is limited to information related to researchers and scientists. A group of in-formation is identified as a cluster and the users will make a choice on the cluster using which the content encapsulated into the cluster will be displayed.

Most of the algorithms meant for doing web searching are based on the number of back links that a site has. The algorithm involves in safe building of the links with each of the link weighted with velocity No compromise on page optimisation as such is made. One of the important strategies is to rank a page on a keyword [5]. Every organisation can find the keywords and link the URLS/pages to the keyword and rank the pages bases on the fre-quency of usage of a keyword for searching. The techniques eliminate the sandboxing of the search into already available search engines.

A tutorial has been made present on the WEB [6] that identifies the participants for WEB surfing system. They have expressed that keywords must be recognized considering the customer require-ments and the competitors choices and design the WEB pages that are inbuilt with keywords. A search engine uses the keywords for indexing and ranking

Finding the exact required information from WEB is tough due to existence of extensive information on the WEB. Search engine optimization has become enormously important for this reason [7]. Search engine optimization involves analyzing the WEB data from the perspective of different users and handling the WEB based on the analysis results. Web logs contain information to certain extent logging the information related to different queries that have been processed. The info log includes the times stamps, URLs processed, User identification etc. The user's experiences can be analyzed through analysis of query logs / web logs. The history contained within the web logs can be used effectively for understanding the user behavior and accordingly uses the same for query optimization. The history contained in the WEB log can be grouped dynamically and in an automated manner. The Groups can be then used for optimization through query alteration, re-ranking, query replacement etc. A method [7] has been proposed that links the query groups with URLs that are related over general information needs. It has been proposed to combine word similari-ty measures with document similarity measures and form into a combined similarity measure. Other measures also are considered that include query reformulation, clicked URLs etc.

Search engines are being used for querying the information required by web sufferers. The users of the Web require only the top most of the results they require. The web sites designed are more bothered about promoting the WEB site for access by search engines. Search engine optimisation has become very important and sometimes the techniques used might break the rules and regulations followed by the search engines. A user for instance might be clicking on the same URL several times misleading the rule that the URLs with high clicked count be ranked on the Top. To avoid this Ranking Algorithm that uses the IP address of the users to track the clicking on the URLs has been presented [8].

Trend related queries can be found when users interact with the WEB. A context search engine considers query trends which can be traced from different types of domains. The requirements of the users can be represented as a series of queries based on the search intentions of the user. The context search engines helps in providing search results as required by the end users [9].

Web mining is actually carried for providing the query results. Both web structure mining and web content mining are usually carried for web searching. Page ranking algorithms are used for web structure mining and some algorithms that include HITS and page content ranking are used for mining both WEB structure and WEB content. A new method has been proposed based on the weighted pages and content ranking which uses all web mining techniques (structure, content, usage mining) [10].

Search engines are the only means available to locate and make available the information required. A combination of manually edited directories, automated algorithms and advertisements on the WEB for generating search results. A new algorithm [11] has been presented that implements a modified page rank algorithm that allocates weights to the in-linked web pages. The weights are distributed to all outbound pages based on the popularity of those pages. This kind of algorithm is called Weightage in-link rage rank algorithm. The algorithm calculates score of every individual web page and the score is used for ranking.

WEB logs provide a source for analyzing the user behavior while transaction on site especially the e-commerce sites. Data mining techniques can be applied to WEB log data for revealing interesting patterns. The user's behavior as such is modeled using web log in most static way. The sequence of operations carried by the user generally is dynamic and the static data is not good enough to depict dynamic behavior of the users. Capturing the behavior of the users while interacting through the WEB site in terms of the process followed is more complex and interesting.

A linear-temporal logic model checking approach has been presented for analyzing the e-commerce web logs [12]. The web log records if can be related to event logs dynamic behaviors of the user can be traced.

Driving traffic on the WEB sites has become quite complicated as the completion is increasing to find the data related to an estab-lishment to be right on the top of search results. It is usual that the internet users navigate through the pages which are on the top of the list. Indexing and ranking of the searched pages are usually done, to order the fetched pages in ascending of their ranking. Some of the WEB site owners apply search engine optimisation techniques for optimising the content to ensure that their pages are reflected on top of the search results. Serval methods [13] have been presented in the literature for optimising the search process.

Google SEO On-page and Off-page techniques are one such technique that can be used for search engine optimisation. The per-formance of a search engine can be computed by using the SEERP metric.

3. Investigations and findings

The search process that is generally followed includes a web crawler which is made to visit every web site and find the pages if it contains the desired content by the users. The URLs fetched are stored in a database as it crawls various WEB sites. One optimization technique used is to match the key words entered by the users with the Meta words stored within a web site. Only when the key words and Meta words matches, the content checking is done fetching the URLs of the documents that has the content matching the keyword entered. All the URLS that are fetched are indexed into the database using the Key words.

The URLs are ranked based on the importance of the URLS by following some criteria, for instance, the clicked count on the URLs.

Many algorithms exist in literature for ranking the URLs means the WEB pages. The URLs are weighted based on some criteria. The URLS are ranked based on the weighted criteria and the URLs are returned to the user in the ascending order of Ranks assigned to the URLs.

The process followed is shown in Figure 1. The searching is done in the lines of above mentioned process using sample key words and just browsing through a Known WEB site

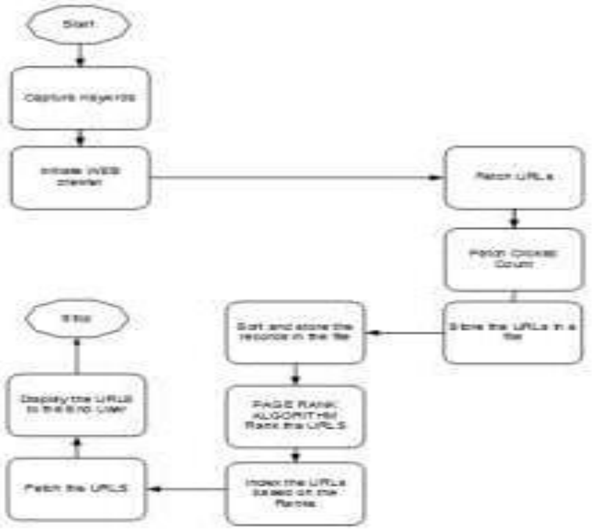
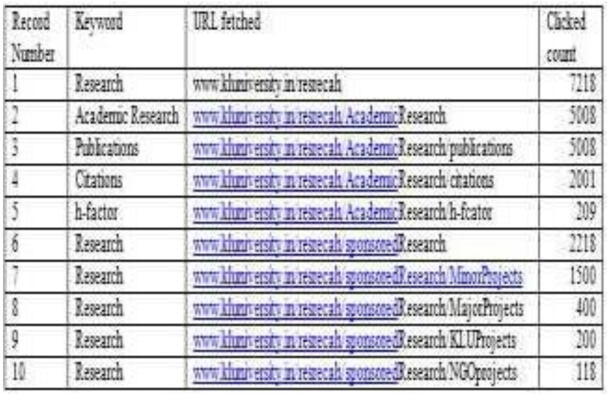

Start
Capture Keywords
Initiate WEB crawler
Fetch URLs
Fetch Clicked Count
Store the URLs in a file
Sort and store the records in the file
PAGE RANK ALGORITHM Rank the URLS
Index the URLs based on the Ranks
Fetch the URLS
Display the URLS to the End User
Stop

Fig. 1: General Search Process.

One of the main processes that are inbuilt into a search engine is storing the URLs fetched into the database and then Index the same on a keyword, generally the snippets using which the URLs have been fetched. A file organization method and indexing meth-od are required to store the data on a database and index the same for easy retrieval. In most of the search engines, the data is stored in the ascending order of the key which in this case is the snippet word. The records are accessed either using a binary or sequential search. Binary search or sequential search is time consuming. Table 1 shows the information fetched by the WEB crawler in identifying the URLS that has the snippet words and also by pro-cessing the WEB log to determine the clicked count.

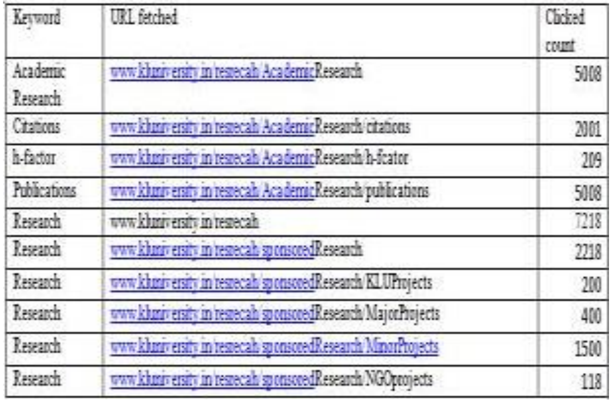
Table 1: Fetched URLS with Clicked Count



Record Number	Keyword	URL fetched	Clicked count
1	Research	www.kluniversity.in/resecah	7218
2	Academic Research	www.kluniversity.in/resecah/AcademicResearch	5008
3	Publications	www.kluniversity.in/resecah/AcademicResearch/publications	5008
4	Citations	www.kluniversity.in/resecah/AcademicResearch/citations	2001
5	h-factor	www.kluniversity.in/resecah/AcademicResearch/h-fcator	209
6	Research	www.kluniversity.in/resecah/sponsoredResearch	2218
7	Research	www.kluniversity.in/resecah/sponsoredResearch/MinorProjects	1500
8	Research	www.kluniversity.in/resecah/sponsoredResearch/MajorProjects	400
9	Research	www.kluniversity.in/resecah/sponsoredResearch/KLUProjects	200
10	Research	www.kluniversity.in/resecah/sponsoredResearch/NGOprojects	118

The data fetched is stored in a Clustered file using the snippet keyword as the key. Many URLS can be fetched that satisfy the same key word. The snippet keyword as such cannot be used as the primary key as many URLS exists for the same key value leading us to consider the URL along with the snippet word to form the Primary key for the clustered file. The primary key now becomes a complex key. Fetching a record based on a complex key having string strength of about 256 characters is quite complex and also time taking. The records are to be fetched either sequentially or by implementing a binary search as the key fields appear in the ascending order of key values. The data in the Table 1 is first sorted and then written on to the clustered file. Table 2 shows the way the records are written into file on the disk. To fetch a record, one has to consider the keyword and URL fetched which forms into long string. Binary search as such becomes infeasible due to the ling key string. Fetching a record as such becomes complex and takes enormous time. No separate index as such needs to be stored in this case.

Table 2: Ordering the Records Fetched by the Search Engines



Keyword	URL fetched	Clicked count
Academic Research	www.kluniversity.in/resecah/AcademicResearch	5008
Citations	www.kluniversity.in/resecah/AcademicResearch/citations	2001
h-factor	www.kluniversity.in/resecah/AcademicResearch/h-fcator	209
Publications	www.kluniversity.in/resecah/AcademicResearch/publications	5008
Research	www.kluniversity.in/resecah	7218
Research	www.kluniversity.in/resecah/sponsoedResearch	2218
Research	www.kluniversity.in/resecah/sponsoedResearch/KLUProjects	200
Research	www.kluniversity.in/resecah/sponsoedResearch/MajorProjects	400
Research	www.kluniversity.in/resecah/sponsoedResearch/MinorProjects	1500
Research	www.kluniversity.in/resecah/sponsoedResearch/NGOprojects	118

4. Proposed algorithm

Yet another efficient method is to store the fetched URLS in to unclustered heap file and develop an index by using a hashing algorithm. In this case a separate storage is used for indexing using an hashing algorithm which is simple to implement. The revised process flow for the proposed system is shown in Figure 2.

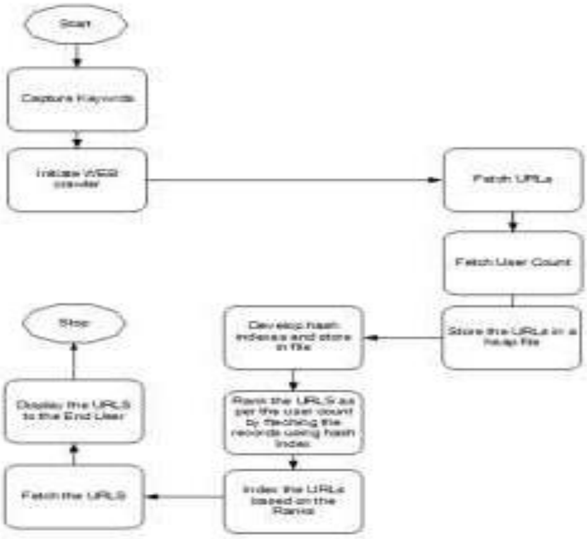

Start
Capture Keywords
Initiate WEB crawler
Fetch URLs
Fetch User Count
Store the URLs in a heap file
Develop hash indexes and store in file
Rank the URLS as per the user count by fetching the records using hash index
Index the URLs based on the Ranks
Fetch the URLS
Display the URLS to the End User
Stop

Fig. 2: Revised Search Flows.

In this case no pre-ordering is required. The data records are write as they arrive as in Table 1. Indexes are generated for each of the record based on the hash function. Table 3 shows the layout of the index file that has the indexes stored in it and also the records that are pointed by the Index. Hash value can be generated by adding the alphabetical values of the characters contained in the Snippet words and URL strings. When a record is needed using its snippet string, has value is computed, and suing it record entry is fetched and using record ID the record can be fetched. This process s such is quite faster.

Table 3: Hash Index File Entries

Hash value	Record Entry	Record Entry	Record Entry	Record Entry	Record Entry	Record Entry
2018	1	6	7	8	9	10
2001	2	-	-	-	-	-
2016	3	-	-	-	-	-
2003	4	-	-	-	-	-
2008	5	-	-	-	-	-

5. Comparative analysis

Both the approaches above are compared to find the effective ness of the algorithm so as to arrive at overall view of the approaches. Table 8 shows the Comparison of the above mentioned approaches.

Table 8: Comparison of the Two Basic Approaches

Parameters	Approch-1	Approch-2
Number of Keywords used	5	5
Number of URLS fetched	9	9
Whether Pre-sorting required	YES	NO
Number of additional operations required for Pre-Sorting	O(N)	-
Whether index file required	No	Yes
Storage area required for storing data and Index (10% Extra Storage for Index)	X	X*1.1
Number of operation required for storing and fetching records	O(N+N)	2

From Table 8 it can be seen storing and fetching records is quite faster when heep organization is used for storing the records and hash indexing is used for fetching the records

6. Conclusions

Every search engine involves in finding the URLs at which the user expected content is stored. The fetched URLS are to be writ-ten on to data file as the crawler moves from one server to the other. The way data is written on to the file actually dictates the speed of the search engines. The problem becomes compacted when a search engines returns too many matches of the URLs. The use of heap files and hash indexed files will handle any number of matches and also help in grouping of the matches into different display pages.

References

[1] Joeran Beel, Bela Gipp, Google Scholar’s Ranking Algorithm: An Introductory Overview International Conference on Scientometrics and Informatics (ISSI’09), volume 1, pages 230– 241, (Brazil), July 2009.

[2] P. R. Chinagongjun, Analysis the idea of personalized search engine based on user behavior International Conference on Com-puter Application and System Modeling (ICCASM 2010).

[3] Zhang Feng and LiXia-Long, Research in Automatic Search Engine Replacement Algorithm For Web Caching Based on User Behaviour, International conference on WEB information systems and applications, 2010, pp. 142-145.

[4] BidishaRoy, Joy Machado, Melicia raj, Gnana Sonica Nadar, Exploiting Web Search to Access IEEE papers, International Journal of Computer Applications, (IJCA), 2012.

[5] Saravankumar S, Ramanath K, Ranjitha R, Ghokul V G, A new methodology for search engine optimization without getting sand boxed, International journal of advanced research in comput-er and communication engineering, Vol. 1, Iss. 7, 2012.

[6] JOHN B. KILLORAN, How to Use Search Engine Optimiza-tion Techniques to Increase Website Visibility IEEE transactions on professional communication, VOL. 56, NO. 1, 2013.

[7] Jayasree M, Analyzing and Classifying User Search Histories for Web Search Engine Optimization International Conference on Eco-friendly Computing and Communication Systems-IEEE DOI 10.1109/ICECCS.2014.19, 2014.

[8] Roop Kaur, Development of a Ranking Algorithm for Search Engine Optimization, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 IJERTV3IS041023Vol. 3 Issue 4, April – 2014.

[9] Shogo Kori, Yanjun Zhu, Koichi Yamaguchi, Satoru Takigu-chi, Yasufumi Takama, Analysis of User's Behaviour Based on Search Intentions for Information Retrieval Using Search Engines, TAAI2015 Tainan, Taiwan Nov. 20-22, 2015.

[10] Ekta Bhardwaj1, Shiv Kumar2, Kuldeep Tomar3, Enhancing Page Rank Algorithm, International Journal on Recent and Inno-vation Trends in Computing and Communication, Volume: 3 Is-sue: 5, 2015.

[11] Rekha Singhal , Enhancing the Page Ranking for Search Engine Optimization Based on Weightage of In-Linked Web Pag-es IEEE International Conference on Recent Advances and Inno-vations in Engineering (ICRAIE-2016), December 23-25, 2016.

[12] Pedro A´ lvarez, Analysis of users’ behavior in structure de-commerce websites, IEEE, DOI 10.1109/ACCESS-2017.

[13] Dukagjin Sadrijaj Investigating Search Engine Optimization Techniques for Effective Ranking: A Case Study of an Education-al Site, Mediterranean conference on embedded computing (MECO), 11-15 JUNE 2017.