IJRRECS/December 2013/Volume-1/Issue-8/1836-1841 **ISSN 2321-5461**

INTERNATIONAL JOURNAL OF REVIEWS ON RECENT ELECTRONICS AND COMPUTER SCIENCE

SCHEMING OF THE QUERY PROCESSING SYSTEM FOR WEB SEARCH ENGINE

N.Sree Divya[1]

[1]Assistant Professor, Mahatma Gandhi Institute of Technology, Hyderabad, A.P, India

sridivya1234@gmail.com

ABSTRACT:

The web search engine is software that indexes web documents which are collected from the Internet in addition to giving instructions to them in accordance with their query relevancy regarding an entered query of user. Nowadays, the web search engine is extensively used as a general way to discover information of interests and its documents of indexed has attained the extent of multiple billions. To build such cooperation resourceful, the query processing system is intended as clustered servers and the server clusters are associated to each other by means of high-speed LANs. To design the structural design of the query processing system, it is essential to recognize how a user query is responded in the system. The summarization information for document ID is included of its URL, title, as well as the passages of keyword-including extracted from the equivalent document's body text and such data for a document ID is known as Document Summarizing Text.

Keywords: Web search engine, Query processing system, Clustered servers, Document ID.

1. INTRODUCTION:

A lot research has been made to work out various problems associated to the web search engine, for instance crawling web documents, analysis of hyperlink, topic sensitive searching and high-performance indexing. In view of the fact that the system has to index an enormous size of data, the outlay intended for yielding a query result could be extremely high. Consequently the public service of web searching might be infeasible for excessive server costs, if there were no query processing system that can economically decrease the resource

utilization throughout query processing. Nowadays, the web search engine is extensively used as a general way to discover information of interests and its documents of indexed has attained the extent of multiple billions [4]. In view of the fact that the amount of used CPU and resources of I/O resources is so gigantic for query processing, in excess of one server has to effort in parallel even for handing out a single query. To build such cooperation resourceful, the query processing system is intended as clustered servers and the server clusters are associated to each other by means of high-speed LANs [8]. The cache system is devised to contain hierarchical 4-level cache data and in the cache of top-level, the current search result pages are accumulated in main memory, and the outstanding lower levels of caches exist in the disk for saving more results of query. By means of parsing the query URL, the web server takes out the keyword and the range of retrieval, and subsequently sends them to other server of query processing. To make a query result page subsequent to the evaluation of ranking, it is necessary to generate the summarization information of the document ID [1]. In view of the fact that the hit rates of multi-level caches extremely depend on the attributes of the queries of entered user, it is not probable to estimate the accurate cache hit rate of each web search engine. The server of coordinator receives user queries by means of web servers and carries out two-phase query processing in synchronization with ranker severs as well as document summarizing text servers [11]. Since the outlay for query processing by means of cached data is insignificant regarding that for queries of non-cached, the multi-level cache gets better the throughput of the implement system.

2. METHODOLOGY:

To design the structural design of the query processing system, it is essential to recognize how a user query is responded in the system. When a user inputs a query of web, the query is sent to a server of web within the system and it is symbolized by an URL, which encloses user's keyword and the range of retrieval [3]. The query processing system is intended as clustered servers and the server clusters are associated to each other by means of high-speed LANs. The range of retrieval identifies the documents of rank range of query-matching to be revealed in the page of query result returned to the user. The range of retrieval is

altered with clicks of user on the link of next/previous page specified in the current page of query result. By means of parsing the query URL, the web server takes out the keyword and the range of retrieval, and subsequently sends them to other server of query processing [9]. The server of query processing is accountable for performing the equi-join and the evaluation of rank. The equi-join is to decide on the documents of query-matching relating to specified keywords. For this, inverted files are used to recognize the documents where the complete specified keywords take place at least once. Subsequent to the operation of equi-join, rank evaluation is carried out to provide rank scores to the documents of query-matching in accordance with the relevancy of documents' query [7]. Our ranking system refers to several index data together with the analysis results of hyperlink, occurrence positions of keywords, and the information of HTML-tag related. This index data is also accumulated in inverted files all along with other information used for equi-join. To make a query result page subsequent to the evaluation of ranking, it is necessary to generate the summarization information of the document ID. The summarization information for document ID is included of its URL, title, as well as the passages of keyword-including extracted from the equivalent document's body text and such data for a document ID is known as Document Summarizing Text [2] [12]. To make a Document Summarizing Text, a HTML-tag free body text saved in the disk was read and chooses pieces of text enclosing numerous keywords and this generation of Document Summarizing Text is very I/O demanding and CPU task of consuming, for the reason that we have to read body texts of documents that are at random positioned in a huge volume of disks and a number of operations of string matching are carried out over the body texts. After congregation of the Document Summarizing Text, a single page of result can be generated by means of assemblage of them and introduction of proper tags of HTML. In fig1 the query processing system is installed at a site of Internet data center intended for constant Internet connectivity [5]. The port of Internet data center in the figure makes available network bandwidth in addition to a firewall reside behind it for the purpose of security. The load balancer of a switch of L4 acts as a load balancer that dispatches queries of user toward four web

servers in the fashion of round-robin. The performance monitor constantly collects the performance information for instance response times, entered queries rate, workloads of servers. If any difficulty is noticed, subsequently it sends a caution message towards the administrator. Below the web servers, the most important components of our query processing system are represented as coordinator servers, document summarizing text servers and ranker clusters [10]. The server of coordinator receives user queries by means of web servers and carries out two-phase query processing in synchronization with ranker severs as well as document summarizing text servers. During the initial phase, the coordinator server transmits a query towards the four ranker servers promptly. In view of the fact that the hit rates of multi-level caches extremely depend on the attributes of the queries of entered user, it is not probable to estimate the accurate cache hit rate of each web search engine [13]. The whole information of index is separated into four index files and they are accumulated in four ranker clusters, correspondingly. The ranker servers in the similar ranker cluster contain the similar partition of files of index. By sending a query towards four ranker servers of dissimilar clusters, the time for the equi-join can be expected and operations of ranking is reduced owing towards parallel processing of ranker servers. At the end of the initial phase, the coordinator combines the returned consequences [6]. At the subsequent phase, the coordinator server transmits the lists of document ID attained at the initial phase to document summarizing text servers in parallel. The document summarizing text servers generate document summarizing text data for every received document ID and return it to the server of coordinator. The server of document summarizing text accumulates URLs, titles, in addition to tag-free body text of all the documents of crawled web within the disk, and makes use of a scheme of hash to read each of them. By integration of the document summarizing text the coordinator server terminates the second phase.

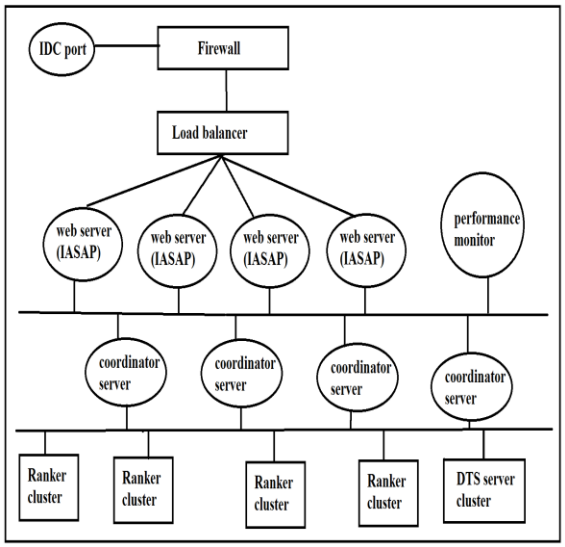

Fig1: An overview of Architecture of the QPS.

3. RESULTS:

In view of the fact that the hit rates of multi-level caches extremely depend on the attributes of the queries of entered user, it is not probable to estimate the accurate cache hit rate of each web search engine. The operational information collected from the real web search engine may possibly be an instruction for similar systems. Since the outlay for query processing by means of cached data is insignificant regarding that for queries of non-cached, the multi-level cache gets better the throughput of the implement system.

4. CONCLUSION:

The public service of web searching might be infeasible for excessive server costs, if there were no query processing system that can economically decrease the resource utilization throughout query processing. To design the structural design of the query processing system, it is essential to recognize how a user query is responded in the system. To build such cooperation resourceful, the query processing system is intended as clustered servers and the server clusters are associated to each other by means of high-speed LANs. The hit rates of multi-level caches extremely depend on the attributes of the queries of entered user, it is not probable to estimate the accurate cache hit rate of each web search engine.

REFERENCES:

[1] Sergey Melnik, Sriram Raghavan, Beverly Yang, and Hector Garcia-Molina. Building a Distributed Full-text Index for the Web, In Proc. of the 10th International World Wide Web Conference. pp. 396-406, 2001.

[2] Tiziano Fagni, Raffaele Perego, Fabrizio Silvestri, and Salvatore Orlando, Boosting the performance of Web search engines: Caching and prefetching query results by exploiting historical usage data, ACM Trans. On Information Systems, Vol. 24(1), pp. 51-78, 2006.

[3] Maxim Lifantsev and Tzi-cker Chiueh, I/O-Conscious Data Preparation for Large-Scale Web Search Engines, In Proc. of the 28th VLDB Conf., pp. Hong Kong, 2002.

[4] Arvind Arasu, et al., Searching the Web, ACM Trans. on Internet Technology, Vol. 1(1), pp. 2-43, August 2001.

[5] Taher H. Haveliwala. Topic-sensitive PageRank, In Proc. of the 11th International Conf. on World Wide Web, 2002.

[6] Search Engine Report, Http://www.searchenginewatch.com, 2005.

[7] Zheng Chen, Shengping Liu, Liu Wenyin, Geguang Pu, and Wei-Ying Ma, Building a web thesaurus from web link structure, In Proc. of the ACM SIGIR' 03, pp. 48- 55, Toronto, Canada, 2003.

[8] Alfred V. Aho and Margaret J. Corasick, Efficient String Matching: An Aid to Bibliographic Search, Communication of the ACM, Vol. 18(6), pp. 333-340, 1975.

[9] Soumen Chakrabarti, Kunal Punera, and Mallela Subramanyam., Accelerated focused crawling through online relevance feedback. In Proc. of the 11th International Conf. on World Wide Web, pp. 148-159, 2002.

[10] Maxim Lifantsev and Tzi-cker Chiueh, Implementation of a modern web search engine cluster, In Proc. of the USENIX Annual Technical Conference, Texas, 2003.

[11] Andrei Z. Broder, Marc Najork, and Janet L. Wiener, Efficient URL Caching for World Wide Crawling, In Proc. of the 12th WWW Conference, Budapest, Hungary, 2003.

[12] Larry Page, Sergey Brin, R. Motwani, and T. Winograd, The PageRank Citation Ranking: Bring Order to the Web, Stanford Univ. Technical Report, 1998.

[13] Sriram Raghvan and Hector Garcia-Molina. Crawling the Hidden Web. In Proc. of the VLDB Conference, pp. 129-138, 2001.