Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080645

A New Replacement Algorithm of Web Search Engine Cache based on User Behavior

Zhang Yong-Heng[1,∗], *Zhang Feng*[1,2,∗] *and You Fei*[1]

[1] School of Information, Yulin University, Yulin 719000, P. R. China
[2] School of automation, Northwestern Polytechnical University, Xian 710072, P. R. China

Received: 15 Nov. 2013, Revised: 13 Feb. 2014, Accepted: 14 Feb. 2014
Published online: 1 Nov. 2014

Abstract: The efficiency of retrieval system is crucial for large-scale information retrieval systems. By analyzing the documents and the users query logs of a real search engine based on the Web caching, through a large number of statistical analyzed of user behavior and found that the search engine query terms entered by the user in the process of clicking and queries to the URL of the page showed a clear temporal locality, and the distribution of user queries characteristics meet power function and has a good self-similarity. In this paper,analyzed of the search engine ranking based on the user behavior investigated mass distribution of information on the website and use the URL into the mirror degree, directory depth parameters and other web related degree feedback,then a replacement algorithm for Web caching is proposed. Through the establishment of retrieval performance model for analysis and simulation results show that this approach under the search algorithm can effectively reduce the execution time of retrieval, and the optimal parameter selection for this blocking organization is discussed.

Keywords: behaviour characteristics, information retrieval, search engine, search algorithms, cache replacement

1 Introduction

With network and the rapid development of information resources, Web search engines have become the main way to access network information. But now people are usually just simple words through a short query and retrieval systems to communicate, and in the network information resources, such large-scale cases such communication is not enough, more accurate retrieval systems often do not return to the user the real needs of information [1]. Therefore, the search engine user behavior analysis is very necessary. General search engine system major maintenance of two kinds of information, one is the Web page and collected related information, the other is the user behavior information collected in the service process (recorded in the log file) [2]. The former refers to the robot from the web crawl webpage through analyzer to analyze the obtained after processing information, including keywords, abstract information, meta information contained in the webpage (such as webpage author, length, modify time) as well as URL hyperlink information, this information is usually used as the output information to the user. The query term, then a kind of information including the user input query time, the IP address of the user, the user clicked on the output page of interest in the page's URL.

In the analysis of the statistical results, we found that user behavior exhibited very strong locality, which inspired us to use the query cache and cache hot Click to improve system performance. We log user behavior as input, analog implementation FIFO, LRU and LFU-band attenuation of three kinds of cache replacement strategy tested its cache hit ratio, compares their advantages and disadvantages. In addition, we found that the adjacent N terms query term deviation distribution is stable, so guess user query entry process in line with the distribution of self-similarity, and thus we are validated to prove that our guess was right. Network traffic on the Internet is similar to the self-similar characteristics, the conclusions for the design and evaluation of a search engine system with high significance. The user query distribution statistics analysis showed that the user's query are very focused, indicating the feasibility of using a cache in a query: users often query is actually very few, the query frequency higher word query result caching, cache can use a very small capacity hit most of the user queries, so we can obtain larger cache hit rate with less space.

[∗] Corresponding author e-mail: 709863637@qq.com, tfnew21@sina.com

2 Existing research results

Cache technology is to improve the system performance and scalability is a kind of important means, has been widely used in computer application in the field of each [3]. How to effectively search using cache technology service system is also in recent years attracted a great deal of attention in the search engine. The literature [4] detailed analysis of search engine user query logfind user query has strong locality, the feasibility of proposed caching query results. Literature [5] was studied of the cache replacement algorithm, cache size and other factors on the performance of the system. The literature [6] proposed the semantic cache, the Boolean query results as the cache object, and using the accelerated subsequent query semantic relations between the executions of query results. This method can take full advantage of the correlation between different queries improve the cache hit rate, the disadvantage is limited to Boolean queries, it may affect the results sorted by relevance. [7] studied the IR context of interactive user queries inverted file caching and query execution method of combining, [8] studied an actual search engines (TodoBR) in inverted file cache on the system efficiency.

3 Web cache replacement algorithms

3.1 Inverted file performance model

Performance model is to give information relationship about N, M, p (i), d, B, r, and k, which can in a given system under the conditions of the internal parameters of its external behavior (throughput) is estimated. here p(i) and B, as well as several hypotheses to explain.

p (i) is the length of the table inverted statistical distribution function, the $M \times p(i)$ is the length of the table indicates the number of records i,$i \in [0,N]$.Then the average length of inverted table is $a = \sum_{i=0}^{N} i \times p(i) = \frac{1}{M} \sum_{j=1}^{M} s_j$

B is the support of the lower run inverted file system bottleneck bandwidth. Depending on the circumstances, it may be the disk I/O bandwidth, network bandwidth may be, we do not distinguish. The idea of the model discussed here is based on the amount of queries arrive simultaneously k, the amount of data to get a D, and then see whether there $\frac{D}{B} \leq r$.

For simplicity, we assume that the query $q_{1,}q_2,\ldots,q_k$ are simple, that is they are directly attributable to the collection TERMS; also assume that they are on the TERMS random and independent distribution.

Now consider the k-th output of the query result data, D. Each query is likely to reach the M words in any one of items, M query may involve any of 1, 2, ..., or the k-th, so the amount of data corresponding to different. If we can calculate the probability related item i, denoted by $f_{M,K}(i), i = 1,2,\ldots,k$, then we can have

$$D = d \times \sum_{i=0}^{n} p(i) \times f_{m,k}(i) \times a \tag{1}$$

The following focus on fm, k (i). First calculate the k random query words fall M the total number of all possible items, which is equivalent to from the set 1,2, ..., k to 1,2, ..., M of the number of mappings, that is M^k.

Then for i = 1, 2, ..., k, k query investigated the i-th inverted falls exactly on the table, and this is equivalent to consider the set 1,2, ..., k of the number of i-division, coupled with the i-th inverted table M may fall in any of the i-th on; former is the number of second Stirling S(k, i). Noting queries between different inverted lists are distinguishable, and therefore need to consider is arranged, so we can write

$$f_{M,k}(i) = \frac{S(k,i) \times P_M^i}{\sum_{i=0}^{k} S(k,i) \times P_M^i} \tag{2}$$

This is what we get an inverted file model performance. It is given directly to the k concurrent query data. Query processing algorithm of inverted files are usually not based on the complex, is not computationally intensive tasks.

For large-scale inverted file, the data from disk to memory or from memory is sent across the network where the main time-consuming and $\frac{D}{B}$is the time required to complete the response output. If we let $\frac{D}{B} \leq r$, i.e. $D \leq B \times r$, we may be discuss various situations in the D, such as the effect of M, p (i) effects, and the like. Do the following discussions, the data movement between disk and memory.

The system of document information retrieval support size generally can be divided into ”index” and ”non full-text index” two classes [9]. Non full-text index only need to tell us what documents containing the term specific, and full-text index also need to give the term appears in the relevant documents in the position information, appeared several times to several records. Thus, the D in formula (1), which is proportional to the size of the full-text index case and the average number of words in different documents, that is $\frac{\sum_{j=1}^{M}\sum_{i=1}^{N} f_{i,j}}{N \times M}$, In the case of non-full-text index is essentially constant (core message is a document number), we denoted c, which is usually a few bytes.

We can also be considered $a \times d$. For each inverted list (corresponding to a specific word t_j), the amount of data it in the full-text index case is proportional to $N \times DF(t_j) + T_N \times TF(t_j)$, the front part is inverted table document number and frequency of the length of occupancy, part the length of the position information of occupation. Because T_N is higher than that of N, so the system each term inverted table length is mainly determined by the word frequency of T_F and the scale of the data T_N. Under the condition of non full-text index is only $N \times DF(t_j)$. In the average case,

$$d \times a = \begin{cases} c \times a & ,\text{nonfull} - \text{textindex} \\ c \times \left(a + \frac{T_N}{M}\right) & ,\text{full} - \text{textindex} \end{cases} \quad (3)$$

In formula (3), for simplicity (but no substantial influence), we also use c to record a term in a document of a desired position data.

3.2 Inverted file caching strategies

Cache technology is to improve the system performance and scalability is a kind of important means, has been widely used in computer application in the field of each [10]. How to effectively search using cache technology service system is also in recent years attracted a great deal of attention in the search engine.

Caching techniques is to establish the validity of the cached object access sequence in the presence of localized characteristics [11]. With the operating system memory management, database systems, and Web proxy cache compared to a lot of research in these areas, search engines cache on the system relatively little research. Commonality between them, but the object is cached object features and differences in access mode, and each has its own characteristics. Search engine retrieval system typically cached objects to be studied can be divided into three types, namely query results, the intermediate results of Boolean operations, as well as inverted file [12].

Inverted file caching retrieval system uses a distributed architecture, organized by document divided into multiple index data service nodes, they are independent of the parallel execution of user queries submitted to the respective search results returned to the user query server rollup. All levels of cache location as shown in Figure 1.

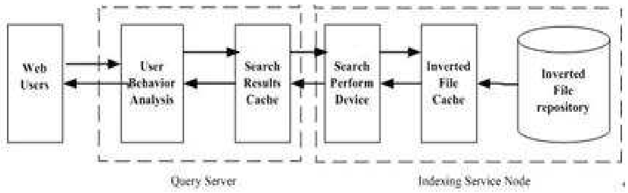

Fig.1: Search engine retrieval system cache structure

Inverted file cache is located in the indexing service node, the query executor access in executing user query process in the inverted file cache data [13]. A large number of statistical studies show that the user query sequence has good locality, can be expected to query execution device sends the read these query inverted data sequence also has the same properties, this is the basic starting point for people to study the inverted file cache.

3.3 Statistical analysis of user behavior

The major search engine system maintains two types of information, one kind is the Web page and collected related information, and the other is in the service process to collect information on user behavior. Inverted file caching retrieval system log file consists of log 1 and user click log queries for the user. The query log is recorded when a user submits a query, it records the user query keywords, submit the submission time, user IP, page number, whether the medium information in the cache. A simple user query log to record format is shown in Table 1.

Table 1: A simple user query log to record format

Name	Notes
2013-09-10 12:21:21	Submission time
208.20.400.45	The user IP
Database	If hit in the cache
Java	Query words
5	Page number

The user clicks the log is when users browse search results page click when recording, it records the time the user clicks on the page, click on the page's URL, user IP, click on the page number (the page's position in the query results), the click on the corresponding query terms and other information. The user clicks on a simple record of the log format is shown in Table 2.

Table 2: The user clicks on a simple record of the log format

Name	Notes
2013-09-12 14:42:55	Click time
208.20.400.45	The user IP
oracle.com /index.html	Click a URL
yulinu	Query words
13	Click on the page sort

In the analysis of the statistical results, we found that user behavior exhibited very strong locality, which inspired us to use the query cache and cache hot Click to improve system performance. We log user behavior as input, analog implementation FIFO, LRU and LFU-band attenuation of three kinds of cache replacement strategy tested its cache hit ratio, compares their advantages and disadvantages. In addition, we found that the adjacent N terms query term deviation distribution is stable, so guess user query entry process in line with the distribution of self-similarity, and thus we are validated to prove that our guess was right. Network traffic on the Internet is similar to the self-similar characteristics, the conclusions for the design and evaluation of a search engine system with high significance.

4 Performance tests

4.1 Mass analysis of web information

We sorted 1000000 collected webpage access by user in accordance with the number in descending order in the 2000 early April of baidu , set the URL sequence $U_1, U_2, \ldots, U_{1000000}$, its corresponding user clicks followed $V_1, V_2, V_{3,\ldots,\ldots,V_{999999}}, V_{1000000}$,their corresponding degree is $H_1, H_2, H_3, \ldots, \ldots, H_{999999}, H_{1000000}$,web mirroring number$C_1, C_2, C_3, \ldots, \ldots, C_{999999}, C_{1000000}$, URL directory depth is $D_1, D_2, D_3, \ldots, \ldots, D_{999999}, D_{1000000}$, in addition, we also added a reference sequence, its per an important degree conferred URL, that $S_1, S_2, S_3, \ldots, \ldots, S_{999999}, S_{1000000}$, where, Figure 2, Figure 3 illustrates were accessed by the user according to the user’s behavior over 156,000 pages (by clicks), web-degree, degree and directory mirroring depth sorting after the distribution. As can be seen, the user clicks the more URL, its pages, and mirror-degree relative degree higher directory depth performance is not obvious.

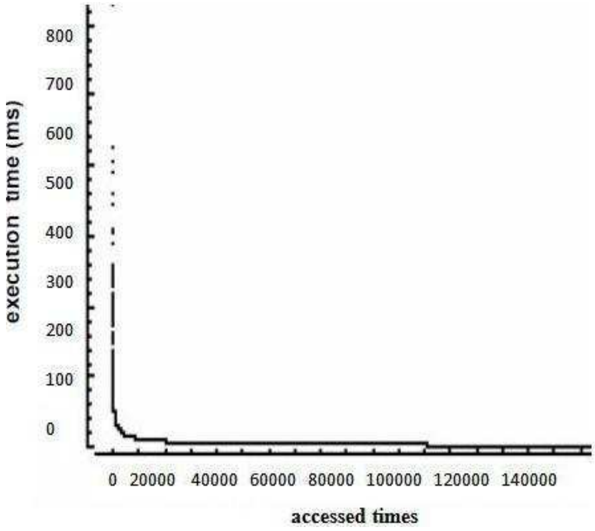

Fig.2: Data access object size distribution

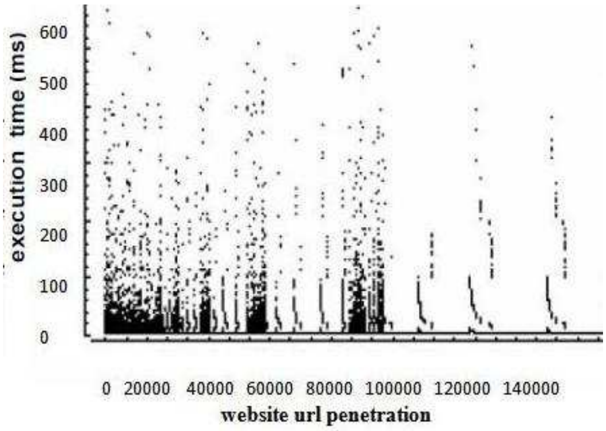

Fig.3: I/O and PAGE frequency distribution

4.2 Sequence of objects in the time interval distribution and repeat the pattern

Sequence of temporal locality can be an object on the same sequence of two consecutive visits to examine the distribution of time intervals. Use to access the location in the sequence interval; instead of using the absolute time, you can shield the user queries density in each time period for analysis of the impact of cyclical. The I/O sequences and the distribution of time intervals PAGE sequence where shown in Figure 4. As the direct distribution of time intervals are very scattered, the processing is to graph the data in 2000 as a unit from the group, showing that the frequency of each group. I/O sequence of slope 1.039, PAGE sequence is 0.764, indicating that the cache size in the same proportion of the next, I/O sequences can be expected to be higher than in PAGE sequence cache hit rate. A strong temporal locality is conducive to cache design, object access Freshness (freshness) is the replacement algorithm to be considered an important factor.

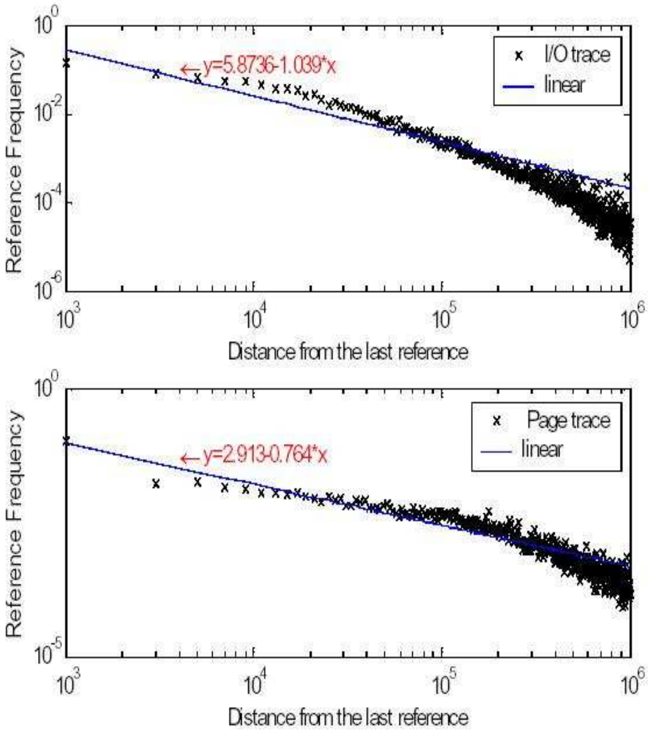

Fig.4: I/O interval distribution of PAGE

Sequence of spatial locality refers to the fixed pattern of repeat sequence, which can be arranged through the original sequence and random sequence after processing only the number of fixed-length strings to illustrate. Spatial locality is also a cache design factors to be considered.

Take I/O sequences and PAGE sequence the first 10 million data, to handle any length from 1 to 9 in which a continuous string, statistics only the number of strings. Then sequence of random rearrangements, repeat statistics. Is the sequence specified length string of only the number shown in Figure 5.

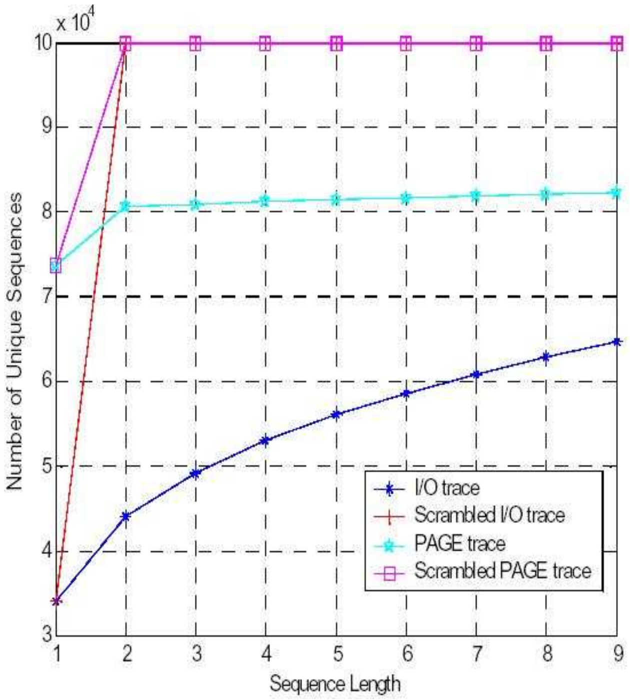

Fig.5: Model in the sequence string PAGE

5 Conclusions

Randomly ordered sequence destroyed the repetitive pattern, that is, destroyed the local characteristics of sequence space. The figure increases as the string length, only the number of strings has also increased. Random permutation of the sequence of the fastest rate of increase, and the spatial locality worst; I/O sequence of increased even the most gentle, its strong spatial locality, PAGE sequence followed.

In a retrieval system of the two efficiency indicators, the response time as the individual performance indicators is relatively easy to satisfy, it has also been the impact of system throughput, if the user queries the system may exceed the number of the load will result in increased query response delay. Therefore, the throughput of the system design and operation are more important. According to the previous discussion, the query lexical items, the average frequency, I/O performance and the determinants of the efficiency of a query, are using them to estimate the size of the system throughput and data relationships. Most of the user's query in the relatively small number of lexical items, query a topic with 2-3 words can describe the query topic, the article may have 10 or more lexical items. The user may wish to set up L_q item number of words in the query, it is estimated an average L_q is equal to 5. Come to the following inequality: m Throughput (a) In this inequality, the right word inversion table entries only consider the length of the main part of the $L_N \times TF$, The I/O disk access time is estimated at an average delay time and data transmission time. Suppose each inversion table is read into memory with a single I/O, to spend time can be estimated as $L_{latence} + T_n \times \frac{T}{IO_{bandwidth}}$, each row of the table to read the time by inverted $L_q \times m$ must be not more than 1 second, when the system's I/O Performance $T_{latency}, IO_{bandwidth}$ and TF finalized, and we get with the inverse relationship between m.

This article is based Web search engine study cache replacement algorithm for the application background of user behavior, puts forward a kind of inverted file buffer to replace strategy. First of all, a search engine in the statistical documentation and user queries based on log data,set up a search engine cache inverted index retrieval performance model.

Through the model analysis and simulation study, shows the method of inverted file implementation of the retrieval algorithm can significantly reduce the time. Further work will improve the performance model, to study for the inverted file compression, query optimization of the index near the inverted file structure block organizational issues, and further verified in the actual system.

Acknowledgement

This work is partially supported by Natural Science Basic Research Plan in Shaanxi Province of China(Program No.2012K12-03-08, 2013JM8005) , Funding Project for Department of Education of Shaanxi Province (No.2013JK1151, No.2013JK1167). Research and Cooperation Project for Department of Yulin city (No.2011SKJ07,No.2012cxy3-26,No.2013ky0271). Thanks for the help.

References

[1] B-S. Jeong and E. Omiecinski,Inverted file partitioning schemes in multiple disk systems, IEEE Transactions on Parallel and Distributed Systems, **6**, 142-153 (1995).

[2] D. Hawking, N. Craswell, P. Thistlewaite, and D. Harman, Results and challenges in Web search evaluation,Computer Networks, **31**, 1321-1330 (2006).

[3] A. I. Aggour, F. E. Attounsi, Fuzzy Topological Properties on Fuzzy Function Spaces. Applied Mathematics & Information Sciences Letters, **1**, 1-5 (2013).

[4] M. Lei, J. Y. Wang, B. J. Chen, and X. M. Li. Improved relevance ranking in WebGather. Journal of Computer Science and Technology, 410-417 (2001).

[5] B. Chidlovskii, C. Roncancio, and M.-L. Schneider, Semantic cache mechanism for heterogeneous Web querying, Computer Networks, **31**, 1347-1360 (1999).

[6] P. K. De, Amita Bhincher, Dynamic Programming and Multi Objective Linear Programming approaches, Applied Mathematics & Information Sciences, **5**, 253-263 (2011).

[7] A Hasegawa, F Tappert, Transmission of stationary nonlinear optical pulses in dispersive dielectric fibers. II. Normal dispersion, Appl. Phys. Lett., **23**, 171 (1973).

[8] Craig Silverstein, Monika Hen zinger, Hannes Marais, et al,Analysis of a very large Web search engine query log, In SIGIR Forum, **33**, 6-12 (1998).

[9] Soboroff I, Nicholas C, Cahan P. Ranking retrieval systems without relevance judgments. In: Kraft DH, Croft WB, Harper DJ, Zobel J, Eds. Proc. Of the 24th Annual Intl ACM SIGIR Conf. on Research and Development in information Retrieval (SIGIR2001). New York: ACM Press, 6673 (2001).

[10] P. Mastorocostas, C. Hilas, A Computational Intelligence Forecasting System for Telecommunications Time Series,Engineering Applications of Artificial Intelligence, **25**, 200-206 (2012).

[11] TP Pedersen, Non-interactive and information-theoretic secure verifiable secret sharing, Advances in Cryptology CRYPTO 91, Springer, **576**, 129-140 (1992).

[12] P.Mastorocostas, C. Hilas, S. Dova, D. Varsamis, Forecasting of Telecommunications Time-series via an Orthogonal Least Squares-based Fuzzy Model, Proceedings of 21st IEEE International Conference on Fuzzy Systems, (2012).

[13] MA Ammar, E Elbeltagi, Algorithm for determining controlling path considering resource continuity, Journal of Computing in Civil Engineering, **15**, (2001).

Zhang Yong-Heng received the MS degree in Computer science from Xidian University in 2010. He is currently a professor in Yulin University. His research interests are in the areas of Big data and Cloud integrated.

Zhang Feng received the MS degree in Computer science from Xidian University in 2009. Now he is a PhD of Northwestern Polytechnical University. He is currently a associate professor in Yulin University. His research interests are in the areas of Cloud integrated manufacturing technology, the modeling of complex systems, the Internet of Things applications.

You Fei received the PhD degree in Mathematical science from Beijing Normal University in 2005. He is currently a professor in Yulin University. His research interests are in the areas of Fuzzy Mathematics and Artificial Intelligence.