On Improving Website Connectivity by Using Web-Log Data Streams

Edmond HaoCun Wu[1], Michael KwokPo Ng[1], and Joshua ZheXue Huang[2]

[1] Department of Mathematics, The University of Hong Kong
hcwu@hkusua.hku.hk,mng@maths.hku.hk
[2] E-Business Technology Institute, The University of Hong Kong
jhuang@eti.hku.hk

Abstract. When people visit Websites, they desire to efficiently and exactly access the contents they are interested in without delay. However, due to the constant changes of site contents and user patterns, the access efficiency of Websites cannot be optimized, especially in peak hours. In this paper, we first address the problems of access efficiency in Websites during peak hours and then propose new measures to evaluate access efficiency. An efficient algorithm is introduced to detect user access patterns using Website topology and Web-log stream data. Adopting this method, we can online modify a Website topology so that the new topology can improve the Website connectivity to adapt current visitors' access patterns. A real sports Website is used to evaluate the effectiveness of our proposed method of accelerating user access to related contents. The results of the evaluation presented in this paper suggest that this method is feasible to online improve the connectivity of a Website intelligently.

Keywords. Data Streams, Optimization, User Access Patterns, Website Topology

1 Introduction

Nowadays, more and more people rely on the World Wide Web to acquire knowledge and information by browsing Websites, so how to organize the content and the structure of a Website so that users can easily access and find what they want, has raised the main concern of Web research.

Much of previous work has focused on Web usage mining [2, 5, 7, 8]. Web usage mining is the application of data mining techniques to discover usage patterns from Web-log data, in order to understand and better serve the needs of Web-based applications [8]. In [8], J.Srivastva et. al also propose a three-step Web usage mining process which are called preprocessing, pattern discovery, and pattern analysis. Web-log data, which include the URLs requests, the IP addresses of users and timestamps, provide much of the potential information of user access behavior in a Website. Usually, we need to do some data processing, such as invalid data cleaning and user and session identification. Then, the original Web logs are transferred into user access session datasets for analysis. Many

Y. Lee et al. (Eds.): DASFAA 2004, LNCS 2973, pp. 352–364, 2004.

researchers have proposed different data mining algorithms for mining user access patterns or trends from the user access sessions [6, 7, 9, 12]. For instance, Mobasher et al. [6] used association rules mined to realize effective Web personalization. Shen et al. [9] suggested a three-step algorithm to mine the most interesting Web access associations. Zaiane et al [13] proposed to apply OLAP and data mining techniques for mining access patterns based on a Web usage mining system.

Recently, data-intensive applications in which the data is modeled best not as persistent relations but rather as transient data streams have become widely investigated. Traditional Web-log mining focuses on off-line data mining, however, in practice, Web logs are generated in the form of continuous, rapid data streams and then stored in Web severs. Therefore, Web-log mining based on Web-log data streams is more important in some Web applications, such as on-line monitoring user behavior, on-line performance analysis, detecting traffic problems. However, few researchers investigate how to develop on-line Web usage analytical algorithm that can handle huge volumes of Web-log data streams. In this paper, we investigate the problem of dynamic redesign Website topology to improve user access efficiency and system performance based on Website topology and Web-log data streams.

The rest of the paper is organized as follows: In Section 2, we first introduce some new measures to evaluate access efficiency in a Website. In Section 3, we suggest a novel method of mining access patterns with connectivity problems for Website connectivity enhancement. In Section 4, experimental results are given. Then, we will apply our proposed method into a real case study. Finally, we conclude the paper and give some future remarks.

2 Access Efficiency of Website

2.1 Problem Statement

We first investigate the access efficiency problem in a Website. In practice, large number of users will visit certain Websites in a particular period of time. For example, some immediate information-based Websites, such as stock Websites and sports Websites, will attract the attentions of many people when some important or smashing events have happened. However, excessive Web pages requested will make Web servers ineffectively. As a result, users may suffer from the low Website connective speed and even cannot access the Website.

On the other hand, we observe that the Website linkage design will also affect the Website access efficiency. Usually, many users will spend a lot of time on searching the contents they are interested in. We remark that the unnecessary pages requested can be reduced if the proper navigation information is provided in the Website. Therefore, from the system point of view, decreasing the redundant pages requested at the peak hours will be of great help to improve the system performance. As to the visitors, they can access quickly the contents they are really interested in. To sum up, how to track such changes and improve

the Website access efficiency become an interesting research problem. In such cases, the objective of a Website should be able to guide users to point to the pages they want to access in as few clicks as possible.

2.2 Motivated by Website Topology

Website topology is the structure of a Website. The nodes in a Website topology represent the Web pages with URL addresses and the edges among nodes represent the hyperlinks among Web pages. Mathematically, a Website topology can be regarded as a graph. We assume that there is at least a path to connect every node, that is, every Web page in a Website can be visited through at least one path. Figure 1 shows an example of a Website topology. All the Web pages are assigned with unique labels. A Website topology contains linkage information among the Web pages. The hyperlinks establish an informative connection between two Web information resources. The original design of a Website topology represents the expectation of access patterns according to a Web designer. However, it may not be true to visitors. Hence, a Website topology combining with Web usage mining techniques can help us to understand the visitors' behavior.

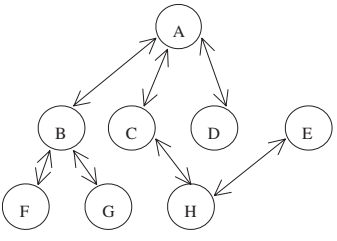

Fig. 1. Website Topology

2.3 Access Efficiency of User Sessions

From the above analysis, we need to find new measurements to evaluate the Website access situation under a Website topology. For a given Website topology, visitors must follow certain traversal paths to access the Web pages that they are interested in. For instance, if a visitor wants to sequentially visit Web page {A, F, E} (See Fig 1), the shortest traversal path is {A, B, F, B, A, C, H, E}. The corresponding access sequence is S = {AB, BF, FB, BA, AC, CH, HE}. Thus, the visitor should click at least seven times to access the target pages {A, F, E}. A access P1 P2 is defined as the access from page P1 to P2. If a visitor wants to browse the same target pages in a different order {A, E, F}, another traversal path {A, C, H, E, H, C, A, B, F} with eight clicks is needed, the corresponding accesses are {AC, CH, HE, EH, HC, CA, AB, BF}. But if one want to access other 3 pages {A, B, G}, just two accesses are enough. We observe that the access efficiency of {A, F, E} is low due to the redundancy of accesses. In general, there

are two types of access redundancy. One is the jump-track access. For example, starting from A, the target page is F, we must access B first before we access F. The other type of redundancy access is the backtrack access, e.g., in order to access C from B, a visitor must click the back button in the browser to go from B to A and then to C, even though A has been accessed previously. The difference between the jump-track access and backtrack access is determined by looking into whether the access has been performed in a visitor access session, e.g., in accessing {A, F, E}, AB is a jump-track access while BA is backtrack access. In this paper, we call both of them non-target accesses, comparing with target accesses which acquire the target Web pages directly.

In order to measure the access efficiency of a visitor access session for a Website topology, we define the User Access Efficiency (UAE) as follows:

Definition 1. *Given a user access session* $S = \{s_1, s_2, s_m\}$, $A = \{a_1, a_2, a_i\}$ *is the jump-track access session,* $B = \{b_1, b_2, b_j\}$ *is the backtrack access session, where* $A \subset S$ *and* $B \subset S$, $A \cap B = \Phi$. *The User Access Efficiency (UAE) is given*

$$UAE(S) = 1 - \frac{|A| + |B|}{|S|} \tag{1}$$

where $|S|$ *is the length of session* S.

If Web pages are cached in the Web browser, the Web server will not record the backtrack accesses in the Web-log. For example, a visitor follows a traversal path A, C, H, E, H, C, A, B, F, the Web-log will only contain A, C, H, E, B, F, which is not a complete traversal path. Therefore, we propose a new measure called Server Access Efficiency to evaluate the access efficiency from the Web server side of view.

Definition 2. *Given a user access session* $S = \{s_1, s_2, s_m\}$, $A = \{a_1, a_2, a_i\}$ *is the jump-track access session,* $B = \{b_1, b_2, b_j\}$ *is the backtrack access session, where* $A \subset S$ *and* $B \subset S$, $A \cap B = \Phi$. *The Server Access Efficiency (SAE) is given*

$$SAE(S) = 1 - \frac{|A|}{|S| - |B|} \tag{2}$$

where $|S|$ *is the length of session* S.

Example 1: Given target Web pages T= {A, E, F}, eight accesses S={AC, CH,HE, EH, HC, CA,AB, BF}are needed in the given Website topology. AC, CH, AB are jump-track accesses and EH, HC, CA are backtrack accesses. The remaining two accesses HE, BF are target accesses. Hence, $UAE(S) = 1 - (3 + 3)/8 = 25\%$, $SAE(S) = 1 - 3/(8 - 3) = 40\%$. We can also calculate the access efficiency of a group of access sessions. Given a access session database D, the UAE and SAE of D are calculated as follows:

$$UAE(D) = \frac{1}{|D|}\sum_{i=1}^{|D|} UAE(S_i) \quad SAE(D) = \frac{1}{|D|}\sum_{i=1}^{|D|} SAE(S_i) \tag{3}$$

We notice that UAE is always lower than SAE, it can be explained that UAE will consider the effect of backtrack clicks to the access efficiency, while SAE will not count. We note that both UAE and SAE measures have practical meanings. If the UAE is low, it means that a user is hard or slow to access Web pages they wanted. If the SAE is low, it suggests that the Web server return too many irrelevant or uninteresting Web pages to the end users. If given access session S is a frequent requested patterns, then it will cause the inconvenience of a large portion of visitors. What's more, if this happens at the peak hour of a day, much more requests will be sent to the Web server, the overload of Web server will significantly slow down the users accesses to the Website. The situation will result in the loss of visitors. Hence, it is particularly important to improve the User Access Efficiency and Server Access Efficiency according to the concerns of improving visitors' satisfaction or maintaining the robustness of Web servers.

We remark that if a Website topology is a complete graph, then we always have the highest access efficiency. However, such topology is implausible for large websites. In our design, we would like to construct a new topology that can achieve better access efficiency than the original one without increasing the number of links among Web pages. The new topology is more adaptive to the user access patterns.

In the calculation of UAE and SAE, we assume that there are some target pages in each session. The assumption is compatible with the prevailing hierarchical structure of a Web site. Usually, we can identify most of the target pages based on the visiting time and categories of Web pages. With more information about user visiting habit or content information, we can get higher accuracy of identifying target pages.

3 Mining Access Patterns from Web-Log Data Streams

3.1 Topology Probability Model

In our previous work [11], we propose a topology probability model to measure the probability among Web pages in a Website topology. Let us first consider association probability between any two Web pages x_i and x_j in a Website. Given a Website topology G containing n Web pages $X = \{x_1, x_2, \ldots, x_n\}$, we assume the number of hyperlinks of x_k is h_k, $k = 1, \ldots, n$, so a set of the Web pages which have hyperlinks to x_k are $X_k = x'_1, \ldots, x'_{h_k}$. When a user has visited current Web page x_i, he or she may continue to visit other Web pages connected to current Web page or just exit the Website. Therefore, there are $h_i + 1$ choices for the user to select after visiting x_i. We consider the probability of visiting x'_j after having visited x_i is given by: $P(x'_j|x_i) = w_{i,j}/h_{i+1}$, where $w_{i,j}$ is the weighting parameter between x_i and x'_j, $j = 1, \ldots, h_i$ (usually we take $w_{i,j} = 1$). We should note that x'_j must be connected to x_i, that is $x'_j \in X_i$. The distance measure between two Web pages x_i and x'_j is defined by $D(x_i, x'_j) = \log(1/P(x'_j|x_i)) = \log(h_{i+1}/w_{ij})$.

If x_i and x_j do not have a hyperlink between them, we consider the shortest path between x_i and x_j in the Website topology. Since the Website topology is

a connected graph, there must be a sequence of nodes connecting x_i and x_j. We can employ classical Floyd algorithm to calculate the shortest paths between any two nodes in a Website topology. Assume the shortest path passes through m Web pages $x_1^*, \ldots, x_m^* \in X$, the length of the shortest path is calculated by $D(x_i, x_j) = D(X_i, x_1^*) + \sum_{k=1}^{m-1} D(x_k^*, x_{k+1}^*) + D(x_m^*, x_j)$. Therefore, the probability of visiting x_j after having visited x_i is given by:

$$P(x_j|x_i) = P(x_1^*|x_i) \prod_{k=1}^{m-1} P(x_{k+1}^*|x_k^*) P(x_j|x_m^*) \ . \tag{4}$$

Using the topology probability model, we can add a new page attribute $P_{ij} = P(V_j|V_i)$ to indicate the transitive probability from page V_i to V_j.

3.2 Counting Page Sequential Accesses

Since we want to investigate the users navigational behaviors under certain Website topology, the user access sessions reconstructed are the users traversal paths. A user access session is equivalent to a traversal path in this paper. As we have mentioned, if Web pages are cached in the Web browser, the Web server will not record the backtrack accesses in the Web server logs. However, we can recover backtrack accesses by checking the Website topology.

Then, we present a data structure of aggregating access sessions as follows: Given a Website topology $G = \{V, E\}$, where $V = \{V1, V2, ..., Vn\}$ represents the page set and $E = \{E1, E2, ..., Em\}$ represents the link set. Also, given a user access session dataset $D = \{S1, S2, ..., Sn\}$, for each access session $S \in D$, $S = \{P1, P2, ..., Pr\}$, where $Pi \in V$, $i = 1, ..., r$. We set $C_{ij} = Count(Pj|Pi)$ to indicate the number of accesses from Pi to Pj in the access session dataset D. We notice that the access session is the traversal path that a user follows in a Website topology, any two adjacent pages in the session must be accessible by one click.

Therefore, there are two cases of such counting, one is that Pi and Pj are adjacent in session S which also means the link of $Pi, Pj \in E$. The other case is that Pi and Pj are not adjacent which means there exist at least a page between Pi and Pj. For the first case, we will count it whenever it appears. As to the second case, we will only count once in the access session.

Example 2: Given access session S={A, B, C, B}, in the first case, we will count AB, BC, and CB once since they are adjacent in the session, respectively. As to the second case, we count AC once. But we won't count AB again since it has been counted once in the first case. So, $Count(B|A) = 1$, $Count(C|B) = 1$, $Count(B|C) = 1$, $Count(C|A) = 1$.

Using the page frequency counting model, we can add another page attribute $C_{ij} = Count(Vj|Vi)$ to indicate the access frequency from page V_i to V_j.

3.3 Criteria for Access Pattern Discovery

In practice, Website managers are eager to know which contents can attract the visitors. It is natural that visitors will spend more time on the Web pages they

interested in. Hence, the temporal factor should also be taken into consideration. Since the Web log also contains the access timestamps, so we can record the time period spent on each page. Thus, we can use page access frequency, page transitive probability, and page staying time as the major factors in a new index to identify really interesting access patterns. For this purpose, we suggest another interestingness measure named Access Interest (AI) as below:

Definition 3. *Given page transitive frequency matrix C and topology probability matrix P, page staying time T, the access interest matrix is given by:*

$$AI(i,j) = In(\frac{C_{ij}T_{ij}}{P_{ij}} + 1) \tag{5}$$

where C_{ij} is the number of accesses from Web page V_i to V_j, P_{ij} is the topological probability from V_i to V_j and T_{ij} is the total staying time of visiting V_i and V_j. Here, we define $T_{ij} = T_i + T_j$, where T_i and T_j are the average staying time on page V_i and V_j respectively.

The main contribution of the Access Interest is that the index is capable of identifying frequent but not efficient access patterns. Access Interest contains three variables C, T and P. We can regard C as page sequential access frequence. If some pages are often visited by users, then the sequential accesses among these pages are the frequent user access patterns. similarly, T, denoting the staying time on pages, can be used to measure the importance of access patterns. The consideration is based on our observation that users will spend more time on the Web pages interested in. In formula 5, $C \times T$ can represent the total time spent on a particular access pattern during a time-period. As to P, this factor is used to measure the 'efforts' in searching the pages related to a given access pattern. From the calculation of AI, we can see that if a access pattern is frequently accessed by users but also it is hard to access for users, then the access pattern will have high AI value. Based on our empirical study, we take a logarithm transformation in the fomula to constrain the value range. The AI is a non-negative index. A access pattern with higher AI value has more necessity to improve its access efficiency.

For example, in Fig 1, we need to click at least five times to access {F,E} sequentialy. However, only one click is enough to access {F,B}. From the calculation of P, we can see that $P(F,E)$ is much smaller than $P(F,B)$. If {F,E} and {F,B} are both frequent access patterns measured by T and C, then P is useful in finding that the access pattern {F,E} is not efficient enough. Using AI, we can automatically detect such patterns with potential connectivity problem and then use them to improve the Website topology.

3.4 Dynamic Website Connectivity Enhancement

In this section, we will present a Website reconstruction method on improving Website connectivity by Web-log data streams. Recall that in section 2.3, we introduce two access efficiency measures UAE and SAE to measure the goodness

of a Website. In formula 3, we can see that UAE and SAE can also be used to measure the access efficiency of a group of access patterns. Hence, the most important benefit of these two measures is that they can on-line monitor the system performance based on the current access patterns and Website topology. In practice, we can adopt UAE and SAE as interactive connectivity performance indexes in our Website connectivity enhancement method.

The process of on-line monitoring Website connectivity as follows: First, the current Web-log data streams will be turned into access sessions by our cube model purposed[12]. Then, based on the current topology model and session information, we can calculate the access efficiency of current user access patterns. If the current system access efficiency(UAE or SAE)is lower than a pre-set threshold (e.g.,60%), the monitoring system can give a signal to warn that the current Website connectivity is not efficient.

By getting this information, we can perform Website connectivity analysis process. First, calculate the AI values of current access patterns, then select access patterns with high AI values for connectivity enhancement(e.g.,build new hyperlinks or merge related pages). After modifying the Website topology, recompute the access efficiency of access patterns, if the system access efficiency increases, it indicates the the Website connectivity has been improved.

From above analysis, we can see that the interaction and utilization of UAE, SAE and AI are the core of our Website connectivity enhancement method. UAE and SAE are used to monitor the Website connectivity on the whole. AI is the key function to detect the exceptional access patterns, providing the guideline to modify the Website topology in order to improve the efficiency of Web browsing activities.

We summarize the key steps in the Website connectivity enhancement method as follows:

Begin

1. Input Website topology G=(V, E) and Web-log data streams.
2. Convert clickstream into access sessions by using proposed cube model and and compute topology probability matrix.
3. Collect session information and on-line report Website access efficiency. If Access efficiency(UAE and SAE) satisfies minimal pre-set threshold, then keep on monitoring. Otherwise, go to Step 4.
4. Discover Website connectivity problems, send out warning message, start on-line Website connectivity enhancement process.
5. Compute AI values of access patterns, then mine user access patterns with high Access Interest, evaluate the priorities of access patterns for connectivity optimization.
6. Input top k access patterns most urgent to improve access efficiency.
7. For each pattern $PiQi$, $i = 1, ..., k$, if there is no existing hyperlink between Pi and Qi, build direct hyperlink from Pi to Qi to connect the sequential access pattern, where k is number of patterns for optimization. If $PiQi$ and $QiPi$ both exist, merge two pages as an alternative solution.

8. Output the optimized Website topology, recompute access efficiency UAE and SAE. If current Website satisfies minimal access efficiency requirement, then go to step 3, otherwise go to step 4. Restore optimization results for further Web usage analysis.

End

4 Experiments

In this section, we used the Web usage data from ESPNSTAR.com.cn, a famous sports Website in China, to test and validate the performance and effectiveness of our Website connectivity enhancement method proposed. All experiments reported were performed on a PC with a Pentium 4 CPU of 2.0 GB and 256 MB main memory.

4.1 Datasets

In our experiments, we take the data stream as a sequence of data items $X = x_1, ...x_n$, where the sequence is scanned only once in the increasing order of the indexes. Each data stream contains a set of continuous access sessions during some period of time. We consider the on-line Web-log data arriving when proceeding session identification. Using our cube model, we can use the access session streams instead of original Web-log streams to analyze Website access efficiency.

With permission, we got the Website topology and related user information for analysis. We use two months Web log data to do the experiments. The original Web logs contain millions of continuous access records. After data preprocessing, we got the user access session datasets needed for optimization. In the datasets, we take all the continuous sessions during a period of time (e.g., one day) as the input data streams.

We also select the sessions from particular users for analysis in some datasets. Table 1 lists some of the datasets for experiments. ES1, ES2 and ES3 are the access session datasets from the logs during Dec, 2002 and ES4 and ES5 are the logs from April, 2003.

Table 1. Characteristics of real datasets

Dataset	No.Accesses	No.Sessions	No.Visitors	No.Pages
ES1	583,386	54,300	2,000	790
ES2	2,534,282	198,230	42,473	1,320
ES3	6,260,840	517,360	50,374	1,450
ES4	78,236	5,000	120	236
ES5	7,691,105	669,110	51,158	1,609

4.2 Performance Analysis

We first evaluate the efficiency of aggregating access sessions for on-line monitoring access efficiency. In this experiment, we used dataset ES3 to test the running time when increasing the input access sessions. From the result (See Fig 2), we can see that the increase of running time exhibits a linear relationship with the increase of input access sessions. This experiment demonstrated that our cube model is robust enough to handle massive volumes of Web-log data streams.

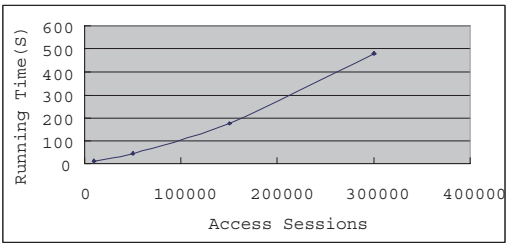

Fig. 2. Increasing Number of Access Sessions

The second experiment is to test the scalability when increasing the number of access patterns for Website connectivity enhancement. We used ES1 to do this experiment. The results show that the computational cost of AI is quite low, thus AI index is feasible to apply into practical application for on-line Web usage analysis(See Fig 3).

The last experiment using the largest dataset ES5 to test the scalability when increasing the number of pages involved in the Website connectivity enhancement. Theoretically, the complexity of computing topology probability matrix and access frequency matrix is $O(N^2)$. However, through our experiment results, even for a Website with more than one thousand and five hundred Web pages, the running time is still acceptable(See Fig 4).

Above performance experiments validated that our proposed method for improving Website connectivity is feasible. Its scalability and stability ensure that this method can be adopted in practical applications.

5 A Real Case Application

We investigated the effectiveness of our method in increasing Website connectivity through a real case study. Fig 5 and Fig 6 show the number of visitors and pages requested from a sports Website ESPNSTAR.com at each hour on April 1, 2003. We can see that the number of visitors and pages requested on each hour is not even. In peak hours, the pages requested can be five times of off-peak hours. What's more, we noticed that during the ongoing period of some important matches, such as World Cup, thousands of visitors will focus on browsing

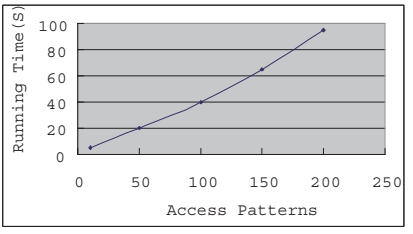

Fig. 3. Increasing Number of Access Patterns

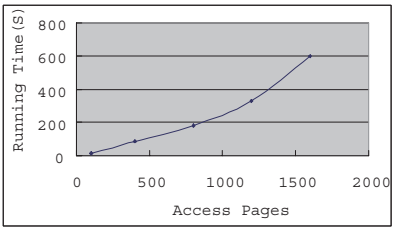

Fig. 4. Increasing Number of Access Pages

certain pages interested in around the same time. It causes the traffic problems of the Web servers. Hence, some real time connectivity optimization solution is needed to improve the Website access efficiency during the peak hours.

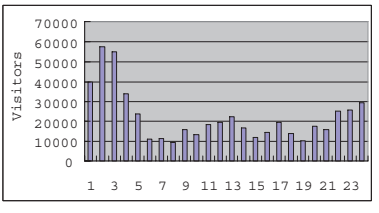

Fig. 5. Distribution of Visitors

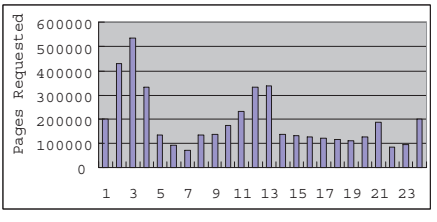

Fig. 6. Distribution of Pages Requested

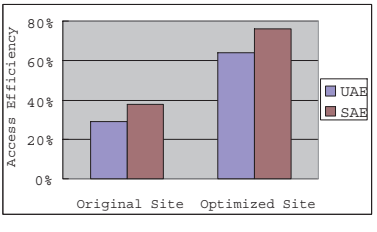

Fig. 7. Comparison of Access Efficiency

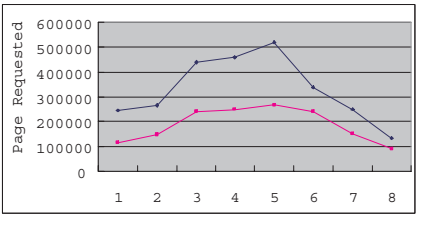

Fig. 8. Comparison of Pages Sent by Servers

We used our proposed method to on-line discover user patterns with low access efficiency. We employed the same data streams to compare the performance. Through our analysis,during the peak hours of that day, many visitors focused on one European football match. However, the original Website topology was not so efficient for the visitors to browse related pages. After discovering such

patterns, we optimized correlative pages and content so that visitors can easily access the on-line Website contents. In Fig 7, we can see that the optimal Website topology has better access efficiency than the original one.

We also discovered that football fans tend to check the ranking of their favorable teams in the scoring board frequently. However, the location of the scoring board is not obvious in the original design of Website. Detecting this access pattern by our method, we put the scoring board in the homepage. As results, more and more people visited the Web content and the access efficiency of the access patterns has also greatly improved.

In Fig 8, we further compare the total pages sent by Web server. Through this simulation, we can see that the workload of Web server at peak hours has been reduced(in Fig 8, one unit equals 15 minutes, the lower curve represents the flow of pages sent by using Website connectivity enhencement while the upper curve represents the original pages sent by servers). Meanwhile, the visitors can still access the same contents with faster connective speed.

The interesting simulation confirms the usefulness of improving access efficiency by using UAE, SAE and AI indexes. We also noticed that if not in peak hours, the effect of improving access efficiency is not so obvious. It can be explained that during normal days or hours, visitors follow their individual access patterns. The variety of access patterns makes it not so effective to construct an easily accessible Website topology to meet the needs of all the people. In these cases, apply this optimization technique into Website personalization is more suitable.

6 Conclusion

In this paper, we first investigate new measures for evaluating access efficiency in Website. Then, we propose an efficient method for aggregating access sessions and mining access patterns from Web-log data streams. The core of the method focuses on improving the Website topology based on current user access patterns. The experiments show that our purposed method is effective and efficient to handle large-scale Web-log data streams. We can use it to monitor Website access activities on-line to enable performance monitoring, load-balancing. In the future, we intend to extend the Website optimization model and implement it in the CRM component of a business intelligence platform.

References

1. Bettina Berendt, Bamshad Mobasher, Miki Nakagawa, and Myra Spiliopoulou, *The Impact of Site Structure and User Environment on Session Reconstruction in Web Usage Analysis*, Proceeding of the WEBKDD 2002 Workshop, Edmonton, Canada, 2002.
2. Robert Cooley, Bamshad Mobasher, Jaideep Srivastava(1999). *Data preparation for mining World Wide Web browsing patterns.*

3. Robert Cooley, Pang-Ning Tan, and Jaideep Srivastava. *Websift: The web site information filter system.* In Proceedings of the Web Usage Analysis and User Profiling Workshop, 1999.
4. Ron. Kohavi, *Mining E-Commerce Data: The Good, the Bad, and the Ugly.* Invited talk at KDD 2001 industrial track, San Francisco, California, USA, 2001.
5. Bamshad Mobasher, Honghua Dai, Tao Luo, Miki Nakagawa, Yuqing Sun, Jim Wiltshire (2000). *Discovery of aggregate usage profiles for Web personalization.*
6. B. Mobasher, H. Dai, T. Luo, Y. Sun, J. Zhu,*Combining Web Usage and Personalization*, Proc. the Technologies (2000)
7. B. Mobasher, N. Jain, E. Han, J. Srivastava. *Web mining: pattern Transactions*, Tech. Rep. 96-050,
8. J. Srivastava, R. Cooley, M. Deshpande, and P. N. Tan. *Web Usage Mining: Discovery and applications of usage patterns from web data*, SIGKDD Explorations, 1:12-23, 2000.
9. L. Shen, L. Cheng, J.Ford, F.Makedon, V. Megalooi-konomou, T. Steinberg,*Mining the most interesting web access associations*, Proc. the 5th International Conference on Knowledge Discovery and Data Mining (KDD'99) (1999) pp.145-154
10. Ramakrishnan Srikant, Yinghui Yang, *Mining web logs to improve website organization.* World Wide Web Conference 2001: 430-437.
11. Edmond H. Wu, Michael, K. Ng, *A Graph-based Optimization Algorithm for Website Topology Using Interesting Association Rules*, Proc. the Seventh Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'03) , 2003.
12. Q. Yang, J. Huang and M. Ng, *A data cube model for prediction-based Web prefetching*, Journal of Intelligent Information Systems, 20:11-30, 2003.
13. Osmar R. Zaiane, Man Xin, Jiawei Han, *Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs*, in Proc. ADL'98 (Advances in Digital Libraries), Santa Barbara, April 1998.