Proxy Server Systems Improvement

Using Frequent Itemset Pattern-Based Techniques

Saranyoo Butkote*, Jiratta Phuboon-op, Phatthanaphong Chomphuwiset

Information Technology Department, Faculty of Informatics,
Mahasarakham University Thailand 44150
E-mail: Saranyoo.b@msu.ac.th

Abstract

In the Internet, proxy servers are tools that are used to facilitate clients for accessing websites and save network bandwidth. Basically, proxy servers use cache to store visited sites in order to deliver faster web accesses to users, without requesting to external target servers. With the limitation of cache size, however, a number of techniques have been proposed to potentially increase the performance of cache replacement methods. The objective of this study is to apply a data mining technique to generate frequent itemset patterns of the internet access by users. Data transactions are generated based on the allocated time slots of internet usage. As the public Internet cluster, different users can use the same single computer; therefore, generating data transaction base on computer IP address is used by many users and may not extract the behavior of users. Association rules are generated using FP-growth algorithm. The frequent itemsets (rules) are then used to implement and extension of LRU cache-replacement technique. The experimental results show that the proposed technique provides promising result and is superior to the base-line cache replacement techniques (i.e. FIFO and LRU).

Keywords: FP-growth, Cache Replacement, Association.

1. Introduction

With the rapid growth of the Internet, World Wide Web in the recent era is popular medium to share and exchange the information. Even though the speed of network has been improved substantially in recent years, increasing popularity of using the Internet, the Request/Response working mode of network still makes the internet traffic very slow. These might result to extreme congestion on the network and load on the servers, all resulting give no

guarantee on the Quality of Service (QOS) at the user end [1]. Because the web server cannot know the users' demand and the users' requests cannot be predicted. One of possible solution is to increase network infrastructure but that will not only increase the economic cost but will also increase the demands of more network applications. So, the solution lies in efficient use of existing network resources.

Proxy Servers are popular tools which are used to facilitate clients for accessing websites. Proxy servers help in improving response latency and reduce network traffic by storing copies of web objects accessed in local temporary memory storage area (also called cache) and providing to other users the same page on demand. Due to the limitation of the storage space of cache, some web objects (pages) are needed to be replaced by different web objects in order to facilitate the faster accessing of the Internet. To replace web objects in cache a number of algorithms have been proposed such as First in First Out (FIFO), Least Recently Used (LRU) and etc. The fundamental strategy of implementing cache replacement techniques is to remove a stored websites (or web objects) and replacing them by newly requested websites. As a result, the efficiency is removing from caches because the deleted websites might be requested again. In the literatures, numerous works have been studied in attempting to improve the Web caching performance. Soonthornsutee R. et al. proposes a cache replacement technique based on Data Mining approaches to generate the model of cache replacement policy [2]. The work predicts the life time of web objects in caches. The replacement of a new web object is performed based on the predicted life time of objects in the caches. In addition to predicting-based technique, Haung Y. et al. proposes a frequent-itemset discovery method to generate an associate rules to replace web objects in caches [3-5]. The work examines data instances based on the IP of computers.

Associate rules are generated, using Apriori-based algorithms, before implementing them to cache replacement mechanism. Bonchi F. al. proposes a technique that extent Least Recently Used (LRU) to increase the hit rates of cache replacement by comparing two data mining techniques, i.e. association rule and decision tree [6]. They generate data transaction base on the access of the Internet of a single computer, considering by the Internet IP. The results of their show that decision trees outperform LRU. Clustering-based techniques can also be applied for increasing the performance of cache replacement mechanism. Poornalatha G.[7] invent a clustering-based technique that separate data transaction into different groups, association repository [8]. The, pre-fetching web pages are loaded into cached based of the repository information. In addition to cache replacement using data mining techniques, data cleaning can be a potential step to increase the performance of cache replacement [9]. The few research in data mining have integrated caching and web log mining. Most previous work on integrating prediction models, applied to predict the future web access, including our work has focused on behavior of users. It remains to show how prediction models can be used to extend the best prediction models.

In this paper, we apply a data mining technique to generate frequent itemset patterns of the internet access by users. We generate data transactions based on the allocated time slots of internet usage. As the public the Internet cluster, different users can use the same single computer; therefore, generating data transaction base on computer IP may not extract the behavior of users. Association rules are generated using FP-growth algorithm [10]. The rules (frequent itemset patterns) are then used to implement and extension of LRU cache-replacement techniques. This paper is organized as follows: Section 2 explains the proposed method for improving cache replacement approach, which is separated into (i) data collection, (ii) data cleaning, (iii) data transformation and data transaction, and (iv) rule generation. Section 3 demonstrated the experiments and results before the conclusion is given in Section 4.

2. Materials and Methods

There are 5 process steps in this study to generate the cache replacement using data mining techniques, as shown in Figure 1. Data collection is the first step to be

performed before data cleaning is carried out. The output of data cleaning is then processed to perform data indexing. Data transaction generates transactions, which are used to generate associate rules by applying FP-growth algorithms. The next section provides intensive details of each process to generate the propose cache replacement technique.

Fig. 1. The overall process of the proposed cache replacement technique.

2.1 Data Collection

The data set was collected from the cache of the College of Politics and Governance, Mahasarakham University using a Squid proxy Server [11]. The data collection period was from November 2013 to April 2014. Users of this cache include graduate and undergraduate students. The data set contains one million records. Each record contains 10 attributes.

Table 1. The attributes of the collected data records

Attributes	Example
Time Stamp	102399377.615
Duration	1
Client Address	10.200.74.126
Result Code	TCP_HIT/200
Byte	493
Request Method	GET
URL	http://www.sanook.com
Rfc931	NONE
Hierarchy Code	-
Type	page

2.2 Data Cleaning

The collected data is not able to directly process to generate rules (which are used in the cache replacement process). Data Cleaning eliminates inconsistent or inessential items in the data. Consequently, this data is substantially required to remove some unrelated data. From an initial set of 10, there are only 3 attributes selected. The

attributes include Time Stamp, URL, Result Code. Cleaning data can be a sensitive method for generating frequent itemset patterns, as they can eradicate some amount significant information that is useful for generating associate rules. In this paper, we derive some respects of data cleaning as follows:

- non-response URLs : if a data records receiving to 200 code. It will be retained in the data; remove otherwise.
- image files and banners: all image files are exclude from the data records, such as *.jpg, *.png and *.gif.
- dynamic pages are the page executing at server side, therefore they are cleaned from the data record.
- flash files(*.swf) are eradicated from the data records
- document files such as *.ppt and *.doc are remove from the data recodes.

2.3 Data Transformation and Transactions

Before generating associate rules, cleaned data is processed to generate data records. Based on the data collection and source, this paper intends to observe the behaviors of users. Therefore, data transactions are extracted from the cleaned data by dividing the data into different time intervals or time slots. Each user is allowed to use the Internet for 2 hours. Therefore, time slots are divided into 6 periods ($t_1, t_2, \dots, t_6$), $t_1 = 08.00-10.00$ am, $t_2 = 10.01-12.00$ am, $t_3 = 01.00-03.00$ pm, $t_4 = 03.01-05.00$ pm, $t_5 = 05.01-07.00$ pm, $t_6 = 07.01-09.00$ pm. By this, the time stamps of each record are checked to generated transactions.

2.4 Rule Generation

After generating transactions, the transactions are then used to build association rules. This work applies FP-growth algorithm to generate rules. The algorithm can be divided into steps, i.e. (i) FP-tree construction and (ii) mining frequent patterns [4]. Let L denotes the list of frequent items and F denotes the set of frequent items and their supports. In FP-tree construction step, frequent item F is first collected and their supports. A root of an FP-tree is then created and set as "null". For each transaction, select and sort the frequent items and their supports according to the order of L . Let the sorted frequent item list in $Trans$ be P and let p is the first element and P' is the remaining list. Then, each node will be inserted to the tree. If T has a child N such $N = p$, then increase the count of N by 1; otherwise, create a new node N , and links its parent node to T . The item in P will be recursively inserted to T until P' is empty.

After generating FP-Tree a complete set of frequent patterns can be constructed. The preliminary requirement of generating frequent pattern is to initial a minimum support (denoted as Ω). The complete set of frequent patterns can be generated as follows:

- (1.) If T contain a single path then for each combination (denoted as β) of the node in the path do following:
 - 1.1 Generate pattern $\beta \cup \alpha$ with support = Ω of nodes in β .
- (2.) Else for each a_i in the head of T , do followings:
 - 2.1 generate pattern $\beta = a_i \cup \alpha$ with support = $\Omega(a_i)$
 - 2.2 construct β 's conditional pattern base and then β 's conditional FP-tree Tree β
 - 2.3 IF Tree $\beta \neq \emptyset$ then go to (1)

2.5 Cache Replacement

Cache replacement is considered as one of issue that can improve the performance of accessing to the Internet in organizations. This section demonstrates in the implementation of the proposed cache replacement technique and the cache replacement algorithms used in this study (i.e. LRU and FIFO). Figure 2. depicts the overview of LRU cache replacement technique. Users send a request to a cache server. The server examines the requested URL; if it resides in the cache (hits), return the requested sites (stored in server) to the users; otherwise, the requested sites will be stored in the cache by replacing the least recently used sites in the cache.

Fig. 2. The diagram of cache replacement using LRU technique.

FIFO is one of basic cache replacement technique that has been applied in cache sever. FIFO is a simple cache

replacement technique. The technique replace the stored site in first-in-first-out fashion. The overview of FIFO cache replacement is shown in Figure 3. A user sends a URL to a cache server. The server examines the requested URL; if it resides in the cache (hit), return the requested sites (stored in server) to the users; otherwise, the requested sites will be stored in the cache by replacing the sites in the cache based on first-in-first-out scheme.

Fig. 3. The diagram of cache replacement using FIFO technique.

The proposed cache replacement method is illustrated in Figure 4. The technique incorporates a frequent itemset patterns (explained in Section 2.4) to replace related URLs in cache server in order to increase hit rates of the system. A user requests a URL and send the request to a cache server. The cache server checks the requested URL with both frequent itemset patterns and cache, which can be explain as following steps:

- Step 1: the requested URL is check with the cache sever; if the URL is resided in the cache, the server responds the corresponding stored-sites to the users; otherwise go to step2.
- Step 2: the requested URL is examined with the frequent itemset patterns (explained in Section 2.4) The frequent itemset patterns extract the patterns of accessing URLs of users, for example, $A \rightarrow B$ signifies the frequent of accessing A and B – where A is the source URL and B is the destination URL. Therefore, if the requested URL is a source URL then load its destination URL and insert it to the cache server using LRU-based technique.
- Step 3: if the requested URL is not in the frequent

pattern, insert the to the cache server using LRU-based technique.

3. Experiment and Results

Cache replacement is one of issues that can improve the performance of accessing to the Internet in organizations. Designing proper cache replacement technique can potentially improve the overall network system in organization. This paper proposes a cache replacement technique by applying frequent itemset patterns (FIP). To evaluate the proposed technique, this section demonstrates the experiment of implementing the proposed techniques (as discussed in Section 2) comparing 2 based-line techniques of cache replacement, i.e. LRU and FIFO. This section I organized as follows: Section 3.1 explains the data setting and environments that are used to evaluate the proposed technique. Section 3.2 limns the evaluation and results before the discussion is presented in Section 3.3.

Fig. 4. The diagram of cache replacement using Data mining-based and LRU technique

3.1 Data

We collected 1,432,918 records of data using Squid proxy Server. Data cleaning is firstly performed to eradicate some unrelated data (as discussed in Section 2.2). After cleaning we obtain 219,706 data records which would be used to generate frequent itemset discovery, so called training data. After generating frequent itemset patterns, the patterns are used to perform the cache replacement algorithms. In addition to the training data, we collected 647,145 records within 2 months for a testing dataset. The 3

replacement techniques were implanted – as explained in Section 2.5.

3.1 Evaluation and Results

To evaluate the propose method, we performed and evaluated the technique by comparing the performance of the propose method with the two base-line cache replacement algorithm, which are (i) FIFO and (ii) LRU. Before evaluating, the initial cache space (size) was need to initiate. We determined the initial size of caches from the total testing data. We experimented the size of initial caches with 5 different sizes (determined as the percentage of the testing data), i.e. 5%, 10%, 15%, 20%, and 25% of the total of the testing data [7]. Table demonstrates that results of the experiment by varying the size of a cache.

Table 2. The comparison result (hit rates) among three replacement algorithms: FIP+LRU, LRU and FIFO.

Cache Replacement	% cache space/ testing data				
	5	10	15	20	25
FIP+ LRU	64.67	69.50	67.57	65.77	60.86
LRU	54.70	59.59	57.67	55.90	51.00
FIFO	43.65	46.47	44.85	43.37	38.40

The results presented in Table 2 shows that using frequent itemset patterns is superior to the base-line techniques. The proposed technique achieves 69.50 of hit rates. Figure 5 show the graphical comparison of the three techniques when varying the size of the cache.

Fig. 5. The graphical comparison of the three techniques when varying the size of the cache.

In addition to demonstrating the overall performance of cache replacement algorithms, this study raises the issue of data cleaning influencing the performance of cache servers. Therefore, 2 cleaning patterns was performed, i.e. (i) image-cleaned and (ii) non-image-cleaned. Image-cleaned excludes image URLs from the dataset (see Section 2.2 for details). Non-image-cleaned includes that image files to the dataset. Table 3. shows the results of data cleaning with respect to the image URL issue.

Table 3. The comparison result (hit rates) of cleaning strategy: image-cleaned and non-image cleaned.

Cleaning Strategy	% cache space/ testing data				
	5	10	15	20	25
Image-cleaned	64.67	69.50	67.57	65.77	60.86
Non-image -cleaned	51.14	57.36	50.55	49.04	48.72

The results shown in Table 3. Indicates that image-cleaned strategy (applied during cleaning process) produces higher hit rates that non-image-cleaned strategy.

3.2 Discussion

This paper presents data mining techniques for improving proxy server based on the relation of web access. Section 3.1 shows the results experiments that evaluate the proposed techniques. Table 2. indicates that applying frequent itemset patterns and LRU is better than using the cache replacement FIFO and LRU alone. The generated patterns (rules) captures the frequent patterns of users that access to websites. Therefore these patterns are useful for cache servers as they can simply predict upcoming requests that users may acquire.

Table 3. summaries the performance of cache replacement using FIP+LRU when applying different data cleaning strategies. The results show that image-cleaned method is superior to non-image-cleaned. Image URLs are sensitive to pattern generations. Non-image-cleaned results a large number of data records in the datasets than thus a number of generated frequent itemset patterns are conveyed image URL, which are not useful for cache replacement.

4 Conclusions

In this paper, we apply a data mining technique to generate frequent itemset patterns of the internet access by

users. Data transactions are generated based on the allocated time slots of internet usage. As the public the Internet cluster, different users can use the same single computer; therefore, generating data transaction base on computer IP address may not extract the behavior of users. Association rules are generated using FP-growth algorithm. The frequent itemsets (rules) are then used to implement and extension of LRU cache-replacement technique. The experimental results show that the proposed technique provides promising result and is superior to the base-line cache replacement techniques (i.e. FIFO and LRU). The results obtained are compared with 2 results available in the literature to demonstrate the generalization of the proposed method. In the future, the work is aimed at studying data cleaning strategies to improve the cache replacement method. We plan to improve prediction model and predict the future web access.

5 References

- (1) Songwattana, A., Mining Web logs for Prediction in Prefetching and Caching in Convergence and Hybrid Information Technology, 2008. ICCIT '08. 2008.
- (2) Rudeekorn Soonthornsutee, P.L. Web Log Mining for Improvement of Caching Performance. in Proceedings of the International MultiConference of Engineers and Computer Scientists. 2012. Hong Kong.
- (3) Qiang Yang and H.H. Zhang, Web-Log Mining for Predictive Web Caching, in IEEE Transactions On Knowledge And Data Engineering. 2003. pp. 4.
- (4) Renáta Iváncsy, I.V., Frequent Pattern Mining in Web Log Data. 2006.
- (5) Yin-Fu Huang, J.-M.H. Mining Web Logs to Improve Hit Ratios of Prefetching and Caching. in Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05). 2005.
- (6) Bonchi F., et al. Data Mining for Intelligent Web Caching. in Information Technology: Coding and Computing. 2001.
- (7) Poornalatha G and P.S. Raghavendra, Web Page Prediction by Clustering and Integrated Distance Measure, in IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. 2012.
- (8) Baowen Xu, et al. Application of Data Mining in Web Pre-Fetching. in Multimedia Software Engineering. 2000.
- (9) Theint Theint Aye, Web Log Cleaning for Mining of Web Usage Patterns, in Computer Research and Development (ICCRD), 2011 3rd International Conference. 2011. pp. 490 - 494.
- (10) Han J, Pei J, and Yin Y. Mining Frequent Patterns without Candidate Generation. 2000.
- (11) Saini, K., Squid Proxy Server 3.1 Improve the performance of your network using the caching and access control capabilities of Squid. 2011.