Maejo Int. J. Sci. Technol. **2013**, *7*(02), 232-247

Maejo International Journal of Science and Technology

ISSN 1905-7873
Available online at www.mijst.mju.ac.th

Full Paper

Integration of recommender system for Web cache management

Supawadee Hiranpongsin and Pattarasinee Bhattarakosol *

Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Bangkok 10330, Thailand

* Corresponding author, e-mail: pattarasinee.b@chula.ac.th

Received: 23 July 2012 / Accepted: 8 May 2013 / Published: 3 June 2013

Abstract: Web caching is widely recognised as an effective technique that improves the quality of service over the Internet, such as reduction of user latency and network bandwidth usage. However, this method has limitations due to hardware and management policies of caches. The Behaviour-Based Cache Management Model (BBCMM) is therefore proposed as an alternative caching architecture model with the integration of a recommender system. This architecture is a cache grouping mechanism where browsing characteristics are applied to improve the performance of the Internet services. The results indicate that the byte hit rate of the new architecture increases by more than 18% and the delay measurement drops by more than 56%. In addition, a theoretical comparison between the proposed model and the traditional cooperative caching models shows a performance improvement of the proposed model in the cache system.

Keywords: Web cache farming, proxy server, recommender system, Web classification, Web usage pattern

INTRODUCTION

Due to the ease of use and the high volume of information that is available over the Internet via Web-based systems, the growth rate of Web usage is rapidly enlarging, seemingly with no end in sight. As a result, the quality of service (QoS) of every Internet service provider (ISP) is highly affected and various strategies have been employed to maintain their service levels. Unfortunately, none of these strategies can completely satisfy their customers. Thus, long delays and/or low throughput rates may occur.

One important metric of the QoS is performance. Different variables are applied to measure this indicator, such as average delay, average hit rate and average byte hit rate. The values of these

metrics are influenced by different factors such as limitation of hardware and unqualified service management systems. As a consequence, termination of the browsing transactions may occur [1-2] and performance drops.

In order to avoid this problem, many techniques have been proposed and applied to the management of the cache. Originally, the cache replacement algorithm was proposed for managing a single cache [3-5]. Currently, cache management is based on the cache farming architecture wherein many caches are combined to work together. As a rule, there are two different management models for cache farming: a hierarchical model [6] and a distributed model [7]. The hierarchical caching model consists of intermediate caches located in different levels of the network. Thus, each retrieved Web is copied to the intermediate caches along the traversal path. On the other hand, the distributed caching model allows the search to be performed on the distributed caches over the network.

As a consequence of the above methods, high duplication of objects in hierarchical caching and high bandwidth usage in distributed caching occur [8]. Moreover, when the ISP needs to increase their performance to satisfy their customers, the expanding of hardware and an adjustment of cache management policy always occur. These modifications cause system complexity for administrators and lead to the increasing of management expenses that is not appreciated by organisational managers.

Based on the problems mentioned above, a cache management mechanism that considers types of transaction has been proposed, including the user behaviour being counted as a management factor when managing the cache [9]. This paper proposes a new architecture of cache farming that integrates the concept of a recommender system where all arising behaviours are considered in managing the incoming transactions so the desired retrieval time and bandwidth can be maintained. Consequently, the proposed solution can be seen as an economical system because the number of caches need not expand when the enhancement of performance is desired [10-11]. Moreover, system complexity can be avoided and the cache management policy does not need to be altered.

RELATED WORK

Proxy server is the common solution to serving Web users for an organisation. The number of users and the resultant Web usage over the Internet has rapidly grown over the years. Thus, a long delay may occur for every browse since the amount of browsed content cannot be served or stored in the proxy's cache. To work around this, various researchers have proposed new caching algorithms to manage contents in the caches. In late 2005, Frequency Recency and Size Cache Replacement (FRES-CAR) [12], which considered the document reference recency, frequency and size was proposed. In addition, Bian and Chen introduced the Least Grade Replacement (LGR) [13] that brought the perfect history into account for Web cache optimisation. Other replacement policies exist, such as Semantic and Least Recently Used (SEMALRU) [14], which ignores the objects that are less related to an incoming object or least recently used; and Dump and Clear [15], which was developed to solve the locality problem based on Petersen Graph topology. In addition, many new cache management policies developed to perform in the mobile environment have been recently proposed, for example Distributed Alternative Binding Cache mechanism (DABC) [16] and Cache Replacement Policy Based on Neighbour Nodes' Condition (CRBNC) [17]. The DABC was presented to reduce the problem of limited cache size and also to eliminate the rate of registration

signalling. The CRBNC, focusing on increasing neighbour hits, used the cached data of neighbour nodes as an effective criterion in replacement.

Due to the fact that there are multiple caches with different sizes to be managed, two fundamental cache models have been proposed and implemented; these are the hierarchical and the distributed structure of a Web cache farm. With respect to the hierarchical structure, Foygel and Strelow [18] applied a prefetching algorithm to the hierarchical Web caches. Moreover, the Internet Small Computer System Interface (iSCSI) protocol [19] to communicate between a lower-level and its higher-level proxy server was proposed with a new Web caching scheme. In 2005, the Hierarchical Web Caching Placement and Replacement (HCPR) algorithm was drawn up [20]. This algorithm placed the most frequently referenced documents close to users in the leaf nodes of the hierarchy. Then the Content Distribution Network (CDN) architecture was applied to the hierarchical Web caching technology by Yang and Chi [21]. Additionally, the evaluation of cooperative caching was studied and used as background information for a better understanding of the current CDNs [22].

The cache content placement in emerging scenarios, such as Internet Protocol Television (IPTV) services, has drawn attention in several studies. For example, Borst et al. [23] developed the cooperative cache management algorithms that aimed to minimise the bandwidth costs. In the same year, Applegate et al. [24] created collaborative caching as a global optimisation solution. Later, Dai et al. [25] investigated the capacity provisioning problem in hierarchical caching networks based on a real-world IPTV system. In order to address this problem, an efficient collaborative caching mechanism with dynamic request routing for massive content distribution was proposed. This proposed mechanism achieved better results compared to conventional cache cooperation with static routing schemes.

Originally, the searches on the hierarchical caching model were usually reserved for depth-first search (DFS) exploration. Thus, a general framework of hierarchical caches that can be used by breadth-first search (BFS) was proposed [26]. In order to evaluate the performance of web caching, performance metrics were captured using Hit Rate (HR) and Byte Hit Rate (BHR) [27-28]. Additionally, a cost function model was implemented based on both performance metrics to investigate the suitability of applied policies over the two-level hierarchical cache model [29]. The results indicated that the performance obtained was higher when the lower-level cache used the Least Frequently Used (LFU) or Least Recently Used (LRU) and the upper-level cache used the Greedy Dual-Size (GDS).

In contrast to the hierarchical caching model, in the distributed caching architecture all browsing requests must be distributed over the network. Thus, Summary Cache protocol [30] was proposed to exchange the messages among caches in order to find documents in other caches. Unfortunately, the investigation [31] pointed out that this model caused congestion over the network. Therefore, the techniques of web caching and web prefetching were implemented in the proxy system to solve the problems of server load and congestion control [32].

After the implementation of the hierarchical caching and distributed caching architectures, a hybrid architecture that is the combination of both models has been proposed. One interesting hybrid model was proposed by Baek et al. [33], in which the reference table was employed at each level in the hierarchy except at the lowest level. Another solution in this model focused on some significant factors that were related to the QoS, such as communication between the caches and cache contents [34].

Maejo Int. J. Sci. Technol. **2013**, *7*(02), 232-247

Since the development of Recommender System (RS) in the area of information retrieval is widely implemented in the information search algorithm to shorten the search time, the METIOREW system has been implemented for the Web search mechanism [35]. This system applies the document evaluation results from users as the 'intelligent bookmark' to finding the most relevant Web pages under the given search topics. In contrast, the Amazon.com system uses Item-to-Item Collaborative Filtering [36] as its recommender system. Similar to Amazon.com, the RS is widely implemented in many Internet activities and services such as course management systems [37] and e-learning systems [38-39]. Furthermore, a modification of the RS has been applied on some product or service systems to improve the efficiency of the RS, for example the RS implementation with fuzzy scatter difference [40]. Additionally, time context and group preferences were used to improve the customer profile in collaborative systems [41]. In 2011, Pukkhem and Vatanawood [42] employed learning styles and a word analysis technique to create a learning object recommendation model which provided learners with personalised learning object selection service. The results of experiments demonstrated that the accuracy resulted in high student satisfaction.

Although various caching management mechanisms are continuously proposed, the limitation of the QoS has not been eliminated because the retrieval process must rely on hardware and cache management policies. Thus, every ISP always extends the volume of its service caches for certain periods of time. This management method causes system complexity and increases the cost of management. In this paper the Behaviour-Based Cache Management Model (BBCMM) is therefore implemented to manipulate the cache farm in an economical way. In the following section, details of this mechanism are discussed.

ANALYSIS OF BROWSING BEHAVIOUR

Since the BBCMM is implemented based on the browsing behaviour of users under the service organisation, the first step in applying this model is to identify types of browsing characteristics of the organisation. The browsing methods can be identified by considering all existing transactions that pass through the cache farm in which a record is stored in the proxy log.

In this research samples of retrieval behaviour were collected from Chulalongkorn University with its many faculties and research activities, both in pure sciences and social sciences, and national and international subjects. As a consequence, there are a large number of categories of transactions during Internet access. In addition, under the Internet management process of the university, every browsing query must run through the proxy system and some part of this query is stored in the squid log file referred to as access.log. The data in the access.log file, consisting of 1.5 TB of compressed files, was sampled from January to August 2009 for this study. After uncompressing the files, the number of transactions was approximately 50 million per day.

The format of the access.log allows many fields to be stored, such as the retrieval time or the timestamp, the file size, the destination URL and the access situation (hit/miss). However, in this research only two fields are used: the destination URL and the file size. Since the types of website can be classified as static and dynamic [43], the URLs of all dynamic websites consider only the first part of the URL's name. For example, there is a CNN URL that presents the news of girl brides being abducted as a fabled HIV cure (http://thecnnfreedomproject.blogs.cnn.com/2012/05/27/girl-brides-abducted-as-fabled-hiv-cure/?hpt=hp_c2). This address will be truncated and considered only as thecnnfreedomproject.blogs.cnn.com. All these URLs will be counted for frequency usage during the time of the study. This method is called frequency-based analysis. The result of this

Maejo Int. J. Sci. Technol. **2013**, *7*(02), 232-247

method is the 'Frequency Pattern (FP)', which shows the popularity of websites in a certain period of time.

It is a fact that congestion over the communication channel occurs because of a large volume of transferred information. Thus, values in the file size field of the access.log are analysed to find the load pattern of the retrieved URLs that is the real load of the cache. This load pattern is called 'Loading Size Pattern (LSP)' and this value can be applied as a weighting value in the architecture designing process of the cache farm.

After considering the existing values of the FP, three different groups can be identified. The first group, the low frequency (LF), is the group of URLs that are browsed less than three times per week. This group represents approximately 58% of all URLs. Most of these websites are sites where their pages were obtained from various server locations such as facebook.com or live.com. In addition, websites with frequency of only one or two times per week are authorised websites. The second group, the medium frequency (MF), accounts for 17% of the total URLs and are retrieved from four to eight times per week. The last group, the high frequency (HF), is the remaining 25% of the total URLs and they are retrieved more than eight times per week.

As a consequence of the LSP-based analysis, the Web size can be grouped into three specific ranges. The first range, the normal size (NS), accounts for 69% of all websites with the file size of 0-4,000 bytes. This NS can be further categorised into two sub-ranges as small size (SS) and medium size (MS). The SS is the group for the browsed files with sizes less than 1,100 bytes while the MS is the group for the browsed files with sizes of 1,100-4,000 bytes. The second range, the large size (LS), includes over 30% of all websites with the file size of 4,000 bytes - 10 Mbytes. The third range, the extra size (ES), is less than 1% of all websites and includes all files that are larger than 10 Mbytes. The ES range is independently grouped as another group due to the large size of the websites even though the number of sites browsed is small. Table 1 shows the grouping for Web cache based on the FP and the LSP.

Table 1. Grouping for Web cache

Type	Grouping criteria	Loading ratio (%)
1	LF with SS, MF with LS	25.11
2	LF with MS, MF with NS	23.37
3	HF with all size ranges except ES	25.62
4	All frequency groups with ES	25.89

Referring to Table 1, the probability that a retrieved URL will be a member of a group is approximately equal to 0.25 for every group. These criteria are employed to set up the proxy caches in the BBCMM. Nevertheless, groups of websites are altered when the pattern of browsed websites is changed. Consequently, the system configuration will occur in the classification process. This group classification must be performed on a fixed schedule, e.g. weekly or monthly, depending on the business objective and the policy of each organisation. Details of the BBCMM are presented in the following section.

PROPOSED ARCHITECTURE

Owing to the limitation of proxy caches of the ISPs and the existing policy of cache management, the number of caches must be increased when the number of users is expanded. Consequently, the complexity of cache management increases. Thus, in this investigation a new

Maejo Int. J. Sci. Technol. **2013**, *7*(02), 232-247

architecture, BBCMM, is proposed to eliminate this weakness. As can be seen from Figure 1, the BBCMM consists of different groups of specific proxy server (SPS) whose main part is the proxy manager (PM) that works with the data in the Web profile database system (WPDB). Following is the description of each part of the BBCMM.

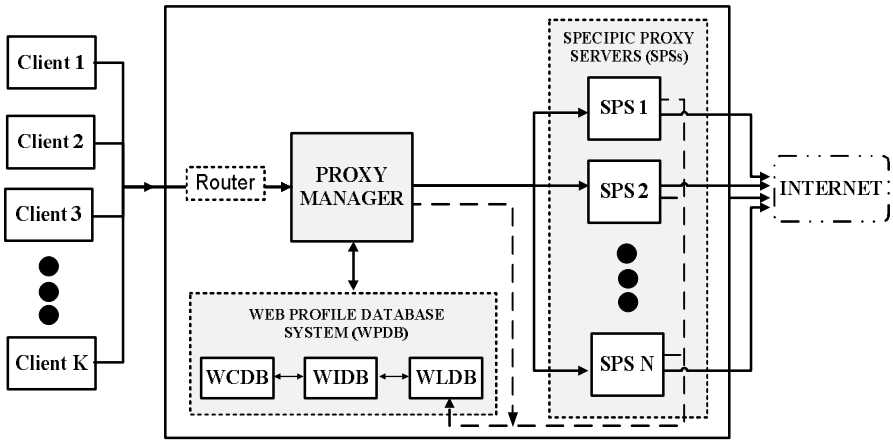

Figure 1. Behaviour-Based Cache Management Model (BBCMM)

Web Profile Database System

The WPDB consists of three sub-databases: a Web classification database (WCDB), a web identification database (WIDB) and a log database (WLDB), as illustrated in Figure 1. These sub-databases are implemented for Web classification when users call for websites. Of these three databases, the WCDB and the WIDB are used in the Web classification process when retrieving URLs. The WCDB contains a boundary value of both the browse frequency (bv_F) and the Web size (bv_S). These data are calculated and used to identify the group whenever a request by a user arrives at the PM. The WIDB is the database that stores the information for each Web, e.g. host name of web, browse frequency and size of web. The information stored in WIDB is used with both boundary values stored in the WCDB to identify the pattern of browsed websites by an automatic classification module (ACM) in the PM. Figure 2 shows the classification of the requested Web that is sent to the corresponding group or the SPS using the two boundary values, bv_F and bv_S, as indicators. Each group has bv_F and bv_S, which are denoted by $bv_F(i)$ and $bv_S(i)$ respectively, for the i^{th} classified group. The $bv_F(i)$ and the $bv_S(i)$ are defined in equations 1 and 2 respectively.

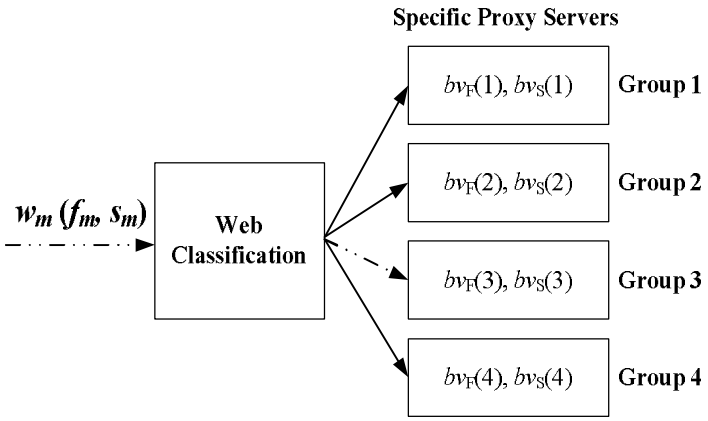

Figure 2. Web classification with boundary values

Maejo Int. J. Sci. Technol. **2013**, *7*(02), 232-247

Given $F_l(i)$, $F_u(i)$, $S_l(i)$, $S_u(i) \in \Re$, and $F_l(i) < F_u(i)$ and $S_l(i) < S_u(i)$, let $F_l(i) = avg_F(i) - std_F(i)$ and $F_u(i) = avg_F(i) + std_F(i)$, where $avg_F(i) \in \Re$ is the average of the browse frequency of group (i) and $std_F(i) \in \Re$ is the standard deviation of the browse frequency of group (i). In addition, $S_l(i) = avg_S(i) - std_S(i)$ and $S_u(i) = avg_S(i) + std_S(i)$, where $avg_S(i) \in \Re$ is the average of the Web size of group (i) and $std_S(i) \in \Re$ is the standard deviation of the Web size of group (i). Then,

$$bv_F(i) = [F_l(i), F_u(i)] = \{x \in \Re \mid F_l(i) \leq x \leq F_u(i)\} \quad (1)$$

$$bv_S(i) = [S_l(i), S_u(i)] = \{y \in \Re \mid S_l(i) \leq y \leq S_u(i)\} \quad (2)$$

Let m be the set of different requested websites, named $\{w_1, w_2, \ldots, w_m\}$; each w_m has the browse frequency denoted by f_m and the Web size denoted by s_m.

Since $f_m, s_m \in \Re$, and $f_m \in bv_F(i) \leftrightarrow F_l(i) \leq f_m \leq F_u(i)$ and $s_m \in bv_S(i) \leftrightarrow S_l(i) \leq s_m \leq S_u(i)$, thus, if $f_m \in bv_F(i)$ and $s_m \in bv_S(i)$, then the w_m is classified to the group (i). However, the data management and maintenance are recognised to handle data overloading in the database. For this reason, if the classified w_m is inactive for a specific time it will be removed from the database.

Consider the situation when a request is issued from a client. This request will be recorded into the WLDB so the system can be recovered when a failure or an unexpected event arises. The records in the WLDB are not only obtained from the retrieved command from the ACM, but also from the retrieved commands recorded in the local log database of each SPS, or caches. The transferring of data from all local log databases from each SPS is in the offline mode and is performed only before the RAM starts its evaluation. Nevertheless, all data of the WPDB system are necessary for the PM processes that are described in the following section.

Proxy Manager

The PM is an assigned proxy server in the cache farm, responsible for classifying websites into groups depending on the browsing patterns mentioned previously. This system consists of six modules: Record Analyser Module (RAM), Automatic Classification Module (ACM), Gateway-like Module (GM), Internet Communication Module (ICM), Squid Cache Module (SCM) and WPDB Communication Module (WPDB CM). One significant function of the system is to adapt the recommender system in the transaction identification when the transaction is indefinable in the normal classification process. Figure 3 demonstrates the PM's architecture in performing its tasks. The details of these modules are explained below.

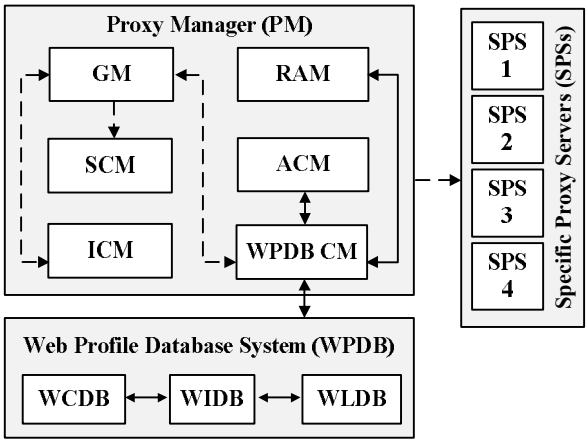

Figure 3. Proxy manager's architecture

Maejo Int. J. Sci. Technol. **2013**, *7*(02), 232-247

Record Analyser Module

The RAM is the only module that works individually for pattern classification. This module deals with all URLs stored in the WLDB. The outcomes of the RAM computing are new values of browse frequencies and Web sizes that must be updated to the WIDB at the end of every day. These values are used to justify all unknown websites received from the ACM. Additionally, the re-analysing of all boundary values, $bv_F(i)$ and $bv_S(i)$, for every group (i) will be performed every weekend because websites are constantly inserted, updated and deleted. These new values will overwrite the previous values in the WCDB. However, these new boundaries may not be suitable since the number of websites in the existing groups under the newly defined indexes might break the load balancing policy of the network. Thus, websites with their frequencies and sizes that are close to the upper or lower bounds of their neighbours are changed to their neighbours' value until the number of websites in each group is significantly equivalent. Then the new boundary values, $bv_F(i)$ and $bv_S(i)$, for every group (i) will be recalculated and stored in the WCDB. As a result of the RAM process, the size of available caches for each group when passing the process of the ACM may change to obtain high performance and maintain the load balance of each group properly.

Automatic Classification Module

After all boundaries are calculated by the RAM, the ACM will use them for Web grouping. As mentioned in the RAM section, the pattern analysis of Web browsing will be performed every weekend. Thus, the process of the ACM, like the RAM, will automatically run every weekend. The group categorisation of the ACM is performed based on the popularity system called the recommender system. Since this system has two different approaches, the content-based (CB) approach and the collaborative-filtering (CF) approach [44], the ACM integrates both of them to gain the highest efficiency in the Web categorisation process.

Based on the calculation values obtained from the RAM process, the CF and the CB are applied to sort websites into suitable groups. Since there are various methods of the CF classification process, the user-based collaborative filtering technique is applied to identify websites in the WIDB. The real content of each website is unidentified; thus, in this investigation the browsing frequency is used as the indicator for users' preference based on the CF rule. Moreover, the file size is used as the Web's characteristic based on the CB concept.

Since the values in the WIDB contain the real browse frequency and average file size of retrieved websites during a week, every browse frequency and the average file size of each site will be compared with $bv_F(i)$ and $bv_S(i)$ in the WCDB for grouping. There are two possible situations in this comparison. The first is the normal case in which websites satisfy the condition of group (i) in the WCDB, $f_m \in bv_F(i)$ and $s_m \in bv_S(i)$; those websites are classified as websites of group (i). The second situation occurs when there is no suitable condition in the WCDB for websites in the WIDB, either $f_m \notin bv_F(i)$ or $s_m \notin bv_S(i)$ or neither of them. These websites will be assigned to the group with highest frequency under the assumption that the more popular the request is, the more chance occurs to meet the requirement.

Gateway-like Module

The GM classifies all arriving transactions from the Internet so they will be sent to suitable proxy servers. The situation of a transaction can either be classifiable or unclassifiable. If the requested transaction is classifiable, then the request will be sent via the ICM to the classified SPS that is linked to the PM system as shown in Figure 1. Otherwise, the request will be sent to the SCM

to retrieve the required information. The GM can classify the request using information in the read-only mode of the WIDB.

Internet Communication Module

The ICM sends a message from the PM to the SPSs according to the classification result of the ACM. The connection between this module and an SPS is connection-oriented, using Transmission Control Protocol/Internet Protocol (TCP/IP) to guarantee the delivery of data. The format of the sent message is the same as that of the packet received from the GM. The communication type of the ICM to any SPS is the simplex method because every SPS will return the requested page directly to the users.

Squid Cache Module

A cache module in the PM, this module is to serve all unidentified websites. The load of this module therefore depends on the existence of the number of undefined URLs.

WPDB Communication Module

Every process of the PM must use data from the WPDB system. This system consists of three sub-databases: WLDB, WIDB and WCDB. Thus, the WPDB CM creates a connection to the WPDB system to retrieve the required data. The connection performed by this module is connection-oriented using TCP/IP to prevent the occurrence of network congestion and reduce maintenance costs due to transmission loss.

Specific Proxy Server

The SPS is a group of caches defined based on the WIDB contents. Since there are many types of defined boundaries in the WCDB that are related to websites in the WIDB, there are many installed SPSs in the PM. However, the size of each SPS depends on the weekly calculation of the RAM. As a consequence, the size of each SPS is dynamic in order to maintain the performance of each server which affects the performance of the entire system.

EVALUATION METHOD

In this section, the implementation of the evaluation system environment is performed on HP-2 Quad Cores with XEON Processors and 16-GB main memory running Ubuntu 10.04 desktop. The database management system employs MySQL server version 5.1, and the phpMyAdmin running on Apache2 Web server is used to deal with MySQL server. All mechanisms maintaining the system are developed using Java.

In order to demonstrate that the proposed caching mechanism, the BBCMM, performs better than the single caching mechanism, a comparison between both systems is performed. Although the comparison is based on single caching mechanism, this mechanism is a general method that is implemented for the actual use in every organisation. The performance estimation of the BBCMM and the existing caching mechanism (EM) is based on the results of the trace-driven simulation implemented in a virtual machine environment. The outcomes of performance evaluation between the BBCMM and the EM are therefore trustworthy and acceptable. The details of simulation experiments are explained as follows.

Based on the defined groups in the previous section, the WCDB contains four pairs of boundaries in accordance with the four groups of websites in the WIDB. Thus, there are four SPSs in the simulation system and one SCM for undefined websites as mentioned in the GM section.

Maejo Int. J. Sci. Technol. **2013**, *7*(02), 232-247

Each SPS and the EM employ the Least Recently Used (LRU) algorithm as their cache replacement policy. As a result, the similarity in individual cache management can be directly compared without bias or any external influence. The data for this evaluation is based on the real Internet usage of students and staff of a small-size government university in Thailand during October 2010. The simulation data are approximately 18 million records. Figure 4 illustrates the testing environment for the BBCMM.

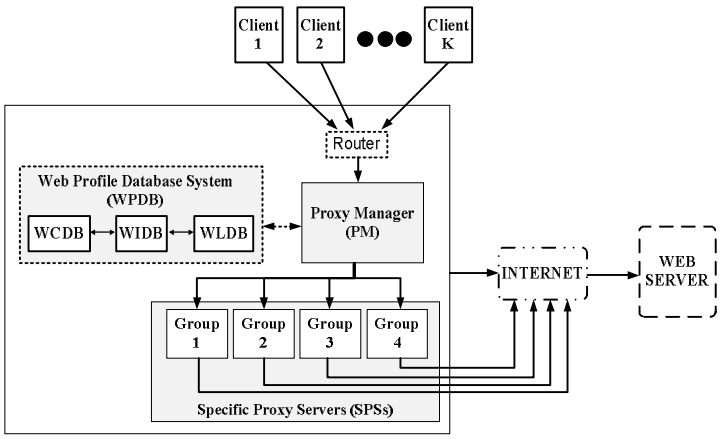

Figure 4. Simulation system environment for the BBCMM

SIMULATION RESULTS

To indicate the performance of the BBCMM and the EM, the performance metrics in this research constitute hit rates, miss rates, byte hit rates, byte miss rates and response times of hits and misses. These metrics are measured from running data in the simulation system obtained from the university. The results of this run are presented in Table 2.

Table 2. Result overview of simulation experiments

Metric	Implemented model		Difference (%)
	BBCMM	EM	
Hit rate (%)	41.82	39.53	+5.80
Miss rate (%)	58.18	60.47	-3.79
Byte hit rate (%)	43.39	36.76	+18.05
Byte miss rate (%)	56.61	63.24	-10.49
Average response time of hit (ms.)	85.44	196.20	-56.45
Average response time of miss (ms.)	2842.12	2431.25	+16.90

Referring to the results in Table 2, the hit rate of the BBCMM is 5.8% higher than the EM while the miss rate is also reduced. This can primarily be interpreted that the new architecture and caching mechanism are serving users better than the original concept. In addition, the byte hit rate shows an 18% increase from the EM and the byte miss rate is also reduced by approximately 10.5%. The average response time of the hit mode is more than 56% lower than the EM. Unfortunately, the simulation result shows that if websites are unidentified, users must wait for their requests longer than in the normal situation. The daily performance of the BBCMM can be drawn as in Figure 5, where every significant metric is measured within 30 days.

Maejo Int. J. Sci. Technol. **2013**, *7*(02), 232-247

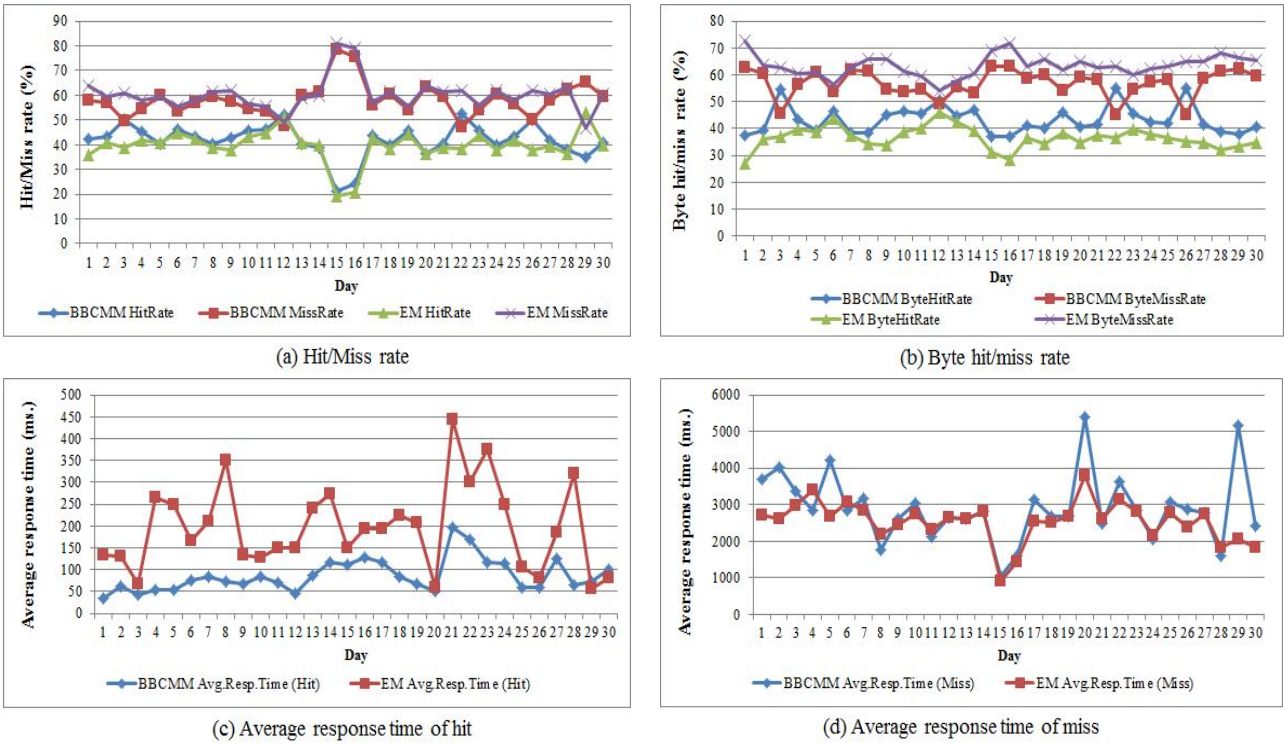

Figure 5. Daily performance comparison between BBCMM and EM

Referring to Figure 5(a), in the beginning of the measurement process the hit rate of the BBCMM is slightly higher than that of the EM. Nevertheless, after 15 days of this experiment the number of hits based on the BBCMM has the potential to be much better. This is because the variety of websites in each cache is small when starting the BBCMM, but all requested websites based on the defined patterns will be accrued to the system daily until most of the requests are available in the cache. However, if retrieved URLs have no pattern as expected, the daily hit rates of both the BBCMM and the EM will be dropped.

Since the hit rate is increased under BBCMM system, the simulation also illustrates that the number of byte hits is increased as shown in Figure 5(b). This is the result of the grouping mechanism of the cache because websites with similar characteristics are stored and manipulated in the same area. Thus, the possibility of the required website being found in the managed caches is higher compared to when different groups of websites are stored in the same area as the EM.

With respect to finding the requested web in the caches, the BBCMM is faster than the EM and the response time to obtain the required information is shorter for the BBCMM as shown in Figure 5(c). However, the retrieving information from the external sources of the BBCMM architecture is greater than the retrieval time of the EM. This is a result of the overhead required to search the non-existent content in each cache as shown in Figure 5(d).

The test results confirm that the cache management policy of BBCMM is highly efficient and QoS can be achieved. This is the result of categorising and grouping websites based on the available retrieval pattern; the classified websites are then stored in the individual caches based on the defined groups. In order to provide more insights into the proposed BBCMM, a comparison with models other than EM is performed. A theoretical comparison with two common cooperative

Maejo Int. J. Sci. Technol. **2013**, *7*(02), 232-247

caching models, namely hierarchical model (HM) and distributed model (DM), is discussed in the following section.

THEORETICAL COMPARISON

The World Wide Web can be considered as a universal warehouse of web pages and links that offer large amounts of data for the Internet users. Thus, no one can imagine the number of users accessing that data. As a result, it is a major challenge for the proposed BBCMM, the HM and the DM to manipulate retrieved transactions. In fact, the main challenge and design principle in the cooperative caching architecture is to rapidly locate requested cached websites. To this end, this research focuses on two significant issues: the number of hops and the processing time for both hit and miss cases. The number of hops for traversing multiple levels in the cache system should be optimised by efficient locating and accessing of data, while the processing time handling each request may be reduced to shorten the response time.

The cache system is assumed to contain k clients and m web pages. A request for a specific web page originates from one of the clients. In addition, the client will not send another request before obtaining the previously requested page. Therefore, the maximum number of requests in the system is k. For a fair comparison, the HM, DM and BBCMM contain n nodes or caches and each web-caching model uses the LRU for page replacement.

Number of Hops

In all types of web caching model when the cache receives a requested transaction, it first checks the local cache for the requested page. If the requested page has a current copy of the object in the cache—the hit state, it simply returns the page to the user. Thus, the time complexity of the number of hops for both the HM and the DM is obviously $O(1)$ since they can return the page obtained from the local cache. With the results of classification and grouping mechanism of the BBCMM, after GM of the PM classifies the retrieved page to a suitable cache, it must send the transaction to the SPS or the SCM as mentioned previously. Therefore, the time complexity of the number of hops for the BBCMM is $O(1)$. As a result, the time complexity of the number of hops for all three caching models for the hit case is a constant value.

On the other hand, if the search in the cache returns a miss state, the required object will be sent to other cooperative caches and also to the original web server if the request cannot be found in the cache system. Since all caches that are arranged in a tree-like structure are compulsory objects for the HM, the time complexity of the number of hops in the miss state is $O(\log n)$, where n is the number of nodes. In contrast to the HM, the DM has no intermediate caches and its connection is performed among n peer cooperative caches. So when the miss state occurs, the content search will be performed on other $n-1$ distributed caches. For this reason, the time complexity of the number of hops for the DM is $O(n)$. For the BBCMM, each SPS deals with the specific contents that are cached. If the request transaction is not satisfied by the SPS, this transaction will be fetched from the original web server without accessing other SPSs. Thus, the time complexity of the number of hops for the BBCMM when a miss occurs is $O(1)$.

Therefore, the proposed BBCMM can minimise the number of hops in the cache system on a miss case under the defined assumptions. Moreover, the number of hops can be maintained on a hit case when the BBCMM is implemented.

Maejo Int. J. Sci. Technol. **2013**, *7*(02), 232-247

Processing Time

The processing time is measured from the time that the request is sent until the time when the user receives the response. Given that t_u is the processing time between the user and the local cache for the request, t_c the processing time between cooperative caches for the request, and t_o the processing time between the cache and the original web server for the request. Let t_{PM} be the processing time between the PM and the SPSs or the SCM inside the PM. There are two situations to consider: hit and miss.

Under the hit state, both the HM and the DM contain the requested page in their own caches. Thus, it is clear that the time complexity of the processing time for both of them is $O(t_u)$, where t_u is a constant value. In the case of BBCMM, the time complexity of the processing time is $O(t_u + t_{PM})$, where t_u and t_{PM} are constant values. As a consequence, with BBCMM, the transaction issued from the user will be sent first to the PM for the web page classification. Then the classified web page will be forwarded to the identified SPS or handled by the SCM. So two processing times, t_u and t_{PM}, are included in the elapsed time of the BBCMM.

The other situation, the miss state, occurs when the issued request cannot be satisfied by the local cache. Consequently, the search will be performed over cooperative caches for the HM and the DM. In order to search for the object among siblings, the total processing time t_c for the HM is $t_c\log(n)$ and that for the DM is t_cn. However, there is no processing time t_c for the BBCMM. Thus, the processing times for handling each transaction of the HM, the DM and the BBCMM are $t_u + t_c\log(n) + t_o$, $t_u + t_cn + t_o$ and $t_u + t_{PM} + t_o$ respectively, t_u, t_c, t_{PM} and t_o being constant values and n the number of caches. Consequently, the time complexities of the processing time for the request that is served by the caches in HM, DM and BBCMM are $O(\log n)$, $O(n)$ and $O(T)$ respectively, where n is a number of caches and T is a constant value.

In summary, the proposed BBCMM reduces the response time with shorter processing time on both of hit and miss states when comparing with the traditional caching models, the HM and the DM.

CONCLUSIONS

Retrieving information from any website within an organisation requires proxy and cache management systems to filter suitable information flowing in and out the organisation's networks. Thus, accessing performance depends on the cache management mechanism of the proxy server and the cache farm. Even though various techniques have been proposed and implemented to increase the service quality, there are still difficulties in maintaining the service quality desired as the number of transactions expands. The proposed BBCMM integrates the concepts of the recommender system, using frequencies and browsed file sizes, in order to maintain the quality of service, which is the desired response time for the Internet users. The components of the BBCMM are PM, WPDB and SPSs. Within the PM, there is an important module known as the ACM, which deals with the classifying of all retrieved websites.

This experiment has demonstrated that under a systematic organisation with a clear objective of browses, most of the important metrics have been changed positively. This can be accomplished while neither adding nor adjusting hardware or implementing changes in the cache management policy. These positive changes are, among others, the reduction of the response time of the BBCMM by more than 56% while the hit rate and the byte hit rate of the BBCMM are increased by more than 5% and 18% respectively. The comparison results measured by the time complexity of the number of hops and the processing time have shown that the proposed BBCMM can improve

the caching performance in the value of the time constant. The BBCMM can thus be an eco-proxy system that provides high service performance for an organisation.

ACKNOWLEDGEMENTS

This research was financially supported by: 1) the Inter-University Network (UniNet) Unit under the Office of the Higher Education Commission (OHEC), Thailand; 2) Chulalongkorn University's 90th Anniversary Scholarship: Ratchadapisek Sompote Endowment Fund; and 3) the University Development Committee (UDC) Scholarship Programme of the OHEC. The authors would also like to express their appreciation to Nakhon Pathom Rajabhat University (Thailand) for providing the simulation data. Mr. Tony Criswell of the English Department, Faculty of Humanities and Social Sciences, Khon Kaen University is thanked for English proofreading.

REFERENCES

1. R. P. Wooster and M. Abrams, "Proxy caching that estimates page load delays", *Comp. Netw. ISDN Syst.*, **1997**, *29*, 977-986.
2. J. Mardesich, "The Web is no shopper's paradise", *Fortune*, **1999**, *140*, 188-198.
3. P. Cao and S. Irani, "Cost-aware www proxy caching algorithms", Proceedings of USENIX Symposium on Internet Technology and Systems, **1997**, Monterey, USA, pp.193-206.
4. S. Jin and A. Bestavros, "GreedyDual* Web caching algorithm: Exploiting the two sources of temporal locality in Web request streams", *Comp. Commun.*, **2001**, *24*, 174-183.
5. B. R. Haverkort, R. E. A. Khayari and R. Sadre, "A class-based least-recently used caching algorithm for world-wide web proxies", *Lect. Notes Comp. Sci.*, **2003**, *2794*, 273-290.
6. A. Chankhunthod, P. B. Danzig, C. Neerdaels, M. F. Schwartz and K. J. Worrell, "A hierarchical internet object cache", Proceedings of USENIX Annual Technical Conference, **1996**, San Diego, USA, pp. 153-164.
7. R. Tewari, M. Dahlin, H. M. Vin and J. S. Kay, "Beyond hierarchies: Design considerations for distributed caching on the Internet", Proceedings of the 19th IEEE International Conference on Distributed Computing Systems, **1999**, Austin, USA.
8. P. Rodriguez, C. Spanner and E. W. Biersack, "Analysis of Web caching architectures: Hierarchical and distributed caching", *IEEE/ACM Trans. Netw.*, **2001**, *9*, 404-418.
9. P. Bhattarakosol and V. Ngamaramvaranggul, "An Internet web management policy for government organization", Proceedings of Network Research Workshop in 18th APAN Meetings, **2004**, Cairns, Australia, pp.249-255.
10. W. Srisujjalertwaja and P. Bhattarakosol, "Customer-oriented policy for proxy management system", Proceedings of International Computer Symposium, **2004**, Taipei, Taiwan, pp.1168-1173.
11. S. Hiranpongsin and P. Bhattarakosol, "Intelligent caching algorithm for Web cache farming system", Proceedings of International Conference on Wireless Information Networks and Business Information System, **2009**, Kathmandu, Nepal.
12. G. Pallis, A. Vakali and E. Sidiropoulos, "FRES-CAR: An adaptive cache replacement policy", Proceedings of International Workshop on Challenges in Web Information Retrieval and Integration, **2005,** Tokyo, Japan, pp.74-81.
13. N. Bian and H. Chen, "A least grade page replacement algorithm for Web cache optimization", Proceedings of the First International Workshop on Knowledge Discovery and Data Mining, **2008**, Adelaide, Australia, pp.469-472.

14. K. Geetha, N. A. Gounden and S. Monikandan, "SEMALRU: An implementation of modified Web cache replacement algorithm", Proceedings of World Congress on Nature and Biologically Inspired Computing, **2009**, Coimbatore, India, pp.1406-1410.
15. B. Zhang and H. Wu, "A new distributed caching replacement strategy", Proceedings of 3rd IEEE International Conference on Communication Software and Networks, **2011**, Xi'an, China, pp.167-170.
16. S. S. Hasan, R. Hassan and F. E. Abdalla, "A new binding cache management policy for NEMO and MIPV6", *J. Theoret. Appl. Inform. Technol.*, **2012**, *36*, 113-117.
17. A. S. Ghasemi and A. M. Rahmani, "A new cache replacement policy based on neighbor nodes' condition in mobile environments", *J. Comp.*, **2012**, *4*, 164-169.
18. D. Foygel and D. Strelow, "Reducing Web latency with hierarchical cache-based prefetching", Proceedings of International Workshop on Scalable Web Services, **2000**, Washington, DC, USA, pp.103-108.
19. H. Lim and D. H. C. Du, "Design considerations for hierarchical Web proxy server using iSCSI", Proceedings of Symposium on Applications and the Internet, **2003**, Orlando, USA, pp.414-417.
20. W. Li, K. Wu, X. Ping, Y. Tao, S. Lu and D. Chen, "Coordinated placement and replacement for grid-based hierarchical Web caches", *Lect. Notes Comp. Sci.*, **2005**, *3795*, 430-435.
21. F. H. Yang and C. H. Chi, "Using hierarchical scheme and caching techniques for content distribution networks", Proceedings of 3rd International Conference on Semantics, Knowledge and Grid, **2007**, Xi'an, China, pp.535-538.
22. J. Zhang, "A literature survey of cooperative caching in content distribution networks", **2012**, arxiv.org/pdf/1210.0071 (Accessed: October, 2012).
23. S. Borst, V. Gupta and A. Walid, "Distributed caching algorithms for content distribution networks", Proceedings of IEEE INFOCOM, **2010**, San Diego, USA, pp.1-9.
24. D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee and K. K. Ramakrishnan, "Optimal content placement for a large-scale VoD system", Proceedings of 6th International Conference on Emerging Networking Experiments and Technologies, **2010**, Philadelphia, USA.
25. J. Dai, Z. Hu, B. Li, J. Liu and B. Li, "Collaborative hierarchical caching with dynamic request routing for massive content distribution", Proceedings of IEEE INFOCOM, **2012**, Orlando, USA, pp.2444-2452.
26. R. Mateescu and A. Wijs, "Hierarchical adaptive state space caching based on level sampling", *Lect. Notes Comp. Sci.*, **2009**, *5505*, 215-229.
27. M. Busari and C. Williamson, "ProWGen: A synthetic workload generation tool for simulation evaluation of web proxy caches", *Comp. Netw.*, **2002**, *38*, 779-794.
28. L. Shi and Y. Zhang, "Optimal model of Web caching", Proceedings of 4th International Conference on Natural Computation, **2008**, Jinan, China, pp.362-366.
29. L. Shi, P. Yao, L. Wei and Y. Tao, "Cost-benefit analysis of the Web hierarchy caching model", *Inform. Technol. J.*, **2012**, *11*, 364-367.
30. L. Fan, P. Cao, J. Almeida and A. Z. Broder, "Summary cache: A scalable wide-area Web cache sharing protocol", *IEEE/ACM Trans. Netw.*, **2000**, *8*, 281-293.
31. M. Piatek, "Distributed Web proxy caching in a local network environment", **2004**, www.acm.org/src/subpages/papers/piatek.src.2004.pdf (Accessed: March, 2005).
32. C. V. Manikandan, P. Manimozhi, B. Suganyadevi, K. Radhika and M. Asha, "Efficient load reduction and congession control in Internet through multilevel border gateway proxy caching",

Proceedings of IEEE International Conference on Computational Intelligence and Computing Research, **2010**, Coimbatore, India, pp.1-4.
33. J. Baek, G. Kaur and J. Yang, "A new hybrid architecture for cooperative Web caching", *J. Ubiq. Converg. Technol.*, **2008**, *2*, 1-11.
34. M. S. E. Oneis, H. Barada and M. J. Zemerly, "Towards an efficient Web caching hybrid architecture", Proceedings of 4th International Conference on Information Technology, **2009**, Amman, Jordan.
35. D. Bueno, R. Conejo and A. A. David, "METIOREW: An objective oriented content based and collaborative recommending system", *Lect. Notes Comp. Sci.*, **2002**, *2266*, 310-314.
36. G. Linden, B. Smith and J. York, "Amazon.com recommendations: Item-to-item collaborative Filtering", *IEEE Internet Comp.*, **2003**, *7*, 76-80.
37. J. Itmazi and M. Megías, "Using recommendation systems in course management systems to recommend learning objects", *Int. Arab J. Inform. Technol.*, **2008**, *5*, 234-240.
38. J. Bobadilla, F. Serradilla, A. Hernando and MovieLens, "Collaborative filtering adapted to recommender systems of e-learning", *Knowl.-Based Syst.*, **2009**, *22*, 261-265.
39. K. Souali, A. E. Afia, R. Faizi and R. Chiheb, "A new recommender system for e-learning environments", Proceedings of International Conference on Multimedia Computing and Systems, **2011**, Ouarzazate, Morocco, pp.1-4.
40. Y. Gong and Q. Xue, "Study on internet recommendation system of collaborative filtering based on scatter difference", Proceedings of International Conference on Computer, Mechatronics, Control and Electronic Engineering, **2010**, Changchun, China, pp.160-163.
41. M. Julashokri, M. Fathian, M. R. Gholamian and A. Mehrbod, "Improving recommender system's efficiency using time context and group preferences", *Adv. Inform. Sci. Service Sci.*, **2011**, *3*, 162-168.
42. N. Pukkhem and W. Vatanawood, "Personalised learning object based on multi-agent model and learners' learning styles", *Maejo Int. J. Sci. Technol.*, **2011**, *5*, 292-311.
43. J. Ravi, Z. Yu and W. Shi, "A survey on dynamic Web content generation and delivery techniques", *J. Netw. Comput. Appl.*, **2009**, *32*, 943-960.
44. M. Balabanovic and Y. Shoham, "Fab: Content-based, collaborative recommendation", *Commun. ACM*, **1997**, *40*, 66-72.