Journal of Engineering Science and Technology
Vol. 4, No. 2 (2009) 206 - 219

INTELLIGENT CACHE FARMING ARCHITECTURE WITH THE RECOMMENDER SYSTEM

S. HIRANPONGSIN*, P. BHATTARAKOSOL

Department of Mathematics, Faculty of Science, Chulalongkorn University
254 Phyathai Road, Patumwan, Bangkok Thailand. 10330
*Corresponding Author: supawadee.h@student.chula.ac.th

Abstract

The Quality of Services (QoS) guaranteed by the Internet Service Providers (ISPs) is an important factor for users' satisfaction in using the Internet. The implementation of the web proxy caching has been implemented to support this objective and also support the security procedure of the organizations. However, the success of guaranteeing the QoS of each ISP must be depended on the cache size and efficient caching policy. This paper proposes a new architecture of cache farming with the recommender system concept to manage users' requirements. This solution helps reducing the retrieval time and also increasing the hit rate although the number of users increases without expanding the size of caches in the farm.

Keywords: Cache farming, Cache management, Proxy system, Recommender system, Users' requirement

1. Introduction

Presently, browsing webs and files through the Internet are normal activity of the Internet users. Various services have been provided by the Internet Service Providers (ISPs), such as AT&T, Telstra BigPond, Lexicon, CAT, etc., in order to offer and maintain their Quality of Services (QoS). One mechanism that has been implemented by every ISP is the installation of the proxy system to serve the Internet users. This mechanism helps reducing the retrieval time and also performs security assurance for the installed organizations. However, the performance of proxy implementation is depended on two factors: the cache size, and the management policy.

Abbreviations

ACM	Automatic Classification Module
ICFA	Intelligent Cache Farming Architecture
ICM	Internet Communication Module
ISP	Internet Service Provider
PM	Proxy Manager
PMI	Proxy Manager Interface
QoS	Quality of Services
RAM	Record Analyzer Module
SPS	Specific Proxy Server
TCP/IP	Transmission Control Protocol/ Internet Protocol
UIDB	URL Identification Database
ULDB	Log Database
UPDB	User Profile Database
URDB	User Registration Database

Currently, the World Wide Web (WWW) is viewed as the most successful application for providing simple access to a wide range of information and services. As the result, the popularity of Internet usage constantly grows with web strategies developed to support those demand. Most clients browsing webs and loading files through the Internet expect that they should obtain the fast services. Unfortunately, clients often experience delays when accessing the Internet due to the hardware limitation, or low qualified service management system for clients of Internet Service Providers.

In order to serve users' expectations, the latencies must be maintained in a small number. Among numerous studies for enhancing performance of the Internet services, one effective solution is the caching popular web documents at proxy server. This standard technique improves the QoS over the Internet. Additionally, the web proxy caching is one ways to reduce latencies, bandwidth usage, and increase the hit ratio.

Initially, web proxy caching is focused on a single proxy server [1-3]. For this model, each individual proxy server acts independently. Furthermore, there is no cooperation among the caching proxies. Since the number of clients linked to a single cache continuously increases, making the amount of information in cache rapidly increases, the caching performance must be dropped in a certain amount of time.

According to the problem of the single cache stated above, two common approaches using the concept of cooperation among the caching proxies: a hierarchical [4] and a distributed [5] caching, are employed over the Internet. With the hierarchical caching architecture, this architecture requires intermediate caches in the network. Each retrieval process leaves a copy of the requested web document in each intermediate cache through the traversal paths. Unfortunately, there are several problems associated with a caching hierarchy, such as additional overhead at every hierarchy, bottom necks issue at upper level caches making high client latency, and the wastes of the proxy cache space.

Different from the hierarchical model, the distributed caching architecture has no intermediate caches. When a request is issued, the content search will be performed over the distributed caches. The distributed caching has very good performance in cooperative proxy caching unless its implementation encounters several problems, such as having complicated network system, high bandwidth usage, and administrative issues.

From the problems mentioned earlier, this paper proposed a new architecture of cache farm integrated with the concepts of the recommender system to enhance or maintain the QoS when the number of users increases. This system performs all necessary process in transactions' classification and management based on the interested areas of end users. Although, URL-based classification is available, it should be desirable to classify based on user behavior mentioned in [6]. Therefore, if a suitable architecture and an appropriate algorithm are developed without adding any hardware, it is said to be an economical system that supports the required QoS when the number of users is large and continuously increases [7].

The remainder of this paper is organized as follows. In the next section, a survey of the related works is presented. Then, the performance problem that is considered to be improved of the browsing process is addressed in Section 3. The proposed system architecture for cache farming integrated with the concepts of the recommender system is described in Section 4. Finally, discussion and conclusions are given in Section 5 and 6, respectively.

2. Related Works

Since the number of clients linked to a single cache continuously increases, making the amount of information presented in cache rapidly increases, the caching performance must be dropped in a certain amount of time. To serve the larger amount of clients with only one proxy server, most of researches proposed a new caching algorithm, e.g., [1-3, 8], to manage various requests from clients, or analyzed the factors affecting to the caching performance with existing well-known caching strategies [9], to suggest ways for improving the caching performance. Since the cache hit ratio can be increased significantly by sharing the interests of a larger area [4], several caches can cooperate to increase the effective client population using a cache.

Additionally, there are several works focusing on hierarchical and distributed web caching especially. In [10] has studied and analyzed the performance parameters of hierarchical and distributed caching such as request latency, hit rate, and bandwidth usage. This work also considers a hybrid caching architecture where caches cooperate at every level of a caching hierarchy using distributed caching. In [11] applied hierarchical web caching technology to the Content Distribution Network (CDN) architecture as well as investigate the potential performance gain. In [12] proposed distributed caching in a local network environment to reduce server load, and to relieve problems of scalability and reliability of the proxy caching. There are many research papers focusing on the performance evaluation of several replacement policies within the hierarchical web caching, e.g., [13, 14]. Otherwise, [15] proposed a new algorithm for hierarchical web caching which is a novel coordinated placement and replacement algorithm for hierarchical web caching.

Although very little research effort has been made on the study of fundamental design principles for hierarchical web caching, H. Che et al [16] aim to find that principles. Moreover, this research also proposed a cooperative hierarchical web caching architecture based on these principles to guide the caching algorithm design. Moreover, there are researches aim to investigate whether caching a web replica in all intermediate caches on the reverse path is a good idea. In [17] has studied the performance of various meta algorithms that are responsible for deciding whether a new document will be accepted in a cache. Finally, [18] proposed a new hierarchical web caching scheme by using iSCSI protocol which provides more improved performance than existing web caching scheme.

About the recommender systems, the collaborative method becomes the most popular method even though it is limited to certain conditions. Many solutions of this problem have been researched by trying to combine the collaborative method with others. Combining collaborative filtering with personal information filtering agents is proposed in [19]. This approach aids users to avoid selecting any recommendation among agents. The users can use all of agents, and let the collaborative filtering framework selecting the best ones for them. In [20] proposed an objective oriented content based and collaborative recommender system to find the most relevant web pages for the current user's objective. Additionally, item-to-item collaborative filtering, an advance recommendation algorithm focuses on finding similar items, not similar customers, used by Amazon.com is discussed in [21]. Mentioned in [22] is an introduction of the agent-based information recommending system based on the statistical information for personal user. The last proposal for recommender system is using a hybrid system based on the Intelligent Neighbor Formation Algorithm (INFA) which uses an automated mechanism to update the user's preference details into the user profile whenever a user enters the preference details, and the Modified Naïve Bayes Theorem (MNBT) that has improved to support the situation of no opinion from neighbors, or no existing neighbors [23].

According to the efficiency of the recommender system that narrows the search results and, probably, reduces the delay time, this technique will be applied to manage the search in a new cache farming architecture. The expected results of proposed architecture are to reduce user request latency, increase the cache hit ratio, and assist the maintained cost.

3. Stated of Problems

Since there are various methods to perform cache management, the final solution to maintain QoS of all Internet service providers (ISPs) and other organizations when the number of users is enlarged is to expand the size of cache from time to time. This solution is not an optimized solution because there is no boundary for cache expansion for unlimited time interval. Additionally, adding more caches cause the complexity of the system as well as increasing the cost of maintenance.

Since the conclusion from [9] stated that the performance of a browse that is a part of QoS does not related to the types of browsed files under all existing cache management algorithms mentioned in Section 2. Thus, it is possible that the performance of the browsing process might depend on types of files that are proper classified and managed in the cache farm.

This paper will focus in the situation that the number of users increases and the ISPs would like to obtain real optimize solution for cache farming that satisfy the QoS without changing the cache management algorithm of cache itself. Therefore, this paper proposed a new architecture to manage caches in the farm that can maintain the QoS without expanding sizes of caches when the number of users increases by applying the concepts of recommender system.

The modified recommender system proposed in this paper will consider content of browsed files and apply the content to classify users' interests. Moreover, the proposed architecture is dynamic according to the time interval and users' requests.

4. Proposed Architecture

Generally, most ISPs arrange their cache farm based on types of their customers (end users who request the web document). Although the cache management policies are automatically implemented into the cache box using various methods, there is no guarantee that most required data will be stored to serve users as needed, especially when the volume of browsing webs is expanded. Therefore, each time the number of customers' usages continuously increases, the cache sizes must be increased to meet their service level agreements.

In order to maintain the available cache size in the farm to serve an increasing usage from users, this paper proposed a new architecture, called as Intelligent Cache Farming Architecture (ICFA), for cache farming that guarantees their services as requested. Figure 1 shows the proposed architecture that consists of gateway system, user profile database system, main proxy manager, and special proxy servers.

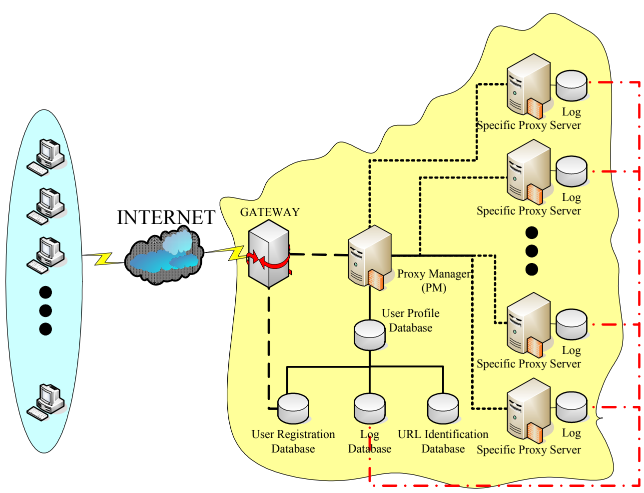

Fig. 1. Intelligent Cache Farming Architecture.

Referring to Fig. 1, ICFA consists of four main systems. The first system is the gateway system where every request from the Internet will be basically classified to the proxy system inside the farm cache. The second system is the user profiles database that consists of three sub-database systems: user registration database, log database, and URL identification database. The third system that is important system in this architecture is called a Proxy Manager (PM). This system is responsible for classifying all requests from users of the entire organization, or ISPs' customers. The last system consists of various proxy servers; each server is responsible for specific requirements, or URLs' types. Each system will be described as follows.

4.1. Gateway system

The gateway is responsible to classify all arrived transactions from the Internet to a suitable proxy server according to the class of users who delivered the transaction. If the request transaction can be classified in a specific proxy server then the request will be sent directly to the specific proxy server that links to the proxy manager (PM) system, as shown in Fig. 1. Otherwise, the request will be sent to the PM to make a decision to which proxy server to send.

The gateway can classify the request using data in the user registration database (URDB). Since a function in PM will responsible for user classification, the result of this classification will be kept in the URDB for references. However, the gateway can use data in the URDB in the read-only mode only.

4.2. User profile database system

As mentioned previously that the user profiles database system (UPDB) consists of three sub-databases: URL identification database (UIDB), user registration database (URDB), and log database (ULDB).

Generally, every web site must be identified its type using the postfix in the URL. For examples, .org represents all organizations, .com represents commercial organizations, .gov represents government organization, etc. However, using only the postfix to identify types of webs is not enough. Thus, contents under those web sites are used to determine specific type of each web. For example, http://doi.acm.org/10.1145/332040.332491 must be classified as a document under the academic organization, while http://amazon.com must be identified as an e-commerce web site, etc. The UIDB is a database that stores both web sites' names and specific types of those webs. This database is used with other information to identify the user's appreciation by the automatic classification module (ACM) in PM. The URDB contains general information of each user, including the rating of user's appreciation in web categories. This rating is used to identify the user whenever a request of the user arrives at the PM. The scale for rating the appreciation of each web category is running from 1 to 5; the maximum 5 refers to the most preferences, and 1 refers to the lowest interest.

Consider the situation that a request is issued from a client. This request will be copied into the ULDB, so the system can be recovered when a failure or an unexpected event occurs. However, records in the log can be used to identify the user's interesting area, or groups of web sites, using frequency of each browsed web as an indicator. For example, if webs in group A, B, and C were browsed 14, 46, and 20 times per day respectively for a user X, then these numbers indicate that user X is interested in webs under the group B because webs in the group B were frequently browsed.

The records in the ULDB are not obtained from only the retrieved command from the ACM but also 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 the offline mode and performed only before the RAM starts its evaluation.

4.3. Proxy manager

Proxy Manager (PM) is an assigned proxy server in the cache farm, responsible in classifying users into groups of users depending on user's interests. The main procedure of PM is applied from the method of recommender system; the module performs such task is called the automatic classification module (ACM). All requests that PM received from the gateway must be classified by the ACM in order to monitor the user's behavior and also keeps records in the UPDB attached to the PM. Figure 2 demonstrates the PM's architecture to perform all tasks. Each module in Fig. 2 is described as follow.

4.3.1. PM interface

Referring to Fig. 2, the PM interface (PMI) receives the requests from the gateway and sends all of them to ACM to classify users into group of user's interest. The input data received from the gateway will be rearranged in a suitable format and send as parameters to the ACM afterwards. After receiving requests from the gateway, the only message that PMI will return to the gateway is the acknowledgement.

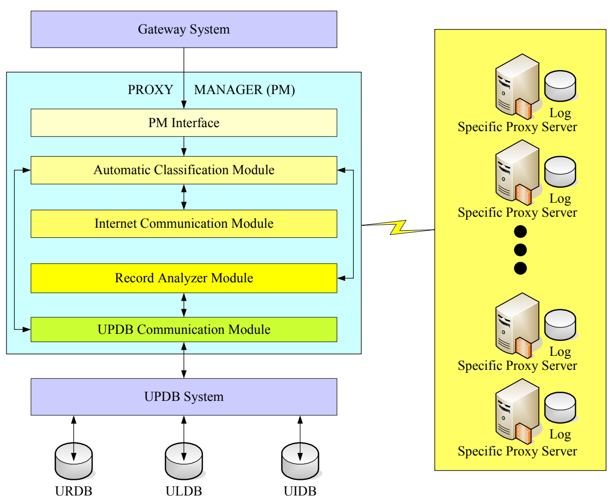

Fig. 2. Proxy Manager's Architecture in the Intelligent Cache Farming Architecture.

4.3.2. Automatic classification module

This function is responsible for classifying the Internet users and is called by PM. As the fact that every new user will have no record in the log database of their first accessing time, therefore, the ACM function will not classify the user unless they have used the system for a certain period of time. This period is called the precaution period because the ACM will be aware not to classify users until the real characteristics of users have shown out.

The concept of this module is modified from the recommender system where suitable objects are proposed to users in a short period of time. As the fact that there are two basic approaches of the recommender systems: content based approach, and collaborative filtering approach [24]. Thus, the ACM will integrate these two approaches to identify types of users as described in the following paragraphs.

During the precaution period, all requests of users will be directly send to the UPDB to identify the group of the requested web. After the URL was classified its group, the transmission of URL will go to the specific proxy server (SPS) of that group through the Internet communication module. However, the record of using each SPS will be stored in the log database of the UPDB for future process of the Record Analyzer Module (RAM).

After the precaution period, all users will have log database. As the fact that there are various types of users who browse webs or access to the Internet, therefore, applying the collaborative filtering approach of the recommender system, users can be grouped into three types. The first group refers to users who have no direction in browsing webs. This situation occurs in some places, such as Internet Café or Starbuck Internet, etc; in these locations, each customer gains the Internet access to many webs according to customer's interest. The second group can be classified from the customers of Internet subscribers of the home network. Consider the Internet services for a home network. It is possible that there are various interested groups of webs existing under one subscriber. Since people in the same house who have different interested areas always share the Internet link under the same subscription from an ISP, thus the interested groups of the Internet access will be more than one group under this Internet subscriber. The third group, the last group, can be considered from any particular organizations, such as financial organizations, import/export organizations, etc. These organizations always have a clear picture of information to be used from the Internet. Thus, the browsed information will drop into a particular group of interest without overlapping to others.

Consider the users in the first group whose browsed webs are undirected. All requests delivered from this user will be handled as same as the users in the precaution period because of no direction can be classified. Similarly to users in this group, the users who are in the second group also be difficult to be classified.

For users in the third group, the modification of the recommender system is applied using the content-based approach with the rating data of user. Basically, there are two situations to be considered before making decision to classify; both situations relate to the rating data of users in the URDB. Since users must identify their interest during the registration period, this value must be consistent with the browsed group of users during the precaution time. However, in some situations, the browsed locations of webs were not fitted into the groups stated in the URDB of users. Thus, the results from the RAM are inconsistent with the rating value in the URDB. The other situation is quite straightforward; the requested webs are in the same category as the rating group in the URDB.

In the case that the downloaded webs are inconsistent with the rating groups, the administrator will use this analytical result to perform a re-rating process with the user. The user must choose to either change the original rated group, or remain the original rated group. The algorithm of the ACM that covers all cases is shown in Fig. 3.

Automatic Classification Module (ACM) Algorithm

```
INPUT: User's requests (IP, URL, ...), Log Database (ULDB),
          User Registration Database (URDB),
          URL Identification Database (UIDB)
1. BEGIN
2. SET UR        User's request
3. IF (user is in the precaution period) THEN
4.     Send UR to a identified SPS directly
5. ELSE IF (user is not in the precaution period) THEN
6.     IF (user is in the first group OR
7.            user is in the second group) THEN
8.              Send UR to a identified SPS directly
9.     ELSE IF (user is in the third group) THEN
10.          CASE comparing browsing webs to rating OF
11.              Consistence: Send UR to a corresponding SPS
12.              Inconsistence: Perform a re-rating process with the user
13.                     CASE a re-rating process OF
14.                         Changing: Send UR to a corresponding SPS
15.                         Unchanging: Go to Step 6.
16.                     END CASE
17.          END CASE
18.     END IF
19. END IF
20. END
OUTPUT: Group of user's interest
```

Fig. 3. Automatic Classification Module Algorithm.

4.3.3. Internet Communication module

After ACM classified the group of the requested URL, the request must be sent to the Internet Communication Module (ICM). This ICM is responsible for sending queries from PM to SPSs according to the classification results of ACM. The connection between this module and a SPS is the connection-oriented using TCP/IP. Moreover, the format of the sending message is the same format of the packet received from the gateway system. The communications from the ICM to any SPSs are one-way direction because every SPS will directly be responsible to return the requested page to users.

4.3.4. Record Analyzer module

This module reads records in the log database to analyze users' behavior in order to identify group of interest for the users. The RAM generally performs its task every constant period, such as every 3 months or 6 months, etc. The process of RAM is the offline process, so it will not interfere with normal process of PM. The result from the RAM process is the identification of a group that a user belongs. However, if there is no relation among browsed information, RAM will not identify group of interest for the user and leave that user as an undefined group.

Since each SPS has its local log database, these records will be merged into the ULDB and be analyzed to find the usage ratio of every group of interests. Thus, in every round that RAM runs its process, the ratio of interest may change; nevertheless, this change may not be necessary affect to the change of the grouping in the cache farm.

4.3.5. UPDB communication module

All processes of PM must use data from the UPDB system that consists of three databases mentioned previously. Therefore, the UPDB communication module of PM is responsible for making a connection to the UPDB system to retrieve the required data. The connection performed by this module is the connection-oriented type using TCP/IP because the transmitted data cannot be lost.

4.4. Specific proxy servers

Specific proxy servers are groups of caches in the farm that serve users' needs. Additionally, these groups must be the same as groups of webs defined in the UIDB. The rule to install a cache for serving interested groups is that a cache can serve more than one interested group but not vice versa.

According to the result of RAM, ratios of web usages among groups can be calculated. This ratio can be applied to setup the size of each cache in the farm to maximize the services without expanding the cache size for the long term use. However, ratios of caches can also be changed when the time past. Thus, the change of each cache can be re-determined using the result of the RAM process mentioned in the previous section.

After sending records in its local log database to the ULDB, the SPS will reset its log database and start recording new uses for next evaluation time.

5. Discussions

Since the number of Internet users is continuously increasing, the issue of QoS is an important concern of users. Every ISP generally groups their customers based on their applications or organizational profiles. However, this criterion does not fit well for cache management to serve users when the size of users is expanded. Therefore, various cache management algorithms are implemented to maintain retrieval process of users over the Internet environment.

In this paper, it has shown that the proposed architecture, ICFA, is easy to manage and reduce the retrieval time comparing with the hierarchical and distributed cache systems. The distributed web proxy caching, e.g., [5, 12, 25], employs sophisticated caching and searching schemes to distribute and search the cached web documents. Using sophisticated caching scheme increases the complexity of proxy's management, while the proposed architecture is designed as one layer of distributed architecture with one management scheme, PM. Therefore, the complexity of ICFA is less than the existing distributed web proxy caching.

Considering the hierarchical caching architecture, e.g., [11, 16], this architecture needs intermediate caches with the high-quality algorithms to avoid vastly loading in the caches that will result in high retrieval time. However, this concept does not support economical issue when the number of usages increases. Therefore, ICFA has many advantages than the hierarchical system because of the function of recommending system. According to efficiency of the recommending function, the increasing number of users will not affect to the retrieval time. Thus, there is no obligatory to expand size of available caches of ICFA although the number of users is increasing. Furthermore, there is no change in the retrieval time under such situation.

Referring to the efficiency of the recommender system, users can save time to obtain their required files from the file repository or database. Therefore, applying the concepts of recommender system, called a modified recommender system, for classifying groups of interests and users' requests in cache management policy will also decrease the retrieval time from the cache farm without effects from the implemented cache management algorithm. However, according to the conclusion of [9] that the best cache management algorithm is GDS, so implementing technique with this cache algorithm would increase the performance of the service to be in a high acceptable rate of users.

Comparing the proposed architecture with general cache farm architecture, the proposed architecture arranged number of caches based on the number of interesting groups and also the ratio of these existing groups while most of the cache architecture do not consider in the existing usage ratio among usage contents, e.g., [11, 12, 15, 17]. Thus, the adaptation of cache sizes will perform only within the available caches in the farm, based on the result from RAM. Unlike the proposed cache farming technique, the existing cache farm will be re-implemented or add more cache size for every increasing number of users. Therefore, the proposed technique can save the maintenance cost in buying new hardware and time to install new cache into the legacy system.

6. Conclusions

Information retrieval over the Internet through the Internet Service Providers (ISPs) is increased dramatically. All users expect a high QoS from their ISPs for every request delivered from the client systems. The general solution for every ISP is to implement a cache farm to manage all requests for web browsing. This technique is implemented for decades to increase the performance of the services and also filter suitable packet in and out the network system.

One major problem of the ISPs is the unlimited increasing size of their customers where this can affect to the performance of the entire service system when cache management mechanism cannot serve all requests as suitable as it should. Various transactions must be delivered out to the Internet to retrieve information from the original source where some can find contents in the cache area. Thus, the response time for users will not be fast as expected, or sometimes, the transaction has gone down according to the congestion traffic of the Internet. So, the QoS of ISP can be dropped according to users' disappointments. Many cache management techniques were proposed to solve the problems stated above. Unfortunately, none of them can control the size of cache when the number of users is increasing.

This paper proposed the Intelligent Cache Farming Architecture, ICFA, of a cache farm where every transaction must pass to the gateway for transaction classification. The main proxy, called a PM, will manage all un-classified users before being classified by RAM using the modified recommender system. The modified recommender system is an adaptive concept of the recommender system responsible for classifying groups of interests based on the browsed content captured in the log database of the entire service system. Furthermore, the new cache farm architecture is presented with various management modules to control all requests and also classes of interests. Therefore, the response time is reduced as needed.

Since the updating of these interested groups will be performed in a certain period of time using the result from the RAM. Then, the ratio of each cache size will be changed within the farm without adding external cache. Thus, the maintenance cost can be controlled when the number of users increases.

References

1. Cao, P.; and Irani, S. (1997). Cost-aware WWW proxy caching algorithms. *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, Monterey, CA, 193-206.
2. Williams, S.; Abrams, M.; Standridge, C.R.; Abdulla, G.; and Fox, E.A. (1996). Removal policies in network caches for world-wide web documents. *Proceedings of ACM SIGCOMM*, CA, 293-305.
3. O'Neil, E.J.; O'Neil, P.E.; and Weikum, G. (1993). The LRU-k page replacement algorithm for database disk buffering. *Proceedings of ACM SIGMOD Record*, 22(2), 297-306.
4. Chankhunthod, A.; Danzig, P.B; Chuck Neerdaels; Schwartz, M.F.; and Worrell, K.J. (1996). A hierarchical internet object cache. *Proceedings of the 1996 annual conference on USENIX Annual Technical Conference.*
5. Tewari, R.; Dahlin, M.; Vin, H.M.; and Kay, J.S. (1999). Beyond hierarchies: Design considerations for disturbed caching on the Internet. *Proceedings of the 19th International Conference on Distributed Computing Systems*, Austin, TX.
6. Bhattarakosol, P.; and Ngamaramvaranggul, V. (2004). An Internet web management policy for government organization. *Proceedings of the International Conference on APAN*, Cairns, Australia, 249-255.

7. Bhattarakosol, P.; and Srisujjalertwaja, W. (2004). Customer-oriented policy for proxy management system. *Proceedings of International Computer Symposium*, Taipei, Taiwan, 1168-1173.
8. Haverkort, B.R.; Abdouni Khayari, R.E.; and Sadre, R. (2003). A class-based least-recently used caching algorithm for world-wide web proxies. *Lecture Notes in Computer Science* [Online]. 2794(2003), 273–290. Available: http://www.springerlink.com/content/7khpur60hn9q2npp/.
9. Khayari, R.E.; Best, M.; and Lehmann, A. (2005). Impact of document types on the performance of caching algorithms in WWW proxies: A trace driven simulation study. *Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, Vol. 1, 737-742.
10. Rodriguez, P.; Spanner, C.; and Biersack, E.W. (2001). Analysis of web caching architectures: hierarchical and distributed caching. *IEEE/ACM Transactions on Networking*, 9(4), 404-418.
11. Yang, Fu-Hong; and Chi, Chi-Hung. (2007). Using hierarchical scheme and caching techniques for content distribution networks. *Proceedings of the Third International Conference on Semantics, Knowledge and Grid*, Xi'an, China, 535-538.
12. Piatek, M. (2004). Distributed web proxy caching in a local network environment. *The Student Research Competition*, Available: www.acm.org/src/subpages/papers/piatek.src.2004.pdf.
13. Williamson, C. (2002). On filter effects in web caching hierarchies. *ACM Transactions on Internet Technology*, 2(1), 47–77.
14. Fu, H. ; Au, Pui-On ; and Jia, W. (2004). Performance evaluations of replacement algorithms in hierarchical web caching. *Proceedings of the 5th International Conference on Advances in Web-Age Information Management*, Springer LNCS 3129, 176–185.
15. Li, W.; Wu, K.; Ping, X.; Tao, Y.; Lu, S.; and Chen, D. (2005). Coordinated placement and replacement for grid-based hierarchical web caches. *Proceedings of the 4th International Conference on Grid and Cooperative Computing*, Springer LNCS 3795, 430–435.
16. Che, H.; Wang, Z.; and Tung, Y. (2001). Analysis and design of hierarchical web caching systems. *Proceedings of the IEEE INFOCOM 2001*, Alaska, USA, Vol. 3, 1416-1424.
17. Laoutaris, N.; Syntila, S.; and Stavrakakis, I. (2004). Meta algorithms for hierarchical web caches. *IEEE International Conference on Performance, Computing, and Communications*, Arizona, USA, 445-452.
18. Lim, H.; and Du, D.H.C. (2003). Design considerations for hierarchical Web proxy server using iSCSI. *Proceedings of the 2003 Symposium on Applications and the Internet*, FL, USA, 414-417.
19. Good, N.; Schafer, J.B.; Konstan, J.A.; Borchers, A.; Sarwar, B.; Herlocker, J.; and Riedl, J. (1999). Combining collaborative filtering with personal agents for better recommendations. *Proceedings of the 16th national conference on Artificial intelligence and the 11th Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, Florida, 439–446.

20. Bueno, D.; Conejo, R.; and David, A.A. (2002). METIOREW: An Objective oriented content based and collaborative recommending system. *Lecture Notes in Computer Science* [Online], 2266(2002), 310–314. Available: http://www.springerlink.com/content/2rea65fqq0gk0w04/.
21. Linden, G.; Smith, B.; and York, J. (2003). Amazon.com recommendations item-to-item collaborative filtering. *IEEE Internet Computing* [Online], 7(1), 76–80. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1167344.
22. Abe, K.; Taketa, T.; and Nunokawa, H. (2000). An Idea of the agent-based information recommending system using the statistical information. *Proceedings of the 7th International Conference on Parallel and Distributed Systems: Workshops*, Iwate, Japan, 143–146.
23. Maneeroj, S.; and Bhattarakosol, P. (2006). Hybrid system based on intelligent neighbor formation algorithm. *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, Hong Kong, 765-768.
24. Balabanov´ıc, M.; and Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3), 66–72.
25. Touch, J. (1998). The LSAM Proxy Cache – a Multicast distributed virtual cache. *Proceedings of the 3rd International WWW Caching Workshop*, Manchester, England.