INTELLIGENT CACHE FARMING ARCHITECTURE FOR E-BUSINESS SERVICES

Supawadee Hiranpongsin[1,2] *and Pattarasinee Bhattarakosol*[1,3]

[1]Department of Mathematics, Faculty of Science,
Chulalongkorn University, Bangkok, 10330, Thailand.
[2]supawadee.h@student.chula.ac.th, [3]pattarasinee.b@chula.ac.th

ABSTRACT

One class of the ISP's customers is the group of e-business companies and important this group is providing the fast access for their customers. The standard method that ISPs used to serve their customers for obtaining high performance is the implementation of cache farming with the standard management policy. Unfortunately, the size of cache must be varied based on the size of customers in order to maintain customer satisfaction. This paper proposes a novel architecture, called as Intelligent Cache Farming Architecture, with the concepts of recommender system to enhance or maintain the quality of service (QoS) over the Internet business. This proposed architecture supports fast information retrieval for e-business customers. Additionally, with this recommender system undirected search will be reasonably managed.

Index Terms— Cache farming, cache management technique, recommender system, customer's requirement

1. INTRODUCTION

When a web site becomes overload, customers constantly grow with long waits and rejected requests. This situation leads Internet businesses in each year to lose revenues because of slow and failed Internet services [1]. Therefore, many companies, such as Amazon.com, the book sellers, etc., have rushed to provide efficient technologies for serving the bigger load from their customers. In order to offer and maintain the quality of services (QoS) of those companies, cache technique is a standard technology which stores the web objects for later retrieval. Although, web caching is not a new technique on the Internet services, it can help the companies to recapture those lost revenues. This mechanism helps reducing the information retrieval, and also performs security assurance for the installed organizations. Moreover, web caching which is originated to help companies avoids the need to repeatedly access slow data storage devices in standalone host computers. However, the performance of web caching is depended on two factors: the cache size of proxy server, and the cache management policy.

Currently, there are many reasons for going online to do business online, such as buying, selling, servicing customers, and collaborating with business partners, which are some activities of the e-business system [2], [3]. In order to serve customers' requirement in the e-business system, the caching popular web documents at proxy server is employed. As the mention above, this standard technique can improve the QoS over the Internet business. Therefore the web caching is applied into the e-business model to enhance the performance of doing business online.

Initially, web proxy caching is focused on a single proxy server [4], [5]. 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 increase 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 [6] and a distributed [7] 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 novel architecture of cache farming integrated with the concepts of the recommender system to enhance or maintain the QoS when the number of customers increases

This system performs all necessary process in transactions' classification and management based on the interested areas of customers. Although, URL-based classification is available, it should be desirable to classify based on customer behavior mentioned in [8], [9]. 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 customers is large and continuously increases [10].

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

To serve the larger amount of clients with only one proxy server, most of researches proposed a new caching algorithm, e.g., [4], [5], [11], to manage various requests from clients, or analyzed the factors affecting to the caching performance with existing well-known caching strategies [12], 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 [6], 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 [13] 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 [14] applied hierarchical web caching technology to the Content Distribution Network (CDN) architecture as well as investigate the potential performance gain. In [15] 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., [16], [17]. Otherwise, [18] 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, Z. Wang, and Y. Tung [19] aim to find that principles. Moreover, this research also proposed 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 [20] defines a framework for building and evolving customer networks supporting Internet business solutions. In [21] has studied the performance of various meta algorithms that are responsible for deciding whether a new document will be accepted in a cache. Finally, [22] 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 [23]. 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 [24] 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 [25]. Mentioned in [26] 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 Naive Bayes Theorem (MNBT) that has improved to support the situation of no opinion from neighbors, or no existing neighbors [27].

According to the efficiency of the recommender system that narrows the search results and, probably, reduces the retrieval time, this technique will be applied to manage the search in a novel cache farming architecture. The expected results of proposed architecture are to reduce the retrieval time and to assist the maintained cost.

3. STATED OF PROBLEMS

Since ISPs have various types of customers, one of those customers is classified as the e-business group which has many areas of interests. The important issue in serving this group is providing the fast access for their customers. However, users over the Internet may not have a certain objective in searching for a product. Therefore, providing a recommending system as a part of the ISP services would increase the service value of the ISP itself. Moreover, when the accesses to this e-business group are increasing, maintaining the response to their customers without increasing the ISP investment cost is an interesting problem that will be focused in this paper.

Since the conclusion from [12] 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. Moreover, [9] demonstrates that proposed technique based on usage pattern improves the adaptability of prediction rules comparing with the existing other methods. Therefore, the usage pattern or behavior of customers can be considered as a factor for analyzing the performance of cache management.

This paper proposed a novel architecture to manage caches in the farm that can maintain the QoS without expanding sizes of caches when the number of customers increases by applying the concepts of recommender system. Additionally, the modified recommender system proposed in this paper will consider content of browsed files and apply the content to classify customers' interests to support the requests and growth of the e-business.

4. PROPOSED ARCHITECTURE

Normally, the ISPs provide the service for the e-business organization by giving the storage and customer access to their Internet link. Considering the data from the e-business organization, this can be business profiles, product advertisements, business member club, etc. This data is mainly presented in the form of web pages. Thus, every e-business organization must have their own web sites stored in the rented storages of the ISPs. Therefore, managing this storage related to the cache farm can increase the performance of serving to their customers.

Generally, most ISPs arrange their cache farm based on types of their customers 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 customers 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 customers, this paper proposed a novel 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, customer profile, database system, main proxy manager, and special proxy servers.

Referring to Figure 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 cache farm. The second system is the customer profile database that consists of three sub-database systems: customer 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 customers of the entire organization, or ISPs' customers focusing on the e-business. The last system consists of various proxy servers; each server is responsible for specific e-business groups. Each system will be described as follows.

4.1. Gateway

The gateway is responsible to classify all arrived transactions from the Internet to a suitable e-business server according to the class of customers 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 e-business server that links to the PM system, as shown in Figure 1. Otherwise, the request will be sent to the PM to make a decision to which e-business server to send.

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

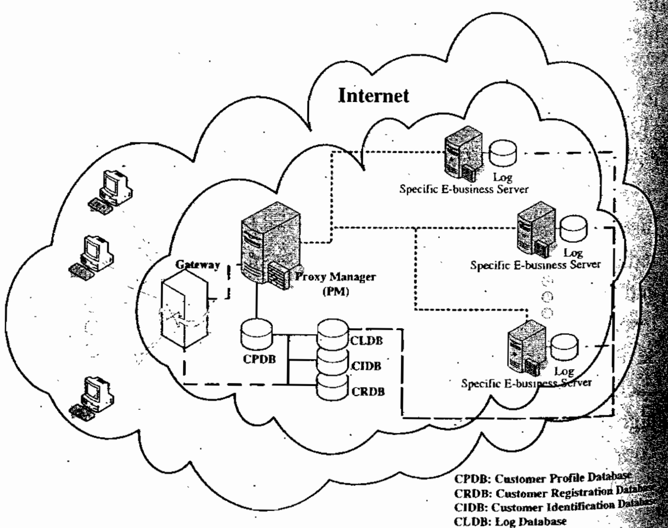

Figure 1. Intelligent Cache Farming Architecture.

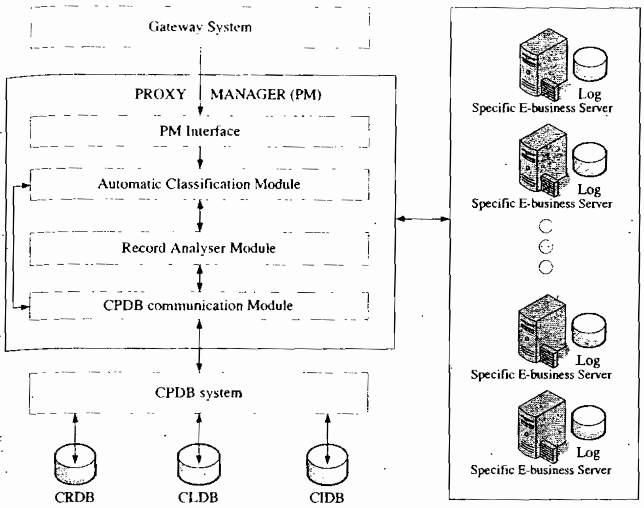

Figure 2. The proxy manager architecture in the ICFA.

4.2. CPDB: Customer profile database system

As mentioned previously that the CPDB system consists of three sub-databases: URL identification database (CIDB), customer registration database (CRDB), and log database (CLDB).

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 CIDB 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 customer's appreciation by the automatic classification module (ACM) in PM. The CRDB contains general information of each customer, including the rating of customer's appreciation in web categories. This rating is used to identify the customer whenever a request of the customer 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 customer. This request will be copied into the CLDB, 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 customer's interesting area, or groups of e-business, using frequency of each browsed web as an indicator. For example, if webs in group I, II, and III were browsed 14, 46, and 22 times per day respectively for a customer C, then these numbers indicate that customer C is interested in webs under the group II because webs in the group II were frequently browsed.

The records in the CLDB are not obtained from only the retrieved command from the ACM but also the retrieved commands recorded in the local log database of each specific e-business server (SES), or caches. The transferring of data from all local log databases from each SES is the offline mode and performed only before the RAM starts its evaluation.

4.3. PM: Proxy manager

The PM is an assigned proxy server in the cache farm, responsible in classifying customers into groups of customers depending on customer'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 customer's behavior and also keeps records in the CPDB attached to the PM. Figure 2 demonstrates the PM's architecture to perform all tasks. Each module in Figure 2 is described as follow.

4.3.1. PMI: PM interface

Referring to Figure 2, the PMI receives the requests from the gateway and sends all of them to ACM to classify customers into group of customer'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.

4.3.2. ACM: Automatic classification module

This function is responsible for classifying the Internet customers and is called by PM. As the fact that every new customer will have no record in the log database of their first accessing time, therefore, the ACM function will not classify the customer 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 customers until the real characteristics of customers have shown out.

The concept of this module is modified from the recommender system where suitable objects are proposed to customers in a short period of time. As the fact that there are two basic approaches of the recommender systems: content-based filtering approach and collaborative filtering approach [28]. Thus, the ACM will integrate these two approaches to identify types of customers as described in the following paragraphs.

During the precaution period, all requests of customers will be directly sent to the CPDB to identify the group of the requested web. After the URL was classified its group, the transmission of URL will go to the SES of that group. However, the record of using each SES will be stored in the log database of the CPDB for future process of the record analyzer module (RAM).

After the precaution period, all customers will have log database. As the fact that there are various types of customers who browse webs or access to the Internet, therefore, applying the collaborative filtering approach of the recommender system which analyzes the CLDB's contents, customers can be grouped into two types. The first group refers to customers who have no direction in browsing webs. In this case, each customer accesses the Internet for doing business according to interested group of business. Another group, the second group, customers have the target of browsing webs. The customers always have an obvious searching over the Internet. Thus, the browsed information will drop into a particular group of interest without overlapping to others.

Consider the customers in the first group whose browsed webs are undirected. All requests delivered from these customers will be handled as same as the customers in the precaution period because of no direction can be classified. For the situation that the customers are not classified into any the specific e-business server, the contents of browsing web are compared to the existing information of the specific e-business server using content-based method of recommending system. After comparison, the list of the specific e-business server that has the closed information is shown.

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

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 customer. The report of this analytical result shown at user interface is one of the recommender system functions. When the user receives the report, the customer 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 Figure 3.

ACM Algorithm:

```
INPUT: Customer's requests (IP, URL, ...),
       Customer Registration Database (CRDB),
       URL Identification Database (CIDB),
       Log Database (CLDB)
1. BEGIN
2. SET CR←Customer's request
3. IF (customer is in the precaution period) THEN
4.    Send CR to a identified SES directly
5. ELSE IF (customer is not in the precaution period) THEN
6.    IF (customer is in the first group) THEN
7.          Send CR to a identified SES directly
8.    ELSE IF (customer is in the second group) THEN
9.          CASE comparing browsed group to customer's group rating OF
10.             Consistence: Send CR to a corresponding SES
11.             Inconsistence: Perform a re-rating process with the customer
12.                   CASE a re-rating process OF
13.                         Changing: Send CR to a corresponding SES
14.                         Unchanging: Go to Step 6.
15.                   END CASE
16.         END CASE
17.   END IF
18. END IF
19. END
OUTPUT: The e-business server
```

Figure 3. Automatic classification module algorithm.

4.3.3. RAM: Record analyzer module

This module reads records in the CLDB to analyze customers' behavior in order to identify group of interest for the customers. The RAM generally performs its task every constant period, such as every 3 months or 6 months, etc. to update the information in the CLDB. 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 customer belongs. However, if there is no relation among browsed information, RAM will not identify group of interest for the customer and leave that customer as an undefined group.

Since each SES has its local log database, these records will be merged into the CLDB 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.

4.3.4. CPDB communication module

All processes of PM must use data from the CPDB system that consists of three databases mentioned previously. Therefore, the CPDB communication module of PM is responsible for making a connection to the CPDB 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. SESs: Specific e-business servers

The SESs are groups of caches in the farm that serve customers' needs. Additionally, these groups must be the same as groups of webs defined in the CIDB. The rule to install a cache for serving interested groups is that a cache can serve more than one interested group but not vise 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 CLDB, the SES will reset its log database and start recording new uses for next evaluation time.

5. DISCUSSIONS

Since the number of Internet customers continuously increase, the issue of QoS is an important concern of those customers. Generally, every ISP or organization groups their customers based on their applications or organizational profiles. However, this criterion does not fit well for cache management to serve customers when the access from customers to the e-business group is expanded. Therefore, various cache management algorithms are implemented to maintain retrieval process of customers over the e-business 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., [7], [15], [29], employ 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 called as the PM. Therefore, the complexity of ICFA is less than the existing distributed web proxy caching.

Considering the hierarchical caching architecture, e.g., [14], [19], 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 customers will not affect to the retrieval time. Thus, there is no obligatory to expand size of available caches of ICFA although the number of customers is increasing. Furthermore, there is no change in the retrieval time under such situation.

Referring to the efficiency of the recommender system, customers 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 customers' requirements in cache management policy will also decrease the retrieval time from the cache farm without effects from the implemented cache management algorithm. Thus, the modified recommender system can solve the problem in the Internet business system. However, according to the conclusion of [12] 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 customers.

Comparing the proposed architecture, ICFA, with general cache farm architecture, the ICFA 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., [14], [15], [18], [21]. 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 customers. 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 ISPs or organizations increases dramatically. All customers, in this paper focus on the e-business group, 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 is called a PM will manage all un-classified customers 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 customers increases.

According to discussions, when the ICFA with the modified recommender system is applied into the e-business system, the Internet business can be improved the performance of Internet services with increasing the information retrieval time, reducing maintenance cost as well as avoiding administrative issues.

7. REFERENCES

[1] C. Kenyon, "The Evolution of Web-Caching markets," *Computer*, vol. 34, no. 11, pp. 128-130, November 2001.

[2] M. Singh, *A Primer on Developing An E-Business Strategy*, First Stop Business Information Center, Springfield, IL.

[3] P. Weill and M. Vitale, *Place to Space: Moving to eBusiness Models*, Boston: Harvard Business School Publishing Corporation, 2001.

[4] P. Cao and S. Irani, "Cost-aware WWW Proxy Caching Algorithms," in *Proc. on USENIX Symposium on Internet Technologies and Systems*, Monterey, CA, pp. 193-206, December 1997.

[5] S. Williams, M. Abrams, C. R. Standridge, G. Abdulla, and E. A. Fox, "Removal policies in network caches for world-wide web documents," in *Proc. ACM SIGCOMM*, Palo Alto, CA, pp. 293-305, August 1996.

[6] A. Chankhunthod et al., "A hierarchical internet object cache," in *Proc. 1996 USENIX Technical Conf.*, San Diego, CA, January 1996.

[7] R. Tewari, M. Dahlin, H. M. Vin, and J. S. Kay, "Beyond hierarchies: Design considerations for disturbed caching on the Internet," in *Proc. ICDCS '99 Conf.*, Austin, TX, May 1999.

[8] P. Bhattarakosol, and V. Ngamaramvaranggul, "An Internet web management policy for government organization," in *Proc. Network Research Workshop*, The 18th APAN Conference, Cairns, Australia, pp. 249-255, July 2004.

[9] L. Jianhui, H. Tianshu, and Y. Chao, "Research on WEB Cache Prediction Recommend Mechanism Based on Usage Pattern," in *Pro 1st Int. Workshop on Knowledge Discovery and Data Mining (WKDD 2008)*, Adelaide, Australia, pp. 473-476, January 2008.

[10] P. Bhattarakosol, and W. Srisujjalertwaja, "Customer-oriented policy for proxy management system," in *Proc. Int. Comput. Symp.*, Taipei, Taiwan, pp. 1168-1173, December 2004.

[11] B. R. Haverkort, R. El Abdouni Khayari, and R. Sadre. (2003, September, 18). A Class-based least-recently used caching algorithm for world-wide web proxies. *Lecture Notes in Comput. Sci. [Online]*. 2794(2003). pp. 273–290. Available: http://www.springerlink.com/content/7khpur60hn9q2npp/

[12] R. E. A. Khayari, M. Best, and A. Lehmann, "Impact of Document Types on the Performance of Caching Algorithms in WWW Proxies: A Trace Driven Simulation Study," in *Proc. 19th Int. Conf. on Advanced Inform. Networking and Applicat. (AINA '07)*, 2005.

[13] P. Rodriguez, C. Spanner, and E. W. Biersack, "Analysis of Web caching architectures: hierarchical and distributed caching," *IEEE/ACM Transactions on Networking*, vol. 9(4), pp. 404-418, August 2001.

[14] Fu-Hong Yang and Chi-Hung Chi, "Using Hierarchical Scheme and Caching Techniques for Content Distribution Networks," in *Proc. 3rd Int. Conf. on Semantics, Knowledge and Grid*, Xi'an, China, pp. 535-538, October 2007.

[15] M. Piatek, "Distributed web proxy caching in a local network environment," *The Student Research Competition (SRC2004)*, 2004. Available: www.acm.org/src/subpages/papers/piatek.src.2004.pdf

[16] C. Williamson, "On filter effects in web caching hierarchies," *ACM Transactions on Internet Technology*, vol. 2, no. 1, pp. 47–77, February 2002.

[17] H. Fu, Pui-On Au, and W. Jia, "Performance Evaluations of Replacement Algorithms in Hierarchical Web Caching," in *Proc. 5th Int. Conf. on Advances in Web-Age Information Management (WAIM2004)*, Springer LNCS 3129, 2004, pp. 176–185.

[18] W. Li, K. Wu, X. Ping, Y. Tao, S. Lu, and D. Chen, "Coordinated Placement and Replacement for Grid-Based Hierarchical Web Caches," in *Proc. 4th Int. Conf. on Grid and Cooperative Computing (GCC2005)*, Springer LNCS 3795, 2005, pp. 430–435.

[19] H. Che, Z. Wang, and Y. Tung, "Analysis and Design of Hierarchical Web Caching Systems," in *Proc. IEEE INFOCOM'01*, Alaska, USA, vol. 3, pp. 1416-1424, April 2001.

[20] Cisco Systems, Inc, "Cisco AVVID-The Architecture for E-Business," *White Paper*, 2001.

[21] N. Laoutaris, S. Syntila, and I. Stavrakakis, "Meta Algorithms for Hierarchical Web Caches," in *Proc. 2004 IEEE Int. Conf. on Performance, Computing, and Communications*, Arizona, USA, pp. 445-452, April 2004.

[22] H. Lim and D. H. C. Du, "Design considerations for hierarchical Web proxy server using iSCSI," in *Proc. 2003 Symposium on Applications and the Internet (SAINT'03)*, FL, USA, pp. 414-417, January 2003.

[23] N. Good, J. B. Schafer, J. A. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl, "Combining collaborative filtering with personal agents for better recommendations," in *Proc. 16th Nat. Conf. on Artificial Intell. and 11th Innovative Applicat. of Artificial Intell. Conf.*, Florida, 1999, pp. 439–446.

[24] D. Bueno, R. Conejo, and A. A. David. (2002, January, 1). METIOREW: An Objective oriented content based and collaborative recommending system. *Lecture Notes in Comput. Sci.* [Online]. *2266(2002).* pp. 310–314. Available: http://www.springerlink.com/content/2rea65fqq0gk0w04/

[25] G. Linden, B. Smith, and J. York. (2003, January/February). Amazon.com recommendations item-to-item collaborative filtering. *IEEE Internet Comput.* [Online]. *7(1).* pp. 76–80. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1167344

[26] K. Abe, T. Taketa, and H. Nunokawa, "An Idea of the agent-based information recommending system using the statistical information," in *Proc. 7th Int. Conf. on Parallel and Distributed Systems: Workshops*, Iwate, Japan, 2000, pp. 143–146.

[27] S. Maneeroj, and P. Bhattarakosol, "Hybrid system based on intelligent neighbor formation algorithm," in *Proc. 2006 IEEE/WIC/ACM Int. Conf. on Web Intell. (WI 2006 Main Conf. Proc.) (WI'06)*, Hong Kong, pp. 765-768, December 2006.

[28] M. Balabanov´ıc and Y. Shoham, "Fab: Content-based, collaborative recommendation," *Commun. of the ACM*, vol. 40(3), pp. 66–72, March 1997.

[29] J. Touch, "The LSAM Proxy Cache – a Multicast Distributed Virtual Cache," in *Proc. 3rd Int. WWW Caching Workshop*, Manchester, England, June 1998.