Multimedia Object Placement for Hybrid Transparent Data Replication*

Keqiu Li [1], Hong Shen [1,2], Francis Y. L. Chin [3], and Liusheng Huang[2]

[1]Graduate School of Information Science, Japan Advanced Institute of Science and Technology, Nomi, Ishikawa, Japan
[2] Department of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui, China
[3] Department of Computer Science and Information Systems, University of Hong Kong, Hong Kong, China

Abstract

In this paper, we address present an optimal solution for the problem of multimedia object placement for hybrid transparent data replication. The performance objective is to minimize the total access cost by considering both transmission cost and transcoding cost. The performance of the proposed solution is evaluated with a set of carefully designed simulation experiments for various performance metrics over a wide range of system parameters. The simulation results show that our solution consistently and significantly outperforms comparison solutions in terms of all the performance metrics considered.

1. Introduction

The World Wide Web has become the most successful application on the Internet since it provides a simple way to access a wide range of information and services. However, due to the dramatic growth in demand, considerable access latency is often experienced in retrieving web objects from the Internet, and popular web sites are suffering from overload. An efficient way to overcome such deficiencies is web caching, by which multiple copies of the same object are stored in geographically dispersed caches. As many mobile appliances are divergent in size, weight, I/O capabilities, network connectivity, and computing power, differentiated devices should be employed to satisfy their diverse requirements. In addition, different presentation preferences from users make this problem more serious. Transcoding, used to transform a multimedia object from one form to another, frequently through trading off object fidelity for size, is a technology that can meet these needs [5, 6, 9, 13].

Transparent data replication [7, 11, 12] has been advocated by both academic and industrial communities because of its low management overheads. Early studies on data replication [12] showed that considerable management overheads were incurred for identifying the optimal locations for the replica before each request is served. For transparent data replication, caches are placed transparently to both the servers and the clients. Each cache intercepts any request that passes through its associated node, and either satisfies the request by sending the requested object to the client or forwards it upstream along the path to the server until it can be satisfied. If the request is for a general object that can be viewed as a single-version multimedia, a copy of this requested object will be cached at each node that the request passes through when it is sent back to the client. When the request is for a specified version of a multimedia object, this method cannot be simply applied since it is not optimal to store the same version at all the nodes that the request passes through.

It can be easily validated that the placement of different versions of a multimedia object at different nodes has significant influence on network performance. Therefore, the study of the multimedia object placement strategies for transparent data replication has significant contribution on both theory and practice. In this paper, we address the problem of multimedia object placement for hybrid transparent data replication, i.e., determining exactly which version should be placed at each node such that the total access cost is minimized. In this paper, we first present a model for the problem of multimedia object placement for hybrid transparent data replication, formulated as an optimization problem. Second, we propose a dynamic programming-based solution to compute the optimal versions to be cached and give some analysis on the proposed solution. Finally, we evaluate our model on various performance metrics through extensive simulation experiments. The implementation results show that our solution consistently and significantly outperforms existing solutions.

The rest of this paper is organized as follows. In Section 2, we formulate the problem of multimedia object placement for hybrid transparent data replication. Section 3 presents an optimal solution for the formulated problem. The simulation model and performance evaluation are described in Sections 4 and 5, respectively. Section 6 summarizes our work and concludes the paper.

2. Problem Formulation

The network topology in this paper is modelled as a graph $G = (V, E)$, where $V = \{v_0, v_1, \cdots, v_n\}$ is the set of nodes or vertices, and E is the set of edges or links. For a multimedia object O, we assume that it has m versions: $A_1, A_2, \cdots, A_m$. For each version of object O, we

* This work was partially supported by Japan Society for the Promotion of Science (JSPS) under its General Research Scheme **B** Grant No. 14380139. Corresponding author: shen@jaist.ac.jp

associate each link $(u, v) \in E$ a nonnegative cost $L_k(u, v)$, which is defined as the cost of sending a request for version A_k and the relevant response over the link (u, v). In particular, $L_k(u, u) = 0$. If a request goes through multiple network links, the cost is the sum of the cost on all these links. The cost in our analysis is calculated from a general point of view. It can be different performance measures such as delay, bandwidth requirement, and access latency, or a combination of these measures. The transcoding cost of a multimedia object from A_i to A_j is denoted by $t(A_i, A_j)$. Obviously, $t(A_i, A_i) = 0$. If a version cannot be transcoded from another version, we consider the transcoding cost as infinity. $\Phi(A_i)$ is the set of all the versions that can be transcoded from A_i, including A_i. Let $f_{i,j}$ be access frequency of version A_j from node v_i.

Now we start to formulate the problem of multimedia object placement for hybrid transparent data replication (*MOP problem*). Consider the snapshot when a request for a specified version of a multimedia object is being served (see Fig. 1). Here v_0 denotes the content server which contains all versions of object O. v_n is the client and $v_1, v_2, \cdots, v_{n-1}$ are the nodes on the path from the client to the server. We can see that a request for a version of a multimedia object from a node can be satisfied either by this node or by upstream nodes (transcoding if necessary) until it arrives at the server at which no transcoding is necessary. Therefore, the total access cost can be decomposed into two parts: transcoding cost and transmission cost. Our objective is to find the exact version of a multimedia object to be placed at each node on the path from v_1 to v_n so that the total access cost is minimized. Note that all requests at node v_0 can be satisfied at zero cost.

If we denote A_{d_i} ($d_i \in \{1, 2, \cdots, m\}$) as the version cached at node v_i, then the total access cost of caching $A_{d_1}, A_{d_2}, \cdots, A_{d_n}$, denoted by $C(X)$, is defined as follows:

$$C(X) = \sum_{i=1}^{n} \sum_{j=1}^{m} f_{i,j} \min_{0 \le k \le i} \{L_j(v_i, v_k) + T(A_{d_k}, A_j)\} \tag{1}$$

where $X = (A_{d_1}, A_{d_2}, \cdots, A_{d_n})$ and $T(A_{d_k}, A_j) = \begin{cases} 0 & \text{if } k = 0 \\ t(A_{d_k}, A_j) & \text{if } k \neq 0 \end{cases}$.

Obviously, our objective is to obtain $X^* = (A_{d_1^*}, A_{d_2^*}, \cdots, A_{d_n^*})$ such that $C(X^*) = \min_{X}\{C(X)\}$.

Before we solve the MOP problem based on the cost function as given in Equation (1), we can make the following assumptions.

- *Assumption* 1. $L_j(v_i, v_k) = (i - k)L$ for all $1 \le j \le m$ as there are $i - k$ links on the path between nodes v_i and v_k, and the cost on each link for each version is L.
- *Assumption* 2. The transcoding graph is a linear array and the transcoding cost between any two adjacent versions is constant, i.e., $t(A_i, A_j) = \sum_{k=i}^{j-1} t(A_k, A_{k+1}) = (j - i)^+ T$, where $x^+ = x$ if $x \ge 0$ else $x^+ = \infty$.
- *Assumption* 3. $(\delta - 1)T \le L$, and $\delta T > L$ for some positive integer δ.

If there does not exist δ such that *Assumption* 3 can be satisfied, i.e., $L \gg T$ or $T \gg L$. Obviously, these are two special cases. If $L \gg T$, then version A_{d_i} should be cached so that no transmission cost is necessary to incur, where $d_i = \min\{j | f_{i,j} > 0, 1 \le j \le m\}$. If $T \gg L$, this case is not trivial and is equivalent to the en-route caching problem of caching m objects on a linear network of n nodes, where transcoding cost is prohibited.

With the above assumptions, the MOP problem can be simplified as follows:

$$C(X) = \sum_{i=1}^{n} \sum_{j=1}^{m} f_{i,j} \min\left\{ \min_{1 \le k \le i} \left\{(i - k)L + (j - d_k)^+ T\right\}, iL \right\} \tag{2}$$

3. A Dynamic Programming-Based Solution

In this section we present an optimal solution for the problem of multimedia object placement.

Assume that version A_j is cached at node v_i, and v_k is the smallest vertex, $k > i$, with a cached version, say A_z, more detailed than A_j, i.e, $z \le j$ (see Fig. 2). n Fig. 2, a square symbol at (d_i, i) indicates that version A_{d_i} is cached at node v_i and a dot indicates the request for a specified version from a node. Each node has exactly one such square symbol. A request for version A_j at node v_i might be either served at node v_i by version A_{d_i} if $j \ge d_i$ with transcoding cost $(j - d_i)T$, or at node v_{i-1} with additional transmission cost L. Assume that A_y is the most detailed version in Block $B_{i,j,k}$, which is cached at node v_x. Let $W(i, j, k)$ denote the minimum total access cost for serving all the requests in Block $B_{i,j,k}$. It is obvious that all the requests in Block C are served by version A_j at node v_i because the versions of all the requests in this block is more detailed than A_y, i.e. there does not exist a version in Block $B_{i,j,k}$ other than A_j that can provide the requested versions in this block since A_y is the most detailed version in Block $B_{i,j,k}$ besides A_j. Similarly, it is easy to see that the minimum total access cost for serving all the requests in Block A, i.e., $B_{x,y,k}$ and Block B, i.e., $B_{i,j,x+1}$ (see Fig. 2), is $W(x, y, k) + W(i, j, y-1)$. With a similar method for partitioning Block $B_{i,j,k}$, Blocks A and B can be divided recursively until the minimum total access cost for serving all the requests in each block, i.e., $W(x, y, k)$ and $W(i, j, y - 1)$, can be finally determined.

Based on the above observation, $W(i, j, k)$ is defined as follows:

$$W(i,j,k) = \begin{cases} \min\limits_{i \le x < k; j \le y \le m} \{W(i, j, y - 1) + W(x, y, k) \\ \quad + \sum\limits_{x \le \alpha < k; j \le \beta \le y} f_{\alpha,\beta}((\alpha - i)L + (\beta - j)T) \\ \quad (\text{for } \ 0 < i < k \le n; 1 \le j \le m) \\ \min\limits_{0 < x \le k; 1 \le y \le m} \{W(0, 1, y - 1) + W(x, y, k) \\ \quad + \sum\limits_{x \le \alpha < k; 1 \le \beta \le y} \beta L f_{\alpha,\beta} \\ \quad (\text{for } \ i = 0, j = 1) \\ 0 \quad (\text{for } \ i = k) \end{cases} \tag{3}$$

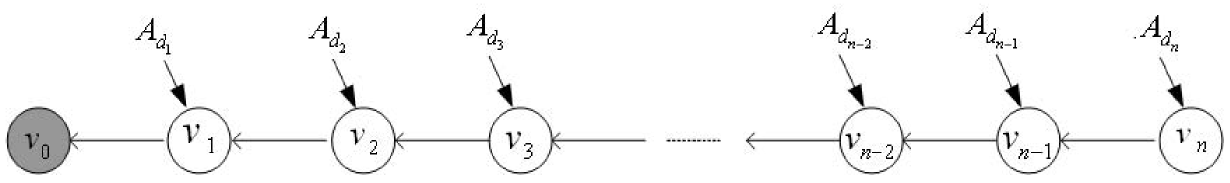

Figure 1. System Model for Multimedia Object Caching

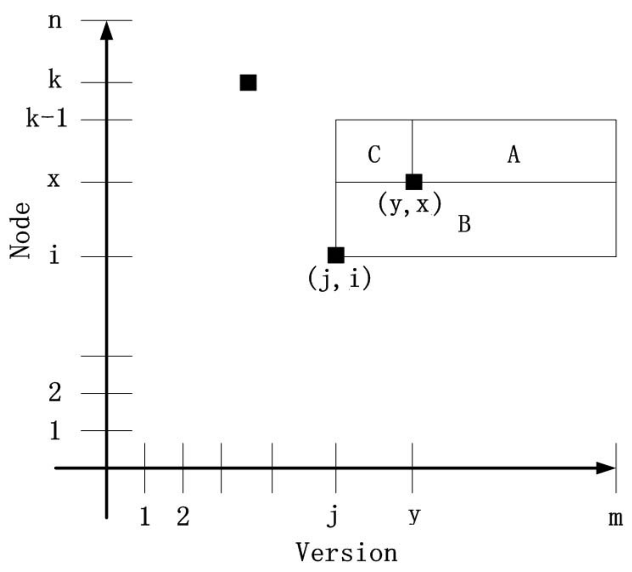

Figure 2. Block Definition for Multimedia Object Placement

Now let us refer to the first equation in the recurrence formula above. The first term $W(i,j,y-1)$ is the total access cost for the requests in Block B and D, the second term is the total access cost for the requests in Block A, and the last term is the total access cost for the requests in Block C. The second equation is for the special case of $i=0$ which denotes the original server, where transcoding is not necessary since all versions are stored there. To obtain the optimal solution, all possible values of i, j, and k must be checked. The following theorem shows the correctness of the above recurrence formula for $W(i,j,k)$.

Theorem 1 *Formula (3) is the correct recurrence formula for $W(i,j,k)$.*

Proof Without loss of generality, we only need to prove the correctness of the first equation in Formula (3) since the second equation can be easily derived in a similar way and the third equation is trivial.

Let $W^{'}(i,j,k)$ denote the value of the right side of the first equation, i.e., $W^{'}(i,j,k) = \min_{i\le x<k; j\le y\le m}\{W(i,j,y-1)+W(x,y,k)+\sum_{x\le\alpha<k; j\le\beta\le y} f_{\alpha,\beta}((\alpha-i)L+(\beta-j)T)$.

We now prove that $W^{'}(i,j,k)$ is the optimal access cost, i.e., $W^{'}(i,j,k) = W(i,j,k)$. Suppose $A_{d_i^*}, A_{d_{i+1}^*}, \cdots, A_{d_k^*}$ is the optimal placement in Block $B_{i,j,k}$, which makes $W(i,j,k) = C(A_{d_i^*}, A_{d_{i+1}^*}, \cdots, A_{d_k^*})$. Thus, we can always divide Block $B_{i,j,k}$ into four parts according to y^* (see Figure 2), where $y^* = \max_{j<y\le m}\{d_y^*\}$ and version A_{y^*} is cached at node v_{x^*}. Therefore, we have $W(i,j,k) = C(A_{d_i^*}, A_{d_{i+1}^*}, \cdots, A_{d_{k-1}^*}) = W(i,j,y^*-1)+W(x^*,y^*,k)+\sum_{x^*\le\alpha<k; j\le\beta\le y^*} f_{\alpha,\beta}((\alpha-i)L+(\beta-j)T) \ge \min_{i\le x<k; j\le y\le m}\{W(i,j,y-1)+W(x,y,k)+\sum_{x\le\alpha<k; j\le\beta\le y} f_{\alpha,\beta}((\alpha-i)L+(\beta-j)T)$.

Now we want to prove $W^{'}(i,j,k) \ge W(i,j,k)$. Suppose there exists $(x^{'},y^{'})$ such that $W^{'}(i,j,k) = \min_{i\le x<k; j\le y\le m}\{W(i,j,y-1)+W(x,y,k)+\sum_{x\le\alpha<k; j\le\beta\le y} f_{\alpha,\beta}((\alpha-i)L+(\beta-j)T) = \min_{i\le x'<k; j\le y'\le m}\{W(i,j,y^{'}-1)+W(x^{'},y^{'},k)+\sum_{x'\le\alpha<k; j\le\beta\le y'} f_{\alpha,\beta}((\alpha-i)L+(\beta-j)T)$. Thus, Block $B_{\alpha,\beta}$ can be divided into four parts according to $x^{'}$ and $y^{'}$. According to the definition of $W(i,j,k)$, we have $W(i,j,k) \le \min_{i\le x'<k; j\le y'\le m}\{W(i,j,y^{'}-1)+W(x^{'},y^{'},k)+\sum_{x'\le\alpha<k; j\le\beta\le y'} f_{\alpha,\beta}((\alpha-i)L+(\beta-j)T) = \min_{i\le x<k; j\le y\le m}\{W(i,j,y-1)+W(x,y,k)+\sum_{x\le\alpha<k; j\le\beta\le y} f_{\alpha,\beta}((\alpha-i)L+(\beta-j)T) = W^{'}(i,j,k)$. Therefore, we have proved that $W^{'}(i,j,k) = W(i,j,k)$. Hence, the theorem is proven. □

The original multimedia object placement problem, i.e., with the cost function based on Equation (2), can be solved using dynamic programming with these recurrences. We can also see that the minimum access cost is $W(0,1,n)$. It can be proved that the time complexity of the relevant solution is $O(n^3m^2)$ time, where n is the number of nodes and m is the number of versions.

4. Simulation Model

To the best of our knowledge, it is difficult to find true trace data in the open literature to execute such simulations. Therefore, we generated the simulation model from the empirical results presented in [1–4].

The network topology was randomly generated by the Tier program [4]. Experiments for many topologies with various parameters were conducted and the relative performance of our solution was found to be insensitive to topology changes. Here, only the experimental results for one topology are presented due to space limitations. The characteristics of this topology and the workload model are shown in Table 1, which were chosen from the open literature and are considered to be reasonable.

Parameter	Value
Number of WAN Nodes	200
Number of MAN Nodes	200
Delay of WAN Links	Exponential Distribution ($\theta = 1.5Sec$)
Delay of MAN Links	Exponential Distribution ($\theta = 0.7Sec$)
Number of Servers	100
Number of Web Objects	1000
Web Object Size Distribution	Pareto Distribution ($\mu = 6KB$)
Web Object Access Frequency	Zipf-Like Distribution ($\alpha = 0.7$)
Relative Cache Size	4%
Average Request Rate Per Node	$U(1,9)$ requests per second
Transcoding Cost	$50KB/Sec$

Table 1. Parameters Used in Simulation

The objects generated are divided into two types: *text* and *multimedia*. Similar to the studies in [3,10], cache size is described as the total relative size of all objects available in the content server. In our experiments, the object sizes are assumed to follow a Pareto distribution and the average object size is $6KB$. We also assume that each multimedia object has five versions and that the transcoding graph is as shown in Fig. 3. The sizes of each version are assumed to be 100 percent, 80 percent, 60 percent, 40 percent, and 20 percent of the original object size. The transcoding delay is determined as the quotient of the object size to the transcoding rate. In our experiments, the client at each MAN node randomly generates the requests, and the average request rate of each node follows the distribution of $U(1,9)$, where $U(x,y)$ represents a uniform distribution between x and y. The access frequencies of both the content servers and the objects maintained by a given server follow a Zipf-like distribution [3,8]. Specifically, the probability of a request for object O in server S is proportional to $1/(i^{\alpha} \cdot j^{\alpha})$, where S is the ith most popular server and O is the jth popular object in S. The delay of both MAN links and WAN links follows an exponential distribution; the average delay for WAN links is 1.5 seconds and the average delay for WAN links is 0.7 seconds.

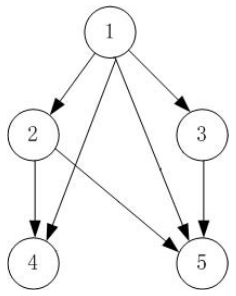

Figure 3. Transcoding Graph for Simulation

The cost for each link is calculated by the access delay. For simplicity, the delay caused by sending the request and the relevant response for that request is proportional to the size of the requested object. Here, we consider the average object sizes for calculating all delays, including the transmission delay and the transcoding delay. We apply a "sliding window" technique, for estimating access frequency, to make our model less sensitive to transient workload [10]. Specifically, the access frequency is estimated by $N/(t - t_N)$, where N is the number of accesses recorded, t is the current time, and t_N is the Nth most recently referenced time (the time of the oldest reference in the sliding window). N is set to 2 in the simulation.

In addition to the solution proposed in Section 3, we also consider the following placement solutions for comparison purposes. SV stores the same version of a multimedia object at each node when the request is sent back to the client from the server. MV stores the most referred version of a multimedia object at each node as the request is returned back to the client from the server. Specifically, if $i^* = \max_{1 \le j \le m} \{f_{i,j}\}$, then version A_{i^*} is cached at node v_i. RV randomly stores a version at each node.

5. Performance Evaluation

In this section, we compare the performance results of our solution with those solutions introduced in Section 4, in terms of several performance metrics. The performance metrics we used in our simulation include delay-saving ratio (DSR), defined as the fraction of communication and server delays which is saved by satisfying the references from the cache instead of the server; average access latency (AST); request response ratio (RRR), defined as the ratio of the access latency of the target object to its size; object hit ratio (OHR), defined as the ratio of the number of requests satisfied by the caches as a whole to the total number of requests. In the following figures, SV, MV, and RV denote the results for the three solutions introduced in Section 4, and OV denotes the optimal solution proposed in Section 3.

In the experiment, we compare the performance results of different solutions across a wide range of cache sizes, from 0.04 percent to 15.0 percent. The first experiment investigates DSR as a function of the relative cache size at each node and Fig. 4 shows the simulation results. As presented in Fig. 4, we can see that our solution outperforms the others since it considers multimedia object placement by determining the optimal versions to be placed at each node, whereas existing solutions, including SV, MV, and RV, consider multimedia object placement heuristically or randomly. Specifically, the mean improvements of DSR over SV, MV, and RV are 4.3 percent, 17.9 percent, 19.8 percent, and 24.5 percent, respectively.

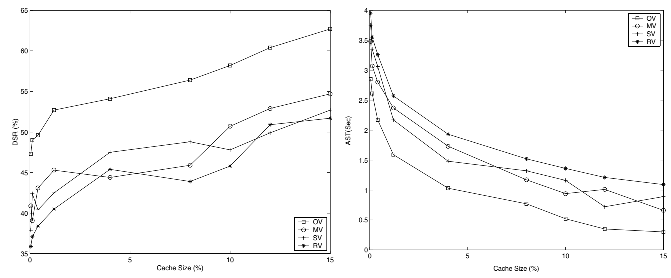

Figure 4. Experiment on DSR and AST

Fig. 4 also shows the simulation results of ASL as a function of the relative cache size at each node. Clearly, the lower the ASL, the better the performance. As we can see, all solutions provide steady performance improvement as the cache size increases. We can also see that OV significantly improves both ASL compared to SV, MV, and RV, since our solution determines the optimal versions to be cached on the path from the client to the server, while the others place multiple versions of a multimedia object in a heuristic or random way. For ASL to achieve the same performance as OV, the other solutions require 2 to 8 times as much cache size.

We describe the results of RRR and OHR as a function of the relative cache size at each node in Fig. 5.

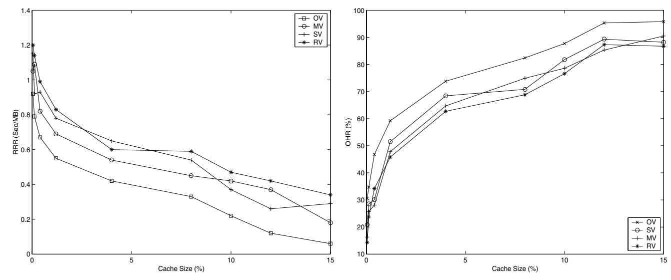

Figure 5. Experiment on RRR and OHR

6. Conclusion

Transcoding is attracting increasing research interest in the environment of mobile appliances, and transparent data replication is receiving more and more attention since it is capable of high system scalability. In this paper, we addressed the problem of multimedia object placement for hybrid transparent data replication. We studied this problem with the objective of minimizing total access cost by combining both transmission cost and transcoding cost. A set of simulation experiments were conducted to study the performance of our proposed solutions. The simulation results showed that our solution can significantly improve network performance compared with existing solutions.

References

[1] C. Aggarwal, J. L. Wolf, and P. S. Yu. *Caching on the World Wide Web*. IEEE Transactions on Knowledge and Data Engineering, Vol. 11, No. 1, pp. 94-107, 1999.

[2] P. Barford and M. Crovella. *Generating Representive Web Workloads for Network and Server Performance Evaluation*. Proc. ACM SIGMETRICS'98, pp. 151-160, 1998.

[3] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. *Web Caching and Zipf-like Distributions: Evidence and Implications*. Proc. IEEE INFOCOM'99, pp. 126-134, 1999.

[4] K. L. Calvert, M. B. Doar, and E. W. Zegura. *Modelling Internet Topology*. IEEE Communications Magazine, Vol. 35, No. 6, pp. 160-163, 1997.

[5] S. Chandra, C. Ellis, and A. Vahdat. *Application-Level Differentiated Multimedia Web Services Using Quality Aware Transcoding*. IEEE Journal on Selected Areas in Communications, Vol. 18, No. 12, pp. 2544-2565, December 2000.

[6] R. Han, P. Bhagwat, R. LaMaire, T. Mummert, V. Perret, and J. Rubas. *Dynamic Adaptation in An Image Transcoding Proxy for Mobile Web Browsing*. IEEE Personal Communications, Vol. 5, No. 6, pp. 8-17, December 1998.

[7] P. Krishnan, D. Raz, and Y. Shavitt. *The Cache Location Problem*. IEEE/ACM Transactions on Networking, Vol. 8, No. 5, pp. 568-582, 2000.

[8] V. N. Padmanabhan and L. Qiu. *The Content and Access Dynamics of a Busy Site: Findings and Implications*. Proc. ACM SIGCOMM'00, pp.111-123, August 2000.

[9] B. Shen, S.-J. Lee, and S. Basu. *Caching Strategies in Transcoding-Enabled Proxy Systems for Streaming Media Distribution Networks*. IEEE Transactions on Multemidia, Vol. 6, No. 2, pp. 375-386, April 2004.

[10] J. Shim, P. Scheuermann, and R. Vingralek. *Proxy Cache Algorithms: Design, Implementation, and Performance*. IEEE Transactions on Knowledge and Data Engineering, Vol 11, No. 4, pp. 549-562, 1999.

[11] X. Tang and S. T. Chanson. *Coordinated En-Route Web Caching*. IEEE Transactions on Computers, Vol. 51, No. 6, pp. 595-607, June 2002.

[12] J. Xu, B. Li, and D. L. Li. *Placement Problems for Transparent Data Replication Proxy Services*. IEEE Journal on Selected Areas in Communications, Vol. 20, No. 7, pp. 1383-1398, 2002.

[13] A. Vetro, C. Christopoulos, and H. Sun. *Video Transcoding Architectures and Techniques: An Overview*. IEEE Signal Processing Magazine, Vol. 20, No. 2, pp. 18-29, March 2003.