A Layered Cache Scheme for Internet Video Proxies*

Chung-ming Huang and Ching-hsien Tsai
Laboratory of Multimedia Networking (LMN)
Dept. of Computer Science and Information Engineering
National Cheng Kung Univeristy
Tainan, Taiwan, 70101, R.O.C

Correspondence: huangcm@locust.csie.ncku.edu.tw

Abstract

Due to the improvement of the computer networks, it becomes more popular to stream video from the server to clients over Internet. To offer more convenient and flexible services to more users, the proxy-based service architecture should be adopted in video transmission over Internet. However, many control schemes, e.g., the cache scheme, that are currently adopted in the web proxy should be modified because of the characteristics of videos, such as the larger size and the real time concern. In this paper, we propose a layered cache scheme and a replacement scheme for video proxy, where clients can specify the quality of the requested video. To meet the QoS concern, delay factor is considered in our cache scheme to reduce the waiting time at the client side, and a corresponding cache scheme that does caching according to a media's deserved storage size is devised.

Key words: Stream, Proxy, Real Time, Layering, Replacement.

*This research is supported by the National Science Council of the Republic of Chain under the grant NSC 90-2213-E-006-108.

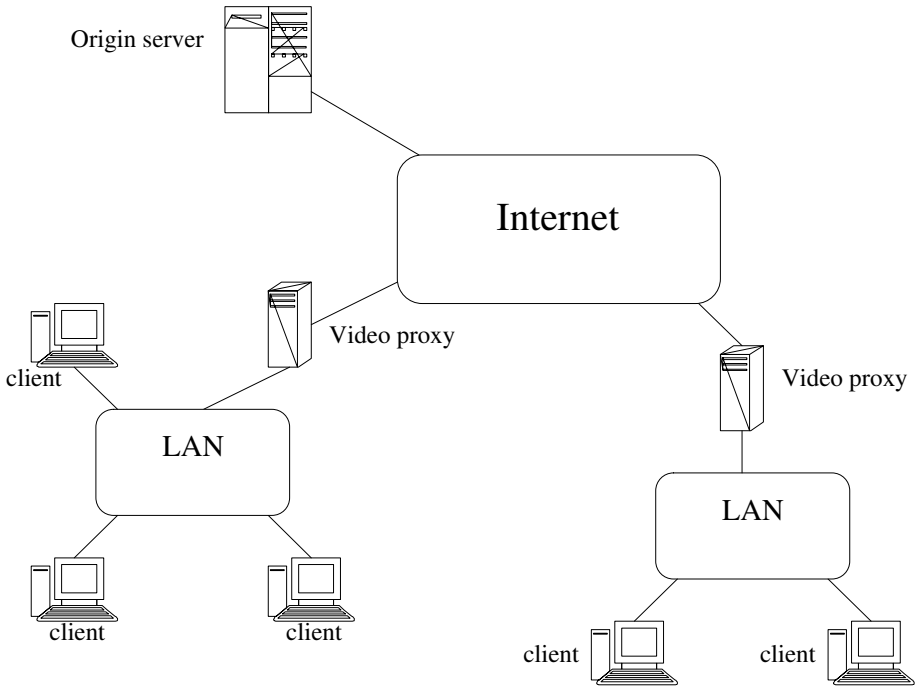

Figure 1. The role of the proxy on the Internet.

1. Introduction

With the dramatic growth of Internet in these years, we can access information resources at any node of Internet, of which the geographical location may be far away from us. The more dependence on Internet, however, has made servers suffering a significant overload on providing their services. Hence, when more and more people are trying to fetch their desiring data on Internet, the waiting time would be also longer because of the network's congestion. The proxy architecture is one of the main solutions to solve the problem. A proxy that is closer to the client can cache recently and frequently accessed data in its storage. When the user nearby tries to access these data, the data will be transmitted from the proxy if they are available in the storage of the proxy. The proxy-based service architecture not only provides faster data transmission to the client, but also reduce the load of the origin server.

Recently, because of the development of the high-bandwidth network connection, users can enjoy video and audio services through Internet. However, existing proxy's caching schemes are mainly designed for traditional web data, such as images and texts [1, 2, 3, 4, 5, 6, 7]. Many algorithms, such as LRU, LFU, and LRFU, were developed to make a better use of the proxy storage. However, for the large size of video media, if the traditional caching and replacement schemes are used, the storage of a proxy will be exhausted by just a few video objects very soon. Thus, the traditional LRU, LFU, and LRFU techniques and the corresponding cache schemes are not suitable for the construction of a video proxy over Internet.

Major concerns that are taken into consideration for the caching and replacement in a web proxy are the requesting frequency, requesting recency, and the size of the object. Due to the different characteristics of the video proxy, e.g., large size and the real-time requirement, other concerns should be

considered to make the multimedia proxy more efficient [8, 9, 10, 11, 12] . In [8], the author proposed that the proxy stores initial frames of popular clips due to the unpredictable delay, throughput, and loss properties of Internet. The size of the video prefix depends on the performance of the path between the server and the proxy. In the study of [12], the proxy stores a portion of a media stream or an entire stream according to the popularity of the stream. In [10], some proposed heuristics are used to determine which video and which layers in the videos should be cached when delivering layered video using caches.

In summary, the "popularity" concern is considered as an important factor for caching continuous media in proxy. Since the current Internet is best-effort-based, many adaptive flow control schemes are proposed to achieve some QoS requirements. The media layering technique is usually adopted to achieve adaptive flow control. Using the media layering technique, a video stream is divided into many layers. For example, an MPEG stream is divided into 4 layers, in which the base layer contains I frames, one enhancement layer contains P frames, and B frames are divided into two other enhancement layers. Using the media layering technique and the adaptive flow control scheme, when the network is not congested, all 4 layers are transmitted; when the network is becoming congested, some enhancement layers are gradually dropped. On the contrary, when the network is becoming less congested, some enhancement layers can be transmitted again to have a higher quality's presentation. Moreover, in some multimedia applications, e.g., Video-On-Demand systems, the media server can choose which layers to deliver according to their priorities, or control the number of the layers to be delivered according to end users' different QoS requirements. Thus, a video proxy that is devised for dealing with layer media should have different concerns in caching and replacement.

In this paper, we propose a caching scheme for continuous layered media using a weighted value considering some related factors, such as the most popular layers, the corresponding size of the layered video, and the delay time from the origin server to the proxy of each video, which enable the client to reduce the waiting time when requesting a playback of video. Our method can also utilize the storage capacity of the proxy to stretch the length of the playback time from the proxy and thus the load of the origin server can be significantly reduced.

The rest of this paper is organized as follows. In Section 2, details of our caching scheme is introduced. In Section 3, the performance evaluation of the proposed caching scheme is presented. In Section 4, conclusion remarks are given.

2. The Layered Cache Scheme Considering Delay-Sensitive and Popularity Factors

In this Section, the proposed layered cache scheme is introduced.

2.1 Popularity of a Video Media

Because of the limited storage of the proxy and the large size of the video media, we cannot store the complete file of a video object in the proxy with their entire length. To provide efficient services that are adaptive to most users' request preferences, caching and replacement within the proxy must be done according to media's popularity. Traditional web caching defines the popularity of an object according to its access recency. But, the access recency is not suitable as the popularity definition for video media because the video media has longer playout time length and users may not watch them, e.g., a 90-minute movie, completely every time. Thus, the popularity definition for a video media is based on the total

amount of playback time that all users have done during a specific time period, e.g., an hour, a day, or a week.

Therefore, the popularity of video media should be based on its access degree [9, 11, 12]. The proxy periodically measures the playback time of each specific video. Based on the historic statistics of a given time interval, we define the $PLAYTIME_{i,j}$ represents the playback time of video i played by client j in a time unit, where a time unit could be hour, day, etc. Thus the total playback time of a video i from all M clients in a time unit is the summation of $PLAYTIME_{i,j}$ for different j values, and we can also use this value as the popularity of video i :

$$Pop_i = \sum_{j=1}^{M} PLAYTIME_{i,j} \tag{1}$$

Let's consider the environment that the service provider can offer differentiated qualities to users. Due to various kinds of video contents in the proxy, such as news clips, advertisement videos, and movies, the user may want to decide the playback quality to his preference when making a request of some videos. For example, when the user makes a pre-view request of a news clip from the server, he may not want to have the full quality of that news clip because reducing the playback quality can also reduce the startup latency and hence he can have a pre-view much faster. When the user decides to have a request of the complete movie video, he will want to have the full quality stream of that video to enjoy the sound and light effects of that movie.

With the illustrated video service scenario described above, if we cache the video based on its popularity given in Formula (1), the proxy will store all layers of each video. As a result, for the videos that users usually make requests with lower qualities, there will be many redundant layers stored in the proxy. For example, if a news clip that consists of 4 layers is usually requested with 2 layers, the other 2 layers seem useless, but they still reside in the proxy's storage. Assuming that each layer of the video has the same size, if there is 40% of layers stored in the proxy are rarely used, the layers that actually work in the proxy is just about 60%. It implies that the hit-rate of the proxy could be efficiently improved if we replace the useless layers with the useful and the potentially popular ones.

In our method, we will cache the most popular layers of the video in the proxy. The most popular layers mean those media layers that are requested by users most often. Let a video have 4 layers. If the requested number of the four types, 1-layer (layer1 only), 2-layer (layers 1+2), 3-layer (layers 1+2+3), and 4-layer (layers 1+2+3+4), are 20, 65, 30, and 5, respectively in a specific time interval, we will cache layer1 and layer2 of that video because there are more than 50% of the users requesting for layer1 and layer2 of that video. Thus, for each video i, we use $poplayer_i$ to represent the "most popular layers " of the video and use the representation $size_i(poplayer_i)$ as the size of video i when it's cached in the proxy with its most popular layers.

Based on the above two parameters, i.e., the popularity of the layered media and the more popular layers, we can define a weighted value to calculate the storage size that a layered video can reside in a proxy when it is cachable. The weighted value of layered video i is as follows:

$$W_i = \frac{Pop_i}{size_i(poplayer_i)} \tag{2}$$

2.2 The Delay-Sensitive Factor

Let two video clips V_i and V_j have similar popularities and the weighted value of V_i be a little bit larger than that of V_j, but the geographical position of the origin server of V_j be so far away from the proxy server and therefore the response time is much longer than that of V_i. Under the circumstance, the origin server of V_i is relatively much closer to the proxy server. When V_i is requested by a client, it can be streamed from the origin server to the proxy in time because it's just nearby. Thus, V_j should be given more cache capacity because of its longer response time even though its popularity value is not larger. It means that it is more valuable if we cache a video clip with longer response time (delay time).

With the consideration of the delay factor, we define a new weighted value with delay-sensitive factor D_i, which determines the delay time of video i when it is fetched from the origin server. The weighted value with the delay-sensitive factor dW is as follows:

$$dW_i = \frac{D_i \cdot Pop_i}{size_i(poplayer_i)} \tag{3}$$

where dW_i is the weighted value of video i with the delay-sensitive factor, D_i is the delay time of video i from its video server to the proxy, Pop_i is the popularity of video i, and $size_i(poplayer_i)$ is the size of video i when it is cached in the proxy with its most popular layers.

2.3 The Caching Scheme

Let the deserved storage size of video m be dC_m, where m must be in the descending sequence of the weighted value, dW_m. Based on the weighted value with the delay-sensitive factor dWi that is given in Formula (3), dC_m is derived as follows:

$$dC_m = \min\{(C - \sum_{p=1}^{m-1} dC_p) \times \frac{dW_m}{dW - \sum_{q=1}^{m-1} dW_q}, size_m(poplayer_m)\} \tag{4}$$

where C is the total capacity of the cache storage, and dW is the summation of all video's weighted values, i.e., $dW = \sum_{i=1}^{n} dW_i$.

After the value of each dC_m is known, the proxy begins to cache videos according to their deserved storage size. Since a video stream may be very large, e.g., 400MB, it may be divided into many segments. For example, one segment contains 1MB. To ensure that the cached streams of the cached layers can be played well, the caching sequence is from layer 1 to x and from segment 1 to y, which is depicted in Figure 2. In Figure 2, the number of the most popular layers of the video is 3, which means layer1, layer2, and layer3 of the video will be cached in the proxy. When the video is going to be cached into the proxy's storage, the cache process should be as follows: layer1 of segmentation1, layer2 of segmentation1, layer3 of segmentation1, then, layer1 of segmentatin2, layer2 of segmentation2, layer3 of segmentation2, and so on. This process continues until the summation of all cached segmentations is equal to the deserved storage size of the video clip.

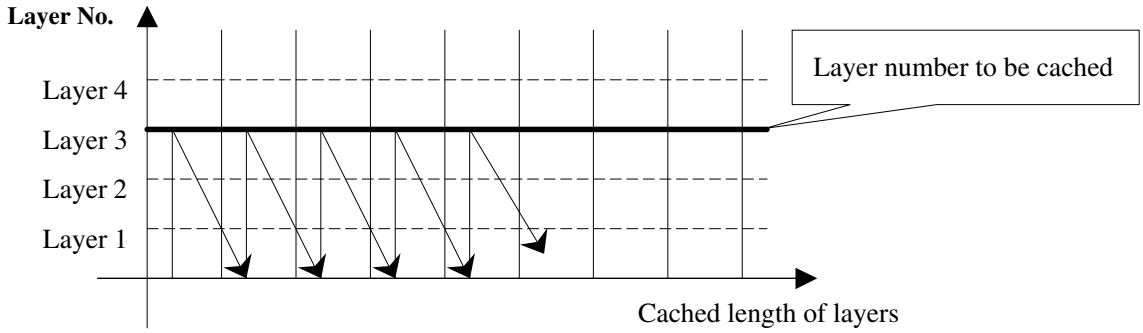

Figure 2. The caching scheme of the layered media.

3. Performance Evaluation

In our cache scheme, only the most popular layers of a video are cached in the proxy. The caching precedence and the deserved storage size of each video is based on weighted values, which are calculated with the formula introduced in Section 2. We compare our cache scheme with the original popularity-based cache scheme, which caches a media with all layers according to its popularity. We mainly use Byte Hit Rate (BHR) to measure the performances of our proposed layered cache scheme (LC), and the popularity-based cache scheme (PC). In addition, the comparison on each video's playback time cached in the proxy and the waiting time at the client side will also be depicted.

Parameters of our simulation environment are as follows. Let there be 250 video objects in the proxy, and their sizes be from 400MB to 650 MB. Each video are layered into 5 layered streams. The processing time of the simulation is the duration of 10000 requests by users, and the popularity distribution is conformed to Zipf distribution with the parameter equals 0.6 [13]. The response time (delay) of the origin server is from 100ms to 5000 ms.

3.1 Playback Length of the Video Cached in the Proxy

In Figure 3, the ratio of the total cached playback length of LC and PC are shown. The layered cache scheme caches videos according to their deserved sizes calculated using the weighted function depicted in Formula (4) of Section 2. Using LC, since the proxy only caches the most popular layers of the video, the playback lengths of videos using LC are longer than those using PC. In Figure 3, we can see the total playback lengths using LC are about 58 to 86 percent longer than those using PC, but this does not mean that the cached playback length of each video using LC is always longer than that using PC. There may be some exceptions because the weighted function in LC also takes the delay factor into consideration, whereas the PC does not. Therefore, if a video's delay factor is relatively small when comparing with the others, the weighted value of the video will also be relatively smaller than that without considering the delay factor. Thus, the corresponding deserved storage size of the video will also be smaller, and so will the playback length of the video. It is the reason that the playback lengths of videos using LC are not always longer than those using PC.

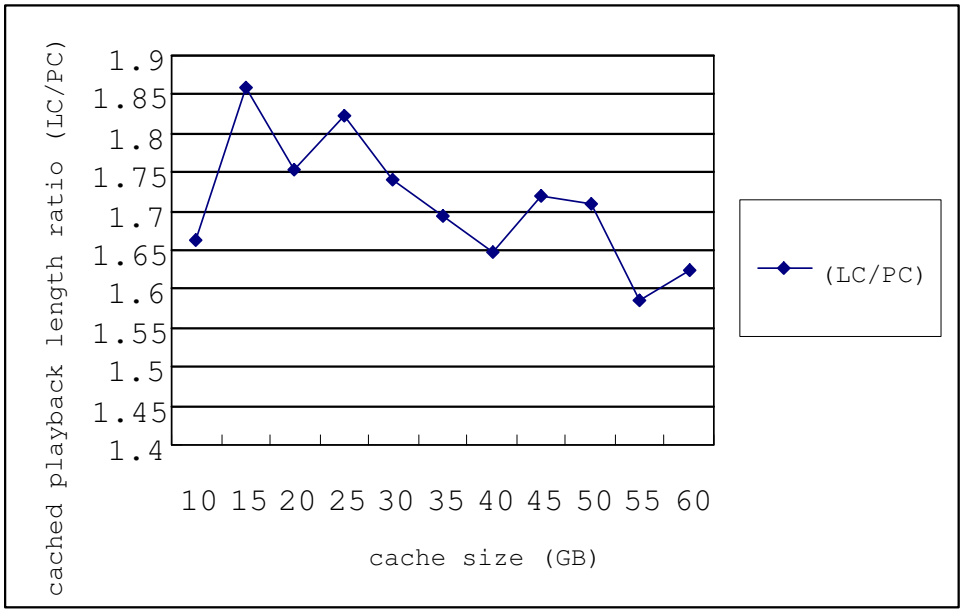

Figure 3. The ratio of the total playback length cached in the proxy of LC and PC, according to different cache sizes.

3.2 Byte Hit Rate (BHR)

First let's see the effect on BHR of a single object cached in the proxy. The main difference between the layered cache scheme and the traditional popularity-based caching scheme is that the former only caches the most popular layers of a video, while the later caches all layers of a video. It means that given the same storage size, the layered cache scheme can have the proxy provide longer playback length than the cache scheme that caches all layers. However, some tradeoffs may occur. Considering the following case. The number of the most popular layers of a video is 3, with 50% of all requests. Thus, using the proposed layered cache scheme, the proxy will cache layer 1, layer 2, and layer 3. When users request layer1, layer2, or layer3 quality of the video, there will be byte hit in the proxy; but when some users request for layer4 quality, for example, 10% of all requests, then the BHR will decrease because the proxy doesn't cache layer4 in the proxy. On the contrary, the traditional popularity-based cache scheme caches all layers of a video, therefore, the BHR in the case will not decrease if the user's playback time is not long. Hence, if the user plays a video for very short time, the BHR of the traditional cache scheme is better; but for normal playback time, because the layered cache scheme can satisfy most of the requests with longer playback time, the BHR is better than that of the traditional polarity-based cache scheme.

Figure 4 shows the BHRs of using the two schemes. Figure 4 shows the BHR of a single video object cached in the proxy according to the average playtime of that video from the client. The video is divided into 4 layers, and the most popular layer is set as 2. The "p(L2)" in Figure 4 denotes the percentage of the requests that preferring layer2 quality. For example, in data1 and data2, 50% of the requests are set to acquire layer2 quality. Other parameters in data1 and data2 are set as follows: p(L1)=0.2, p(L3)=p(L4)=0.15. Likely, in data3 and data4, p(L1)=0.25, p(L3)=0.25, p(L4)=0.2. From Figure 4,

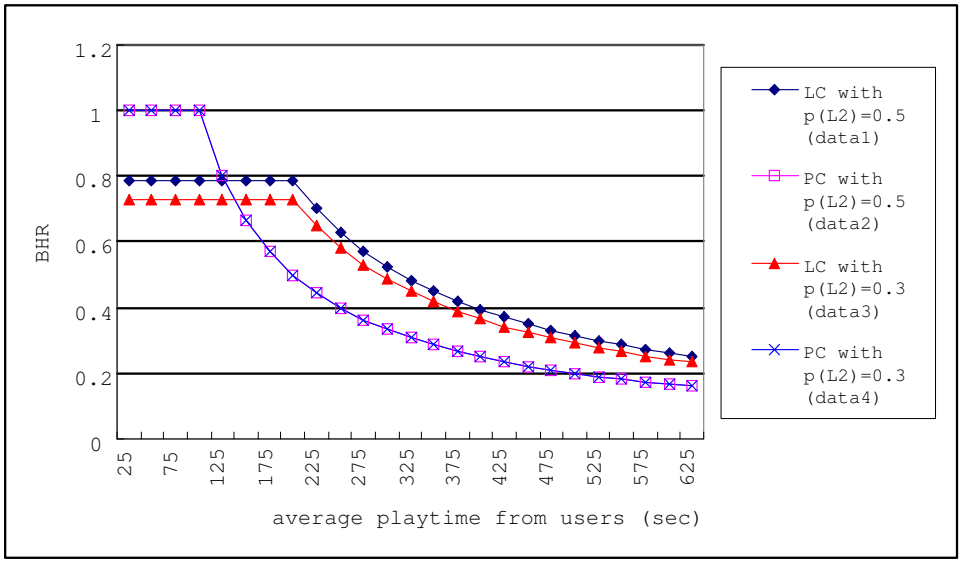

Figure 4. The BHR comparison of the two schemes for a single video.

we can observe that when the average playtime of the user is more than 100 seconds, the BHR of the popularity-based cache scheme decrease significantly, and when playtime is more than 150 seconds, the BHR of the layered cache scheme becomes better than that of popularity-based cache scheme. It means that the BHR of the LC is more stable than that of the PC and thus LC is more suitable for users' longer playtime of the video. Moreover, in Figure 4, we can also find that the curve of data2 and data4 are overlapped, whereas the curve of data1 is above data3. It means that the variance of popularities between layers does not affect the BHR of PC, but in LC, if the degree of popularity variance among layers are larger, the BHR is better. The BHR of LC is better than PC in average.

In Figure 5, the BHRs of using LC and PC after the simulation of 10000 requests are depicted according to the cache size of the proxy. From Figure 5, we find that LC performs 4% to 12% better than the PC and hence the LC can efficiently reduce the load of the origin server. The result also conforms the advantage of LC depicted in Figure 4.

3.3 Waiting Time after 10000 Requests

In the weighted function depicted in Formula (3) of Section 2, we consider the waiting time (delay factor) when requesting a video from the origin server. It means that using LC, the proxy would prefer caching videos that have larger delay values because if a user requests this video and the video is not cached in the proxy, the user needs to wait a longer time to receive the data transmitted from the origin server.

Figure 6 shows the amount of waiting time using LC and using PC after 10000 requests, in which the cache size is from 10 GB to 70 GB. When the cache size is under 50GB, with the consideration of the delay factor, the amount of waiting time in LC is relatively small comparing with PC. In Figure 6, we can also observe that the value of waiting time in both schemes are very small when the cache size is above 50GB. It is because when the cache size is larger, the number of cachable videos will also increase. As a result, the total waiting time will decrease because there are more videos cached in the proxy. In the

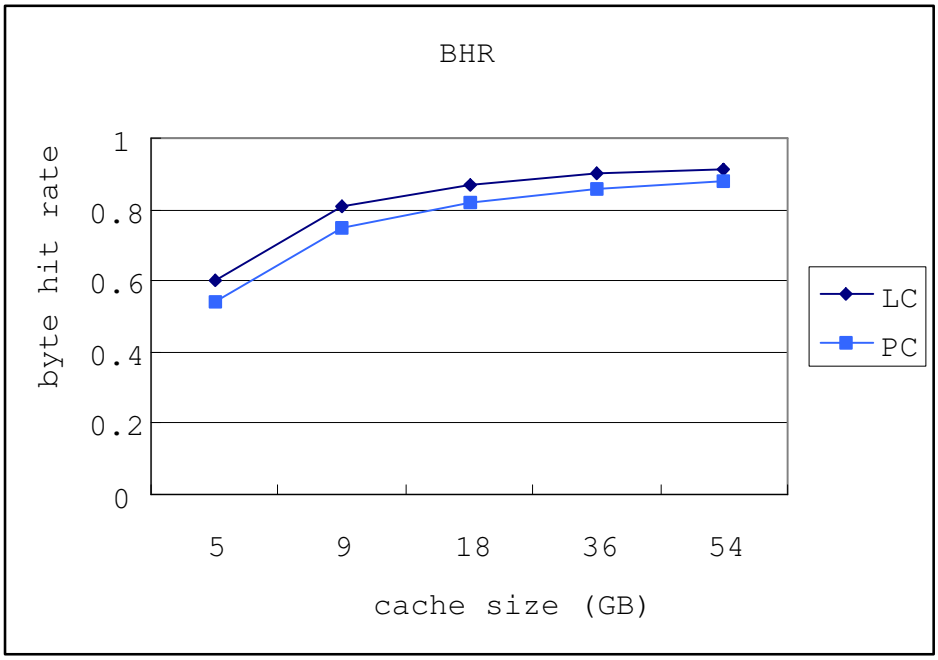

Figure 5. The BHR of the two schemes through our simulation.

simulation, the LC works much better than PC does when the cache size is small.

4. Conclusion

The cache technique, which stores the frequently accessed data in the proxy can significantly reduce the load of the origin server and the startup latency at the client side. Existing Web caching techniques, such as LRU and LFU, are object-level caching which is suitable for the caching of text and image files, and can not work well for dealing with continuous media.

The popularity-based cache is hence developed to do the caching of continuous media files. In this paper, the layered cache scheme is proposed to handle the caching of layered media. In the layered cache scheme, only the most popular layers of a video are cached, and the deserved storage size of the video is calculated from the weighted function, which combines three factors: delay factor, popularity of the video, and the size of the video when it is cached with its most popular layers. Through the simulations, we have evaluated the performance of the layered cache scheme. When the proposed scheme is compared with the traditional popularity-based cache scheme, we find that the the layered cache scheme works better than the popularity-based cache scheme in all experiments: BHR, playback length cached in the proxy, and the waiting time at the client side.

References

[1] A. Chankhunthod, P. B. Danzig, C. Neerdaels, M. F. Schwartz, and K. J. Worrell, “A Hierarchical Internet Object Cache,“ Proceedings of 1996 Usenix Technical Conference, pp. 153 -164, January

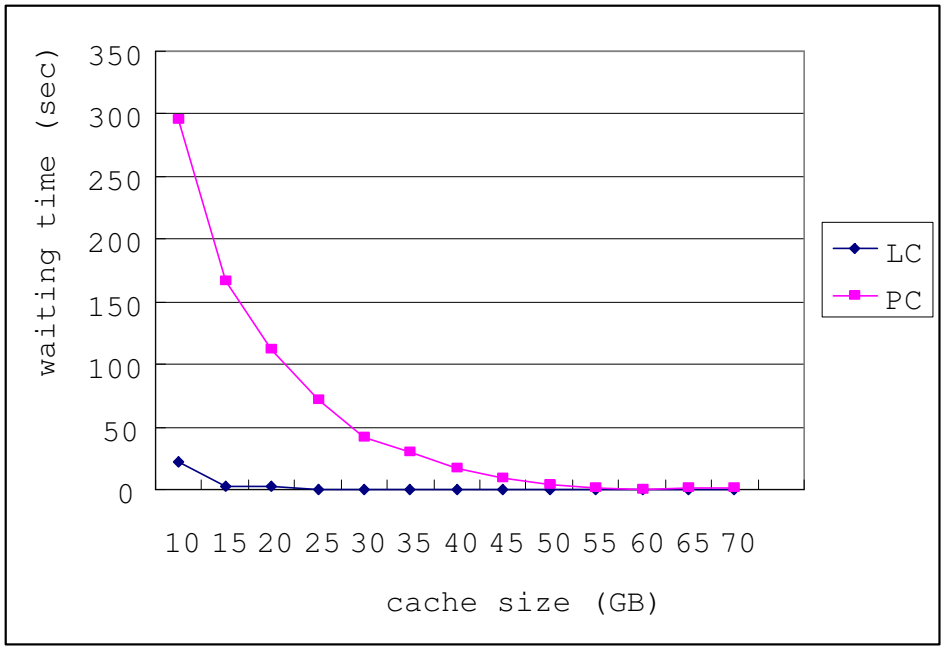

Figure 6. The total waiting time of the two schemes after 10000 requests' simulation.

1996.

[2] M. Abrams, et al., "Caching Proxies: Limitations and Potentials," Proceedings of the 4th International Conference on the WWW, pp. 312-319, 1995.

[3] J. Bolot and P. Hoschka, "Performance engineering of the World Wide Web: Application to Dimensioning and Cache Design," Proceedings of the 5th International Conference on the WWW, pp. 1397 -1405, 1996.

[4] S. G. Dykes and K. A. Robbins, "A Viability Analysis of Cooperative Proxy Caching," Proceedings of 2001 IEEE INFOCOM, VOL. 3, pp. 1205 -1214, 2001.

[5] M. Busari and C. Williamson, "On the Sensitivity of Web Proxy Cache Performance to Workload Characteristics," Proceedings of 2001 IEEE INFOCOM, VOL. 3 , pp. 1225 -1234, 2001.

[6] A. Luotonen and K. Altis, "World Wide Web Proxies," Proceedings of the 1st International Conference on the WWW, pp. 147- 154, May 1994.

[7] J. T. Robinson and N. V. Devarakonda, "Data Cache Management Using Frequency-based Replacement," Proceedings of ACM SIGMETRICS Conference, pp. 134- 142, 1990.

[8] S. Sen, J Rexford and D. Towsley, "Proxy Prefix Caching for Multimedia Streams," Proceedings the 18th IEEE INFOCOM, VOL. 3, pp. 1310 -1319, March 1999.

[9] Reza Rejaie, Haobo Yu, Mark Handely, and Deborah Estrin, “Multimedia Proxy Caching Mechanism for Quality Adaptive Streaming Applications in the Internet,“ Proceedings of the 19th IEEE INFOCOM, VOL. 2, pp. 980 -989, March 2000.

[10] J. Kangasharju, F. Hartanto, M. Reisslein, and K. W. Ross, “Distributing Layered Encoded Video Through Caches,“ IEEE Transactions on Computers, VOL. 51, NO. 6, pp. 622 -636, June 2002.

[11] Eun-Ji Lim, Seong-Ho Park, Hyeon-Ok Hong, and Ki-Dong Chung, “A Proxy Caching Scheme for Continuous Media Streams on the Internet,“ Proceedings of the 15th International Conference on Information Networking, pp. 720 -725, 2001.

[12] S. H. Park, E. J. Lim, and K. D. Chung, “Popularity-based Partial Caching for VOD Systems Using a Proxy Server ,“ Proceedings of the 15th Conference on InternationalParallel and Distributed Processing Symposium, pp. 1164 -1168, 2001.

[13] B. Lee, C. Pei, F. Li, P. Graham, and S. Scott, “Web Caching and Zipf-like Distributions: Evidence and Implications,“ Proceedings of the 18th IEEE INFOCOM, pp. 126-134, 1999.