The Transmitted Strategy of Proxy Cache Based on Segmented Video

Zhiwen Xu, Xiaoxin Guo, Yunjie Pang, Zhengxuan Wang

Faculty of Computer Science and Technology, Jilin University,
Changchun City, 130012, Jilin Province, China
`xuzhiwen@public.cc.jl.cn`

Abstract. Using proxy cache is a key technique that may help to reduce the loads of the server, network bandwidth and startup delays. Basing on the popularity of clients' request to segment video,we extend the length for batch and patch by using dynamic cache of proxy cache for streaming media. Present transmission schemes using dynamic cache such that unicast suffix batch, unicast patch, multicast patch, multicast merge and optimal batch patch by proxy cache based on segmented video. And then quantitatively explore the impact of the choice of transmission scheme, cache allocation policy, proxy cache size, and availability of unicast versus multicast capability, on the resultant transmission cost.

1   Introduction

We consider that video stream from a server travels through Internet to the end clients. We assume that clients always request playback from the beginning of a video. The proxy receives the client request and, if the prefix video is available in the proxy, streams the prefix directly to the client. If the video is not present in the proxy, the latter will contact the server and streams the received data from the server to the client. We, according to the popularity of video application from the clients, store the segmented videos in proxy cache, therefore enhance the byte hit ratio of proxy cache.

In order to raising the efficiency of proxy cache for streaming videos, we present the video segmented cache scheme based on the popularity of video requests from the clients. We segment the video into small parts, for being cached and substituted. Suppose the minimal cache allocation unit length is b, we segment video into the multiple of b. The value of b is determined according to the startup delays in the Internet environments. In later discussion, we simply assume that the video length unit is b, therefore video unit can be independently cached and substituted. The length of the prefix cache unit is also b. According to the popularity of video requests, we cache the video segmentation of different size and insure that segmented cache can obtain maximal percentage of byte hits radio. We extend the length for batch and patch by using dynamic cache of proxy cache for streaming media. Present transmission schemes using dynamic cache such that unicast suffix batch, unicast patch, multicast patch, multicast merge and optimal batch patch by proxy cache based on segmented video.

2   Proxy Cache Based on Segmented Video

The f_i measures the relative popularity of a video: every access to the video repository has a probability of f_i requesting video i. Let λ_i be the access rate of video i and λ be the aggregate access rate to the video repository. The storage vector $V = (V_1, V_2, \ldots, V_N)$ specifies that a prefix of length V_i seconds for each video i is cached at the proxy, i=1,2,…,N. The storage vector $U = (U_1, U_2, \ldots, U_N)$ specifies that a segmented cache of length U_i seconds for each video i is cached at the proxy, i=1,2,…,N. The length of video i cached at the proxy is $V_i + U_i$. Let C_s and C_p represent respectively the costs associated with transmitting one byte of video data on the server-proxy path and proxy-client path. $C_i(V_i, U_i)$ is the transmission cost per unit time for video i when the length of proxy cache is $V_i + U_i$. The r_i represent the stream rate of video i resource.

2.1   Unicast Suffix Batch Based on Segmented Video

Unicast suffix batch using dynamic cache based on segmented video is a simple batch scheme, which makes use of proxy cache to provide immediate playback. It is designed for the unicast transmission of video from the proxy to the clients. Suppose the first video request i arrives at time 0. The proxy will immediately transmit the video prefix to the client. Unicast batch processing makes the transmission time from server to proxy be as late as possible, and ensures it the playback on the client side is continuous. That is to say, the first frame of suffix is scheduled to reach the client at time V_i. The length of prefix cache which depends on the network environment, but the other length U_i of cache in proxy cache is relative to the speed of customer's request for this video, that is to say, we determine the length of cache U_i according to the popularization of video. For any request arriving in time $(0, V_i + U_i)$, the proxy just forwards the incoming suffix (of length $L_i - V_i - U_i$) to the client, and new transmitted suffix should come from server. Actually, several video suffix requests are transmitted in batch using dynamic cache. Segment-based unicast transmission and prefix cache can solve the problem of startup delays. But batch transmission using dynamic cache can increase transmission efficiency because its maximum unit is realized by dynamic cache windows and different from the others. Suppose there is a Poisson distribution process, and the average request number is $1 + (V_i + U_i)\lambda_i$ in time $[0, V_i + U_i]$, these requests cause the transmission of suffix $[V_i + U_i, L_i]$ from the server, and the average transmission cost of video i is :

$$c_i(V_i, U_i) = (c_s \frac{L_i - V_i - U_i}{1 + (V_i + U_i)\lambda_i} + c_p L_i)\lambda_i r_i \tag{1}$$

2.2   Unicast Patch Based on Segmented Video

Unicast patch for proxy cache based on segmented video using dynamic cache can save network resource. The first request of video i arrives at time 0, and the video non-stored in proxy cache arrives at time $V_i + U_i$ from server (Fig.1).

Suppose another customer's request arrives at time t_2, $V_i + U_i < t_2 < L_i$. One method is to read the video non-cached in proxy from server directly. The other is to use patch processing technique. We suppose to handle $[V_i + U_i, t_2]$ from the patch of server, since segment $[t_2, L_i]$ have been scheduled to transmit, the patch is set at time $t_2 + V_i + U_i$, then the client must receive at the same time from two channels and deliver the content of the suffix and patch. So a patch depends on a suffix threshold G_i. Measured from the beginning of the suffix, if we begin delivering from the nearest suffix, and request the arrival within G_i, the proxy will schedules a patch from the server for it. Otherwise, it starts a new complete transmission of the suffix. Suppose a Poisson arrival process, between the initiations of the two consecutive transmission of the suffix, the average number is $1 + \lambda_i(V_i + U_i + G_i)$. These requests are only in one transmission of the suffix $[V_i + U_i, L_i]$ from the server, the total length of patches from the server for these requests is:

$$\lambda_i G_i \int_0^{G_i} x/G_i dx = \lambda G_i^2/2 \tag{2}$$

This is because, the distribution of arrivals in time interval $[V_i + U_i, V_i + U_i + G_i]$ follows a uniform distribution in a Poission arrival process. The average transmission cost of the video i is:

$$c_i(V_i, U_i) = c_s \lambda_i r_i \frac{\lambda_i G_i^2/2 + L_i - V_i - U_i}{1 + \lambda(V_i + U_i + G_i)} + c_p \lambda_i r_i L_i \tag{3}$$
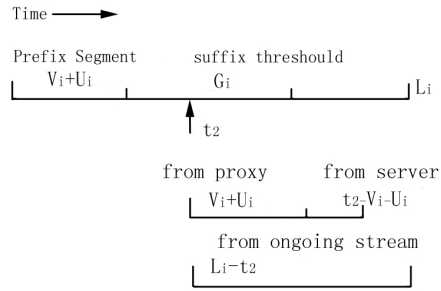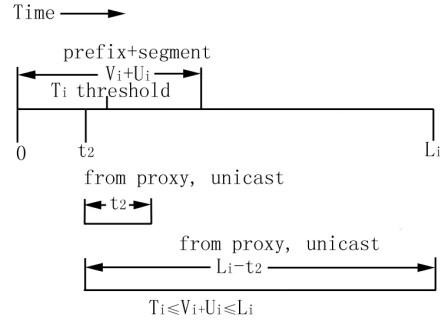

Fig. 1: Unicast patch   Fig. 2: Case 1

2.3 Multicast Patch Based on Segmented Video

If the path from the proxy to the client is multicast transmission, proxy can use the scheme of segmented multicast patch using dynamic cache. Suppose

the request of video i arrives at time 0 (fig.2), proxy begins to transmit video $V_i + U_i$ by multicast at time 0, the server begins to transmit suffix of video to the client at time $V_i + U_i$, and the proxy transmits received data to client by multicast patch using dynamic cache. If that T_i is the domain value that control the transmission frequency of whole stream. Suppose a request shortly after the stream transmitting arrives at $t_2(0 < t_2 \leq T_i)$, the video transmission to the clients can be classified into two cases according to the relationship between $V_i + U_i$ and T_i:

Case 1: $T_i \leq V_i + U_i \leq L_i$ (Fig.2), the client receives $[0, t_2]$ segment by using unicast from proxy through a single channel, and receives $[t_2, L_i]$ segment by using a processing multicast. Suppose this is the Poisson distribution, in this circumstance, the value transmission function $g_1(V_i, U_i, T_i)$ is:

$$\frac{\lambda_i r_i}{1 + \lambda_i T_i}[(L_i - V_i - U_i)c_s + L_i c_p + \frac{\lambda_i T_i^2}{2}c_p] \tag{4}$$

Case 2: $0 \leq V_i + U_i < T_i$ (Fig.3), if $0 < t_2 \leq V_i + U_i$, transmission construction is the same as case 1. If $V_i + U_i < t_2 \leq T_i$, the client receives $[0, V_i + U_i]$ segment by using unicast from proxy through a single channel, and receives $[t_2, L_i]$ by using the processing multicast. The server transmits $[V_i + U_i, t_2]$ through proxy to client by using unicast. Suppose its process is the Poisson distribution, then the transmission cost function $g_2(V_i, U_i, T_i)$ is:

$$\frac{\lambda_i r_i}{1 + \lambda_i T_i}[(L_i - V_i - U_i)c_s + L_i c_p + \frac{\lambda_i(V_i + U_i)^2}{2}c_p + \frac{\lambda_i(T_i - V_i - U_i)^2}{2}(c_s + c_p)] \tag{5}$$

We denote the cost of transmission with k(k=1,2), $h_k(V_i, U_i)$ is the minimum transmission cost of the transmission value function: $h_k(V_i, U_i)$=min$\{g_k(V_i, U_i, T_i), 0 \leq T_i \leq L_i\}$. For a given $V_i + U_i$ cache proxy, the average transmission value is: $c_i(V_i, U_i)$=min$\{h_1(V_i, U_i), h_2(V_i, U_i)\}$.
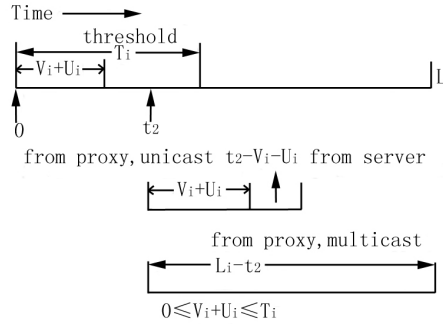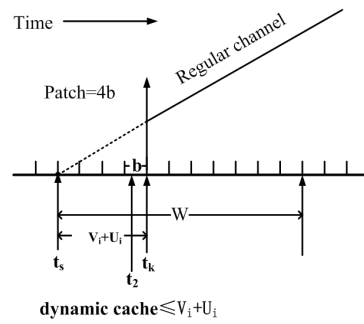

Fig. 3: Case 2   Fig. 4: optimal multicast patch

2.4   Multicast Stream Merging Based on Segmented video

The key issue of stream merging is deciding how to merge a later stream into an earlier stream. Closest Target policy [5] is one online heuristic policy whose performance is close to optimal offline stream merging. Our scheme for multicast Merge integrates dynamic cache and stream merging. It uses Closest Target policy to decide how to merge a later stream into an earlier stream. For a video segmentation required by the client, if a prefix of the segmentation is at the proxy, it is transmitted directly from the proxy to the client. The suffix not stored at proxy cache is transmitted from the server as late as possible while still ensuring continuous playback at the client. Let P_j be the probability of requiring a j-second , $0 \leq j \leq L_i$. This can be obtained by monitoring a running system. The average transmission cost of video is:

$$c_i(V_i, U_i) = \sum_{j=b}^{V_i+U_i} jP_j b_i c_p + \sum_{j=V_i-U_i+b}^{L_i} (j(c_p + c_s) - (V_i + U_i)c_s))P_j r_i \quad (6)$$

2.5   Batch Patch Based on Segmented Dynamic Cache

Recently White and Crowecroft [8] have introduced the concept of optimal batch patch for prefix cache in order to minimize the average stream rate of the server. We present optimal batch patch using dynamic cache based on segmented video. The common clients, before applying to the server for conventional channel RM, are transmitted in batches as one interval. This interval is fixed, symbolized as V_i+U_i, with an optimal patch window W. RM denotes regular channel. It exceeds this window, initiating a new conventional regular channel RM is more efficient than sending the patch within $V_i + U_i$(Fig.4). A length of RM is transmitted as a startup interval of RM, while a non-RM length is transmitted as a non-startup interval. The RM length and the average interval of startup patch, it is that in the interval of two adjacent RM periods random chosen by computer. The average interval of RM length is:

$$R = \frac{(1-P)rW^2 + (1-P)(V_i + U_i)rW + (V_i + U_i)rL}{2(V_i + U_i)W + \frac{2(V_i+U_i)^2}{1-P}} \quad (7)$$

Here $P = P(0)$ denotes the probability of getting 0 request in the batch transmission interval $V_i + U_i$. L is the sustained time of video, r represent the stream rate of video resource, the optimal patch window is achieved by differentiating R and letting the result to be 0 request. The result is:

$$W = \frac{-(V_i + U_i) + \sqrt{P(V_i + U_i)^2 + 2(1 - P)(V_i + U_i)L}}{(V_i + U_i)(1 - P)} \quad (8)$$

The optimal batch patch uses dynamic cache to transmit the request of batch and patch in interval V_i+U_i. The value of b for prefix cache can not be too large, generally in seconds. If the value is too large, that will extend the clients' waiting

time. In proxy cache based on segmented video, by utilizing dynamic cache technique, the total length of prefix and segmented cache in the proxy cache is considered as the unit of batch transmission and patch. If $t_2 < V_i + U_i \leq W$, there exists multiple patch streams with batch of $V_i + U_i$, and the dynamic cache, which is in proxy cache, completes processing the patch within the exact time of the patch streams, then the value from the server to the proxy becomes $1/(1+((V_i+U_i)\lambda_i)$. If $t_2 < W < V_i+U_i$, there exists multiple patch streams with W, and use of length W for dynamic cache in proxy cache, completes processing the patch within the exact time of the multiple patch streams, then the value from the sever to the proxy becomes$1/(1+W\lambda)$. These two cases can both reduce the output from the server and enhance the efficiency of proxy cache.

3   Conclusions

We present the segmented proxy cache technique based on the popularity of video, and extend the length of batch transmission by using dynamic cache. Further more, we put forward several schemes which are based on video segment such as unicast batch transmission, unicast patch, multicast patch, multicast stream merging and optimal batch patch schemes, and study the impact on the transmission cost caused by the choice of transmission schemes. The evaluation exposits that even to relative small proxy cache, elaborately designed transmission scheme, with dynamic cache based segmented video, can produce significant cost saving. Its performance is prior to the optimal prefix cache, proportional priority cache and 0-1 cache schemes.

References

1. C. Aggarwal, J. Wolf, and P. Yu : On optimal batching policies for video-on-demand storage servers. In Proc. of IEEE International Conference on Multimedia Computing and Systems, June 1996.
2. K.L.Wu, P.S.Yu: Segment-Based Proxy Caching of Multimedia Streams. In: Proc. of IEEE INFOCOM, May, 2001.
3. S.Sen, J.Reforrd, D.Towsley: Proxy Prefix Caching for Multimedia Streaming. In: Proc. Of IEEE INFOCOM, Mar.1999.
4. K. Hua, Y. Cai and S. Sheu: Patching: A multicast techniquefor true video-on-demand services. In: Proc. ACM Multimedia, September 1998.
5. D.Eager, M.Vernon and J.Zahorjan: Optimal and efficient merging schedules for video-on-demand servers. In Proc.ACM Multimedia, November 1999.
6. B. Wang, S. Sen, M. Adler and D. Towsley: Proxy-based distribution of streaming video over unicast/multicast connections. In: Tech. Rep. 01-05, Department of Computer Science, University of Massachusetts, Amherst,2001.
7. O.Verscheure, C.Venketramani, P.Frossard, L.Amini: Joint server scheduling and proxy caching for video delivery. In: IBM Technical Report Number RC21981,2001.
8. Paul P.White,Jon Crowcroft:Optimised Batch Patching with Classes of Service. In: ACM Cpmputer Communication Review (2000), Vol 30.