Cache Replacement Algorithm for P2P Media Streaming

Masahiro SASABE Naoki WAKAMIYA Masayuki MURATA Hideo MIYAHARA

Graduate School of Information Science and Technology, Osaka University

1 Introduction

Peer-to-Peer (P2P) is a new network paradigm to solve problems in the client-server architecture. By using the P2P communication technique, media streaming can be expected to flexibly react to changes in network conditions and user demands for media streams in a scalable and robust manner. We have proposed several methods for on-demand media streaming on pure P2P networks where there is no server [1]. Through simulation experiments, we have shown that our mechanisms could accomplish continuous media play-out for popular media streams without introducing extra load on the system. However, we also have found that completeness of media play-out deteriorates as the media popularity decreases. To improve the continuity of media play-out, we propose an effective cache replacement algorithm considering the supply and demand for media streams.

2 P2P Media Streaming

We consider media streaming on pure P2P networks where there is no server. For efficient use of network bandwidth and cache buffer, a media stream is divided into blocks. A peer participating in our system has blocks of media streams it watched or is watching in its cache buffer. It searches a P2P network for a media stream in a block-by-block basis. When it can successfully find blocks cached at other peers, it directly retrieves them from those peers. Then, retrieved blocks are played out at the peer as they arrive.

Since a block-by-block search apparently increases the number of queries in the network and causes the congestion, we proposed two scalable per-block search methods. First, taking into account the temporal order of reference to media blocks, a peer sends a query message for a group of consecutive blocks, i.e., group-by-group search. We call the group as a round. Second, based on the preceding search results, the range of search is dynamically regulated so that only those peers expected to have desired blocks are inquired. On receiving response messages, a peer determines a peer from which it can retrieve blocks in time for continuous media play-out.

3 Bio-inspired Cache Replacement Algorithm

When there is no room to store a newly retrieved block in a cache buffer, a peer must replace cached blocks with a new block. Although LRU is a simple and widely used cache replacement algorithm, it has been proved to fail to accomplish continuous media play-out [1]. In this paper, we propose a bio-inspired cache replacement algorithm considering the balance between supply and demand for media streams. Since there is no server in a pure P2P network, a peer has to conjecture the behavior of other peers by itself. It is not preferred that a peer aggressively collects information about supply and demand by communicating with other peers, because it introduces extra loads on the system and deteriorates the scalability. In our scheme, a peer estimates the supply and demand from locally available information: search results it obtained and messages it relayed. Then, it discards blocks of a media which is considered to excessively exist in a P2P network. As a result, the supply and demand for media streams will be well-balanced in the P2P network. Such a distributed system is also found in an insect society where a highly structured organization is accomplished only through indirect communications among individuals. In particular, a recently proposed model of division of labor in a colony of primitively eusocial wasps can be transformed into a decentralized adaptive algorithm of task allocation [2].

In the model of the division of labor, the ratio of individuals that perform a task is adjusted in a fully-distributed and self-organizing manner. The demand to perform a task increases as time passes and decreases when it is performed. The probability that an individual i performs a task is given by the demand, i.e., stimulus s, and the response threshold θ_i as $\frac{s^2}{s^2+\theta_i^2}$, for example. When the individual i performs the task, θ_i is decreased and thus it tends to devote itself to the task. After performing the task several times, it becomes a specialist of the task. Otherwise, θ_i is increased. Through threshold adaptation without direct interactions among individuals, the ratio of individuals that perform a specific task is eventually adjusted to some appropriate level.

By regarding the replacement of media streams as a task, we propose a cache replacement algorithm based on the division of labor and task allocation model. The proposed algorithm is organized by following two steps.

Step1 Estimate the supply and demand for media streams per round. For a set of cached media streams M, a peer calculates supply $S(i)$ and demand $D(i)$ for media stream $i \in M$ from search results it received and messages it relayed at the previous round. $S(i)$ is the ratio of total number of blocks for media stream i in received and relayed response messages to the number of blocks in media stream i. Here, to avoid the overestimation, only response messages received are taken into account for $S(i)$ when a peer watches stream i. $D(i)$ is the number of query messages for media stream i, which the peer emitted and relayed.

Step2 Determine a media stream to replace. Based on the "division of labor and task allocation", we define probability $P_r(i)$ of replacement of media stream i as follows:

$$P_r(i) = \frac{s^2(i)}{s^2(i) + \theta^2(i) + l^2(i)}, \quad (1)$$

where $s(i)$ is derived as $\max\left(\frac{S(i)-1}{D(i)}, 0\right)$, which indicates the ratio of supply to demand for media stream i after the replacement. $s(i)$ means how excessively media stream i exists in the network if it is discarded. $l(i)$ is the ratio of the number of locally cached blocks to the number of blocks in media stream i. $l(i)$ is used to restrain the replacement of fully or well-cached streams. Among cached streams except for a stream being watched, a victim is chosen based on $P_r(i)$, $i \in M$. Then, a peer discards blocks from the head or the tail of the stream at random. As in [2], threshold values are regulated using Eq.(2). Thus, media i is to be discarded more often once it is chosen as a victim.

$$\forall j \in M,\ \theta(j) = \begin{cases} \theta(j) - \xi & \text{if } j = i \\ \theta(j) + \varphi & \text{if } j \neq i \end{cases} \quad (2)$$

By sequentially replacing blocks of the same media stream, fragmentation of media streams can be avoided.

4 Simulation Evaluation

We randomly generated P2P logical networks with 100 peers. At first, all 100 peers do not watch any media stream. Then, peers randomly begin to request a media stream one by one. The inter-arrival time between two successive requests for the first media stream follows an exponential distribution whose average is 20 minutes. Forty media streams of 60-minute length are available. Media streams are numbered from 1 (the most popular) to 40 (the least popular), where popularity follows a Zipf-like distribution with $\alpha = 1.0$. Each peer watches a media stream without interactions such as rewinding, pausing, and fast-forwarding.

A media stream is divided into blocks of 10-sec duration. Each peer sends a query message for a succession of six blocks. Each peer has a cache buffer whose size corresponds to three media streams. At the start of the simulation, each peer stores three whole media streams chosen following a Zipf-like distribution with $\alpha = 1.0$. Considering the values used in [2], we set the parameters of the cache replacement algorithm as follows: $\xi = 0.01$ and $\varphi = 0.001$. $\theta(i)$ is initially set to 0.5, but it dynamically changes between 0.001 and 1. $s(i)$ is normalized by dividing by $\sum_i s(i)$. To prevent the initial condition of the cache buffer from influencing system performance, we only use the results after the initially cached blocks are completely replaced with newly retrieved blocks for all peers.

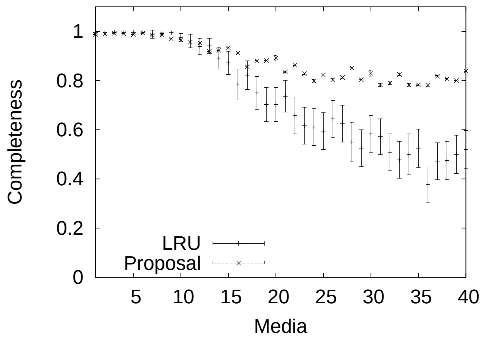

Figure 1 Completeness (LRU vs. Proposal)

We define the completeness of media play-out as the ratio of the number of retrieved blocks in time to the number of blocks in a media stream. Figure 1 depicts the completeness with a 95 % confidence interval of each media stream. We find that our proposed algorithm can improve the completeness of unpopular media streams without affecting popular streams. We conducted several experiments and verified that the improvement could be attained on variety of conditions.

5 Conclusion

In this paper, we proposed a bio-inspired cache replacement algorithm for P2P media streaming. Through several simulation experiments, we have shown that our proposed mechanism can accomplish continuous media play-out independent of media popularity. As future research works, we should evaluate our proposed mechanisms in more realistic situations where a peer randomly joins and leaves our system.

Acknowledgements

This research was supported in part by "The 21st Century Center of Excellence Program" and a Grant-in-Aid for JSPS Fellows.

Reference

[1] M. Sasabe, N. Wakamiya, M. Murata, and H. Miyahara, "Scalable and continuous media streaming on peer-to-peer networks," in *Proceedings of P2P2003*, (Linkoping), pp. 92–99, Sept. 2003.

[2] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press, 1999.