Supporting VCR in Peer-to-Peer Video-On-Demand

Supporting VCR in Peer-to-Peer Video-On-Demand

Yuan He
Hong Kong University of Science and Technology
heyuan@cse.ust.hk

Yunhao Liu
Hong Kong University of Science and Technology
liu@cse.ust.hk

1. INTRODUCTION

Video-On-Demand (VOD) is an interactive multimedia service, which delivers video content to subscribers (users) [2]. Due to the frequent VCR controls from users, such as play, pause, fast forward, fast search, reverse search and rewind, existing approaches either present long latencies on the user side, or incur excessive stress on the server side. In order to provide a "play-as-download" VOD service, stream reuse techniques, such as batching, patching and chaining are proposed. Generally, overlay construction with these techniques is tightly coupled with the peers' playback progress. The stream reusability will be under-utilized unless partnering peers keep persistent connections with each other. As a result, users' experience will be seriously degraded when they take frequent VCR controls. To address this problem, prefetching is also proposed. Different strategies are adopted, such as sequential, random and global rarest strategies [1], but fail to consider the content-based associations inside videos.

We propose a scheme of P2P VOD supporting VCR interactivities. By mining associations inside a video, the segments requested in VCR interactivities are accurately predicted based on the information collected through gossips. Together with a hybrid caching strategy, a collaborative prefetching scheme is proposed to optimize resource distribution among neighboring peers. We evaluate our scheme through extensive experiments. Results show that it is scalable and effective, providing short startup latencies and good performance in VCR interactivities.

1.1 Observations on VOD user behavior

First, VOD users rarely view the movie continuously from the beginning to the end. As peers with transient lifespan account for a large proportion, streaming connections among peers are unstable and altered often. Second, accesses to different periods of a video are not uniformly distributed. Some parts always attract more accesses than the others. Hot scenes are always appreciated by most audience and lead to a consensus on the popularity of a video. Third, VCR interactivities occur frequently. Many VCR controls are forward-looking, due to the users' skips of the video, ignoring the uninteresting periods.

1.2 Associations inside a Video

The technique of association rule mining is used to discover elements that co-occur frequently within a data set consisting of multiple selections of elements. Considering the case of VOD, a video can be regarded as a sequence of media segments. Due to the difference in individual interests, the segments watched might vary a lot with different users. However, segments of a video are not completely independent to each other. In other words, if the *x-th* and the *y-th* segment are strongly associated, users who watch the *x-th* segment will probably watch the *y-th* segment. Such associations can be inferred from the history information of VOD user behavior.

2. SYSTEM DESIGN

2.1 Policies of Streaming and Caching

Our design requires that each peer keeps two types of downloading: urgent downloading and prefetching. As for the management on buffer, a hybrid caching strategy is adopted. Every peer caches the latest 5 minutes of the video played and uses the Least-Recently-Used (LRU) policy for cache replacement. Moreover, all peers hold the initial 5 minutes of video and never replace this part before departure.
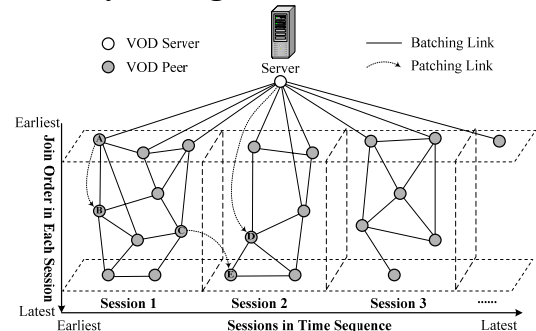
2.2 Overlay Management
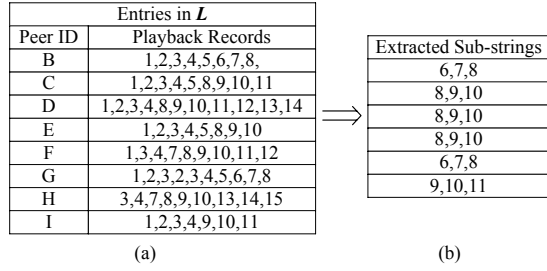

Figure 1. Batching and patching

Entries in _L_

Peer ID	Playback Records
B	1,2,3,4,5,6,7,8,
C	1,2,3,4,5,8,9,10,11
D	1,2,3,4,8,9,10,11,12,13,14
E	1,2,3,4,5,8,9,10
F	1,3,4,7,8,9,10,11,12
G	1,2,3,2,3,4,5,6,7,8
H	3,4,7,8,9,10,13,14,15
I	1,2,3,4,9,10,11

Extracted Sub-strings
6,7,8
8,9,10
8,9,10
8,9,10
6,7,8
9,10,11

(a)   (b)

Figure 2. Association rule mining

As illustrated in Fig. 1, the VOD server _S_ uses _batching_ to serve asynchronous peers. Every _m_ minutes the server _S_ triggers a new _batching session_ to broadcast the video from the beginning, where _m_ is a predefined _session width_. The server _S_ allocates a certain amount of dedicated outgoing bandwidth for each batching session. Early joining peers become direct children of the server. After the allocated bandwidth is fully occupied, late peers are redirected by the server and become the descendants of the early ones. We also reinforce the batching scheme with _patching_. When a peer joins in a session late and misses the initial part of the video, it picks up a few existing peers as _patching sources_ and immediately starts to download the missing part.

2.3 Mining the Association Rules

While playing, a peer maintains a _playback record_ of its playback history. Peers exchange their records through gossips, so that the state of peers is efficiently propagated and updated throughout the networks. Based on the information collected from other peers, a peer dynamically predicts its future behavior using the technique of association rule mining. We find all the association rules that have a support and a confidence greater than specified thresholds. As shown in Fig. 2, we get two eligible association rules for a peer who just plays segments 1, 2 and 3. They are (1, 2, 3)→(6, 7, 8) and (1, 2, 3)→(8, 9, 10). Segments appearing most frequently in the extracted sub-strings, such as 6, 7, 8, 9 and 10 in Fig.2, become the potential targets of prefetching. Afterwards, neighboring peers prefetch these segments using a collaborative strategy. In the scope of neighborhood, every peer respectively counts the copies of each prefetched segment and performs prefeching according to the _local-rarest-first_ policy. Due to the constraints in time and bandwidth, a peer is probably unable to download all the segments in the prefetching set before they are requested. Using the _local-rarest-first_ policy, the probability is maximized for a peer to find the requested segments locally.
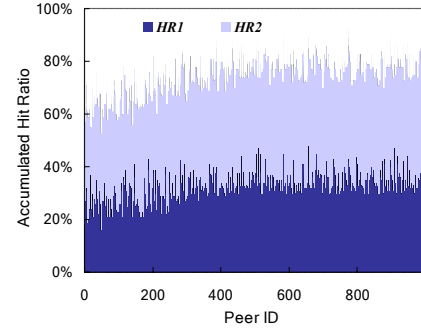

Figure 3. Accumulated hit ratio

3. ANALYSIS AND EVALUATION

We theoretically compare the hit ratios produced by random prefetching and our scheme. Results show that $\frac{HR1}{HR1'} = \frac{HR2}{HR2'} \approx \frac{|V|}{|V_i|}$, where _HR1_ and _HR2_ of our scheme respectively represent the percentage of VCR requests satisfied by the segments prefetched on the current peer and those prefetched by the neighbors. _HR1'_ and _HR2'_ represent the counter parts in random prefetching. V = {all the segments possibly to be requested by the VCR control}. V_i is the prefetching set of peer _i_. As usually $|V_i| << |V|$, we can clearly see the advantage of our scheme over random prefetching. Moreover, we have _HR1_+ _HR2_ ≈ 1, which means our scheme is able to provide near-optimal hit ratios in the scope of neighboring peers.

We evaluate the proposed scheme through extensive experiments. The results are listed below: First, Fig. 3 plots the accumulated hit ratio. Superior to existing approaches, our scheme provides accurate and efficient prefetching for P2P VOD. Second, the server is capable to serve much more concurrent users than its original capacity. Third, our scheme provides short and equitable latencies in startups and VCR interactivities. Detailed analysis and further experimental results can be found in our technical report.

REFERENCES

[1] S. Annapureddy, S. Guha, C. Gkantsidis, D. Gunawardena, and P. Rodriguez, "Is High-Quality VoD Feasible using P2P Swarming?" in Proceedings of _ACM WWW_, Banff, Alberta, Canada., 2007.

[2] C. Huang, J. Li, and K. W. Ross, "Can Internet Video-on-Demand Be Profitable?" in Proceedings of _ACM SIGCOMM_, Kyoto, Japan, 2007.