Probabilistic Prefetching Scheme for P2P VoD Applications with Frequent Seeks

Yifeng He	extsuperscript{1}, Guobin Shen	extsuperscript{2}, Yongqiang Xiong	extsuperscript{2} and Ling Guan	extsuperscript{1}

	extsuperscript{1}Department of Electrical and Computer Engineering, Ryerson University, Toronto, Canada

	extsuperscript{2}Wireless and Networking Group, Microsoft Research Asia, Beijing, P. R. China

Abstract—In Peer-to-Peer Video-on-Demand (P2P VoD) applications, users tend to seek to the positions that they are interested in. The frequent seeks raise a great challenge to the design of the prefetching scheme. In this paper, we propose a probabilistic prefetching framework to reduce the seeking distance. Each peer performs prefetching based on the segment access probability, which is estimated from the seeking statistics in the previous sessions. It is a challenging task to collect the seeking statistics in a distributed P2P network. In the proposed framework, we employ FM sketches to represent the seeking statistics, thus greatly reducing the space and time complexity. The simulation results show that the proposed prefetching scheme can approach closer to the desired seeking positions compared to the prefetching scheme neglecting the user viewing pattern.

I. INTRODUCTION

With advances in broadband Internet access technology and the coding techniques, media streaming services have become increasingly popular. Video-on-Demand (VoD) streaming is one such service where videos are delivered to asynchronous users with minimal delay and free interactivity. Recently, peer-to-peer (P2P) technology has become a promising approach to provide VoD streaming service [1, 2] to a huge number of the users over the global area.

Most of the existing work on P2P-based VoD systems has made an implicit assumption that a user who has joined a streaming session will keep on watching till it leaves or fails the session. Unfortunately, based on the analysis of a large amount of real user viewing logs, we found that this is not the truth [3]. Users usually do not play the video successively and passively. Instead, users perform seeks quite frequently. The reasons for these seeking behaviors are: 1) some users feel that the current segment are boring and jump away from it; 2) some users do not have time to watch the whole video and just want to browse some exciting segments.

In P2P VoD applications, each peer caches some segments in its buffer or replicates some segments in its storage such that it can contribute these segments to other peers. In order to combat the traffic congestions or dynamic departures of peers, a prefetching scheme can be used to increase the playback continuity [4]. However, most of the existing systems assume that peer will watch the video normally without any seek. Therefore, peers do sequential prefetching [4, 5]. However, in the VoD applications with frequent seeks, the next position the peer will access may not be the segment next to the current segment, it will be probably any other segment in the whole video. Therefore, the prefetching scheme in P2P VoD systems with frequent seeks needs to be re-examined.

A peer has no knowledge about the content when it watches a video at the first time. Therefore, it performs seeks randomly. In that case, we cannot predict the access probability at a given moment. In this work, we introduce a concept of guided seek. In the VoD systems with guided seeks, each peer can learn the segment popularity from the previous seeking statistics, and then perform seeking based on this information. For example, a user viewing a video of soccer match may just want to watch the segments with goals. However, he or she does not know which segments are the scenes with goals. In the VoD systems with guidance, the system can provide the segment-popularity bar in the media player, with grey scale indicating the degree of the popularity for each segment. This user can just drag the sliding-bar to the segments with dark color since these segments will contain the goal scenes with a higher probability.

We can guide the user seeks and design a prefetching scheme based on the segment access probability, which is a two-dimensional probability density function (PDF), denoted as \(P(x,y)\), where \(x\) is the start segment of a seek and \(y\) is the end segment of a seek. \(P(x,y)\) can be learnt from the user seeking statistics in the previous sessions. In a centralized VoD system, a server can be used to collect the user seeking statistics. However, this approach may overload the burden of the server. Collecting user seeking statistics in a distributed P2P network is a challenging task.

In this paper, we propose a probabilistic prefetching framework in P2P VoD systems with frequent seeks. The probabilistic prefetching scheme is based on the segment access probability, which is a summary of user seeking
Probabilistic prefetching scheme

Fig. 1. The proposed prefetching framework in P2P VoD systems with frequent seeks

statistics in the previous sessions. In order to obtain the segment access probability in a distributed P2P network, we employ Flajolet-Martin (FM) [6] sketches to represent the seeking statistics to reduce the space and time complexity. The seeking statistics are collected at each peer via gossip protocol.

The proposed framework is shown in Fig. 1. It consists of two modules: the statistics collection and probabilistic prefetching scheme. The module of statistics collection collects user seeking statistics, which are used to estimate the segment access probability \( P(x, y) \), based on which the prefetching scheme will determine and prefetch those segments which are most likely to be accessed next. The segment access probability is also fed back to the user, such that the user can perform efficient seeks with the guidance of segment popularity information.

II. STATISTICS COLLECTION

Gossip protocol is one of the efficient approaches to aggregate the information in the P2P networks [7]. Gossip occurs periodically. In each round of random gossip, every peer talks to one or more randomly selected neighbors and exchanges the information with them. It turns out that, after approximately \( \log N \) rounds of computation where \( N \) is the number of the peers, all the peers can obtain the global information with a high probability [8].

Suppose a video is uniformly divided into \( M \) segments, each segment lasts a length of \( t \). We assume a segment is the smallest unit in the time scale. A peer can either playback a segment or skip it. A seeking behavior \( s(x, y) \) is characterized by the start segment, denoted by \( x \), and the end segment, denoted by \( y \) \((y \neq x)\). The sequential playback is a special seeking behavior, in which \( y = x + 1 \). A seeking behavior \( s(x, y) \) contains two variables. It is inefficient in terms of message size. Therefore, we map a seeking behavior into a corresponding seeking type \( v_i \in \{1, 2, ..., M(M - 1)\} \), which is given by

\[
v_i = \begin{cases} 
(M - 1)(x - 1) + y, & \text{if } y < x, \\
(M - 1)(y - 1) + y - 1, & \text{if } y > x.
\end{cases}
\]  

A peer collects the seeking statistics in the P2P network and put them into a data stream. A data stream at peer \( k \) is denoted by \( R_k \), which contains a series of seeking records \( R_k = \{v_i\} \). Each seeking record is denoted by a pair \( v_i = (v, u_i) \), where \( v_i \) is a seeking type, \( u_i \) is a unique record ID associated with this seeking type \( v_i \) and it can be formed as the concatenation of the unique peer ID and the sequence number issued by this peer. In gossip-based information aggregation, there are risks that a peer may receive duplicate information. By carrying the record ID in each seeking record, we can avoid the double-counting problem.

A. Intuitive approach

We first look at an intuitive approach, in which each peer exchanges its data stream with other peers via gossip. In the intuitive approach, the procedure of information aggregation between two peers (e.g., peer 1 and peer 2) is as follows. First, peer 1 sorts its data stream \( R_1 \). Second, peer 1 searches each seeking record in peer 2, and puts it into \( R_1 \) if it has not appeared in \( R_1 \).

The intuitive approach can accurately aggregate the information. However, the space complexity and time complexity is large. Let \( S \) denote the number of seeking records in \( R_i \), \( S' \) denote the number of distinct seeking records in \( R_i \). For the information aggregation between two peers (peer 1 and peer 2), the time complexity at peer 1 is \( O(S_1 \log S_1' + S_1' \log S_2') \). The space required to store the data stream \( R_i \) is \( O(S_1' \log S_1') \), indicating the space is linearly scaling with the number of the distinct seeking records.

B. FM sketch

To reduce the space and time complexity, we apply sketches to represent the data stream at each peer. Sketch is a compact representation of data stream. It prevents the space from scaling linearly.

FM sketch was introduced for estimating the number of distinct objects in a multi-set in one pass while using only a small amount of space [6]. Given a multi-set \( W \), the FM sketch \( FM_w \) of \( W \) is a bitmap of \( L \) bits. The bitmap length \( L \) is \( \log_2(n) \), where \( n \) is the number of distinct elements in the multi-set. The bit of \( FM_w \) is denoted by \( FM_w[l] \), \( l = 1, ..., L \). All the bits in \( FM_w \) are initialized to zero. A uniform hash function \( h(x) \) maps each element \( \omega \in W \) to \( h(\omega) \in [0, ..., 2^l - 1] \). Let \( I(\omega) \) denote the position of the least-significant 1-bit in \( h(\omega) \). If \( h(\omega) = 0 \), we allocate \( I(\omega) = L \). For each element \( \omega \in W \), we set \( FM_w[I(\omega)] = 1 \). For example, if \( L = 5 \), \( h(\omega) = 6 = 00110 \) (in binary), then \( I(\omega) = 2 \). If \( \omega \) is the first element inserted into \( FM_w \), the FM bitmap will be \( FM_w = 000010 \). Since duplicate elements just set the same bit, FM sketch is duplicate-insensitive.
To estimate the number of distinct elements from an FM sketch, the following approach is used. FM finds the first bit of the sketch that is still 0. Let the position of this bit be \( c \). Then the number of distinct elements is estimated as \( n = 1.2928 \times 2^{c} \) [6]. To reduce the approximation error, Flajolet and Martin proposed a technique called Probabilistic Counting with Stochastic Averaging (PCSA) [6]. PCSA applies a second hash function to choose one of the \( m \) bitmaps and performs the element insertion only to this bitmap. As a result, each bitmap is responsible for approximately \( (n/m) \) distinct elements. The number of distinct elements with PCSA is given by \( n = 1.2928 \times m \times 2^{\frac{c}{\log_{2}m}} \) [6].

We use FM sketches to represent the seeking statistics. Each peer (e.g., peer \( i \)) maintains a set of FM sketches. Each FM sketch is responsible for recording the occurrence frequency of a seeking type. The space complexity at each peer is \( O(D \cdot m \cdot \log_{2} n_{u}) \), where \( D \) is the number of the seeking types, \( m \) is the number of the bitmaps for each FM sketch, and \( n_{u} = \max\{n_{1}, \ldots, n_{m(D-1)}\} \) where \( n_{i} \) represents the number of distinct seeking records having seeking type \( i \). Peer \( i \) periodically updates its FM sketches as follows. First, peer \( i \) inserts the data stream generated by itself since last update-time into FM sketches. For each seeking record \( e_{i} = (v_{i}, u_{i}) \) from the newly generated data stream, we first locate FM sketch corresponding to seeking type \( v_{i} \), then we insert record ID \( u_{i} \) into this FM sketch. Second, peer \( i \) asks from the tracker for \( K \) random neighbors, and then performs pair-wise gossip with each of the \( K \) neighbors respectively. In the pair-wise gossip with peer \( j \), peer \( i \) gets the FM sketches from peer \( j \), and updates each of its FM sketches by performing bit-wise OR operation. The time complexity for each round of pair-wise gossip is \( O(D \cdot m) \).

C. Computation of segment access probability \( P(x,y) \)

After peer \( i \) has collected sufficient seeking statistics, it can compute the segment access probability \( P(x,y) \). First, peer \( i \) estimates the occurrence frequency for each seeking type from the FM sketches. For each seeking type \( v_{j} \), there is a FM sketch associated with it. So we can estimate the occurrence frequency \( c_{v_{j}} \) for each seeking type \( v_{j} \), respectively. Each seeking type is corresponding to a seeking behavior \( s(x,y) \). Therefore, we actually have the occurrence frequency for each seeking behavior \( s(x,y) \). Let \( f(x,y) \) denote the occurrence frequency of seeking behavior \( s(x,y) \). The seeking probability \( P(x,y) \) from segment \( x \) to segment \( y \) is given by

\[
P(x,y) = \frac{f(x,y)}{\sum_{m,n} f(m,n)}.
\]

The conditional seeking probability given that the current segment is \( y \) is given by

\[
P(y|x = y) = \frac{P(y,x)}{\sum_{y} P(y,x)}.
\]

III. PROBABILISTIC PREFETCHING SCHEME

Intuitively, each peer would like to prefetch the segments as many as possible. However, due to the limitation of bandwidth and buffer capacity, each peer can prefetch at most \( T (T \leq M) \) segments.

When a peer is watching segment \( y \), the probability that which segment will be accessed next is given by the conditional probabilities \( P(y|x = y) \), where \( y = 1, \ldots, M \). At most of the moments, the peer will watch the segments sequentially. Therefore, segment \( (y+1) \) typically has the highest conditional probability among all the segments. In our prefetching scheme, we prefetch \( T \) segments which have the highest conditional probabilities, because these \( T \) segments are most likely to be accessed next in the VoD systems with guidance. When a seek occurs, the prefetched segment that is closest to the desired destination is scheduled and played without any delay. There is a deviation between the desired seeking position and the scheduled position. This deviation is defined as seeking distance, which will be used to evaluate the performance of a prefetching scheme.

Since the prefetched segments depend on the current viewing position \( y \), they need to be updated from segment to segment. When the peer moves into a segment, it first determines the desired prefetched segments, and then it needs to quickly locate and download the segments that have not appeared in the buffer. In our framework, each peer actually can collect the information of segment locations during the gossip. Each peer can maintain a segment-location table, which has three entries: the segment No., IP address, and the updated time. With the assistance of the segment-location table, the peer can quickly locate the segments for prefetching. Moreover, since every peer follows the similar guidance, it turns out that the popular segments are easy to be located and hence prefetched more quickly.

IV. SIMULATIONS

In the simulations, we choose a video clip of 30 minutes. The video is evenly divided into 90 segments. The segments in the video follow Zipf distribution. We generate 1000 logs from 1000 previous users. Each session has 20 seeks in average. A new peer collects the seeking statistics from the 1000 logs and estimate the segment access probability \( P(x,y) \). Then the new peer performs seeks following the guidance of \( P(x,y) \). In the proposed framework, the seeking records are represented by FM sketches. Each FM sketch consists of 4 bitmaps, each of which has a length of 11 bits.

The new peer can collect a part or all of the seeking statistics from the 1000 logs. We define a sampling ratio \( \alpha = N_{p}/N \), where \( N_{p} \) is the number of the peers whose seeking statistics have been collected, \( N \) is the number of
We propose a probabilistic prefetching framework to reduce the seeking distance in P2P VoD applications with frequent seeks. In the proposed framework, each peer performs prefetching based on the segment access probability, which is estimated from the seeking statistics in the previous sessions. It is a nontrivial task to aggregate the seeking statistics in a distributed P2P network. We employ FM sketches to represent the seeking statistics, thus greatly reducing the space and time complexity.

V. CONCLUSIONS

We propose a probabilistic prefetching framework to reduce the seeking distance in P2P VoD applications with frequent seeks. In the proposed framework, each peer performs prefetching based on the segment access probability, which is estimated from the seeking statistics in the previous sessions. It is a nontrivial task to aggregate the seeking statistics in a distributed P2P network. We employ FM sketches to represent the seeking statistics, thus greatly reducing the space and time complexity.

REFERENCES