Towards Health of Replication in Large-Scale P2P-VoD Systems

Haitao Li, Xu Ke
Tsinghua National Laboratory for Information Science and Technology
Dept. of Computer Science and Technology
Tsinghua University, Beijing, China
E-mail: {lihaitao, xuke}@csnet1.cs.tsinghua.edu.cn

James Seng
PPLive R&D, Shanghai Synacast Media Tech, Shanghai, China
E-mail: james.seng@pplive.com

Po Hu
China Academy of Telecommunication Research (CATR)
The Ministry of Industry and Information Technology (MII)
Beijing, China
E-mail: hupo@mail.ritt.com.cn

***Abstract*—Unlike P2P-based live streaming, P2P-based video-on-demand (P2P-VoD) has asynchrony in sharing video content among users. To compensate, each peer is expected to contribute a fixed amount of disk space, which forms a distributed P2P storage system. How to regulate this storage system is undoubtedly one of the most important strategies in P2P-VoD systems. PPLive, a large-scale P2P-VoD system, has made great efforts to optimize replication by improving file replacement algorithm, since developed in the fall of 2007. In this paper, we introduce the replacement algorithms PPLive takes and compare their performance when the user request patterns are in a steady state and when they change sharply. These real-world experiences will provide valuable insights into what a healthy replication is and the future design of replication strategy.**

Keywords-Replication; P2P; VoD; Ralacment Algorithm;

I. INTRODUCTION

In recent years, P2P has been applied to reduce bandwidth load for content providers and thus support large-scale Video-on-Demand (VoD) systems [1, 2, 3, 4, 5]. Unlike streaming live content, P2P-VoD has less synchrony in sharing video content among users. Therefore the number of videos that can benefit from P2P comes out relatively small if users just cache temporarily in its memory what they are watching and share them with others. For example in Youtube, even when users share videos for a longer period of time (e.g., 1 day), P2P just assist 60% of videos with at least 10 current users all the time [6]. To compensate, each peer is required to contribute a fixed amount of hard disc storage (e.g., 1GB). A peer watches and at the same time stores video files in its local contributed storage if there is free space. It shares all videos stored in its local contributed storage, including those currently unviewed. The entire viewer population thus forms a distributed P2P storage (or file) sharing system. How to regulate this storage system is undoubtedly the most critical part of the P2P-VoD system, because proper replica distribution among peers' shared disks is the precondition to discover and transmit the desired contends efficiently with each other.

Obviously the replica distribution is mainly determined by users' viewing behaviors, but two main methods can be used to regulate replica distribution over peers: replacement algorithm and pre-fetching [8]. But first, it is critical to know what a healthy replication is if the target of the healthy replication is to make the video data as available as possible to peers by changing the replication distribution over peers.

One method is to study the performance of these algorithms through simulation or mathematic model. But complicated real-world P2P environment always makes their conclusions useless in practice. Another method is to directly deploy pre-fetching and replacement algorithms in real system, and compare their system performance. However, it is usually cost for a commercial product to develop in real world an algorithm that is not well-understood. Existent experiences will be helpful to reduce the experiment cost.

In this paper, we conduct an in-depth study of replication health based on a real-world P2P-VoD system deployed by PPLive in the fall of 2007. As of late November 2008, the number of simultaneously online videos exceeds 4300, with the average video length 59 minutes and the average bitrate 395 Kbps. The peak number of simultaneous peers is around 402000. The peak traffic of all simultaneous online peers is around 158.677 Gbps. Since deployment, PPLive has made great efforts to optimize replication by improving replacement algorithms. These experiences will provide valuable insights into what a healthy replication is and the future design of replication strategy. The contributions of this paper are as follows:

- We define several replication characteristics such as how complete of local replica, and owning ratio of each chunk (unit for storage and advertisement, e.g., 2MB) of the video. We study how these replication characteristics affect the system performance, mainly

by regulating their values and observing the change of system performance in PPLive.

- We propose a decentralized replacement algorithm framework—*PPR*, which considers comprehensive video properties, including video popularity, the number of replicas, how complete of local replica, and owning ratio of each chunk of the video. Comparing native (Least Recently Used) LRU replacement algorithm, bandwidth cost of servers is reduced by several times. However, the cost of collecting such information is limited.
- We present the limitations of replacement algorithm in regulating replication distributions, when the user access patterns change sharply. To overcome these limitations, we give some alternative choices.

This paper is organized as follows. Section II describes related work. Section III gives an overview of PPLive. Section IV defines three *replication health indexes* and points out the system metric used to evaluate these indexes. In Section V, by comparing the performance of three replacement algorithms, we make some conclusions about the impacts of *replication health indexes* on system performance metric *BSR*. We present the limitations of replacement algorithm in regulating replications when the user access patterns change sharply and discuss alternative methods in Section VI. In Section VII, we conclude and present future work.

II. RELATED WORK

P2P-VoD systems have gained great attentions in the recent three years and great efforts have been made in designing effective P2P-VoD systems [7, 8, 9, 15, 16]. However, they mainly focus on peer selection and piece selection strategies and studies on replication strategy in P2P-VoD systems are few. Huang et al. [7] made a first study on replication strategy in large-scale P2P-VoD systems. They pointed out three replication design issues and the original choice of PPLive. The first design issue is taking Multiple Video Caching (MVC) or Single Video Caching (SVC). SVC means a peer only redistributes its currently watching video, while MVC means a peer can store and redistribute a video which was previously viewed but is not currently played. They believed the design of SVC is simpler, but MVC is more flexible in satisfying user demands and thus is the choice by PPLive. The second issue is whether to pre-fetch. Without pre-fetching, only videos viewed locally could be found in a peer's disk cache. The design choice of PPLive is not pre-fetching, because they think it may waste precious bandwidth resource. The third issue is disk replacement algorithm. The favorite choices by many caching algorithms are least recently used (LRU) or least frequently used (LFU). LRU is the original choice of PPLive.

Although Huang et al. gave brief comments on each replication issue, whether their choices work well and how well these choices still need further researches. The first two issues were studied in [8, 9]. Cheng et al. [8] deployed and compared two cache strategies SVC and MVC in real system and showed how MVC improve both scalability and user experience. Cheng et al. [9] proposed a pre-fetching method-lazy replication to improve the replication health. In this paper, we achieve this by improving the replacement algorithm, which has been deployed in PPLive and alleviated the server's bandwidth load by several times than the previously used algorithm-LRU algorithm. Moreover, none of previous works give conclusions on the relationship between replication proprieties and the system performance.

Besides the works on replication in P2P-VoD systems, there are also studies on how many replicas should be allocated for each request object in P2P file sharing system. Lv et al. [10] and Cohen et al. [11] studied optimal replication in an unstructured P2P network in order to reduce random search times. Tewari et al. [12][13] showed that having the number of replicas of each object proportional to the request rate for these objects has both per-node and network-wide advantages for P2P networks. Kangasharju et al. [14] introduced an optimization methodology for maximizing file availability in peer-to-peer content distribution and showed that a simple Top-K Most Frequently Requested (Top-K MFR) algorithm almost always achieves optimal performance. Allen et al. [15] examined LRU and LFU caching algorithms for VoD services in a relatively stable environment, such as cable networks.

Our work is different from these studies in the following three aspects. First, we have different optimization targets. We try to download as much as possible from peers rather than from servers on the condition that users can watch the video smoothly. Secondly, previous work only focuses on the replica distributions from quantity aspect, but ours also considers the replica quality, such as how complete of video replicas and balance of chunk distributions in a video, which play important roles for chunk sharing in P2P-VoD system. Thirdly, different research methods are used. Previous works use mathematic model or simulations while we mainly base on development experience of a real-world large-scale system.

III. PPLIVE OVERVIEW

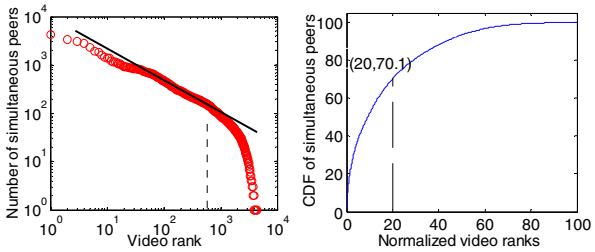
PPLive is one of the leading P2P streaming solution providers in mainland China. Table 1 shows system basic feature of PPLive during our measurement in November, 2008. It shows much difference with user generated content (UGC) video sites such as YouTube [6]. The average video bitrate is around 400Kbps, which is higher than that in YouTube (around 300Kbps to 400Kbps). The average video length is about an hour, which is much bigger than that in

YouTube (around 3 to 5 minutes). However, the number of video is much smaller than that in YouTube.

TABLE I. BASIC FEATURE

Metric	Value
Mean simultaneous peers	206157
Peak simultaneous peers	402734
Video bitrate (Kbps)	101~2024(394)
Video Length (minute)	3.6~298(59)
Number of Videos	4403

When deciding which video will be the next one to be replaced, replacement algorithm of PPLive will consider its popularity as an important factor. Fig. 1(a) shows the number of simultaneous peers against video rank. It exhibits power-law behaviour (a straight line in a log-log plot) across less than three orders of magnitude. However, it shows a sharp decline for the unpopular videos. This truncation at the tail is also evident for YouTube [6]. If the truncated tail is caused by some removable bottlenecks such as recommendation mechanism, the company will gain potential commercial benefit to remove them. Fig. 1(b) shows the CDF of simultaneous peers against video ranks. The horizontal axis represents the popularity of videos, with video ranks normalized between 1 and 100. The graph shows that the top 20% popular videos account for nearly 70% views. It shows a smaller skew than that of YouTube [6], in which the top 10% popular videos account for nearly 80% of views. The strategy of regularly replacing the least requested videos in PPLive might lead to this smaller skew.



(a) Simultaneous peers of each video (b) CDF

Figure 1. Video popularity distribution

IV. REPLICATION HEALTH

A healthy replication is to make the video data as available as possible to peers by changing the replication distribution over peers. In this section, we define three replication health indexes and point out the system metric used to evaluate these indexes.

A. Replication Health Index

Huang et al. [7] used the owning ratio of chunk i (OR_i) and available to demand ratio of video m (ATD_m) to evaluate the replication health of the systems, the definition of which are shown in Formulae (1, 2), where S_m is the size of video m in chunk. However, they overlooked the quality of replicas such as how complete of video replicas, which sometimes is more important than the quantity. Besides, they ignored the need of replicas are also determined by video popularity.

$$OR_i = \frac{Number\ of\ replicas\ of\ chunk\ i}{Number\ of\ video\ owners} \quad (1)$$

$$ATD_m = \frac{\sum_{i=1}^{S_m} Number\ of\ replicas\ of\ chunk\ i}{S_m * Number\ of\ online\ viewers\ of\ video\ m} \quad (2)$$

The qualities of files replicas stored in local disks are different owing to user behaviors such as seeking and unfinished viewing. For example, some files are constructed of sequential chunks, but some have many holes in them. Some files are completely stored, but some just have one or two chunks of a video. Therefore, we introduce another metric-- *Integrity* to reflect the replication health.

$Integrity_m$ reflects the completeness of replicas of video m in system-width. Let S_m be the size of video m in chunk, and then we have

$$Integrity_m = \frac{\sum_{i=1}^{S_m} Number\ of\ replicas\ of\ chunk\ i}{S_m * number\ of\ owners\ of\ video\ m} \quad (3)$$

B. Evaluation Metric

The replication health indexes are just the properties of replication that might affect replication health. Our aim is to find a system evaluation metric, which can straightly reflect the replication health. Then we will try to change the values of these replication health indexes and see effect on this metric. It is different with file sharing, whose target is the minimal download time. For P2P-VoD system, it is expected that all peers download at a speed faster than playback bitrate. It is not expected that some download very fast and some download slow than video bitrate.

In PPLive P2P-VoD systems, a peer will prefer to download data from other peers. And only when the neighboring peers cannot supply sufficient download rate, the content server will be used to supplement the need. As a healthy replication is to make the video data as available as possible to peers by changing the chunk distribution over peers, the fraction of download from peers, or *server bandwidth saving rate (BSR)* can be used to evaluate the replication health. If all conditions other than replication strategy are similar across comparing times, a bigger *BSR* means a better replication. Metric *BSR* also describes the ability to alleviate servers' bandwidth load. And the bigger is *BSR*, the more bandwidth of servers is reduced.

BSR_{mn} is defined as the fraction of bytes of video m downloaded by peer n from other peers out of the whole bytes from both others peers and servers. Let M be the video number of the system, N_m be the number of simultaneous online viewers of video m and N be the number of simultaneous online viewers in the system, and

then the *BSR* of the video *m* (BSR_m) in system-width is:

$$BSR_m = \frac{1}{N_m} \sum_{n=1}^{N_m} BSR_{mn} \quad (4)$$

The *BSR* of the system is:

$$BSR = \frac{1}{N} \sum_{m=1}^{M} (N_m * BSR_m) \quad (5)$$

V. Replacement Algorithm Improvement

Many efforts have been made to improve the replacement algorithm in PPLive P2P-VoD systems. PPLive adopted native LRU algorithm before April 27th, 2008. From April 28th to August 5th, it adopted the *PPLive Replacement* I (*PPR* I) algorithm to regulate *ATD* among videos with different popularities. Since August 6th, 2008, the *PPLive Replacement* II *(PPR* II) algorithm has been adopted to regulate *OR* and *Integrity*. Comparing the performance of these replacement algorithms, we provide insights about the effect of *replication health indexes* on replication health metric *BSR*.

A. Data Collection Methodology

To know the algorithm performance and the relationship between *replication health indexes* and *BSR*, a log server and trackers are responsible to collect replication related reports from peers since April 5th, 2008. Data collection works as follows. The information about which chunks a peer has is kept in a *Chunk Bitmap*. Each peer regularly (e.g., 5 minutes) sends messages to a tracker to report (Video ID, *Chunk Bitmap,* Integrality) of all its replicated videos and sends (Currently Viewing Video ID, *BSR*) to the log server. Using these reports we can count the number of simultaneous online viewers, system-width *Chunk Bitmap* and system-width *Integrity* of each video and see their evolutions every 5 minutes. *ATD* can be calculated from the number of simultaneous online viewers and system-width *Chunk Bitmap. OR* can be calculated directly from *Chunk Bitmap.*

B. Statistics Across Compared Days

In a widely deployed system like PPLive, it is difficult to let all other conditions across compared days compared files or compared peers similar. To know whether the difference in performance is mainly due to the change of replacement algorithm rather than something else, Table 2, Fig. 2 and Fig.3 give the statistics across three compared days, April 26th 2008, May 7th 2008, and October 5th 2008. The main differences are that there are more simultaneous online peers and videos, but more unpopular videos on October 5th. Video popularity distribution is a more important factor that affects the system performance. We find there is not evident change of disk utility distribution across compared days. We also find more than 70% users have more than 99% disk utility, which means replacement algorithm really takes place in a large fraction of peers.

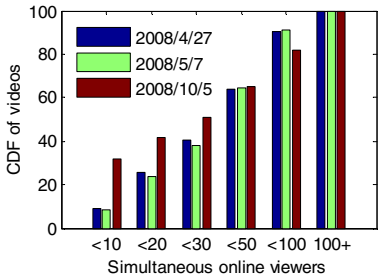


Figure 2. Video popularity distribution across compared days

TABLE II. Basic Statistics Across Compared Days

	April 27th	**May 7th**	**October 5th**
Videos Number	700	700	4300
Average bitrate	380Kbps	380Kbps	395Kbps
Online peers	210000	220000	380000

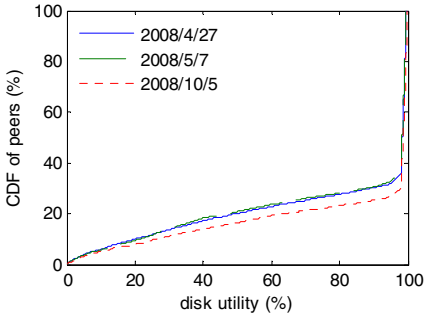


Figure 3. Disk utility across compared days

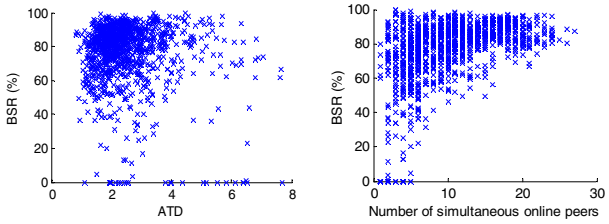
C. Native LRU Replacement Algorithm

Native least recently used (LRU) is the first version of replacement algorithm PPLive P2P-VoD system adopts. It shows some performance deficiency.

1) How to Implement

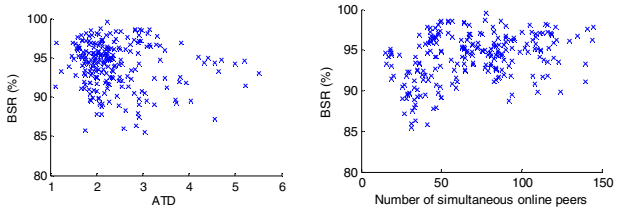
Native LRU works as follows. The peer maintains a list to record local videos. If a video is watched and downloaded, it will be linked to front of the list. If a peer watches a video again before it is deleted, it will be fetched to front of the list. If there is no free space, the peer will choose the video from tail of the list and delete all its chunks.

2) Performance



(a) *ATD* vs. *BSR* (b) Online peers vs. *BSR*

Figure 4. An unpopular video



(a)ATD vs. BSR (b) Online peers vs. BSR

Figure 5. A popular video

Finding that unpopular videos can not be watched smoothly and their *BSRs* are quite small, we try to explore the relationship of *BSR* with *ATD* and video popularity. We collect data of 50 videos of April 26th and 27th 2008, and illustrate *BSR* trend as the increase of *ATD* and the number of online viewers respectively. The results are different for popular and unpopular videos as shown in Fig. 4 and Fig. 5. For unpopular videos (e.g., less than 30 online viewers), the performance is improved with the increase of online viewers, but *BSR* exhibits no obvious trend as the increase of *ATD*. For popular videos (e.g., more than 50 online viewers), the *BSR* exhibits no obvious trend as the increase of *ATD* and online viewers.

3) Insights

Native LRU replacement algorithm has limitations in regulating replication among popular and unpopular videos. Generally, the replicas are redundant for popular videos, and deficient for unpopular videos. For unpopular videos, the performance is improved with the increase of online viewers. For popular videos, the *BSR* exhibits no obvious trend as the increase of *ATD* and online viewers.

Increasing properly replicas for unpopular videos and decreasing replicas for popular videos will improve the performance of unpopular videos, without performance decline for popular videos. Thus, the performance of the whole system will be improved.

D. PPR I

The target of *PPR* I, which is developed on April 28th, is to regulate replicas distribution among popular and unpopular videos, by increasing *ATD* for unpopular videos.

1) How to Implement

PPR I works as follows. A peer records a *Replacement Index* (*RI*) for each local file. If there is no space available in local contributed disk, the local file with the biggest *RI* will be first chosen and all its chunks will be deleted. In our implementation, Formulae (6, 7) are taken to calculate *RI*. When it needs to choose a video to be replaced, the peer will request system-level information of replicas and the number of simultaneous online viewers of its local stored files from its tracker and then *ATD* can be calculated. The videos with bigger *ATD* and bigger *number of simultaneous online viewers* will first be deleted. The target of Formula 7 is to make videos with less than 30 viewers have the similar replicas to videos with 30 viewers, and keep the *ATD* even for popular videos.

$$RI(i) = ATD_i * Fix\_C(number\ of\ simultaneous\ online\ viewers) \quad (6)$$

$$Fix\_C(x) = \begin{cases} x/30 & x < 30 \\ 1 & 30 \le x < 100 \\ 1.2 & x \ge 100 \end{cases} \quad (7)$$

Now, we consider the overhead of this method. For a peer, in order to calculate *RI* for all its local videos based on Formula (6), it must regularly collect the information of possessing number and the online viewer number of all its local videos. However, we find the overhead is low. Assuming the average size of *chunk bitmap* for each video is S bytes, N is the number of peers in the system, each peer stores M videos averagely and each peer sends and receives these information every T second. If all these messages are sent by tracker, the bandwidth cost for the server will be 2* S*M*N/T bps. In a case, we assume that S=500 bytes, M=3, N=100,000 and T=10 minutes. The desired bandwidth will be just 4Mbps. Moreover, if trackers just send video chunk map to several peers for every video and let them forward to other peers using gossip method, the tracker cost will be smaller, growing with the number of videos instead of peers in the system.

2) Performance

At steady state, LRU and other popular cache management algorithms can achieve near-proportional replication [12]. To know the *ATD* regulation effect under *PPR* I algorithm, we count the number of simultaneous online viewers and their *ATDs* of 300 videos at a snapshot on May 7th, 2008. Fig. 6 shows the statistical result of *ATD* versus the video popularity. If some videos have the same number of simulations online viewers, we use the average value of *ATD*. It is evident the *ATD* declines with the increases of video popularity for the videos with less than 30 simultaneous viewers. The *ATD* is similar (around 5) for videos with 100 simultaneous online viewers. This strategy guarantees that every peer can find enough neighbors to download data from even if it just has one simultaneous online viewer.

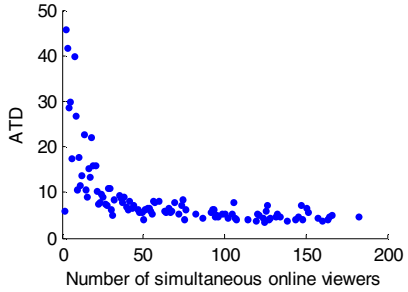


Figure 6. Video popularity vs. ATD under PPR I algorithm

Fig. 7 shows *BSR* evolutions in a two-day period under native LRU and *PPR* I algorithms. There is a significant improvement of *BSR* from 3:00 am to 9:00 am, but from 12 am to 9 pm the *BSR* is even smaller under new algorithm. It can be explained by the difference in the fraction of unpopular videos at different times on May 7th 2008, as is shown in Fig. 8. At 6:00am, more than 90% videos have less than 30 simultaneous online viewers. While at 9:00pm, 40% videos have less than 30 simultaneous online viewers. A high fraction of unpopular videos means a wide space to improve the system performance.

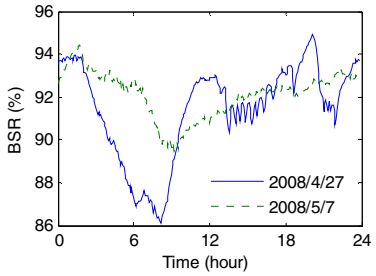


Figure 7. *BSR* comparison across two days

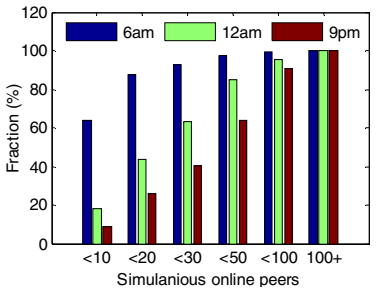


Figure 8. Popularity distribution at different times on May 7th

3) Insights

Whether it is worthy of improving *ATD* by collecting system-level information of replica requests and supplies depends on the fraction of unpopular videos. The original intention of *ATD* regulation is to improve the unpopular videos while maintaining the performance of popular videos. If the system consists of many unpopular videos, even at peak request time (around 9pm), the improvement will be significant. Otherwise, LRU is good enough.

E. PPR II

Finding the tail of videos can not be viewed smoothly and peer reselections occur frequently because of resource miss, on August 6th, we deployed *PPR* II algorithm to improve *Integrity* and *OR*.

1) How to Implement

PPR II works as follows. If there is no space available in local contributed disk, the local file with the biggest *RI* will be first chosen and then replace it one "cluster" after another sequentially, either starting from the first "cluster" or the last "cluster", with the probability of K. In our implementation, K is 1:1. *PPR* II takes Formula (8) as *RI*, which considers the local file *Integrity* in its storage. *Fix_C(x)* takes the same one taken in PPR I as shown in Formula (7).

$$RI(i) = \frac{ATD_i * Fix\_C(number\ of\ simultaneous\ online\ viewers)}{Integrality_i} \quad (8)$$

Besides function *RI*, another main change is that *PPR* II changes replacement granularity from file to "cluster". File and chunk are two common placement granularities. Granularity "chunk" provides maximum flexibility to regulate the distribution of both the files and chunks in certain file. However, it will result in discontinuous chunks (many holes in the file) and short files. Also, too small granularity means there are more things to keep track of and it cost much to select which chunk to be deleted. Granularity "file" can not regulate the distribution of different files to a certain pattern but avoids the cache being overly fragmented among many videos, providing good *Integrity*. Considering the problems of file and chunk replacement granularity, we introduce a middle-class granularity "cluster", a virtual data layer between file and chunk. We make 10 sequential chunks as a "cluster".

Which "cluster" should be first replaced after one video is selected to be replaced can impact the *OR* distribution. It is easy to regulate *OR* based system-level information. For example, a peer regularly requests replica number of each chunk, and then replaces first the most platitudinous chunks. However, the overhead of this method is big. Through development, we find the simple method that replacing the file one "cluster" after another sequentially can achieve good result.

2) Performance

We mainly explore the improvements of *OR*, *Integrity* and *BSR*. Fig. 9 compares the system-level chunk *OR* of a video under native LRU and *PPR* II. There are about five times of replicas for "early" chunks than "late" chunks under native LRU algorithm. This can be explained by special piece selection strategy in P2P-VoD systems. Rarest-first piece selection can regulate *OR*, but it can hardly be applied to P2P-VoD, because the peer downloads data sequentially to guaranty the viewing fluency. Then peers watching "early" sections of a video can not provide "late" chunks, while peers watching "late" sections of a video can provide both "late" and "early" chunks. This may lead to viewing problems for video tails, because it is hard to find neighbors with tail chunks. *PPR* II has solved this problem. We can see that the *OR* is better than that under *PPR* II algorithm. By regulating the value of K, we can regulate *OR*. If just replacing chunk sequentially from front to rear (K=1:0), the *OR* of each chunk is nearly even. If replacing chunk by K=1:1, the replicas of front chunk is about 25% more than the rear chunk. The rate under K=1:1

is better than the uniform distribution, because the requests for "early" chunks is usually a few more than "late" chunks.

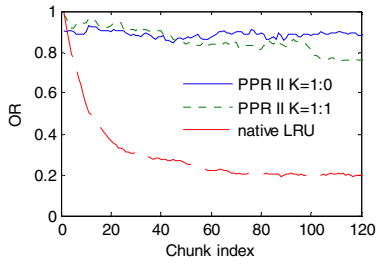


Figure 9. *OR* regulation

Fig. 10 shows the file *Integrity* distribution under native LRU and *PPR* II. Under *PPR* II algorithm, 50% files have 100% *Integrality*. While under native LRU algorithm, only 25% files have 100% *Integrality*. Based on statistical result, the average *Integrity* is increased by 25.5% (from 42.07% to 67.57%) after *PPR* II was adopted.

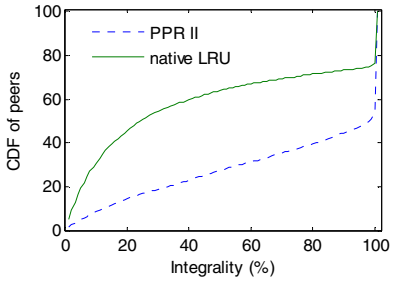


Figure 10. *Integrity* distribution

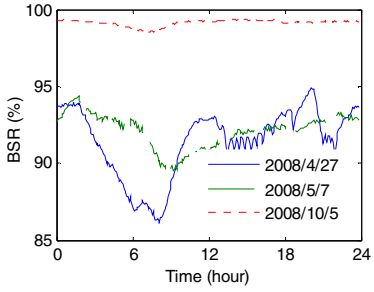


Figure 11. *BSR* comparison across three days

Fig. 11 shows the *BSR* evolutions under three different algorithms in three days. Compared native LRU with *PPR* II, it has a significant increase of *BSR* during all the hours in a day. The *BSR* is above 98% for all the time, and above 99% for most times on October 5th 2008 under *PPR* II. It means only 1% or 2% bandwidth will be supplied by servers.

3) Insights

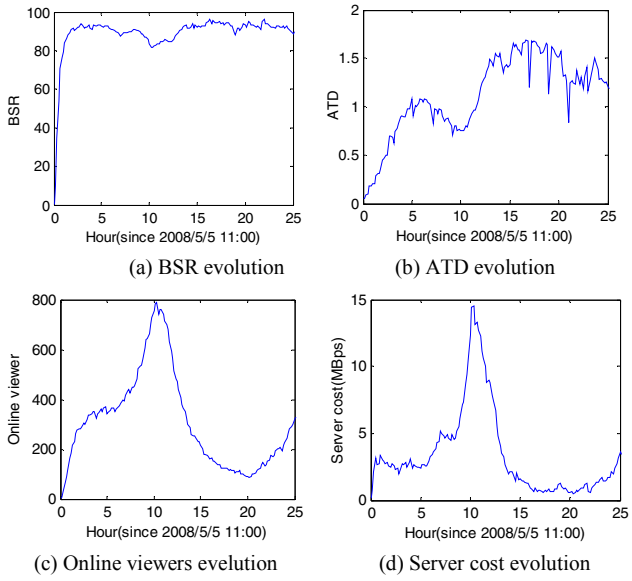
Completeness of video replicas plays important roles in content sharing among peers in P2P-VoD systems. If *Integrity* is small, the P2P overlay network will be very dynamic. And it costs much to rebuild the overly network. We have analyzed the top 50 videos which have smallest *Integrity*. Most of them are movies including erotic scenes. We also analyze the top 50 videos which have biggest *Integrity*. Most of them are classic movies. So if there are many videos with many interactions in a P2P-VoD system, the P2P overlay will be not stable and sharing effect among peers will be bad.

VI. TRANSIENT PERFORMANCE

In this section, we present the processes of video publishing and popularity declining. Pure replacement algorithm shows some limitations, and we give some alternative choices.

A. The Process of Video Publishing

Generally, when a video is just published in PPLive, the number of online viewers will increase quickly. And the *BSR* is relatively low during this time. We expect to learn whether it results from the lack of replicas. Fig. 12 shows the process of 25 hours since a popular video was publishing. We present the evolution of *BSR*, *ATD*, online viewers, and server cost of this video during this process. These values are collected every 10 minutes. We find: (1) the number of online viewers increased sharply since the video was just published, with a relatively small *ATD* and BSR; (2) after two and half hours, the *BSR* reached around 90%; (3) after 10 hours, around 22:00, there is a quick increase of online viewers, with a small valley of *BSR* and *ATD*. And both the peak value of server cost and online viewers appear at this time.



(a) BSR evolution (b) ATD evolution

(c) Online viewers evelution (d) Server cost evolution

Figure 12. The process of the publishing of a popular video

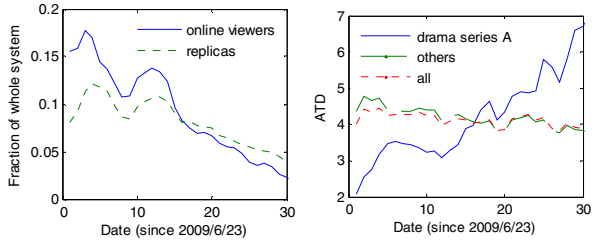
We can conclude that if the number of online viewers increases by a high speed, the *BSR* will be very low because of short-time lack of replicas. And the server might suffer heavy load. However, replacement algorithm seems helpless, during this time. Pre-fetching strategy might be a candidate to settle this problem. For pre-fetching strategy, it allows

some peers who have free bandwidth to download the videos which lack of replicas, regardless whether the video is watched by these peers now. Pre-fetching strategy may unnecessarily waste precious peer uplink resource. However, based on our measurement, pre-fetching is actually feasible. For one thing, we find there are nearly half of free peers, who do not download and upload anything. For another, the possibility of a peer is online is above 30%.

B. *The Process of Popularity Declining*

In June 2009, PPLive published a very popular drama series (*A* for short), which consists of 26 sets. The online viewers of these 26 sets account for nearly 30% of that of all videos (more than 10000 videos). However, after several weeks, their popularity declined sharply. During the process of popular declining, the server bandwidth cost increases sharply and even some videos can be not viewed frequently.

Fig. 13 shows the popularity declining process of *drama series A*, from June 23rd, 2009, to July 24th, 2009. Fig. 13(a) shows evolution of fraction of online viewers and replicas. The number of online viewers declined sharply since June 23rd, 2009, while the replicas didn't decline so sharply. The fraction of replicas is below the fraction of online viewers before July 9th, since which the fraction of replicas exceeded the fraction of viewers. Fig. 13(b) shows evolution of *ATD* of *drama series A*, all videos, and other videos except *drama series A*. The *ATD* of *drama series A* was increasing during these 30 days and exceeded the average line around July 9th.



(a) Fraction of online viewers and replicas (b) ATD evolution

Figure 13. Popularity decline of a very popular drama series

The *ATD* of *drama series A* is 70% higher than the average value and it will result in bandwidth waste. A peer generally watches the drama series one by one, and sharing disks of these peers only store *drama series A*. If these peers have redundant upload bandwidth resources except that for *drama series A*, they can not share them to peers who watch other videos. So for future improvement, PPLive will assign a smaller *Replacement Index* if a video's popularity is on increasing process and a bigger *Replacement Index* if the popularity is on declining.

VII. CONCLUSION

In a large-scale P2P-VoD system deployed by PPLive, we have made great efforts to improve replacement algorithms, so as to regulate replication distributions. Based on development experience, we find: Firstly, Native LRU replacement algorithm has limitations in regulating replica distribution among popular and unpopular videos. Secondly, whether it is worthy of improving *ATD* by collecting system-level information of replica requests and supplies depends on video popularity the fraction of unpopular videos. Thirdly, how complete of replicas and the balance of owing rate of each chunk in a video play important roles for content sharing in P2P-VoD system. Finally, the performance is bad when the video is just published, and when the popularity of videos declines sharply.

ACKNOWLEDGMENT

This research is supported by NSFC Project (60970104), NSFCR-GC Joint Research Project (20731160014), 863 Project of China(2008AA01A326), 973 Project of China (2009CB320501), and Program for New Century Excellent Talents in University.

REFERENCES

[1] "PPLive", http://www.pplive.com/.

[2] "GridCast", http://www.gridcast.cn/.

[3] "Joost", http://www.joost.com/.

[4] "PPStream", http://www.ppstream.com/.

[5] "UUSee", http://www.uusee.com/.

[6] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, "I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System", In Proc. of IMC, 2007.

[7] Y. Huang, T.Z.J. Fu, D.M Chiu, J. C. S. Lui and C. Huang, "Challenges, Design and Analysis of a Large-scale P2P-VoD System", In Proc. of SIGCOMM, 2008.

[8] B. Cheng, L. Stein, H. Jin, and Z. Zhang, "Towards Cinematic Internet Video-on-Demand", In Proc. of EuroSys, 2008.

[9] B. Cheng, L. Stein, H. Jin, and Z. Zhang, "A Framework for Lazy Replication in P2P VoD", In Proc. of NOSSDAV'08, 2008.

[10] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in Proc. of the 16th annual ACM International Conference on Supercomputing, New York, NY, June 22–26, 2002.

[11] E. Cohen and S. Shenker, "Replication strategies in unstructured peer to peer networks," In Proc. of SIGCOMM, 2002.

[12] S. Tewari and L. Kleinrock, "Proportional Replication in Peer-to-Peer Networks", In Proc. of INFOCOM, 2006.

[13] S. Tewari, and L. Kleinrock, "On Fairness, Optimal Download Performance and Proportional Replication in Peer-to-Peer Networks," In Proc. of IFIP Networking, 2005.

[14] J. Kangasharju, K. W. Ross, and D. A. Turner, "Optimizing File Availability in Peer-to-Peer Content Distribution", In Proc. of INFOCOM, 2007.

[15] M. Allen, B. Zhao, and R. Wolski, "Deploying Video-on-Demand Services on Cable Networks", In Proc. of ICDCS, 2007.

[16] C. Huang, J. Li, and K. W. Ross, "Can Internet Video-on-Demand be Profitable?", In Proc. of SIGCOMM, 2007.