PPVA: A Universal and Transparent Peer-to-Peer Accelerator for Interactive Online Video Sharing

Ke Xu, Haitao Li
Dept. of Computer Science&Technology
Tsinghua University
Beijing, China
Email: {xuke,lihaitao}@csnet1.cs.tsinghua.edu.cn

Jiangchuan Liu
School of Computing Science
Simon Fraser University
British Columbia, Canada
Email:jcliu@cs.sfu.ca

Wei Zhu, Wenyu Wang
PPLive R&D
Shanghai Synacast Media Tech
Shanghai, China
Email:{zhuwei,wangwenyu}@pplive.com

***Abstract*—Recent years have witnessed an explosion of online video sharing as a new killer Internet application. Yet, given limited network and server resources, user experience with existing video sharing sites are far from being satisfactory. To alleviate the bottleneck, peer-to-peer delivering has been suggested as an effective tool with success already seen in accelerating individual sites. The numerous video sharing sites existed however call for a universal solution that provides transparent peer-to-peer acceleration beyond ad hoc solutions. More importantly, only a universal platform can fully explore the aggregated video and client resources across sites, particular for identical videos replicated in diverse sites.**

To this end, we develop PPVA, a working platform for universal and transparent peer-to-peer accelerating. PPVA was first released in May 2008 and has since been constantly updated. As of January 2010, it has attracted over 50 million distinct clients, with 48 million daily transactions. In this paper, we highlight the unique challenges in implementing such a platform, and discuss the PPVA solutions. We have also constantly monitored the service of PPVA since its deployment. The mass amount of traces collected enables us to thoroughly investigate its effectiveness and potential drawbacks, and provide valuable guidelines to its future development.

***Index Terms*—Video sharing, peer-to-peer, acceleration, platform.**

I. INTRODUCTION

THE recent four years have witnessed an explosion of video sharing as a new killer Internet application. These new-generation user-generated content (UGC) sites, unlike traditional TV/movie servers, are greatly enriched by constantly updated contents from users worldwide. The most successful site, YouTube, now enjoys more than 100 million videos being watched every day. The success of similar sites like Yahoo Video (new version established in 2008) and Tudou (the most popular video sharing site in China), and the expensive acquisition of YouTube by Google, further confirm the mass market interest in video sharing.

Yet, given the limited network and server resources, and the best-effort nature of the IP network, the user experience with existing video sharing sites are far from being satisfactory, particularly during peak hours. Recent surveys revealed that the average service delay of YouTube is nearly 6.5 seconds, which is much longer than many other measured sites [5]. The situation would only become worse given the rapid content generation from users and the slow server and network infrastructure upgrade.

To alleviate the bottleneck, peer-to-peer delivering has been suggested as an effective tool. With each peer contributing its bandwidth to serve others, a peer-to-peer overlay scales extremely well with larger user bases. Besides file sharing, peer-to-peer has been quite successful in supporting large-scale live streaming (e.g., CoolStreaming [8] and PPLive [1]). Recently, it has also been applied to video sharing with on-demand interactions. A typical commercial example is the PPLive's Video-on-Demand (VoD) service [6]. Unfortunately, each such peer-to-peer accelerator is generally designed for a specific site; a user browsing different video sharing sites has to install customized accelerator for each site, or will suffer from poor performance of non-accelerated sites.

The existence of numerous video sharing sites clearly demonstrates the vigor of this new generation of networked service, but also simply implies that ad hoc accelerators for individual sites are not an ideal solution. For users, installing different accelerators will be time and resource-consuming; for service providers, developing customized accelerators will be costly with a lot of duplicated efforts. Instead, a universal platform that provides transparent peer-to-peer acceleration for different video sharing sites is expected. More importantly, only a universal platform can fully explore the aggregated video and client resources across sites, particular for identical videos replicated in diverse sites.

To this end, we develop PPVA (Peer-to-Peer Video Accelerator), a universal and transparent platform, through which a user can surf any video sharing site with accelerating conducted in background. PPVA extends the widely-used PPLive on-demand streaming engine, and was successfully launched in May 2008. As of January 2010, the PPVA client software (*http://ppva.pp.tv/*) has been installed by over 50 million distinct users, with 3.1 million of them being simultaneous online on average. It has transparently bridged these users and multi-thousand sites (over 48 million daily transactions),

---

This research is supported by NSFC Project (60970104), 863 Project of China(2008AA01A326), 973 Project of China (2009CB320501,2009CB320503), and Program for New Century Excellent Talents in University.

[1] http://www.pplive.com

enabling enhanced yet fully compatible viewing experience.

To the best of our knowledge, PPVA is the first such platform with large-scale deployment. In this paper, we highlight the unique challenges in implementing such a platform, and discuss the PPVA solutions. We have constantly monitored the service of PPVA since its deployment, and have collected data traces of multi-billion acceleration events. The mass amount of traces enable us to thoroughly investigate its effectiveness and potential loopholes, and provide valuable guidelines to its future development.

The rest of the paper is organized as follows: Section II describes the related work. In Section III, we highlight the challenges in developing PPVA and discuss its implementation. Section IV introduces data collection methodology. In Section V, the performance of PPVA is examined. We further identify interesting properties of today's online video sharing in Section VI. Finally, we conclude the paper in Section VII.

II. RELATED WORK

Peer-to-peer (P2P) delivering has been used for accelerating diverse content distribution systems, e.g., for file sharing [4], software updates [3], live streaming [8], and on-demand streaming (P2P-VoD) [1, 6, 7]. Our PPVA is closely related to P2P-VoD, which has attracted significant attention recently [1, 6, 7, 9]. Huang et al. [1] introduced two broad design approaches for P2P-VoD, namely, single video approach (SVA) and multiple-video approach (MVA). They also developed simple mathematical models for P2P-VoD and a prototype based on the MSN video. A large-scale P2P-VoD system was presented in [6], which extends PPLive, one of the most successful peer-to-peer live streaming system. Another working system is GridCast [7, 9], which was deployed on CERNET (China Education Network). These P2P-VoD systems are tailored to specific sites, and a user has to download differently customized client software so as to enjoy accelerated download. There is no coordination nor resource sharing across different video sites.

There have also been measurement studies on video services and user behaviors, in both P2P and client-server modes. Huang et al. [6] presented a number of results on user behaviors and various system performance metrics, including user satisfaction, replication health, server load and NAT related statistics. Cheng et al. [9] presented a comprehensive study on the effectiveness and user experience of P2P-VoD systems. Gill et al. [10] examined the video file, usage, and traffic patterns of YouTube, and compared them with traditional web and media streaming workloads. Cha et al. [11] conducted an in-depth study on YouTube and other similar UGC systems, examining the popularity cycle of videos, the statistical properties of requests and their relationship with video age, and the level of aliases in the system. These investigations are generally confined to a particular or a small collection of sites. Our PPVA platform however enables us to examine collective statistics of diverse video sharing sites and evaluate their impacts.

III. OVERVIEW OF PPVA: A UNIVERSAL AND TRANSPARENT ACCELERATOR

In this section, we overview the architecture and design of our PPVA platform. Besides the general issues that should be addressed in P2P accelerators for individual sites, there are many unique design challenges for a universal transparent platform, and we will elaborate our solutions in the PPVA implementation.

A. Design Challenges And Objectives

Universality. As a universal platform, PPVA is not tailed to any specific video sharing site. Instead, it will provide universal peer-to-peer accelerating services and mask the heterogeneity of the sites, e.g., site architectures and video formats. More importantly, it will make effective use of the aggregated user and video resources across different sites to achieve better performance than with standalone sites.

Transparency. As a universal platform, PPVA is not tailed to a specific site, and does not call for any change in existing video sharing sites. Instead, it has to provide transparent services that do not make any change to existing client and server operations (except for accelerated streaming experience) and mask the the heterogeneity among them, e.g., transmission protocols and video formats. However, lack of information video sites, PPVA can hardly select download strategy. For example, PPVA can not directly get seeking interaction information, including which video is sought and the seeking position. This makes PPVA can hardly find correct neighbors to request data. In Section VII, we will introduce the problem, our distributed solution, and analyze its performance.

Scalability. Popular video sharing sites all have huge user bases. Given that PPVA will serve them universally, the scalability challenge will be enormous, spanning over users (millions of online peers), videos (millions of user-generated videos), and sites (multiple thousand to date). The platform should accommodate them well, and should provide efficient indexing service for peers and videos across the sites of interest.

Server Bandwidth Cost Alleviation. Although video sharing in the Internet has become an immensely popular service in recent years, given the enormous costs associated with client-server distribution, the revenues very possibly will not cover the cost of providing the service. How to reduce bandwidth cost might be the most urgent concern for content providers.

Acceleration Effectiveness. Given limited network and server resources, user experience with existing video sharing sites are far from being satisfactory. The most purpose of this platform is to alleviate bottleneck of server bandwidth and thus improve user viewing experience.

B. PPVA Architecture

Figure 1 depicts the the PPVA architecture, which consists of the following key modules:

Video application: This includes the video repository servers and their web site portals. We do not impose any specific design guidelines on these servers and portals, nor limit their video file format, bitrate, and size. The video sharing sites do not even need to be aware of the existence of PPVA.

Peers: These are the clients running our PPVA client software to fetch video data. These clients can access video sharing sites with conventional operations, and PPVA will intercept the requests and transparently provide accelerated streaming services through peer-to-peer or a combination with server downloading.

Trackers: They are used to manage peers, providing such information as which peers have the desired videos replicas in their local storages. When a peer joins the system, it will register to a Tracker and keep updating about its video of interested and resources to be shared.

Index server: It is used to perceive peers' behaviors, such as which video a peer starts to watch, whether a peer has made a seeking interaction and what position it seeks to. It also provides indexing information about the videos in the repository servers.

P2P cache: The performance of our PPVA can also be enhanced by optional P2P caches, which, as dedicated online nodes, store replicas of the video content and share upon requests.

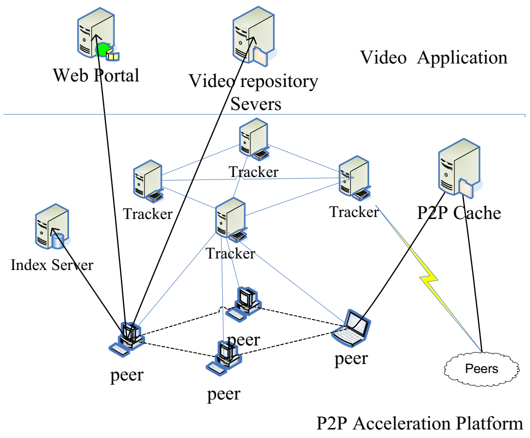

Fig. 1. The PPVA Architecture

Once a client plans to watch a video from a sharing site and the accelerator is invoked, three downloading options are available: server only, peer-to-peer only, and a hybrid of them. By default, the hybrid download is used by PPVA, which achieves the maximum downloading performance and accommodates accesses to both popular and non-popular videos. For the operations of the peer-to-peer part, PPVA adopts the PPLive-VoD engine, but incorporates necessary extensions to achieve universality and transparency, as explained below.

C. System Operations

We now describe the basic operations of PPVA.

Request Interception: Our PPVA serves as a proxy between a client and its web browser, capturing the clients' requests and potentially speeding up the download for video contents. Specifically, it will bypass all requests except for downloading files of video formats (e.g., FLV and MP4). For the latter, peer-to-peer accelerating engine will be invoked.

Join the system: Once the peer-to-peer engine is invoked, the new client will first register its ID, IP address, shared resources list to Trackers and update the resources list from time to time. It will also obtain a list of potential neighbors to fetch video data.

Play: Once the accelerator is invoked, three downloading options are available: from the server only, peer-to-peer only, and a hybrid of them. By default, the hybrid download is used, which achieves the maximum downloading performance and accommodates accesses to both popular and non-popular videos.

For the operations of the peer-to-peer part, PPVA adopts the PPLive-VoD engine, but incorporates necessary extensions to achieve universality and transparency, as described next.

D. Global Video Identification

Each video sharing site has its own local videos identification rules. To share totally identical videos in different sites, the universal platform however should assign a global video identifier(GVID) for each accelerated video. To address this problem, PPVA adopts the hash value of the video content, which is largely unique for each individual video. Based on this GVID, the identical videos in different servers, and even different video sharing sits can shared with each other.

Since a PPVA client cannot directly get such a video identifier from sites, all GVIDs are calculated by peers themselves. To prevent data pollution [12], a peer will calculate GVID again using downloaded data to verify its validity.

E. Caching and Replication

Unlike streaming live content, PPVA peers are not synchronized in watching a video. Hence, if the peers just cache temporarily in their memories what they are watching, the efficiency of PPVA can be quite low. For example, in YouTube, even when peers share videos for a longer period of time (e.g., 1 day), P2P just assists 60% of videos with at least 10 current peers all the time [11]. To compensate, each peer is required to contribute a fixed amount of hard disk 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 then shares all the videos stored in local space. As a result, for a client interested in a particular video, all the peers that have previously downloaded this video serve as potential suppliers, forming an overlay for this video, together with the peers that are downloading this video. We refer to every such overlay as a channel. Obviously, a PPVA peer may appear in multiple channel, and the server is by default in every channel, ensuring that there is always at least one supplier.

The entire viewer population thus forms a larger P2P sharing system with much higher efficiency. 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. The replication strategy mainly includes shared disk size, replication distribution, and replication replacement algorithm. PPVA takes a very similar replication strategy in the PPLive-VoD system [15].

F. Content Discovery

It is not enough to just have good replications in the system. A peer has to find them before downloading from them. So content discovery is an critical function in PPVA.

The process of content discovery is as follows: First, since each video sharing site has its own local video identification rules, a peer will request Index servers to get GVID and Index servers will give back GVID and other security related information. Second, using GVID, the peer will request peer list from trackers and the Tracker servers will assign it a number of existing peers having contents around its starting playback point cached. A peer will report its local replicas to the Tracker system once it joins the system, and update them when some are added or deleted.

To be scalable, it is advisable to provide a decentralized distributed lookup service to find desired peers. However, exit distributed method, Distributed Hash Table (DHT) [14], is designed for file sharing and downloading, and is not adapt to VoD, because it costs too much time to find desired peers. The consequence is the big startup and seeking latency, which are two most important indexes of user experience. PPVA takes distributed Trackers to increase scalability and reduce lookup latency. It is very easy to add a new Tracker to system. And the crash of some Trackers in a group will not lead to the system's crash or even a video channel's crash.

IV. MEASUREMENT METHODOLOGY

The large-scale deployment of PPVA provides us an opportunity to examine the performance of such a universal and transparent peer-to-peer accelerating service in real-world. It also enables us to systematically examine the similarity and differences of diverse video sharing sites. In this section, we discuss the characteristics of interest in our study and the measurement methodology.

A. Characteristics of Interest

Given that server bandwidth and access delays are critical concerns in existing video sharing sites, we mainly focus on the PPVA's effectiveness in minimizing them, as well as the overhead of deploying PPVA. We also investigate the characteristics of online video services, such as video population and site popularity distribution, simultaneous online peer evolution, video file properties, user interaction and sojourn time. Unlike previous works [1, 10, 11], which focused on just individual video sites such as MSN video or YouTube, our measurement has a broad scope, including multi-thousand UGC and traditional video sites. Table I shows the top ranked video sharing sites served by our PPVA. Except for Sina (ranked 4th) - one of the most popular web portals in China, all the others shown here are YouTube-like UGC sites. The top three UGC sites account for over 50% requests. Since PPVA is mainly used in China (account for over 97% downloads), most of these top-ranked sites are in China, and YouTube indeed has a relatively low rank (91st).

Rank	Site	Percentage	Count in one day
1	Youku[2]	30.59%	201,945
2	Ku6[3]	14.07%	92,909
3	Tudou[4]	6.21%	40,985
4	Sina[5]	2.45%	16,201
91	YouTube[6]	0.048%	315

TABLE I
POPULAR VIDEO SITES STATISTICS

B. Data Collection Methodology

To monitor the system operations and analyze its performance, PPVA has deployed a log server to collect reports from peers since November 4th, 2008. Around 3% peers report the information and 3.5 million reports are collected everyday. Other than using a crawler, we directly get data from PPVA's log servers, to which peers report their local information every five minutes.

A peer reports to log server at two time points: First, when a peer closes its PPVA client software, it reports information including the used disk space, upload cache hit ratio, upload data bytes, download data bytes, downloading duration, running duration, CPU peak value, and memory peak value. Second, when the peer finishes a download job, it reports the information including report time, peer ID, VID, request URL, request reference, whether download is finished, average download speed, maximal http speed, maximal P2P speed, average http speed, average P2P speed, bytes downloaded through http or P2P, file size, file bitrate, and file length. To protect users' privacy, we do not collect the their IP addresses.

Besides log servers, we also dump data from the Trackers and Index servers. The data from the Trackers mainly include VID, video owners and viewers. The data from the Index server mainly include video length, video bitrate and so on.

V. PPVA PERFORMANCE

In this section, we show the system perform improvement from three aspects: server bandwidth cost, acceleration effectiveness and client overhead.

A. Server Bandwidth Cost

For PPVA, once the accelerator is invoked, three downloading options are available: from the server only, peer-to-peer only, and a hybrid of them. By default, the hybrid download is used. Thus much bandwidth will be saved for video sharing sites. We first explore how many percentages of downloading chooses peer-to-peer only or hybrid options. We further explore how much it has reduced servers' bandwidth cost.

---

[2] http://www.youku.com

[3] http://www.ku6.com

[4] http://www.tudou.com

[5] http://www.sina.com.cn

[6] http://www.youtube.com

	Exclusive Seeking	All
Youku	93.3%	76.6%
Ku6	40.9%	40.9%
Tudou	76.5%	59.1%
Sina	85.1%	78.1%

TABLE II
PERCENTAGE OF PEER-TO-PEER DELIVERY BEING INVOKED FOR TWO-TYPES REQUESTS

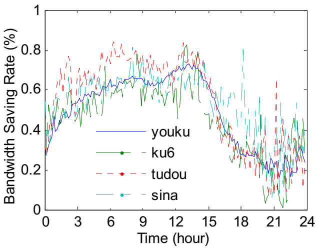

Fig. 2. BSR evolution

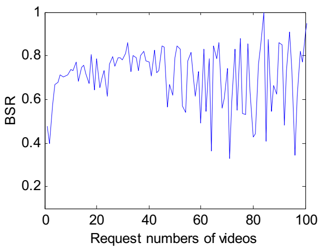

Fig. 3. BSR against video popularity

Table II summarizes the percentage of peer-to-peer delivery being invoked for two-types requests. *Exclusive seeking* means those watchings without seeking interactions. We can see that the most popular site, Youku, has the highest value, because it has a larger user base for peer-to-peer acceleration. Ku6's service however is limited, which does not even enable random seek. The acceleration by PPVA is therefore limited too, but still over 40%. As described in Section III.E, PPVA takes the seeking interaction as a new file request and thus most take server only downloading option. This Table shows that seeking interactions can significantly reduce the effect of peer-to-peer delivery.

To accurately estimate the bandwidth savings, we introduce a metric Bandwidth Saving Ratio (BSR). The higher the BSR is, the more bandwidth is saved for servers. For a particular video i, its BSR is given by

$$BSR_m = \frac{Upload_m}{Download_m} \tag{1}$$

where $Download_m$ is the byte downloaded by peers who watch video m, $Upload_m$ is the the byte uploaded by peers who video m. Thus the difference of $Download_m$ and $Upload_m$ is come from servers. For example, if a peer watches a video of 100 MB length, with 20 MB from resource servers and 80 MB from other peers, then the BSR for this download is 80%. Let M and N be the number of videos and peers in the system respectively, N_m be the number of peers who watch video m. Then the BSR of the system becomes

$$BSR = \frac{1}{N}\sum\nolimits_{m=1}^{M}(N_m * BSR_m) \tag{2}$$

We calculate the average BSR every 10 minutes, and plot the whole-day evolution in Figure 2 for all the sites. The lowest value saving ratio appears around 22:00, which is about 20%, and the highest value appears at 12:00, which is about 70%. Again, Youku has a little higher bandwidth saving ratio. The lower BSR during night is mainly because the disposal ability of Index server is limited. During the request peak time (around 21:00), since it cannot dispose so many request, some are abandoned. So the abandon ratio is relatively high during peak time.

Figure 3 shows BSR of videos with different popularity, where X-axis is the request number of the videos in a one-day period. We can see that BSR increases with video's popularity when the request number of videos in one day is less than 10. This is because the peers watching unpopular videos can hardly find enough replicas to accelerate download. However, when the video popularity reaches to some value, the performance presents no evident improvement. So, the video popularity is not always a crucial factor affecting system performance, especially when the most videos are popular.

B. Acceleration Effectiveness

We focus on two acceleration effectiveness. First, the percentage that PPVA has reduced the unfrequent viewing, which is approximately defined as that average download speed is less than video bitrate. Second, the download speed PPVA has accelerated. Although higher average download speed does not necessarily mean better user experience, it can reflect viewing experience to some extent.

It is easy to measure average download speed after using PPVA, since PPVA client software records and reports this data. However, it is impractical to measure average download speed without using PPVA. We make a approximate measurement. Many viewers can not find peer resources, possibly because they are watching unpopular videos or they make seeking interactions. The download are totally from servers and we define this speed as *speed without PPVA*. The peer downloads data both from peers and servers, and we define this speed as *speed with PPVA*. Actually the real effect might be better. Because PPVA has saved a lot of bandwidth for servers, the peers who directly request from servers can enjoy better bandwidth services.

	Without PPVA	With PPVA
Youku	0.57%	0.57%
Ku6	28.5%	15.1%
Tudou	52.1%	27.7%
Sina	59.5%	9.53%

TABLE III
UNFREQUENT VIEWING RATIO WITH AND WITHOUT USING PPVA

First, Table III shows the unfrequent viewing ratio with and without using PPVA. It shows Youku has the best viewing experience. However, other three video sites have bad viewing experience. Particularly, Nearly 60% Sina videos can not be viewed smoothly. The acceleration effectiveness is obvious for Ku6, Tudou, and Sina.

Second, we measure the download speed PPVA has accelerated. Table IV shows the average download speed of a peer with and without PPVA. We find the download speed increases after using PPVA for Ku6, Tudou and Sina. The same is also with Youku, though this site already has higher speed without PPVA.

	Without PPVA	With PPVA
Youku	1005Kbps	1472Kbps
Ku6	378Kbps	580Kbps
Tudou	442Kbps	1073Kbps
Sina	496Kbps	1199Kbps

TABLE IV
DOWNLOAD SPEED WITH AND WITHOUT USING PPVA

C. Client Overhead

While PPVA can improve user experience by accelerating download speed and reduce server bandwidth cost, it introduces additional cost for peers to participate a peer-to-peer overlay. We now examine the important client costs, including disk, memory, and CPU.

Figure 4 shows the disk cost distribution. We can see that about 29% peers contribute zero spaces, about 80% peers contribute less than 500 MB, and all peers contribute less than 2000 MB, which is reasonable for current personal computers.

Figure 5 shows the memory cost distribution. We can see that all peers use less than 100 MB, and nearly 80% peers use less than 20 MB memories.

Because the peer stores much more than that memory can store, if the request data can not be found in upload memory, it will get it from the disk, which involve a IO operation. Frequent IO operations will result in bad user experience. Figure 6 shows memory hit ratio. Although a small memory is used, the memory hit ratio is still very high, which is also because PPVA employs an improved replacement algorithm rather than a naive FIFO. For active peers, the hit rates above 90% account for 44.54%.

CPU utility (%)	0-5	5-10	10-20	20-100
Percentage (%)	97.38	0.84	0.96	0.62

TABLE V
PEAK CPU COST DISTRIBUTION

Table V shows the peak CPU cost distribution. Again, more than 97.38% peers only use less than 5% CPU time; there are only 0.62% peers use more than 20% CPU time.

VI. CHARACTERISTICS OF VIDEO SERVICES

In this section, we present characteristics of the video services accelerated by PPVA. Unlike previous works [1, 10, 11], which focused on just one video site such as YouTube or MSN video, our measurement has a broad scope, including multi-thousand UGC and traditional video sites.

A. Video Population and Site Popularity Distribution

In this subsection, we explore the video site popularity distribution, which will help to design P2P caching and ISP caching strategies.

Figure 7 shows the skewness of user interests across video. The lower the rank is, the more popular the video is. We can see that the top 10% popular videos account for 82% views, and the top 20% popular videos account for 94% views. This is similar with YouTube [11]. On the other hand, nearly 74% videos are not viewed at all during the one-day period. An immediate implication of this skewed distribution is that caching can be very efficient since storing only a small set of objects can enable high hit ratios. For example, by storing only 10% long-term popular videos, a cache can serve 80% requests.

We use one-day-period reports on November 15th, 2008 to investigate the popularity of video sites. Figure 8 shows the site popularity distribution of over 3000 video sharing sites, which clearly follows the power law distribution. These sites are at least required once on November 15th by PPVA client software.

B. Simultaneous Peers Evolution

In Figure 9, we show the evolution of simultaneous online peers in a typical day, November 15th, 2008. We can see that the number of online peers peeks at 21:00, after that it declines and reaches its lowest at 6:00 in the morning. The highest value is about 3 times of the lowest, and the the number is relatively steady from 12:00 to 18:00. We have found that this daily pattern generally exists since the deployment.

C. Video File Properties

We next explore the important video characteristics classified by sites, namely, video sizes, video durations, and video bitrate.

Figure 10 plots the size distribution by sites. Youku and Ku6 have constraint on file size, and most of them are less than 15 MB. The videos in Tudou and Sina have a long tail, distributing from 0 to 100 MB, though most are still less than 50 MB.

Figure 11 shows the video bitrate distribution by sites. For the top four sites, the bitrates of videos in Tudou and Ku6 are basically around 250 Kbps. Youku however has about 20% videos with rates lower than 170 Kbps, and Sina has a much wider range, likely because Sina's contents are not all generated by users. Overall, the low-bitrate videos are popular in these UGC sites.

We obtain the file duration through dividing file size by file bitrate. Figure 12 shows video length duration distribution by sites. The distributions of different sites are quite different. The videos of Youku and Ku6 are much shorter than that in Tudou and Sina. We find more than 99.8% request videos are less than 8 minutes and 7 to 8 minutes account for 68.2% in Youku. In Ku6, more than 90.5% requested videos are less than 8 minutes. This is mainly because Youku and Ku6 have constraints for the video length. Although they also provide 2-hour-length movies, the videos are divided into multiple shorter files.

Table VI summarizes the average values of important file characteristics, including bitrate, size and length. Compared with Table III, video bitrate is highly related to the unfrequent viewing ratio. It reflects that high bitrate videos can not be easily supported currently because of bandwidth restriction.

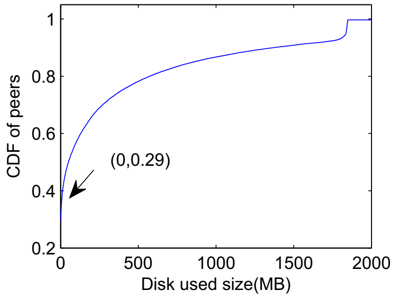

Fig. 4. Disk cost distribution

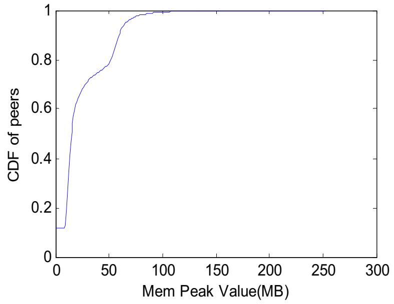

Fig. 5. Memory cost distribution

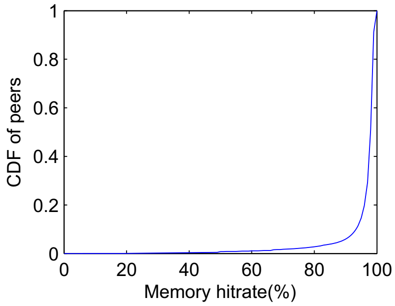

Fig. 6. Memory hit ratio

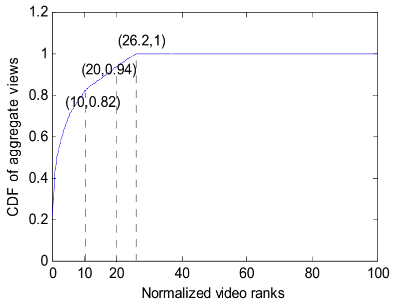

Fig. 7. Skewness of user interests across video

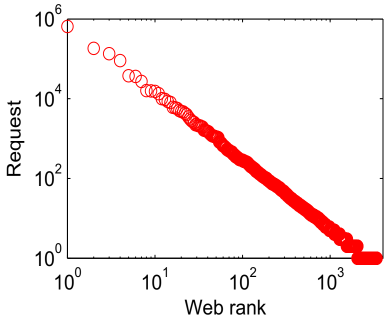

Fig. 8. Site popularity distribution

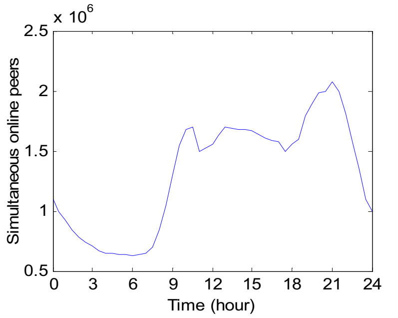

Fig. 9. Simultaneous peers evolution in one day

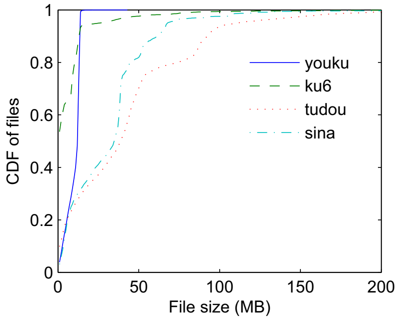

Fig. 10. File size distribution

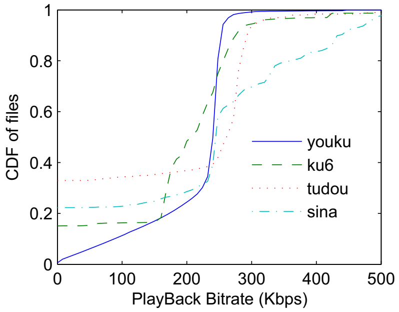

Fig. 11. File bitrate distribution

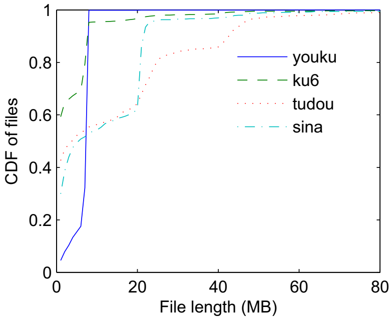

Fig. 12. File length distribution

	Bitrate(Kbps)	Size(MB)	Length(Minute)
Youku	214	9	6.3
Ku6	231	11	7.1
Tudou	274	45	22.5
Sina	319	28	13.5

TABLE VI
AVERAGE FILE CHARACTERISTICS

D. *User Interactions*

When a peer selects a video for streaming, the peer does not need to continuously watch the video from the beginning to the end. The peer may terminate earlier, or perform VCR-like operations. It is important to understand this interactivity while designing peer-assisted solutions for popular video delivery.

In fact, PPVA client software can not capture the precise information of VCR-like operations. When a peer makes a seeking interaction, PPVA client software simply regards it as a general new file request, without knowing whether it seeks, let only what position it seeks to. However, since the PPVA client software can get Content-Length from browsers, it can judge whether the download is complete by comparing real downloading bytes and Content-Length. Figure 13 shows that the percentage of sessions fully completed as a function of video size. The complete ratio generally declines with the file size increases, though fluctuation exists beyond 200 MB.

E. *Sojourn Time*

In this subsection, we explore running, downloading durations and downloading bytes before a peer shuts down its PPVA client software.

The PPVA client software starts up when a peer starts to

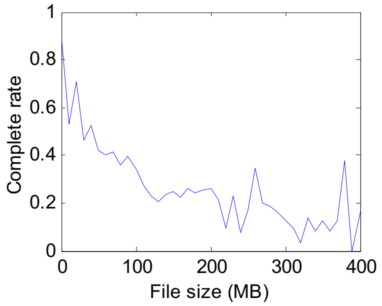

Fig. 13. Complete ratio vs. File size

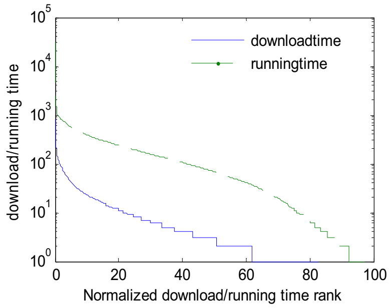

Fig. 14. Downloading and running duration distributions

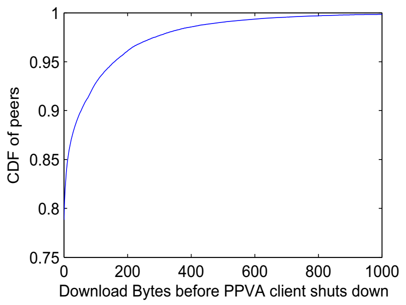

Fig. 15. Download Bytes before PPVA client shuts down

watch a video. Yet, the client software does not shut down automatically if a video viewing is finished. We define the running duration as the time elapsed before the client software is shutdown. We define the downloading duration as the time during which the peer really downloads data before the client software is shutdown. Figure 14 shows the downloading and running duration distributions. First, we can see that the running duration is much longer than downloading duration. Second, about 40% peers download time is 0. The highest running time is up to 10000 minutes and the highest download time is up to 1000 minutes. Rank 50 download duration is about 5 minutes and rank 50 running duration is about 100 minutes.

Figure 15 shows the download bytes before a peer shuts down PPVA client. We can see that nearly 78% peers do not download any data at all. We think it is mainly because that a lot of users make a setting that PPVA will start automatically when they switch on the computer. Usually they do not watch any videos at all before they shut down PPVA. If more users make a setting that PPVA does not start until they watch a video, there will not be so many zero values.

VII. RANDOM SEEKING SUPPORT

Random seeking interactions account for about 18% requests. We find transparent solution poses difficulty on supporting random seeking effectively. In this section, we introduce the problem, our distributed solution, and analyze its performance.

A. *Problem Description*

Our measurement shows that the random-seeking interactions account for nearly 18% requests. Unfortunately, existing video sharing sites do not have a uniform interface for random seeking. As such, PPVA implements random seek by treating it as new request to the video file with the specific playback position. In certain situations, its replication efficiency will be low. For example, a peer skips half of file A and then downloads the other half data; this download will be treated as a new file B rather than the half of file A, and it can neither download data from peers that already have file A, nor upload data to peers that are watching file A. This is could be best addressed by unifying the random seek interfaces of diverse video sharing sites, or at least, making them disclose user behavior information. However, this method should put some demand on eating video sites, and there is not such deals between PPVA and these videos sites now.

VIII. RANDOM SEEKING SUPPORT

Random seeking interactions account for at least 18% requests. We find transparent solution poses difficulty on supporting random seeking effectively. In this section, we introduce the problem, our distributed solution, and analyze its performance.

A. *Problem Description*

Our measurement shows that the random-seeking interactions account for at least 18% requests. Unfortunately, existing video sharing sites do not provide public interface which tells random seeking information such as the seeking position. So, PPVA implements random seeking by treating them as new video requests. Its replication efficiency will be low. For example, a peer skips half of file A and then downloads the other half data; this download request will be treated as for a new file B rather than the half of file A, and it can neither download data from peers that already have file A, nor upload data to peers that are watching file A. This problem could be best addressed by giving a uniform and public interface which discloses the information of user behaviors. However, this method would put some demand on existent video sites. And unfortunately there is not such interfaces between PPVA and these sites now.

B. *Distributed Seeking Identification*

To solve this problem, we propose a distributed solution. First, PPVA client parses whether a request is a seeking interaction. For example, it captures and parses the resource's URL to check whether it contains string '?start='. If so, this is a seeking request of the current watching videos. Second, PPVA client will download a small portion of data(e.g., 2KB) directly from servers. Third, it sends current GVID and the 2KB downloaded data to the peers (which are called neighbors) it is downloading from. Fourth, neighbors will match the 2KB data with its local file which has same GVID. Fifth, the neighbors will then send the offset back or send 'NULL' if it cannot match or matches more than one position. If a peer receives

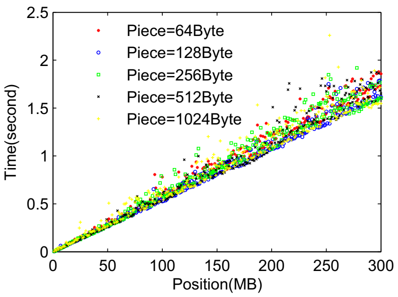

Fig. 16. Matching cost

'NULL' from its neighbors, it will download directly from servers. If a peer receives a offset feedback, it will use this offset and GVID to request more neighbors. Note that there could be multiple matches if the data size for the search is too small. Our experiments will show that 2KB size is good enough to guarantee a unique match in most of the time.

In case of wrong matching results, PPVA provides a mechanism to prevent polluted data diffusing.

C. Performance

First, we explore the overhead of the distributed seeking identification method. The main overhead is matching cost. Figure 16 gives matching cost with different matching data length and matching position. In our experiment, we use KMP[16]fast pattern matching in strings algorithm to match data. The complexity of this algorithm is O(M+K). M is the size of file and K is the size of matching data. Figure 16 shows matching cost is nearly linear to the file size. The cost is less than 2 second when the file is 300 MB. To note that average file length is 10MB, so the average cost is less than 0.1 second.

One important parameter of this method is the match data size. Table VII shows percentage of more than one match with different file size and match data size. It shows that 2K Byte is big enough to uniquely identify the seeking position.

piece=	64B	128B	256B	512B	1KB	2KB
file=5MB	0.17%	0.11%	0.08%	0.06%	0	0
file=10MB	0.23%	0.21%	0.16%	0.09%	0.01%	0
file=20MB	0.25%	0.22%	0.21%	0.11%	0%	0
file=300MB	0.11%	0.08%	0.04%	0.03%	0%	0

TABLE VII
PERCENTAGE OF MORE THAN ONE MATCH

Then, we explore the benefits of this distributed seeking identification method. The main shortage of regarding seeking requests as new files is peers can not utilize actually existed replicas in other peers. And the main advantage of distributed seeking identification method is replicas in peers can be fully utilized. Actually, we can gain the result from Section V. In Table II, the percentage of peer-to-peer delivery being invoked is much less if seeking requests are included. In Table III and IV, the downloading speed is much slower and smooth viewing ratio is much smaller if a viewer can not find other peers to download from and downloads data totally from servers.

IX. CONCLUSIONS

This paper presented PPVA (Peer-to-Peer Video Accelerator), a working platform for universal and transparent peer-to-peer accelerating. PPVA was deployed in May 2008, and, as of January 2010, it has attracted over 50 million unique peers, with over 3.1 million simultaneous online peers and 48 million daily accesses. In this paper, we detailed the design and implementation of PPVA, including the system architecture, the request interception mechanism, the pollution prevention, and the support to user interactions. We constantly monitored the service of PPVA since its deployment. The mass amount of traces collected enabled us to thoroughly investigate its effectiveness and potential drawbacks, and provide valuable guidelines to its future development.

There are many avenues for further optimizing this platform. We are particularly interested in quantitatively optimizing the download share between server and peers. Reducing the inter-ISP traffic to improve user experience and mitigate the impact from peer-to-peer sharing is another direction we are working on.

REFERENCES

[1] C. Huang, J. Li, and K. W. Ross. Can Internet Video-on-Demand be Profitable? In Proc. of SIGCOMM, 2007.
[2] B. Cheng, X. Z. Liu, Z. Zhang, and H. Jin. A Measurement Study of a Peer-to-Peer Video-on-Demand System. In Proc. of IPTPS, 2007.
[3] C. Gkantsidis, T. Karagiannis, P. Rodriguez, and M. Vojnovic. Planet Scale Software Updates. In Proc. of SIGCOMM, 2006.
[4] D. Qiu and R. Srikant. Modeling and performance analysis of bittorrent-like peer-to-peer networks. In Proc. SIGCOMM, 2004.
[5] M. Saxena, U. Sharan, and S. Fahmy. Analyzing Video Services in Web 2.0: A Global Perspective. In Proc. of NOSSDAV, 2008.
[6] 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.
[7] B. Chen, L. Stein, H. Jin, and Z. Zhang. Towards Cinematic Internet Video-on-Demand. In Proc. of EuroSys, 2008.
[8] X. Zhang, J. Liu, B. Li, TSP. Yum. CoolStreaming /DONet: A data-driven overlay network for efficient live media streaming. In Proc. of INFOCOM 2005.
[9] B. Cheng, X. Liu, Z. Zhang, and H. Jin. A Measurement Study of a Peer-to-Peer Video-on-Demand System. In Proc. of IPTSP, 2007.
[10] P. Gill, M. Arlittz, Z. Li and A. Mahanti. YouTube traffic characterization: a view from the edge. In Proc. of IMC, 2007.
[11] M. Cha, H. Kwak, P. Rodriguez, Y. 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.
[12] P. Dhungel, X. Hei, K. W. Ross, and N. Saxena. The pollution attack in p2p live video streaming: measurement results and defenses. In Proc. of P2P-TV, 2007.
[13] J. Liang, R. Kumar, Y. Xi, and K. W. Ross. Pollution in P2P file sharing systems. In Proc. of INFOCOM, 2005.
[14] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In Proc. ACM SIGCOMM, 2001.
[15] H. Li, K. Xu, J. Seng, P. H. Towards Health of Replication in Large-Scale P2P-VoD Systems. In Proc. IPCCC, 2009.
[16] Donald E. Knuth, James H.Morris, Jr., and Vaughan R. Pratt. Fast pattern matching in strings. SIAM Journal on Computing, 6(2):323-350, 1977.