SCIENCE CHINA
Information Sciences

• **RESEARCH PAPERS** •

August 2011 Vol. 54 No. 8: 1677–1690
doi: 10.1007/s11432-011-4288-6

Large-scale P2PVOD system: Focusing on clients

HUANG Yan[1], XU Ke[2*], LI HaiTao[2], CAO Yang[1] & YAO Xin[1]

[1]*Shanghai SynaCast Media Tech Co., Ltd, Shanghai* 201204*, China;*
[2]*Department of Computer Science & Technology, Tsinghua University, Beijing* 100084*, China*

Received April 1, 2010; accepted December 20, 2010; published online June 2, 2011

Abstract Using P2P network to provide video on demand (VOD) service can support large-scale users so as to save the bandwidth cost of servers. As VOD must support VCR characters (drag, pause, randomly choosing watching position, etc.), building a large-scale expansion of high-performance VOD system still faces huge challenges. The paper first presents a P2P based reference model, defines the design object, and analyzes the difficulties. And then taking PPLive demand system as an example, it gives client-based solutions, including disk cache-out strategy, memory cache prefetch strategy, maintaining relations between neighbors, task scheduling, data upload strategies and so on. PPLive-demand systems are putting into practice at present with satisfactory fluency and drag delay. The largest number of simultaneous online users is over 210000, and online video data is over 700. Compared with traditional client-server model, the average bandwidth saving rate of servers is 88% or more, and the bandwidth saving rate of user-number's the peak value can be up to 93%.

Keywords P2P, VOD, server bandwidth saving rate, fluency, drag delay

Citation Huang Y, Xu K, Li H T, et al. Large-scale P2PVOD system: Focusing on clients. Sci China Inf Sci, 2011, 54: 1677–1690, doi: 10.1007/s11432-011-4288-6

1 Introduction

VOD is short for video on demand technology, also known as interactive video on demand system. VOD is the product of computer technology, network technology and multi-media development. Realizing large-scale user base VOD system based on the Internet has long been the target of new Internet business. Currently, there are many living streaming systems, such as Coolstreaming [1], PPLIVE [2], PPStream [3], and GridCast [4]. There are also YouTube [5], YouKu [6] and other websites which provide short videos with relatively low definition. Traditional VOD services are mainly operated based on the servers/clients mode. Media data such as videos mainly assemble on servers. Users set up connection with corresponding servers via a webpage video index, and download data to achieve online viewing. This mode to some extent realizes VOD service. However, under this mode user number that servers can afford is limited. Researches have shown that P2P technology can apply to VOD system to save server bandwidth so as to increase system scalability [7–9]. But it is still difficult to achieve relatively high-definition, high fluency and low drag-delay VOD systems. First, VOD systems are highly dynamic. In addition to random access, it can also be randomly dragged, paused, making P2P topology changed more drastically. Besides, popular and unpopular channels/films coexist. With the increase of channels, the portion of unpopular

*Corresponding author (email: xuke@tsinghua.edu.cn)

channels will be larger. And users scattered in various locations of channels make data sharing even harder. Finally, the upload/download bandwidth of every user differs greatly. And due to network dynamic feature, actual bandwidth capacity changes all the time. So neighbor selection algorithm must be able to dynamically select high-bandwidth neighbors to download data.

PPLive VOD [10] currently has more than 210000 simultaneous online users and more than 700 channels. The average saving rate of server bandwidth has exceeded 88%, and the peak value can be up to 93%. Fluency and drag delay works well. The paper introduces P2PVOD system model, design difficulties and performance evaluation combined with PPLive VOD system.

The main contribution of the paper is as follows:

(1) P2PVOD reference model is introduced in detail, including system deployment, storage and distribution of streaming media data and client operation.

(2) Client plays a key role in P2P system. The paper introduces the client design of P2PVOD, which includes disk cache replacement algorithm, memory prefetching algorithm, neighborship maintaining, task scheduling and data upload strategies. These strategies adopted by PPLive P2PVOD system show good overall performance.

(3) The actual operation system shows that P2P technology can achieve large-scale VOD system. The server bandwidth saving rate can be generally higher than 88%, and reach 93% at peak time.

The structure of the paper is as below: section 2 introduces relative work; section 3 describes P2PVOD system reference model; section 4 gives P2PVOD design goals and corresponding difficulty analysis; section 5 discusses client strategy, mainly including data caching and scheduling; section 6 introduces the operation of current system, mainly including server bandwidth saving rate and user experience; section 7 concludes the paper and gives the prospect of further work.

2 Related work

Traditional server/client mode VOD system cannot support large-scale users. To solve this problem, many schemes are proposed, such as CDN [11] mode and P2P mode. CDN systems construct intellectual virtual network based on current Internet by placing node servers in the network, so that the bottlenecks that might affect data transmission rate and stability can be avoided and transmission can be faster and more stable. For users, not only can responding time be shortened largely, but also connection quality can be greatly improved. So the overall performance of online access can be effectively improved. However, VOD service providers still have to pay huge hardware and traffic cost.

Recent years, P2P has acquired great success in real-time streaming media application. A great many good-quality P2P demand softwares have appeared with high server bandwidth saving rate, such as CoolStreaming [1], PPLive [2], and PPStream [3]. Meanwhile, a great number of measurement analyses and theoretical researches support P2P demand systems as well. Passive detection technology [12] and active scramble method [13] were adopted to test system from client perspective. Wu et al. [14] measured popular P2P demand software from server's perspective. Hei et al. [15] used queuing theory and random flow to model P2P system. Zhou et al. [16] used sliding window technology for modeling so as to analyze buffer filling rate.

Compared with P2P real-time streaming media application, P2PVOD researches are still at the initial stage. The researches mainly include the following aspects:

(1) Measurement, including user behavior features of VOD system and P2PVOD system performance. Huang et al. [7] analyzed user behavior features of VOD system by using MSN Video data log of nine months. They considered that using P2P technology in VOD system can save up 95% server bandwidth. Cha et al. [17] analyzed video's life cycle in VOD system like YouTube and users' different request statistics features according to a great amount of collected video data. At the same time, the bandwidth saving rate brought by applying P2P technology into YouTube is analyzed as well.

(2) P2PVOD shared application layer design. Tree-shape application layer network architecture is widely used in organization nodes [18–21]; Cui et al. [22] proposed Ostream scheme that expands application layer multicast and caches data on peers to support VOD. Annapureddy et al. [8] studied the

flexibility mesh network brought to P2PVOD network by using simulation method.

(3) Support to VCR. Wang et al. [23] proposed a Dynamic Skip List (DSL) to dynamically support VCR operation; Guo et al. [24] supported VCR by dynamically improving VCR operation data request priority. He et al. [25] proposed a mixed caching mechanism and prefetching mechanism to improve VCR's delay; Cheng et al. [26] proposed a RINDY structure to support VCR.

(4) Data storage. Cheng et al. [27] compared the performance between single video sharing and multi-video sharing in disk caching strategies. They considered that multi-video sharing scheme can bring forth great performance improvement. Guo et al. [28] put forward a video distinguishing technology to share videos using BitTorrent method. Theoretical analysis considered that in P2P network, storing data in proportion can optimize user and network performance [29, 30].

Previous researches mostly lay emphasis on local algorithm's improvement on certain performance. The paper gives system reference model combined with large-scale deployed P2P VOD system—PPLive video on demand system from the whole design perspective. Main difficulties of such system design are analyzed, and the most important client design scheme is given. The actual operation shows average server bandwidth saving rate can reach 88% or even higher, and bandwidth saving rate of peak user number can be above 93% with good fluency and drag delay indicators.

3 P2PVOD reference model

Supporting VOD service in the Internet must support large-scale user base. We consider that under current network conditions, the following design goals of P2PVOD system are reasonable and feasible: supporting millions of online users, thousands of online videos, 400 kps play rates, user dragging and other video demand actions. With the uplift of user download bandwidth, especially with ADSL accessing bandwidth exceeding 2 Mbps, play rates above 1 Mbps will become the design goal of next step. From the design goal we can see that, how to design high-effective P2P client is the key to achieve system design. Here we need to propose P2P based system reference model first.

3.1 Architecture

Since P2PVOD systems are required to support stable service and peer program running on user host has strong dynamics, completely relying on user's peer program is unrealistic. Therefore, servers must be introduced in P2PVOD system. Servers set in P2PVOD system can well adjust performance of the whole system under certain condition and provide more reliable services for users. However, the existence of servers may restrict system scalability. Therefore, how to deploy servers and the function undertaken are essential. Integrating various factors, we proposed a mixed P2P network architecture, set index server Tracker, and provide the most basic resource searching function. Meanwhile, unstructured mesh network is set among peers, and peers can also locate resource and search through the network.

The mixed P2P network architecture we proposed is shown in Figure 1. It is mainly composed of Trackers, Publish servers, Image servers, Trans servers, Log servers and Peers.

Trackers: P2P index servers of the system responsible for peer find-and-search and used to manage nodes in different areas, and store data information owned by client nodes. Client needs to register Trackers when starting and periodically reporting resources.

Publish servers: mainly responsible for video publishing, registering published resource information to Trackers, and meanwhile sending video data to Image servers, and setting up copies of many data sources and generating video lists.

Image servers: responsible for storing video source data, diffusing data of the whole P2P network, and working as data supply servers after video data is diffused to some extent.

Trans servers: special for NAT penetrating of UDP message.

Log servers: servers used to collect logs, whose backstage data analysis system can be used for user behavior analysis and statistics.

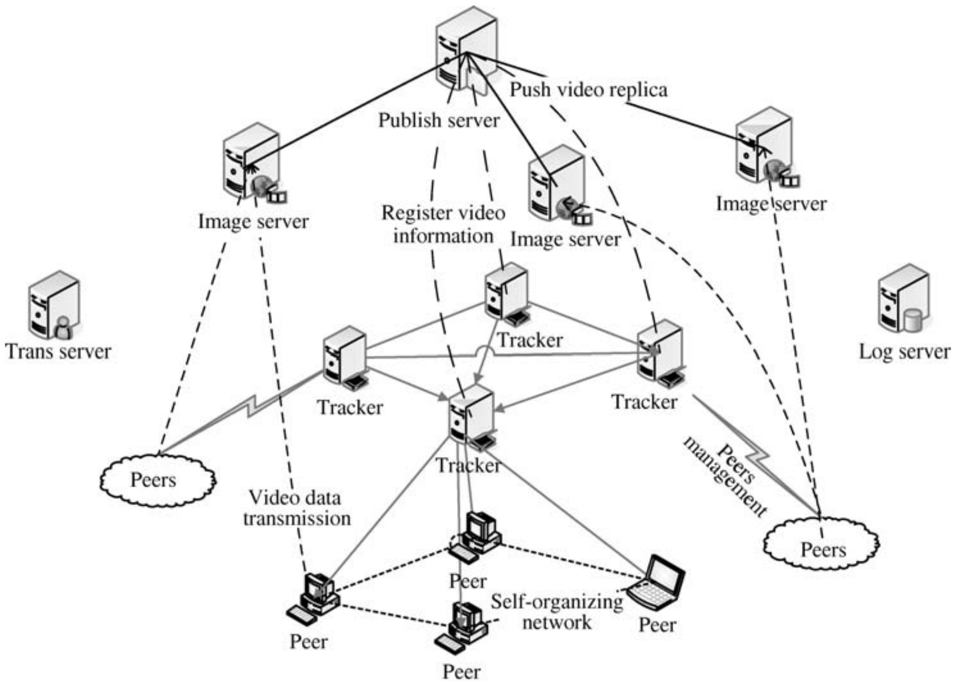

Figure 1 P2PVOD architecture.

In addition to the above servers, start servers can be set to help users' peers join the system, which is not drawn in Figure 1.

PPLive video demand systems carry out the deployment based on the reference model. PPLive as a whole has a Publish server and the system supports across heterogeneous network operation (Education network, Telecom, Netcom and other different ISPs). In each of these networks, there is at least one Tracker, which has its own Image server in heterogeneous network. Data transmission in ISP backbone network is reduced by localizing the whole network. Peers in the area can adopt self-organization (Gossip) method. The network is also easy to manage and maintain and has good controllability. Client (Peers) can directly face user's application software. As a common node in P2Pnetwork, a peer can capture streaming media data to reorganize videos, decode, play, and meanwhile provide physical data storage and service as well as change resources with other nodes in P2P network.

The system deployment model should be used flexibly combined with different requirements. For example, in PPLive deployment, despite the above advantages, we find Tracker autonomy has a performance issue. That is, peers in different areas cannot communicate effectively, especially when certain video has few online users. At this time, peers of one single area cannot constitute effective P2P network, since resource shortage fails to bring fast enough download data. However, if the connection among Trackers can be strengthened, the number of peers in P2P network can be increased so that the download rate among each other can be enhanced. Therefore, system designers should consider the pros and cons of these two different strategies when refering to the model.

As shown in Figure 2, client operation can be mainly divided as login, pause, drag, and logout. The following are brief introduction.

3.2 Login and logout

After clients login, they first join P2P network, connect Bootstrap server, acquire Tracker list, send connection request to these Tracker lists, choose Trackers that agree to join with the shortest responding time, register local information (resource, etc.) to them, from time to time carry out alive registration to Trackers, and then inform Trackers in case local resource has any changes. When clients logout the system, they first logout from P2P network, inform their neighbor peers and all the peers that are requesting data from them, and then inform logout messages to Trackers registered.

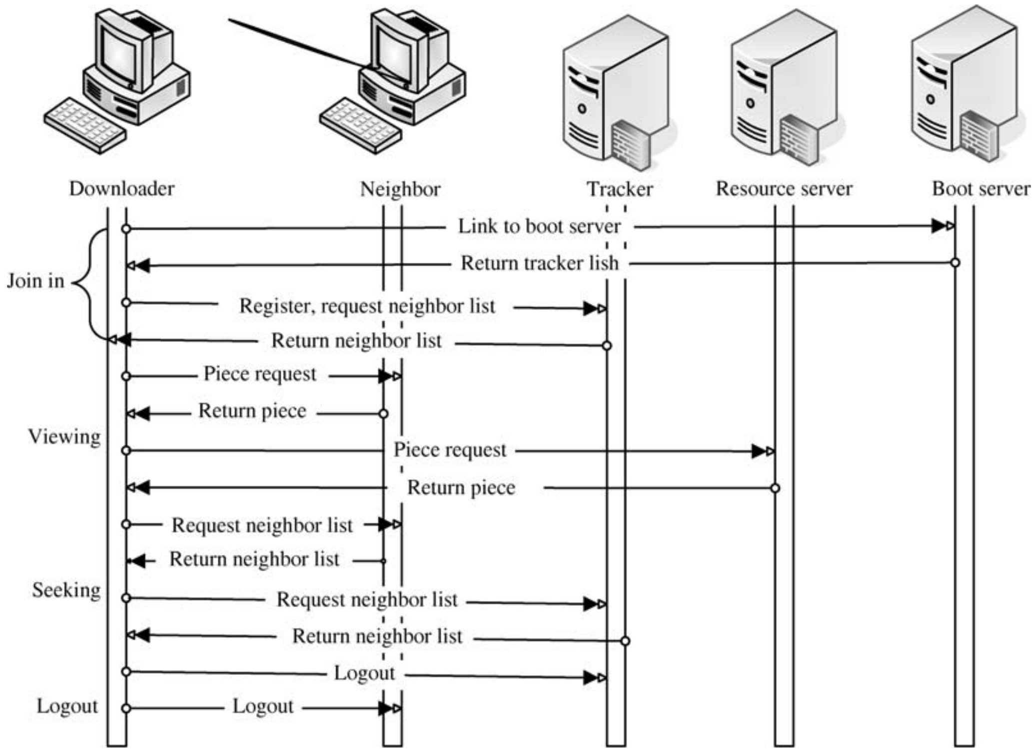

Figure 2 Client operation.

3.3 Play

When clients play a video, they first acquire the addresses of source servers from the video list, and at the same time search addresses of online peers that own the video data (download peers include peers that are currently viewing the video and peers that have local data but are not viewing the video). And then they request BBM information from these peers, and meanwhile add the peers that are viewing the video into neighbor list. Once resource index information is received, send data download request to the peers. When the number of downloadable peers is less than the threshold, the peers they currently use can be acquired from their neighbor list to supply local record. All the peers owning the video resources compose a mesh network. Peers that are viewing the video play a role as network builder and maintainer.

3.4 Drag and stop

VOD system must support drag operation in users' viewing process. When users drag the progress bar forward or backward to decide certain time point to view the video, system will first check if the local disk has data. If yes, directly play the video by quickly starting player; if not, continuously download data needed to peers in use, and start player to view the video until the cache area is sufficient. When the video is finished or pauses, system will inform current neighbor peers to stop. These neighbor peers will delete corresponding records, and the play state in the network is changed as well. At this time, data download task is ended; however, data upload will not be stopped.

4 Design goals and difficulties

The design goals of P2PVOD system mainly include server bandwidth saving rate from system perspective as well as fluency and drag delay from user experience perspective. Meanwhile, design difficulties of each goal are analyzed. Aiming at these difficulties, we give solutions from client design perspective in section 5, and give PPLive measurement of the three performance indicators in section 6.

4.1 Server bandwidth saving rate

Too much server bandwidth cost is the main reason that causes VOD service unprofitable. It directly affects the scalability of P2PVOD system. Therefore, server bandwidth saving rate is the most fundamental indicator. Larger indicator means better performance. The most ideal result is approaching 100%, that is, almost all the download traffic can be solved through data transmission among peers. We define the model as follows.

Suppose there are M videos at time t, video i has N_i online users, and the whole video has totally N online users. U_{ij} denotes the jth user of video i at time t. P_{ij} denotes data downloaded from other peers. S_{ij} denotes data downloaded from server. Then at time t, R_{ij} denotes the server bandwidth saving rate of U_{ij}, R_i denotes server bandwidth saving rate of video I, and R denotes the server saving rate of the whole system respectively.

$$R_{ij} = \frac{P_{ij}}{p_{ij} + S_{ij}}, \tag{1}$$

$$R_i = \frac{1}{N_i} \sum_{j=1}^{N_i} R_{ij}, \tag{2}$$

$$R = \frac{1}{N} \sum_{i=1}^{M} \sum_{j=1}^{N_i} R_{ij}. \tag{3}$$

Suppose the network performance is certain. The difficulties of improving server bandwidth rate lie in: first, peers have enough potential resources to share; second, peers can choose suitable peers to download. The first point is decided by disk caching strategy, and the second point is decided by task scheduling. Realization schemes of the above two issues are given in section 5. The following introduces the specific design difficulties of the two issues.

Disk caching strategy includes the following aspects:

(1) Single video sharing or multiple videos sharing. Single video sharing means that peers only store and share the videos that are currently viewed. And multiple videos sharing means all the viewed videos can be stored and shared as long as there is enough cache space. Single video sharing can provide the simplest sharing strategy, while multiple videos sharing strategy can bring flexibility as well as great complexity to systems. So strategy selection is a difficulty in system design.

(2) Size of sharing disk cache. From system perspective, the larger the better; however, from user perspective, the smaller the better. Therefore, a trade-off design becomes a difficulty. We consider that there exists a turning point that when sharing space exceeds it, performance will not be largely improved through expanding space; however, when sharing space is less than it, performance will be decreased sharply with the fast reduced sharing space.

(3) The optimal distribution of each video copy. Intuitionally, proportional distribution is a good choice. But what makes the best distribution still needs theoretical or practical proofs.

(4) Designing a replacement algorithm to achieve the optimal distribution. Trackers can rely on the overall statistics and demand distribution and send the result to every peer, and guide peer disk to carry out cache replacement. Although this overall replacement algorithm can get good guiding effect, the system scalability is limited. Scalability can be improved by using distributed replacement algorithm that only needs local information. The most direct approach is to replace the earliest viewed video; however, the replacement effect will be influenced for sure. So making a trade-off between the overall replacement algorithm and distributed replacement algorithm is a tough issue.

Task scheduling strategy should satisfy the following conditions:

(1) Stability. When many fellow peers can provide data for peer A, the resource of each fellow peer can be fully utilized. When certain peer leaves or fails to provide data, peer A can still acquire data from other fellow peers to ensure the video stability.

(2) Adaptability. The capacity of fellow peers that can provide peer A with data is different, so the algorithm can continuously self-adapt the change of fellow peer's service capacity.

(3) Load balance. Peer A can fully utilize each fellow peer's resource. In the adjustment, considering service capacity of each peer, data are distributed equally to ensure load balance among peers. Make sure current peers are distributed with high, medium, and low connection speed to ensure high speed download while fully utilizing peers with low upload bandwidth.

4.2 User experience

User experience mainly includes two indicators: one is fluency, that is, the playback pause happening in the order play due to low download speed; the other is drag delay, that is, the play time interval of caching certain play time (20 s for PPLive) after dragging to some position. The model is defined as follows:

Suppose there are M videos in the system, video i has N_i online users, and the whole video has totally N online users. We assume the jth user (denoted by U_{ij}) of the ith video drag C_{ij} times, $T_s(i,j)$ denotes the video play time of the dragged beginning position, and $T_e(i,j)$ denotes the play time of the dragged end position which means the next drag. Then $T_e(i,j) - T_s(i,j)$ denotes effective viewing time, $T_a(i,j)$ denotes the actual viewing time since buffer time after drag and that caused by insufficient download rate. Usually $T_a(i,j)$ is larger than $T_e(i,j) - T_s(i,j)$. F_{ij} denotes the viewing fluency of user U_{ij}, F_i denotes the viewing fluency of video I, and F denotes the viewing fluency of all the videos in the system. $T_b(i,j)$ denotes the starting play time after enough drag buffer. D_{ij} denotes the average drag delay of U_{ij}, D_i denotes the average drag delay of video i, and D denotes the average drag delay of all the videos in the system.

$$F_{ij} = \frac{1}{C_{ij}} \frac{\sum\limits_{k=1}^{C_{ij}} (T_e(i,j) - T_s(i,j))}{\sum\limits_{k=1}^{N_{ij}} T_a(i,j)}, \tag{4}$$

$$F_i = \frac{1}{N_i} \sum_{j=1}^{N_i} F_{ij}, \tag{5}$$

$$F = \frac{1}{N} \sum_{i=1}^{M} \sum_{j=1}^{N_i} F_{ij}, \tag{6}$$

$$D_{ij} = \frac{1}{C_{ij}} \sum_{k=1}^{C_{ij}} (T_b(i,j) - T_s(i,j)), \tag{7}$$

$$D_i = \frac{1}{N_i} \sum_{j=1}^{N_i} D_{ij}, \tag{8}$$

$$D = \frac{1}{N} \sum_{i=1}^{M} \sum_{j=1}^{N_i} D_{ij}. \tag{9}$$

We can see that the core factor affecting fluency is download speed which is also affected by disk cache and task scheduling. When download speed is faster than play speed, high-effective peers provide download data continuously with theoretically 100% fluency. Drag delay is relatively complex. The influential factors mainly include: first, the acquiring of new neighbors after drag and the connection setup time; secondly, new neighbor peers' upload capacity according to downloaders.

Besides random access and leave, there are also random drag and pause in the VOD system. A single user cannot present all the circumstances. For example, a user viewing half of the video can only provide the half data sharing. These make P2P topology change drastically. The high dynamic feature of VOD system will bring great difficulties in improving drag delay. We observed that the peers that are currently downloading probably do not have data of viewing point after drag, especially when the drag is a big-range one. In this scenario, new neighbors with data after drag should be searched and connection should

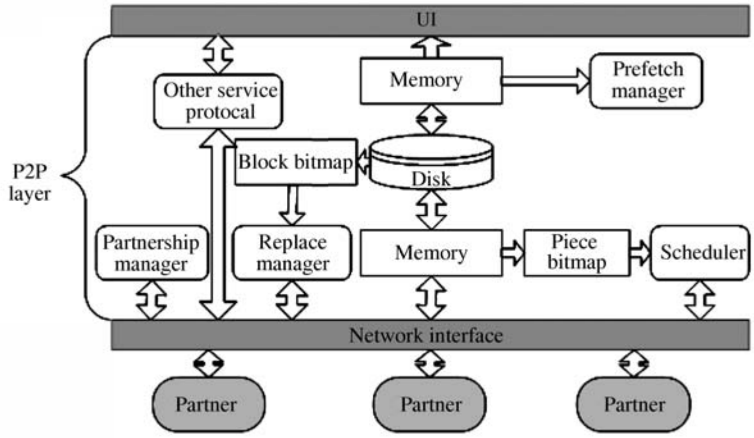

Figure 3 Client architecture.

be reset. Besides, peers with high download speed should be selected as soon as possible. Therefore setting up connection as soon as possible is a design difficulty. In PPLive, this can be solved through maintaining multi-level neighborship, as will be introduced in subsection 5.2.1.

5 Clients design

In P2P system, clients are the most important modules. And data replication and data schedule are two main strategies to optimize system performance.

5.1 Client overview

Client (Peers) is the main body of VOD system to achieve P2P function. It can be used either as data sender or receiver. Logically, we can divide client into three levels: network layer, P2P layer and application layer. Network layer is responsible for receiving and sending all the messages. P2P layer impels network data request and analyzes high-level message types. Application layer is responsible for media play. P2P layer mainly has four key modules: 1) Partnership manager, responsible for neighbor peers' addition, deletion, revision as well as neighborship maintenance; 2) Scheduler, responsible for scheduling data transmission, including data download and upload; 3) Replace manager, responsible for replacement strategy of disk cache; 4) Prefetch manager, responsible for memory data replacement and prefetching strategy. Layered protocol structure is shown in Figure 3.

Media data requests adopt pure 'pull' mode: Application layer sends request sequence to P2P layer; P2P layer acquires resource peers from P2P network to maintain neighborship. Peer selection, multi-point download and task scheduling should meet quality requirement (i.e. video sequence and play fluency) of streaming media real-time play. P2P layer acquires data from data transmission of underlying network, and then sends back to application layer. Application layer then redistributes the data and send to players to play after the cached data are full. The transmission of data pieces and protocol signals all adopts UDP message mode, so do the algorithms mentioned below.

5.2 Data download

Data download process first selects a group of peers as neighbors (potential download objects), and then requests data from a group of them. Others are spare neighbors that cannot be used temporarily. According to responding speed of different neighbors, download tasks assigned to each neighbors are dynamically adjusted. Meanwhile, relationship in the whole process is maintained, new neighbors are introduced, and neighbors either with bad connection or without any data request are dropped. The following discusses neighborship maintenance and task scheduling separately.

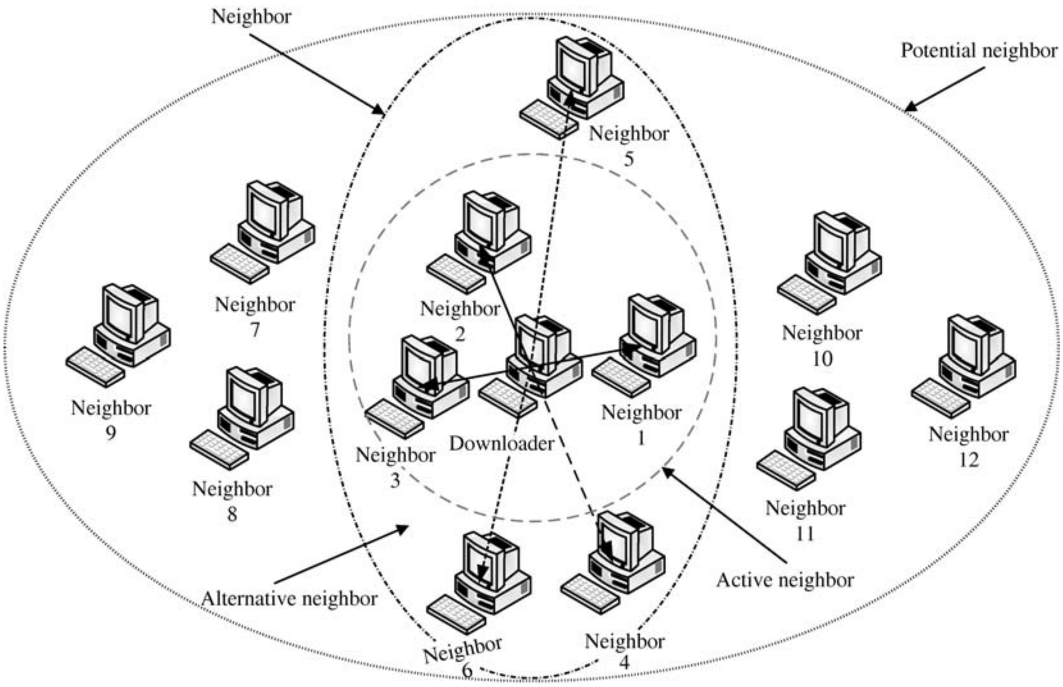

Figure 4 Neighborship.

5.2.1 *Neighborship*

Figure 4 shows four kinds of neighborships:

(1) Potential neighbors: peers that have viewing resources.

(2) Neighbors: peers that are maintained by local client and have certain amount of viewing resources, which include active neighbors and potential neighbors.

(3) Active neighbors: peers that are being downloaded by users.

(4) Alternative neighbors: inactive peers used as spare ones when active neighbors are replaced for some reasons, such as lack of current requested data or a sudden decrease in current active peers' download speed that affects paly fluency. Figure 4 illustrates the relations among these concepts.

Neighborship maintains modules through dynamic adjustment to make the overall download speed provided by active neighbors a little faster than the play speed (the average code rate). Maintaining several spare neighbors is to quickly replace or directly join some active neighbors when the download speed provided by active neighbors is not fast enough, so as to enhance the download speed. Potential neighbors can acquire their list through Trackers or connected neighbors to set up connections when current neighbors in use are too few.

5.2.2 *Job scheduling*

When initializing, select one group of peers as neighbors according to pieces resource ownership of potential neighbors and set the transmission performance of clients and each neighbor as the same. Transmission performance can be described by download speed (i.e., time description of downloading each piece). Send paralleled data request to neighbors according to task assignment, and receive data pieces returned. Then peers periodically update download speed of each neighbor and reassign download task. The ones with higher download speed are assigned more task and send/request data. Periodically judge and update the proportion of neighbors and their tasks assigned based on information assigned and transmission performance changed. The adjustment carries on until data transmission is finished. This guarantees the stability of the overall download data according to current download speed of neighbors.

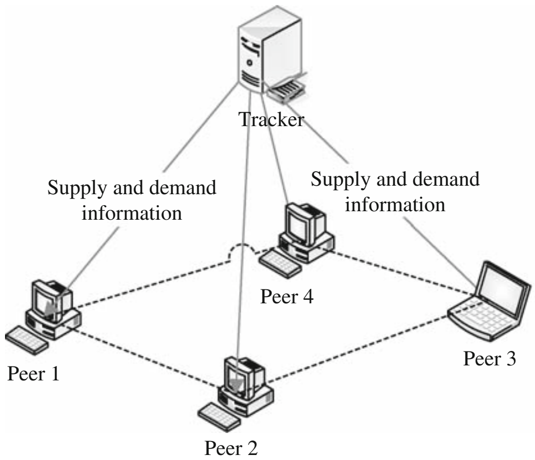

Figure 5 Collection and publishing of supply-demand ratio.

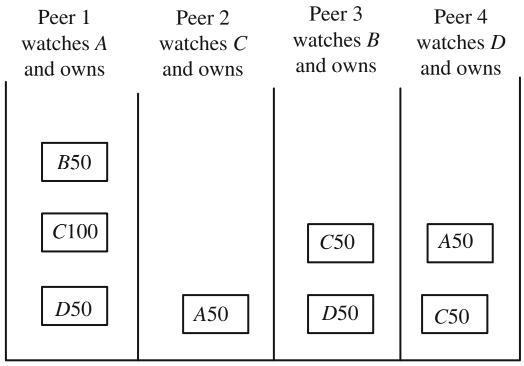

Figure 6 Resource distribution.

5.3 Data upload

Upon receiving data request of other peers, client tries to deal with every request. But there are two restrictions:

(1) Upload restriction: Prevent overflow upload packets from affecting intersection of other packets; real-time monitor other packets' communication status besides upload packets. When packets communication is affected by high upload, reset to reduce the upload speed.

(2) Connection restriction: Provide limited bandwidth to limited peers to ensure peers' quality of service. The maximum connection number of peers that can provide service can be set based on the positive feedback of upload speed. The larger is the upload speed, the larger is the maximum connection number accordingly. When connection number of peers exceeds the maximum, drop the requested packets directly without return.

The strategy is so easy that upload peers only need to fairly dispose all the requested data upload. In actual operation, data upload side has priority to transfer scarce resource in the system only by dynamically adjusting download task.

5.4 Disk cache replacement

Researches show that in P2P network, proportional copies storage can achieve the optimal network performance and user experience [11, 12]. PPLive VOD system currently adopts disk cache replacement algorithm to realize the distribution using global supply-demand ratio. That is, select the video with the highest supply-demand ratio in disk cache space, and then delete all the pieces belonged to this video. The global supply-demand ratio is the ratio between copies of all video pieces and online users that are viewing this video. Receive-send sketch map is shown in Figure 5. Suppose there are four videos A, B, C, D; four peers start to view their selected videos separately. The replicas of the videos are shown as Figure 6. Trackers collect and distribute videos viewed by all the users, and store video information. A, B, C, D are viewed by each of the four users, A: 50+50=100, B: 50, C: 100+50+50=200, D: 50+50=100. Assume the cache of Peer 1 is full at this time. Then the supply-demand ratio acquired is: C, D, A. Then C is replaced.

6 PPLive P2PVOD performance evaluations

This section illustrates the effectiveness of the design scheme by using actual deployment of large-scale P2PVOD system (i.e., PPLive VOD system's performance indicators on bandwidth saving rate, fluency, drag delay, etc.). These performance indicators are defined in section 4. Three videos with different

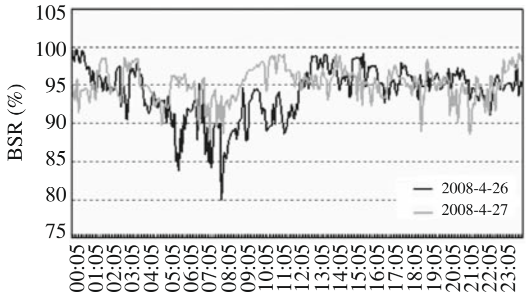

Figure 7 BSR of movie 1.

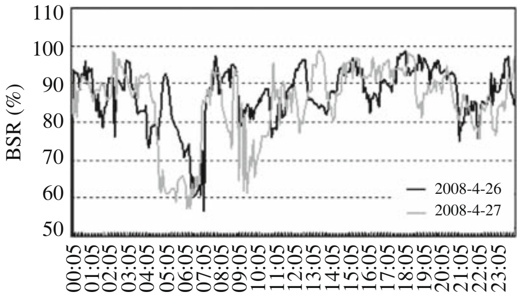

Figure 8 BSR of movie 2.

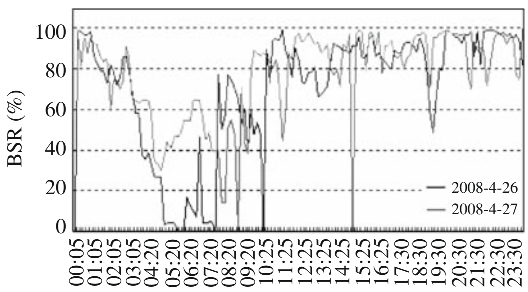

Figure 9 BSR of movie 3.

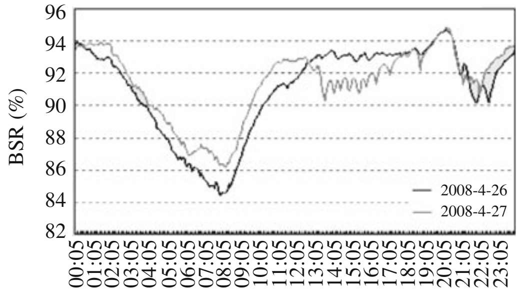

Figure 10 BSR of the whole system.

popularity as well as the overall video data in the system are tested for seven days from Dec. 11, 2007 to Dec. 17, 2007 and another two days from April 26, 2008 to April 27, 2008.

6.1 Server bandwidth saving rate

To quantify bandwidth cost saving rate, three types of videos with different popularities and all the videos in the system are sampled. The videos sampled are high-popular movie1, low-popular movie 3 and medium-popular movie 2. The more online users, the more popular the video is. Sample time is forty-eight hours from 0:00 April 26, 2008 to 24:00 April 27, 2008.

Figures 7–9 are the comparison charts of the three video servers with different popularity. We can see that 1) Bandwidth saving rate has the daily cycle tendency. 2) The bandwidth saving rate of single video server has considerable jitter. The less popular videos usually have more obvious jitter. The jitter of the most popular movie 1 mostly achieves 85%–100%. The bandwidth saving rate of the medium-popular movie 2 is 60%–100%. And that of the lowest popular movie 3 is 0–100%. 3) The popular videos usually have higher bandwidth saving rate, with the average around 95%, 80%, and 60%.

Figure 10 shows the comparison of the overall server bandwidth saving rates in two days. 1) Average bandwidth saving rate is around 90%. 2) Daily cycle has strong regularity with very smooth change. 3) The lowest point appears at about 8:00 approaching 85%, and a small valley appears at about 22:30 approaching 90%, and the peak appears at about 20:00 approaching around 95%. Compared with the change of online user number in one day, we can see that the lowest bandwidth saving rate appears at around 8:00, since P2P mechanism cannot take effect with few users. The small valley appears at 22:30 when the user number reaches the peak. Therefore, we can see that the sudden increase of user number may cause the decrease of bandwidth saving rate. 4) Generally, server bandwidth saving rate increases with the user number. Later we will see that this feature is very useful to actual bandwidth saving rate, as the bandwidth of the peak user number decides whether VOD service providers purchase the bandwidth.

Compared with daily average bandwidth saving rate, bandwidth saving rate at the peak user number time has more practical meaning. This is because the peak time is basically the time when servers pay the highest bandwidth. And the network bandwidth purchased by VOD service providers is a fixed value, which cannot be changed according to the actual situation. To provide good service at all times, the

Table 1 Server bandwidth saving at peak time

Peak users	BSR	Playback rate	Saved bandwidth	RMB/(Gbps*month)	Saves/month
210000	93%	380 kbps	77.406 Gbps	50000	3870300

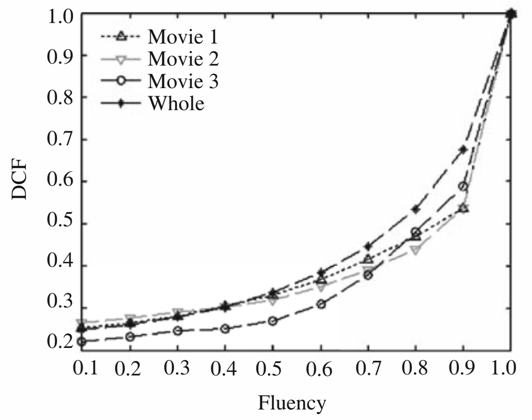

Figure 11 Viewing fluency distribution.

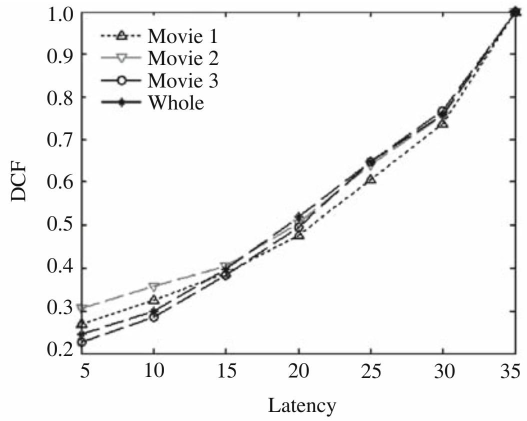

Figure 12 Drag delay distribution.

bandwidth purchased should be the bandwidth of the peak time. And the peak server bandwidth value usually appears at the peak user number time. So the bandwidth saving rate of the peak user number directly decides the saving of the actual bandwidth purchase.

Table 1 shows the bandwidth saving rate at the peak online user number time. It can be seen that the bandwidth saving rate at peak time 93% is higher than the average 88%. With 210000 online users and 380 kbps play code rate, the actual bandwidth saving rate is as high as 77.406 Gbps. According to the current standard RMB 50000 Yuan/Month/Gbps, RMB 3870300 Yuan can be saved monthly.

6.2 User experience

To quantify fluency and drag delay, three types of videos with different popularity and all the videos in the system are sampled using Formulae defined in subsection 4.2. The videos sampled are high-popular movie 1, low-popular movie 3 and medium-popular movie 2. Sample time is forty-eight hours from 00:00 April 21, 2008 to 24:00 April 21, 2008.

Figure 11 shows the fluency distribution of three videos with different popularity. We can see that generally 68% viewing fluency is smaller than 0.9. Fluency is almost the same for videos with different popularity. And these videos can be supported fairly. Service can provide extra download traffic, even though enough download speed cannot be acquired from peers. Videos' low fluency lies in the statistic method used by log servers, where drag and pause time is also counted into waiting time.

Figure 12 shows the drag time distribution of three videos with different popularity. We can see that more that 70% drag time is within 30 s, and the starting speed is evenly distributed from 5 to 35 s which are mainly decided by the download speed of neighbor peers after drag. Since neighbor peers are selected randomly to download after drag, the drag time for videos with different popularity differs slightly.

7 Conclusions and further work

The paper first proposes a mixed P2P network structure, which takes Trackers as servers to provide the most fundamental resource search functions. Meanwhile, unstructured mesh networks are set up among peers, and peers can also locate and search resources through these networks. Based on this structure and model, data storage and distribution of streaming media as well as users' operation behaviors are introduced, and main difficulties in P2PVOD design are analyzed. We deem that the key lies in how to fully take advantages of peers to reduce servers' load. The main technique used is designing corresponding replacement algorithm, dynamically maintaining neighborship and adjusting data download through studying data copies' reasonable distribution. The client strategy design plays a critical role in solving these issues. P2PVOD system combined with PPLive gives the design model for the client, disk cache replacement algorithm, memory cache prefetching algorithm, data download and upload strategies. Finally, performance of PPLive P2PVOD system including server bandwidth saving rate, viewing fluency and drag speed are analyzed based on the results from the overall videos and three videos with different popularity. Data show that the average server bandwidth saving rate can reach 88% or even higher, and the peak value can be up to 93%. The paper verifies the huge advantages of P2P technology in VOD service, and provides a reference model regarding to P2PVOD client design. In the research, we observe that data pieces of a same video are unevenly distributed, usually more in the beginning and less in the end. Next we will study the disk replacement algorithm deployed with smaller granularity. Meanwhile, we will also consider further neighbor filter for Trackers to enhance the responding speed after drag.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 60970104), the National High-Tech Research & Development Program of China (Grant Nos. 2009AA01A334, 2008AA01A326), the National Basic Research Program of China (Grant No. 2009CB320501), and the Program for New Century Excellent Talents in University.

References

1 Zhang X, Liu J, Li B, et al. Coolstreaming/donet: A data-driven overlay network for efficient live media streaming. In: Proceedings of INFOCOM. Miami, FL, USA, 2005. 2102–2111
2 PPLive. http://www.pplive.com/
3 PPStream. http://www.ppstream.com/
4 GridCast. http://www.gridcast.cn/
5 YouTube. http://www.youtube.com/
6 YouKu. http://www.youku.com/
7 Huang C, Li J, Ross K W. Can internet video-on-demand be profitable? In: Proceedings of ACM SIGCOMM'07, Kyoto, Japan, 2007. 133–144
8 Annapureddy S, Gkantsidis C, Rodriguez P R, et al. Providing video-on-demand using peer-to-peer networks. Microsoft Research Technical Report. MSR-TR-2005-147, Oct. 2005
9 Huang C, Li J, Ross K W. Peer-assisted VOD: Making internet video distribution cheap. In: Proceedings of IPTPS. Bellevue, WA, USA, 2007. 1–5
10 PPLIVE-VOD. http://www.pplive.com/subject/20070808pfsvod/
11 Cherkasova L, Lee J. FastReplica: Efficient large file distribution within content delivery networks. In: Proceedings of 4th USENIX Symposium on Internet Technologies and Systems. Seattle, WA, USA, 2003. 7
12 Ali S, Mathur A, Zhang H. Measurement of commercial peer-to-peer live video streaming. In: Proceedings of 1st Workshop on Recent Advances in P2P Streaming. Hilton, Singapore, 2005. 1–6
13 Hei X, Liang C, Liu Y, et al. A measurement study of a large-scale P2P IPTV system. IEEE Trans Multimedia, 2007, 9: 1672–1687
14 Wu C, Li B, Zhao S. Multi-channel live P2P streaming: refocusing on servers. In: Proceedings of IEEE INFOCOM 2008, Phoenix, AZ, USA, 2008. 1355–1363
15 Hei X, Liu Y, Ross K W. Stochastic fluid theory for P2P streaming systems. In: Proceedings of INFOCOM 2007, Anchorage, AK, USA, 2007. 919–927
16 Zhou Y, Chiu D M, Lui J C S. A simple model for analyzing P2P streaming protocols. In: Proceedings of IEEE ICNP 2007, Beijing, China, 2007. 226–235
17 Cha M, Kwak H, Rodriguez P, et al. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In: Proceedings of IMC. 2007, San Diego, California, USA. 2007. 1–14
18 Al Hamra A, Biersack E W, Urvoy-Keller G. A pull-based approach for a VoD service in P2P networks. In: Proceedings of HSNMC 2004, Toulouse, France, 2004, 3079: 995–1006
19 Guo Y, Suh K, Kurose J, et al. P2Cast: Peer-to-peer patching scheme for VoD service. In: Proceedings of WWW 2003, Budapest, Hungary, 2003. 301–309

20 Tai D, Kien H A, Mounir T. P2VoD: Providing fault tolerant video-on-demand streaming in peer-to-peer environment. In: Proceedings of ICC 2004, Orlando, FL, USA. 2004, 3: 1467–1472

21 Wang D, Liu J. Peer-to-peer asynchronous video streaming using skip list. In: Proceedings of ICME 2006, Toronto, Ont, Canada. 2006. 1397–1400

22 Cui Y, Li B, Nahrstedt K. Ostream: A synchronous streaming multicast in application-layer overlay networks. IEEE JSAC, 2004, 22: 91–106

23 Wang D, Liu J C. A dynamic skip list-based overlay for on-demand media streaming with VCR interactions. IEEE Trans Paral Distr Syst, 2008, 19: 503–514

24 Guo Y, Yu S C, Liu H, et al. Supporting VCR operation in a mesh-based P2P VoD system. In: Proceedings of Consumer Communications and Networking Conference. Las Vegas, NV, USA, 2008. 452–457

25 He Y, Liu Y H. Supporting VCR in peer-to-peer video-on-demand. In: Proceedings of ICNP 2007, Beijing, China, 2007. 328–329

26 Cheng B, Jin H, Liao X F. Supporting VCR functions in P2P VoD services using ring-assisted overlays. In: Proceedings of ICC 2007, Glasgow, Scotland. 2007. 1698–1703

27 Cheng B, Stein L, Jin H, et al. Towards cinematic internet video-on-demand. In: Proceedings of the 3rd ACM SIGOPS/EuroSys 2008, Glasgow, Scotland UK, 2008. 109–122

28 Guo Y, Mathur S, Ramaswamy K, et al. PONDER: Providing commercial-quality video-on-demand service using peer-to-peer network. In: Proceedings of GLOCOM 2007, Washington, DC, USA, 2007. 225–230

29 Tewari S, Kleinrock L. Analysis of search and replication in unstructured peer-to-peer networks. In: Proceedings of ACM SIGMETRICS 2005, New York, NY, USA, 2005. 404–405

30 Tewari S, Kleinrock L. Proportional replication in peer-to-peer networks. In: Proceedings of INFOCOM 2006, Barcelona, Spain, 2006. 1–12