Peer-to-Peer Netw. Appl. (2018) 11:1060–1074
DOI 10.1007/s12083-017-0575-3

Migrating big video data to cloud: a peer-assisted approach for VoD

Fei Chen[1] · Haitao Li[2] · Jiangchuan Liu[3] · Bo Li[3] · Ke Xu[4] · Yuemin Hu[3]

Received: 16 February 2017 / Accepted: 30 May 2017 / Published online: 1 July 2017

Abstract Since user demand for a Video-on-demand (VoD) service varies with time in one-day period, provisioning self-owned servers for the peak load it must sustain afew hours per day leads to bandwidth under-utilization at other times. Content clouds, e.g. Amazon CloudFront and Azure CDN, let VoD providers pay by bytes for bandwidth resources, potentially leading to cost savings even if the unit rate to rent a machine from a cloud provider is higher than the rate to own one. In addition, recent studies have presented fog computing as a new paradigm to extend the cloud-based platform for a cost-effective and highly scalable service. In this paper, based on long-term traces from two large-scale VoD systems and temporal development model of content clouds, we tackle challenges, design and potential benefits in migrating both Clients/Server-based and peer-assisted VoD services into the hybrid cloud and edge peers in fog computing environment. Our measurements show that the popularity of the most popular videos decays so quickly, for example, by 11% after one hour that it poses large challenges on updating videos in the cloud. However, the trace-driven evaluations show that our proposed migration strategies (active, reactive and smart strategies), although simply based on the current information, can make the hybrid cloud-assisted VoD deployment save up to 30% bandwidth expense compared with the Clients/Server mode. Moreover, they can also handle the flash crowd traffic with little cost. Leveraging the edge peers in fog computing, we propose a cloud-friendly peer replication strategy, which further reduces the migration cost by a factor of 4. Our simulation also shows that the cloud price and server bandwidth chosen play the most important roles in saving cost, while the cloud storage size and cloud content update strategy play the key roles in the user experience improvement.

Keywords Cloud computing video · Content distribution · Peer-assistance

This article is part of the Topical Collection: *Special Issue on Big Data Networking*
Guest Editors: Xiaofei Liao, Song Guo, Deze Zeng, and Kun Wang

✉ Yuemin Hu
yueminghugis@163.com

Fei Chen
chenf@jiangnan.edu.cn

Haitao Li
haitao.li@ucarinc.com

Jiangchuan Liu
csljc@ieee.org

Bo Li
drlibo@gmail.com

Ke Xu
xuke@mail.tsinghua.edu.cn

[1] School of Digital Media, Jiangnan University, Jiangnan, China

[2] UCAR Technology (China) Limited, Beijing, China

[3] College of Natural Resources and Environment, South China Agricultural University, Guangzhou, China

[4] Department of Computer Science and Technology, Tsinghua University, Haidian Qu, China

1 Introduction

Video-on-demand (VoD) has become an extremely popular service in the Internet [20]. Typically today' ISPs bill a VoD provider for bandwidth usage using the 95 percentile rule, which works as follows: The average server bandwidth is measured every 5 minutes within each month. These bandwidth measurements over a month form a set of values, and the 95 percentile value is the smallest number that is greater than 95% of the values in the set. Since the user demand for a VoD service varies with time in one-day period, provisioning self-owned servers for the 95 percentile value however it must sustain a few hours per day leads to bandwidth underutilization at other times. For example, in PPLive [8, 17], the utilization ratio is less than 20% for more than 50% times with an average value of 40%. The 95 percentile value is 5 times of the lowest value. Moreover, the provision for a flash crowd is extremely expensive even if the flash crowd can be predicted.

Generally, it is sophisticated to design a cost effective VoD system, which is featured with viewer demand dynamics. The conventional service providers have suffered from high bandwidth consumption and large volume of video replication in a high frequent video popularity variation environment. Fortunately, content cloud platforms (e.g., Amazon CloudFront [1] and Azure CDN [3]) with elastic resource provisioning are becoming increasingly popular. They are based on a "pay-as-you-go" paradigm for enabling convenient, on-demand network access to a shared pool of configurable bandwidth and storage resources that can be rapidly provisioned and released with minimal management effort [12, 24]. In addition, the concept of fog computing, which has been suggested since 2014, aims to process certain workloads and services locally on edge devices or edge servers (fog), rather than bing transmitted to the cloud [18]. This paradigm introduces a new intermediate fog layer between end users and cloud, which is composed of geo-distributed fog servers which are deployed at the edge of networks [11]. These fog servers facilitate a wide range of services, such as geo-distributed and latency-sensitive applications [14, 22, 30].

Hence, it is a good idea to develop a hybrid cloud-assisted VoD delivery system, including distributed peers, self-owned servers (also called servers in short), cloud storage and cloud CDN. The self-owned servers, which are owned by the VoD providers, store all the original video files, serve part of user requests and upload videos to the cloud storage. The cloud storage stores part of video files and pushes these videos to its cloud CDN. The cloud CDN delivers streaming content using a global network of edge locations. The requests for videos are automatically routed to the nearest edge location, thus the contents are delivered with the best possible performance. Both the cloud storage and the cloud CDN are owned by cloud providers. Based on the hybrid cloud-assisted VoD development, the clients can download video data either from the servers or from the cloud. The content clouds let VoD providers pay by bytes for the bandwidth resources, potentially leading to cost savings even if the unit rate to rent a machine from a cloud provider is higher than the rate to own one. For example, our analyses show that VoD providers can save more than 30% bandwidth expense by migrating 40% traffic to the cloud, if the unit bandwidth price of the cloud is twice as that of the ISPs. Furthermore, the hybrid cloud-assisted deployment can handle burst traffic with trivial cost compared with over-provisioning in the self-owned servers. It just needs to buy more cloud storage. And if there is burst traffic, the additional requests can be directed to the cloud.

To design such a hybrid cloud-assisted VoD system, a fundamental question is how to balance the video replicas and traffic load among the self-owned servers, the cloud platform, and edge peers in a cost-effective manner. Since a large-scale VoD site can store hundreds of thousands of videos and a large volume of traffic are directed to the cloud in our hybrid solution, it is required that cloud must store a huge amount of files to serve such traffic. While the expense to upload those files is high, including the cloud storage cost and especially the bandwidth cost for uploading video to the cloud, we should carefully design our migration strategy: How much traffic should be directed to the cloud? How many files should be stored in the cloud and what are they? Should we update the set of videos stored in the cloud? And how do we update? Obviously, our target of designing a good migration strategy is to save the aggregate cost while minimize the unmet user requests as much as possible. In order to save the cloud storage cost and updating cost, we choose to store the most popular videos. Even though, the cloud content updating cost can be very high, since the video popularity changes very frequently. For example, our measurements show that 11% of top-5000 videos in Hulu [5] will be changed after an hour. A good update strategy should consider many aspects. For example, it is expected to utilize the uploading bandwidth of self-owned servers for content replication when they are idle. It is also expected to upload videos that will be popular in near future. Furthermore, the difficulty to predict the videos popularity makes our task even harder.

For the first time, this paper studies the challenges, design and potential benefits of the hybrid cloud-assisted VoD deployment. We first propose a cloud-assisted VoD architecture and formulate the problem. Then using the traces from three large-scale video streaming systems, we extract many key characteristics of these systems which are relevant to the hybrid cloud-assisted deployment. We design three heuristic migration strategies and make extensive trace-driven performance evaluation.

Besides Clients/Server-based VoD systems, there also exist many large-scale peer-assisted VoD systems (e.g., PPLive, PPStream [9], and Joost [7]). Considering the limited peers' uplink capacity and the popularity of high definition movies, these systems still need to provide huge server bandwidth. For example, PPLive needs at least 10Gbps server bandwidth to support its peer-assisted VoD service. Similar to Client/Server-based VoD systems, the server bandwidth utilization is less than 40%, with much idle bandwidth in the morning [17]. Therefore, it is also beneficial to migrate partial videos of such systems to content clouds. We analyze the necessary change in current peer-assisted VoD systems when they are migrated to content clouds, and find the migrated videos to cloud can be reduced by a large margin through only small change in peer replication strategy.

The contribution of this paper are as follows:

(1) We collect the traces from two large-scale VoD services, (i.e. Hulu and PPLive), and a crowdsourced video platform (i.e. Twitch.tv). The Hulu trace contains the top-5000 most popular videos information every hour over one month. The PPLive trace contains three parts: the simultaneous online users every 5 minutes over 10 months; the integrated server bandwidth load; and the video popularity distribution. The twitch.tv trace contains the source stream distribution and view demand distribution in one day, respectively. We process these data to extract many of the key characteristics of large-scale video streaming deployments. Particular attention is given to the characteristics relevant to the cloud migration deployment.

(2) Aiming to meet the clients' requests while minimizing the total bandwidth cost, we design three heuristic migration strategies (*active, reactive and smart strategy*), that only need current system information. Our evaluation results show that the smart strategy, which updates the set of videos in the cloud once a day, is sufficient. It is efficient and cost-saving, while the active and reactive strategies, which update multiple times a day, can provide a better user experience at higher costs.

(3) We explore the traces from PPLive and Hulu to drive simulations for the hybrid cloud-assisted deployment. The results show that: (a) The hybrid cloud-assisted deployment can save around 30% bandwidth expense based on current the unit bandwidth price of cloud and that of the ISPs. It also can handle unpredicted flash crowd with very little cost by the cloud storage over-provisioning. (b) The chosen of the server bandwidth capacity play the most important role in the cost savings. (c) The cloud storage size and the cloud content update strategy play the key roles in user experience.

(4) We point out the low efficiency of conventional peer replication strategy in peer-assisted VoD systems, and propose a cloud-friendly replication strategy. Our PPLive trace-based simulations show that the number of migrated videos in cloud can be reduced by a factor of 4, with trivial reduction in peer upload utilization. This is especially beneficial when network bottleneck exits between cloud and self-owned servers.

The rest of the paper is organized as follows. In section 2, we propose a hybrid cloud-assisted VoD delivery architecture, analyze its cost composition and formulate the problem. Section 3 presents characteristics of large-scale VoD services, and shows potentials and challenges of the hybrid cloud-assisted VoD delivery architecture. In Section 4, we propose three heuristic migration strategies to solve these challenges. Section 5 presents evaluation, using real traces from two large-scale VoD systems Hulu and PPLive. Section 6 proposes and analyzes the cloud-friendly replication strategy. We present related work in Section 7 before concluding in Section 8.

2 Hybrid cloud-assisted VoD delivery model

In this section, we describe the architecture, the cost composition, and the problem formulation of the hybrid cloud-assisted VoD systems, based on architecture and pricing of Amazon AWS [1, 2] and Microsoft Azure [3, 4].

2.1 System architecture

As shown in Fig. 1, there are four components in a hybrid cloud-assisted VoD delivery system: clients, self-owned servers (also called servers in short), cloud storage and cloud CDN. The self-owned servers, which are owned by the VoD providers, store all the original video files, serve part of the user requests and upload videos to the cloud

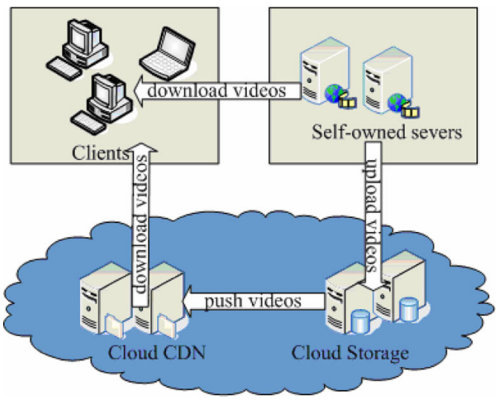

Fig. 1 System architecture

storage. The cloud storage stores a part of video files and pushes them to the cloud CDN in order to get a better user experience. The cloud CDN delivers streaming content using a global network of edge locations. The requests for your objects are automatically routed to the nearest edge location, thus the contents are delivered with the best possible performance. Both the cloud storage and cloud CDN are owned by the cloud providers.

Based on this architecture, there are four kinds of traffic: The clients download the videos from both the cloud CDN and the self-owned servers. The self-owned servers upload videos to the cloud storage. The cloud storage pushes these videos to the cloud CDN.

2.2 Cost composition

The cost under the cloud-assisted VoD delivery mode can be divided into four parts (note that the cloud providers do not charge video providers for the data transfer between the cloud storage and the cloud CDN):

(1) Self-owned server bandwidth cost: it includes the bandwidth consumed to deliver videos to the clients and to upload the videos to the cloud.
(2) Out-cloud bandwidth cost: the bandwidth consumed to deliver videos from the cloud CDN to the clients.
(3) Into-cloud bandwidth cost: the bandwidth cost charged by cloud providers for the video uploading.
(4) Cloud storage cost: the cost for disk space that storage videos in the cloud.

2.3 Problem formulation

Now we formulate the cost and the unmet user requests. To make the problem easy to discuss but without losing essence of this problem, we quantize time into discrete time slots, which may be a few minutes to several hours (e.g., one hour in our experiment). Table 1 gives all the notations of our formation. Equation 1 gives the total cost of the cloud-assisted VoD system during the time T*L. The total cost have four parts including $\sum_{t=1...L}\sum_{v_i\in M(t)} P_1 Z_i$ as the into-cloud data cost, $\sum_{t=1...L}\sum_{v_i\in M(t)} P_2(D(t)-U(t))T$ as the out-cloud data cost, $P_3 C_{server} TL$ as the self-owned server cost, and $P_4 S_{cloud} TL$ as the cloud storage cost. Equation 2 gives the unmet user requests during the time T*L. There will be unmet user requests, if the self-owned servers and the cloud can not provide enough bandwidth capacity for the average system bandwidth demand. Equation 3 gives the constraint for the server bandwidth used for the user requests – it must be less than both the total system bandwidth demand and the total self-owned servers bandwidth capacity.

Table 1 Notations of system parameters

Notation	Definition
T	Time slot size
L	Experiment length in terms of time slots
v_i	Video i
$M(t)$	Set of migrated videos during t^{th} time slot
$D_i(t)$	Average user demand for video i during time slot t
$D(t)$	Average system bandwidth requests during time slot t
Z_i	Size of video i
C_{server}	Self-owned servers bandwidth capacity
$B(t)$	Bandwidth bottleneck between cloud and self-owned servers
S_{cloud}	Cloud storage size
$U(t)$	Average server bandwidth for user requests during time slot t
$S(t)$	Set of videos in the cloud during time slot t
$P_i, i = 1, 2, 3, 4$	The unit price of into-cloud data transfer, out-cloud data transfer, server bandwidth, and cloud storage

Total cost (TC) is defined as Eq. 1:

$$TC = \sum_{t=1...L}\left(\sum_{v_i\in M(t)} P_1 Z_i + P_2(D(t)-U(t))T\right) + P_3 C_{server} TL + P_4 S_{cloud} TL \quad (1)$$

Unmet user requests (UUR) is defined as Eq. 2:

$$UUR = \sum_{t=1,...,L} Max\left(0, D(t) - \left(\sum_{v_i\in S(t)} D_i(t) + U(t)\right)\right) \quad (2)$$

Constraints:

$$U(t) \le Min(D(t), C_{server}), t = 1,...,L \quad (3)$$

The target of a migration strategy is to minimize the total cost while making the unmet user requests zero. Actually, we use normalized cost and normalized unmet user requests as the performance metrics in the Section 5. The normalized cost is defined as the ratio of the total cost under cloud-assisted VoD systems (shown in Eq. 1) divided by the total cost under Clients/Server-based VoD systems. The normalized unmet user requests is defined as the ratio of total unmet user requests under cloud-assisted VoD systems (shown in Eq. 2) divided by total user requests. The measurements of VoD services in the next section will show the potential and challenges in gaining our target. Since

our measurements show the difficulty to predict the system information in the future, we can not expect an optimal solution. Instead, in Section 4, we propose three heuristic update strategies, which only use current information, such as the video popularity. Although simple, they can achieve near-optimal results as is shown in Section 5.

3 Characteristics of large-scale VoD services, potentials and design challenges

In this section, we report the characteristics of large-scale VoD services, which shed insight on an eventual hybrid cloud-assisted deployment for VoD. Then based on these observations, we discuss potentials of hybrid mitigation of VoD services to content cloud and its design challenges. In Sections 4 and 5, we will use this trace data to explore the design and potential benefits of the hybrid cloud-assisted deployment separately.

3.1 Trace collections

We collect the data traces from a leading VoD provider in America, Hulu, and a leading VoD provider in China, PPLive. They are two large-scale VoD applications, which mainly provide movies and TVs. To explore the geo-distributed video content dissemination, we further investigate a crowdsourced video system Twitch.tv, which is the world's leading video platform and community for gamers.[1]

The Hulu trace, which was crawled from its website, contains the information of top-5000 most popular videos, including video name, popularity rank, video length (in terms of time), and category. Each page lists twenty videos, hence top-5000 videos are listed in 250 successive pages.[2] They are collected every hour over one month (starting from November 20^{th}, 2010). The PPLive trace, which was collected by PPLive's log servers, contains three parts: the simultaneous online users evolutions; the aggregate server bandwidth load; the video popularity distribution. The PPLive trace was collected every 5 minutes over 10 months. The Twitch.tv trace was crawled from its website from July 6 to July 12, 2014. It has 14 geo-distributed ingest servers, 1 from Asia area (AS for short), 6 from European area (EU for short), and 7 from United States area (US for short) to serve live broadcast for over 44 million visitors per month in a global scale.

[1] http://www.twitch.tv/

[2] http://www.hulu.com/popular?h=18&page=1&timeframe=today

3.2 User demand evolution and potentials

Based on our long-term measurements, we find the user demand generally exhibits similar daily patterns and similar peak values every day. However, to illustrate how the hybrid cloud-assisted deployment can handle the flash crowd well, Fig. 2 chooses two special consecutive days–November 22^{nd} and 23^{th}, 2010. One 24 set TV series were published on November 22^{nd}. First, we can see that the number of simultaneous users achieves its highest value at about 21:00 and the lowest point appears at about 7:00 with the highest value 5 times of the lowest one. Second, the peak user demand suddenly increases by nearly 25% in next day. Even if service operators can predict the size of this flash crowd correctly, the provision is very costly for them in the self-owned servers. Later in Section 5, we can see cloud-assisted architecture can handle flash crowd very easily and economically.

Typically today the ISPs bills a customer (such as a VoD provider) for bandwidth usage using the 95 percentile rule. Instead, cloud providers charge data transfer with pay-as-you-go mode. For example, Amazon CloudFront charges \$0.15 /GB in United States and \$0.201 / GB in Japan for first 10 TB/month. It only charges \$0.03 /GB in United States and \$0.075 / GB in Japan over 1000 TB/month.

From above measurements, we find that the server utilization is very low. For example, in Fig. 2 we can see that at most time the user demand is lower than 50% on average with only three hours for peak requests in one day. Thus, there is an immense possibility that the VoD providers can get benefits if they buy less server bandwidth from an ISP and let additional user requests be served by the cloud. Table 2 shows the potential bandwidth cost savings under the cloud-assisted VoD delivery mode. The normalized cloud price is defined as the ratio of unit cloud price

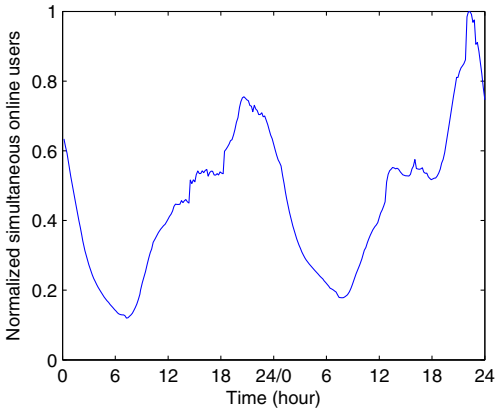

Fig. 2 Simultaneous online users

Table 2 Potential bandwidth cost savings under cloud-assisted VoD delivery mode

Normalized cloud price	1	2	3	4	5	≥6
Normalized bandwidth cost	0.48	0.67	0.75	0.79	0.84	>0.9

divided by the ISP's price. The normalized bandwidth cost is defined as the ratio of server bandwidth cost under cloud-assisted VoD delivery mode divided by that under traditional Clients/Server mode. The unit bandwidth prices of the ISPs [6] and the clouds vary from different providers and we find the normalized unit cloud price is generally from 1 to 10. We only focus on the case where normalized unite cloud price is between 1 and 5, since the benefits might be trivial if the potential bandwidth savings are less than 10%. The unit price of content cloud is expected to be lower with its technology advance.

3.3 Video popularity distribution

Since a large volume of user requests are directed to a cloud in our hybrid solution, it should be guaranteed that the videos in the cloud can attract no less requests than what should be severed by cloud. In order to reduce cloud storage cost, it is always a good idea to store the most popular videos in the cloud. Then, some natural questions raised are: How many videos should we upload to the cloud? How much cloud storage do we need to store those videos? In order to solve those questions, we need to know the video popularity distribution. Figure 3 plots the CDF (Cumulative Distribution Function) 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 10% popular videos attract nearly 50% views and the top 20% popular videos attract nearly 70% views. This result infers that we can employ our hybrid solution with a limited cloud storage cost.

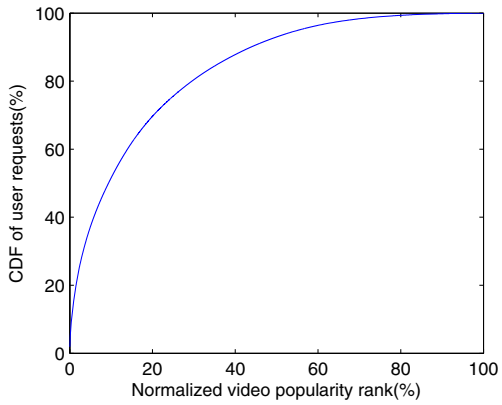

Fig. 3 Video popularity distribution

3.4 Video popularity evolution and design challenges

Although the VoD providers can get benefits from the bandwidth cost reduction for the user requests, they also have to pay additional expense for the data transfer between the cloud and the self-owned servers. Thus, if the set of the most popular videos changes too frequently, it will cost the VoD providers a lot of money to upload current most popular videos to the cloud. In this section, we measure the video popularity evolution of the top-5000 most popular videos provided by Hulu. We investigate how quickly the popularity ranks and the aggregate popularity of top-k most popular videos change over time. Our data analyses show that the video popularity changed very frequently, which means a lot of videos in the cloud should be replaced.

Figure 4 shows the popularity decay of the most popular videos in the cloud under different cloud storage sizes. We assume that the cloud stores a certain number of the most popular videos at the start time, and never updates those videos. We consider different sample times, and gives the average value in Fig. 4. Figure 5 shows the corresponding average, maximum value and minimum value to that in Fig. 4, when the cloud storage size is 1000 files. From Figs. 4 and 5, we can find that: (1) The popularity evolution shows daily pattern. (2) The popularity of the most popular videos decays by 20% after the first day and decays another 20% after anther nine days. (3) Similar popularity decay patterns are shown under different cloud storage sizes. (4) The little difference among average, maximum and minimum decay curves show that popularity decay patterns are not related to the start time.

Since the popularity of the most popular videos demonstrates an obvious decay within the first day, we further examine the popularity variation in a one-day period. We match every current top-k video with all top-k videos an hour(or two hours...) later, and define the total number of

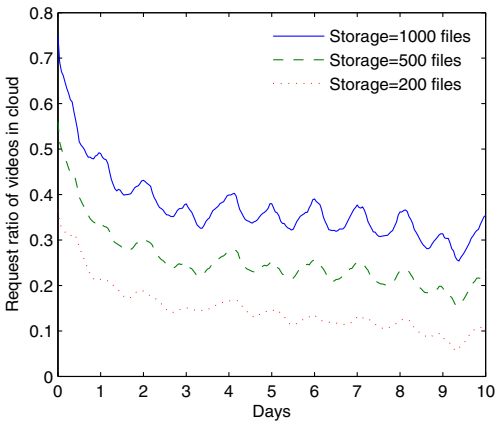

Fig. 4 Popularity decay of videos in the cloud under different cloud storage sizes

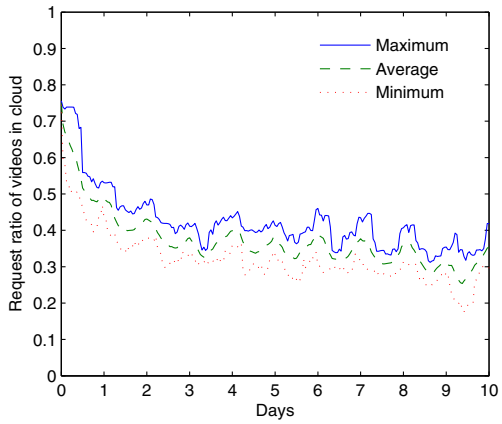

Fig. 5 Popularity decay of videos in the cloud when cloud storage size is 1000 files

unmatched videos as the number of videos leaving top-k. Figure 6 shows the average number of videos leaving top-k after an hour, three hours, six hours, twelve hours and one day. We find that: (1) On average, 11% of videos will leave the top-k most popular videos after an hour, 22% after three hours, 30% after six hours, 35% after twelve hours, and interestingly back to 28% after one day. (2) The update cost will significantly increase, if the cloud updates the top-k videos in each hour instead of every 24 hours. (3) The number of videos that leave top-k is nearly linear to the value of k. (4) It is very interesting that fewer videos leave the top-k list after one day than after six hours and twelve hours. One possible explanation is that people tend to focus on the same kind of videos during the same time next day.

Figure 7 plots the average percentage of the videos that leave top-k after different three-hours slots. We find the values show significant differences for different three-hours slots. The sharp rank changes happen during the office hours (9am-5pm) and midnight (0am-3am). One possible explanation for this may be that Hulu generally publishes new videos during office hours and the beginning of a day.

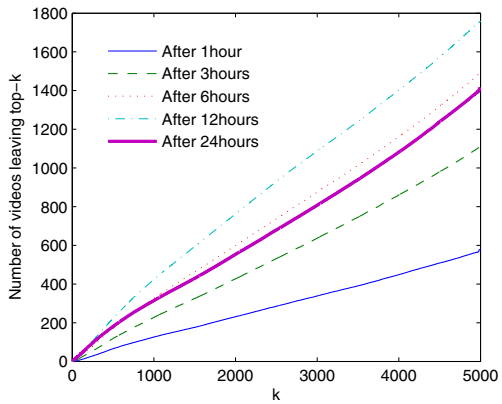

Fig. 6 Average number of videos leaving the list of top-k most popular videos after 1, 3, 6, 12, and 24 hours

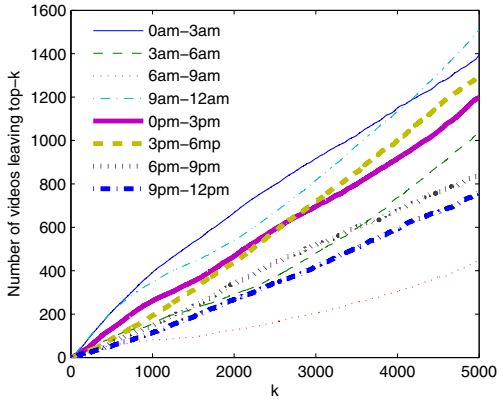

Fig. 7 Number of videos leaving top-k after different each three-hours period

Figure 8 plots the average percentage of the videos that leave top-k after different one-day periods. It is interesting that the values show significant differences under different one-day periods. Particularly, statistically the fewest videos leave top-k list from 0am to 0am next day. Thus we can set 0am to 0am next day as the daily video update cycle, in order to reduce the video update expense.

Figure 9 plots the percentage of the videos that leave the top-5000 list after one day and three hours. Since we take many samples, it plots the average, maximum and minimum values. The horizontal axis is the start time. For example, the vertical value at horizontal axis 0 means that on average, 13% videos leave the top-5000 list from 0am to 3am, and 30% from 0am to 0am next day. We can witness an obvious fluctuation of changes whin one day and a bigger fluctuation of changes whin three hours. This fact means that it is very difficult to predict the popularity change in the future based on the previous statistic information.

To explore the geo-distributed video streaming in Twitch.tv, we divide the locations as AS, EU, and US, and record the percentage of source streams from each region in Fig. 10 and corresponding viewer population in Fig. 11

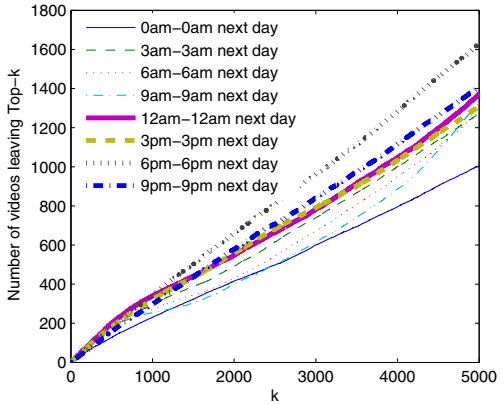

Fig. 8 Number of videos leaving top-k after different each 24-hours period

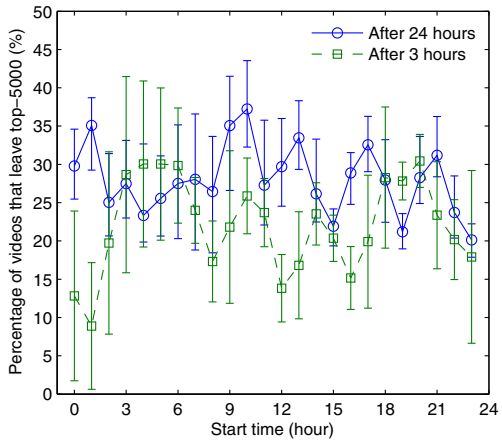

Fig. 9 Percentage of videos that leave top-5000 in one day and three hours

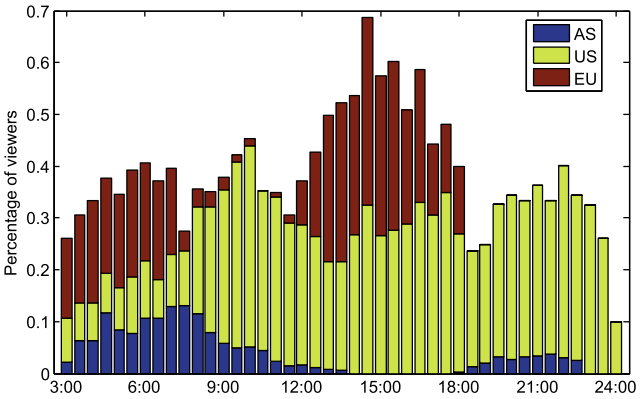

Fig. 11 Viewer demand for the distributed source streams in one day

for every 30 minutes between 3:00 AM to 24:00 PM. In Fig. 10, it can be easily observed that most of the streams from Asia and Europe are during the morning and afternoon, and the number of streams from the United States keeps growing when night falls. In Fig. 11, we can see that in the early morning between 3:00 AM and 7:00 AM, most of the popular streams come from Europe or Asia. We conjecture that it is because the local times in Europe or Asia are in afternoon or evening, and there are more online sourcers from these regions during that time. Meanwhile, the viewer demand from these areas can also be more active during this period. And most of the viewers may prefer the streams with native language speaking sourcers. Similar reasons can also explain the increase of viewer demand for the source streams from the United States after 15:00 PM.

4 Migration strategies

A migration strategy can be divided into three parts: (1) choose a server bandwidth capacity; (2) choose a cloud storage size; (3) choose a cloud content update strategy. In this section, we design three cloud content update strategies and discuss the impact of the server capacity and cloud storage size.

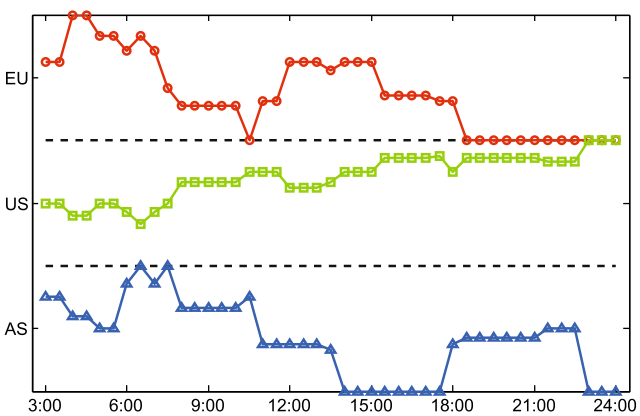

Fig. 10 Source stream distribution in one day

4.1 Cloud resource provisioning

The selection of server capacity is related to unit price of cloud data transfer. Generally higher unit price of cloud data transfer is, more server bandwidth capacity should be provisioned. The selection of cloud storage size should be related to how many user requests will be migrated to the cloud. Generally, more request results in larger cloud storage provisioning. In the next section, we will explore how these two factors affect unmet user requests and total cost.

4.2 Cloud content replication strategies

Since our measurements show that it is difficult to predict the system information in the future, we will propose three heuristic update strategies, which only use current information, such as the video popularity. In the next section, we will find these simple strategies can achieve near-optimal results.

4.2.1 Active strategy

According to Fig. 2, there is much idle server bandwidth in the morning, which provides an opportunity to reduce the cloud content update cost. We can upload more videos in the morning and thus fewer in the evening. This can utilize the free server bandwidth in the morning and hopefully reduce uploading load in the peak time. But it may increase the unnecessary uploading, because video popularity changes so quickly that some videos uploaded in the morning might not be popular any more in the evening. Based on this idea, we design a strategy called active strategy, which works as follows: It uploads current most popular videos to the

cloud and replaces most unpopular videos in the cloud. U_i is equal to total user bandwidth demand when total user demand is smaller than the total server bandwidth C_{server}. But when the total user demand is more than C_{server}, the servers must reserve enough bandwidth to update the most popular videos.

4.2.2 Reactive strategy

To reduce unnecessary uploading, conversely we can upload videos only if the videos in the cloud can not attract enough requests. But this method may introduce very large uploading server bandwidth demand in the peak time. Based on this idea, we design a strategy called reactive strategy, which works as follows: It uploads videos only if when total user demand is bigger than the total server bandwidth C_{server}. U_i is equal to total user bandwidth demand when the total user demand is smaller than C_{server}. But when the total user demand is bigger than C_{server}, self-owned servers must reserve enough bandwidth to update most popular videos to the cloud.

4.2.3 Smart strategy

Exploring the advantages of both ideas, we propose our last strategy called smart strategy, which works as follows: It uploads videos only once in one-day period when there is idle server upload capacity. It replaces the videos so that all videos in the cloud are most popular at that moment.

4.3 Discussion of implementation issues

To deal with the dynamic demands in a large scale, initially we have the geo-distributed self-owned servers. Each self-owned server can monitor its local viewer demands, which can be divided into two parts. Some traffic flow can be served by the dedicated server itself, and the rest can be redirected toward the provisioned cloud servers or CDN. To migrate the traffic flow toward cloud, we will implement the following two steps:

1. Cloud provisioning strategy: We need to provision two types of resources from cloud platform, i.e., bandwidth support from cloud servers or CDNs, and cloud storage for video content. As the traffic load of self-owned server can be released through bandwidth provisioning, the self-owned server can determine whether to provision a new cloud server or CDN according to the traffic load served by itself. As the bandwidth provisioning is dependent to content replication in cloud storage, the self-owned server can determine whether to provision more cloud storage according to the redirected traffic load toward cloud servers or CDNs. In other words, the cloud provisioning strategy can be implemented by self-owned servers through traffic monitoring.
2. Content replication strategy: After the cloud provisioning is completed, the self-owned server can upload the hottest videos to cloud storage. The replication strategy can select any one of the proposed strategy (i.e., Active strategy, Reactive strategy, and Smart strategy). Specifically, we can consider a cooperative solution among multiple self-owned servers in distributed regions. When the traffic load is heavy in a region, the content replication can be completed by an idle self-owned server in another region.

In addition, we can further consider a hierarchical structure to organize the distributed self-owned servers, cloud servers, and CDNs, rather than a flat structure. We can utilize a tree-based method to optimize the server organization, such as allocating the self-owned servers with higher upload capacities closer to the cloud platform, so that the depth of the subtree rooted can be minimized. The heterogenous structure is efficient to avoid the popularity churn and further reduce the total cost through the cooperation of servers in distributed regions. The drawback is that it may lead to cross-boundary traffic.

5 Trace-driven evaluations

In this section, we use the traces of Hulu and PPLive to gain critical insights of the hybrid cloud-assisted deployment. Generally the VoD services have seasonal or other periodic demand variation. But they also face some unexpected demand bursts. We evaluate our migration strategies in both cases. The performance metrics are the normalized cost and normalized unmet user requests, which are defined in Section 2. We use the trace data shown in Section 3 as experiment parameters, such as the video popularity distribution and evolution. We set 5000 files as the system scale.

5.1 Steady-state scenario

In this subsection, we study the performance of our three migration strategies in steady-state scenario. We define the steady-state scenario as where user demand shows predicable periodic demand variation. In the steady-state scenario, we can smartly provision server bandwidth capacity and cloud storage size based on previous user demand information.

Figure 12 shows the normalized cost under different storage sizes and update strategies. The lower bound cost is defined as the cost that excludes the cloud content update cost. We find the performance curves of all three update

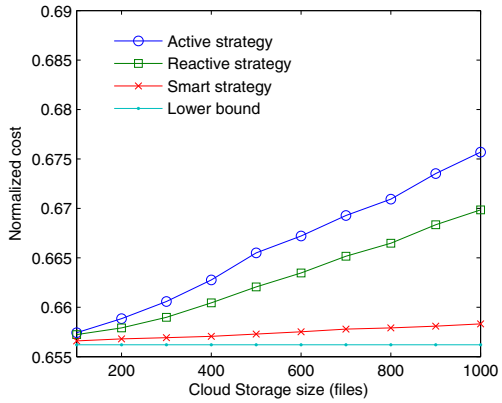

Fig. 12 Normalized cost under different storage sizes and update strategies

strategies are not far from the lower bound curve. Specifically, the cost under the smart strategy is very close to the lower bound value and almost doesn't increase with the cloud storage size. We configure P1=P2=2*P3 in both Figs. 12 and 13.

Figure 13 shows the normalized unmet user requests under different storage sizes and update strategies. The only difference between active and reactive strategies is whether videos in the cloud should be updated when the servers have free bandwidth. Since it does not make different unmet user requests during this period, active and reactive update strategies give exactly the same results. Compared with these two strategies, the smart strategy gives worse results. The performance however becomes much better with a larger cloud storage. From Fig. 12, we know that the aggregate cost increases very little with the increase of cloud storage under the smart strategy. Therefore, the smart update strategy can be a better strategy weighting the trade-off of the cost and user experience.

Figure 14 shows the normalized cost under different server bandwidth capacities and unit bandwidth prices. Since the smart update strategy can be a better strategy

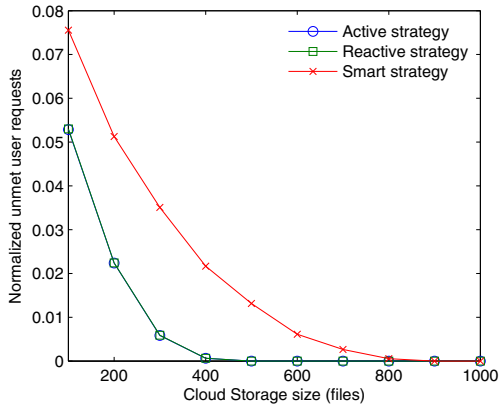

Fig. 13 Normalized unmet user requests under different storage sizes and update strategies

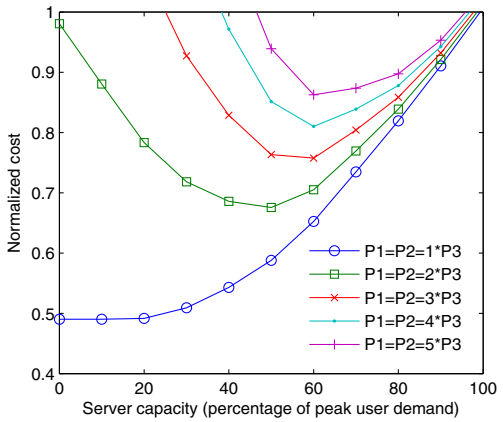

Fig. 14 Normalized cost under different server capacities and unit prices

weighting the trade-off of cost and user experience, we simply configure the *smart strategy* as the update strategy. We set cloud storage size as 1000 files. These settings are also for Fig. 15. We find both server capacity and unit cloud price play the significant roles in cost savings. We also find both a very high or very low of server bandwidth will lead to bad results. Generally the proper server bandwidth is from 40% to 60% of the peak user demand. Hence, we set 50% for experiments of Figs. 12 and 13.

Figure 15 shows the normalized unmet user requests under different server capacities. The unmet user requests will reduce quickly with the increase of the server bandwidth capacity. In this experiment, the unmet user requests become zero when the server bandwidth capacity is more than 30% of the peak user demand.

5.2 Flash crowd scenario

We define the flash crowd scenario as where the daily user demand pattern changes suddenly and the peak value becomes much higher than previous days. In this scenario, the decision is also made based on the previous information.

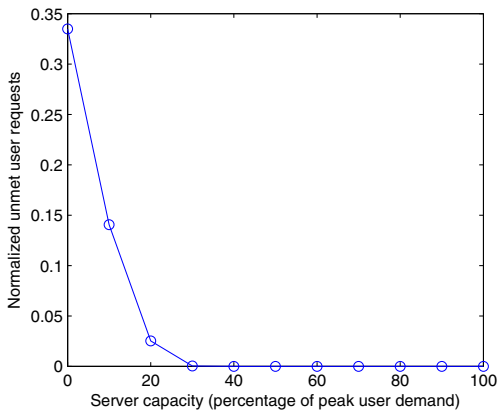

Fig. 15 Normalized unmet requests under different server capacities

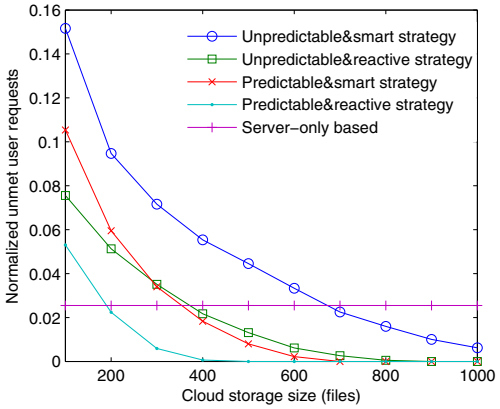

Fig. 16 Normalized unmet user requests under different strategies

We configure P1=P2=2*P3. Here we only use the reactive and smart strategy, since the active strategy shows no advantages against them.

Figure 16 shows the normalized unmet user requests under different content update strategies and cloud storage sizes. The "predictable strategy" refers to the strategy that can correctly predict flash crowds and chooses an optimal server capacity based on the correct user demand. Conversely, the "unpredictable strategy" refers to the strategy that can not predict flash crowds and chooses an non-optimal server capacity based on the previous user demand. We find the performance decreases by around 2% if we do not predict the flash crowd. We also find there is 0.25% unmet user requests under the Clients/Server-based VoD development. The unmet user requests under the reactive update strategy are reduced to zero when the cloud size is more than 900 files. So, the hybrid cloud-assisted VoD development can handle flash crowd easily by setting a bigger cloud storage even if the sudden increased user demand are not correctly predicted.

Figure 17 shows the normalized cost under different cloud content update strategies and cloud storage sizes. It shows that the over-provision of the cloud storage and the wrong forecast of the user demand do not add too much additional cost. In sum, the hybrid cloud-assisted deployment can handle the flash crowd very well with very little cost. For example, we use the reactive update strategy, and set storage size to be 1000 files, The aggregate cost is reduced by more than 32%.

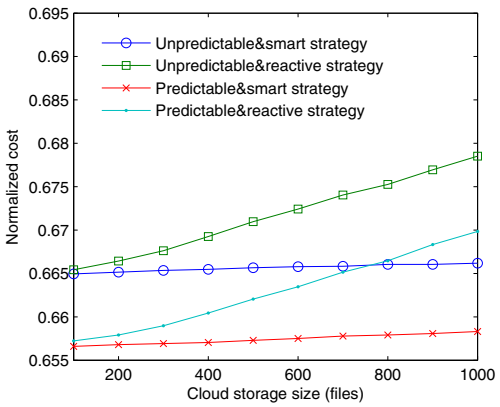

Fig. 17 Normalized unmet requests under different server capacities

5.3 A brief summary

Based on above analyses, we can achieve the flowing findings: (1) The hybrid cloud-assisted VoD deployment can save up to 30% bandwidth expense when the unit price of cloud is twice as that of the ISPs. It can also handle the flash crowd with less than 2% cost by cloud storage over-provisioning. (2) The unit cloud price and server bandwidth chosen capacity play the most important roles in cost savings. (3) The cloud storage size and the cloud content update strategy play the key roles in user experience.

6 Cloud-friendly edge peer replication strategy

Even though the modern cloud platform can provide an elastic and flexible services for the large scale VoD streaming systems, yet the heterogenous demands have presented heavy burden on current cloud based infrastructure. Fog computing has been suggested since 2014, which aims to process certain workloads and services locally on edge peers (fog), rather than being transmitted to the cloud. On the other hand, considering the limited peers' uplink capacity and the popularity of high definition movies, the system still need to provide huge server bandwidth. For example, PPLive needs at least 10Gbps server bandwidth to support its peer-assisted VoD service. Similar to Client/Server-based VoD systems, the server bandwidth utilization is less than 40%, with much idle bandwidth in the morning. Therefore, it is beneficial to migrate partial videos of such systems to content clouds, and further deploy edge peers in a fog computing environment. We point out the low efficiency of current peer replication strategy in peer-assisted VoD systems, and then propose a cloud-friendly edge peer replication strategy. Finally we evaluate our strategy using PPLive trace data and find the number of migrated videos in cloud can be reduced by a factor of 4, with trivial reduction in peer upload utilization.

6.1 Motivation

Current peer-assisted system usually adopts the Multiple Video Caching (MVC) (e.g., PPLive) mechanism which means peer can store and redistribute a video which was previously viewed but is not currently played. 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, and entire viewer population thus forms a distributed peer-assisted storage (or file) sharing system. After peer's local storage is filled with the viewed video replicas, if a new video is requested, disk replacement happens. One viewed video replica is selected to be replaced by the new watching video. How to regulate this storage system is undoubtedly the most critical part of the peer-assisted VoD system, because proper replica distribution among peers shared disks is the precondition to discover and transmit the desired contends efficiently with each other.

However, previous peer-assisted VoD systems is aiming to reduce server bandwidth cost as much as possible while maintaining user experiences [27, 33]. They do not consider the video migration cost from the self-owned servers to the cloud. Now we use a simple example to illustrate how a better replication strategy can reduce video replicas in cloud while not raising total server bandwidth cost.

As is shown in Fig. 18, all replicas of video A, B, and C are stored in three peers (Peer 1, 2 and 3). There is 1 current viewing user for each video. There are two replication strategy, a proportional replication and a skewness replication, namely. The proportional replication means the replicas number of a video is proportional to its viewer number. The skewness replication means all replications expect the proportional replication and Fig. 18 gives an example. We assume a simple schedule strategy: a viewer will request video to all peers who has its replica, and an uploader will upload data to all its requests uniformly. We also assume that all peer has a uniform upload capacity. Under these assumptions, the additional server bandwidth cost of each viewer will be $S_i = max(0, 1 - \rho * R_i / \sum R_j)$.

If $\rho = 2/3$ and taking the skewness replication, then $S_a = 0$, $S_b = 1/3$, $S_c = 2/3$, we only need to store $Video_b$ and $Video_c$ in cloud. While if $\rho = 2/3$ and taking the proportional replication, then $S_a = 1/3$, $S_b = 1/3$, $S_c = 1/3$, we need to store all three videos in the cloud. Both strategies produce same server bandwidth cost:1, but the skewness replication reduces the cloud storage cost and thus migration cost. Note that the skewness replication strategy does not always lead to better performance than proportional replication strategy, particularly if $\rho \geq 1$, although the peer upload capacity is generally less than streaming rate. For example, we assume $\rho = 1$. If we take the skewness replication, then $S_a = 0$, $S_b = 0$, $S_c = 1/2$, and the server bandwidth cost is 1/2. While if we take the proportional replication, then $S_a = 0$, $S_b = 0$, $S_c = 0$, and it do not produce server bandwidth cost.

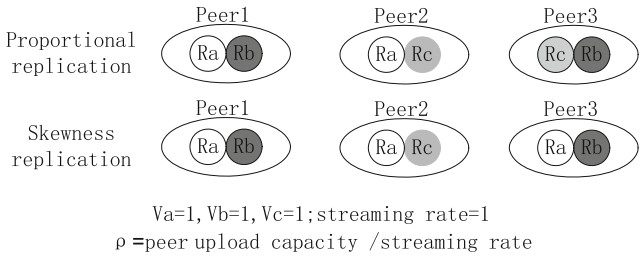

Fig. 18 Illustration of proportional and skewness replication distribution

6.2 Cloud-Friendly replication strategy

From above example, we know a skewness replication distribution probably leads to good performance in terms of least number of videos stored in the cloud. Now we will get some general conclusion. Before that, we first define a cloud-friendly replication strategy called chunked proportional replication distribution. It does not replicate Top-k most popular videos, and all rest videos have proportional replicas to their requests. The parameter K depends on the amount of videos directed to the cloud (denoted as V_{cloud}). k is chosen as the least integer which makes $\sum_{i=1}^{k} V_i > V_{cloud}$.

We make similar assumptions as in the motivation example: a viewer will request data to all peers who has requested replicas; a unloader will upload data to all its requests uniformly; all peer has a uniform upload capacity. Under these assumptions, we can get the following conclusion:

Theorem 1 *If $\rho \geq 1$, proportional replication distribution can get optimal solution. If $\rho \leq 1$, chunked proportional replication distribution can get optimal solution.*

Proof If $\rho \geq 1$, $S_i = 0$, for each *i=1,..,M*. All requests are served by other peers. Thus, neither the server bandwidth nor the migration cost is needed. If $\rho \leq 1$, this k fulfils $\sum_{i=k}^{M} V_i \geq \rho \times N$. Thus all peer's upload capacity can be fully utilized, thus the request server bandwidth (either from self-owned servers or cloud) is least. Meanwhile, since the top-k most popular videos are purely severed by servers, the least videos are needed to be uploaded to the cloud given the amount of migrated traffic that the cloud should serve. Therefore, Theorem 1 is proved. □

One concern is how to achieve the chunked proportional replication distribution by a replacement algorithm. For the proportional part, Wu et al. [27] showed with extensive simulations that, the performance margin enjoyed by optimal strategies over the simplest algorithms (e.g., LRU) is not substantial, when it comes to reducing server bandwidth costs in peer-assisted VoD systems. For the chunked part, we can neither replace all replicas of top-k most popular videos nor produce them in a minute. Thus we can use above algorithms to regulate replication distribution but disable the

replicas of the top-k most popular videos, which should be in the cloud. For example, a peer can do this by not telling other peers that it has such video replicas. Thus, these replicas can not be found and requested until these videos are moved out of the cloud. In the next subsection, we will examine the performance of this simple implementation.

6.3 Performance evaluation

In this subsection, we compare the performance of the proportional replication strategy and our proposed cloud-friendly strategy based on the PPLive video popularity distribution data, which is shown in Fig. 3.

Figure 19 compares the least number of migrated videos in the cloud under different replication strategies. The least number of migrated videos is defined as the least integer k which makes $\sum_{i=1}^{k} V_i > V_{cloud}$. We find that it needs to migrate much fewer videos to the cloud under the cloud-friendly replication strategy. Recall that the system gains the most bandwidth cost saving generally when migrating less than 50% of total traffic. Thus we particularly zoom up this region, and find that only less than 10% of videos should be migrated in this region. When $\rho = 1/2$ and the migrated traffic is 50%, the number of migrated videos can be reduced by a factor of more than 4.

The peer upload capacity may be not fully utilized because the replicas of some videos are disabled by the chunked proportional replication strategy. For example, if all replicas of a peer stores are videos in the cloud. It seems that this peer stores nothing from other peers' perspective, because these replicas are disabled by the system. Figure 20 shows the increased server bandwidth cost by the cloud-friendly replication strategy. We configure each peer stores four videos, which is a generical case in PPLive. Since Fig. 17 shows less than 10% of videos should be migrated to gain most server bandwidth cost saving, we zoom up this region. We find the increased server bandwidth cost is less than 0.25% in this region.

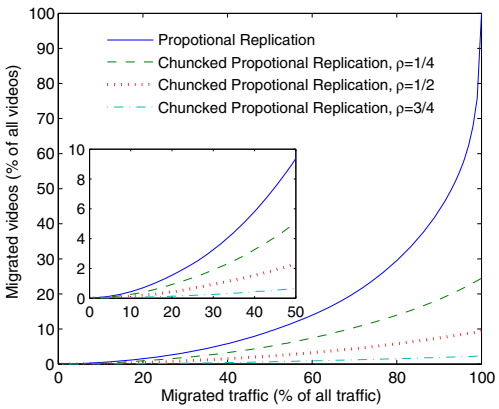

Fig. 19 Least migrated videos in cloud

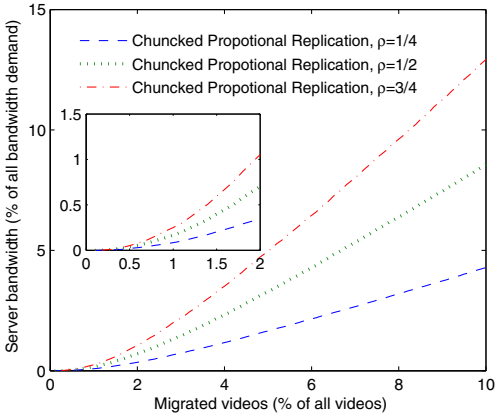

Fig. 20 Increased server bandwidth cost

In sum, our PPLive trace-based simulations show that the number of migrated videos in cloud can be reduced by a factor of 4, with trivial reduction in peer upload utilization. This is especially beneficial when network bottleneck exits between cloud and self-owned servers. Although a lower peer upload utilization increases the server bandwidth cost, it reduces peer upload load on the other hand.

Note that our simple model does not consider the peer churn and the heterogeneous peer upload capacity. Based on a more complicated model, a recent research showed that the proportional replication strategy is not optimal in reducing server bandwidth cost [27]. Based on the real-world development and some trace data, our previous work also showed that the proportional replication strategy and the LRU replacement algorithm does not give an optimal result, although very close to it. A more recent research [33] proposed and analyzed a generic replication algorithm RLB which balances the service to all movies, for both deterministic and random demand models, and both homogeneous and heterogeneous peers. Nevertheless, we can simply replace the proportional replication part of truncated proportional replication strategy by these better replication strategy.

7 Related work

As a novel computing paradigm, cloud services provide flexible resource allocation on demand with the promise of realizing elastic, Internet-accessible computing on a pay-as-you-go basis [13]. We have seen many new generation of cloud-based services that emerged in recent years, which are rapidly changing the operation and business models in the market. A prominent example is Netflix, a major on-demand Internet video provider. Netflix migrated its entire infrastructure to the powerful Amazon AWS cloud in 2012, using EC2 for transcoding master video copies to 50 different versions for heterogenous end users and S3 for content storage [2].

Leveraging the elastic and flexible resource provisioning, many researchers pay attention to develop cloud assisted video streaming systems with the Quality-of-Service (QoS) guarantee to support various video streaming applications. For the social aware video applications, the online social network interaction among users can facilitate to build stable relationship in cloud environment. Wang et al. [25] proposed a cloud-assisted adaptive video streaming with social-aware video prefetching to avoid intermittent disruptions and long buffering delays. Hu et al. [15] presented a social video replication and user request dispatching mechanism in the cloud content delivery network architecture to reduce the system operational cost. Nan et al. [21] developed an efficient multimedia distribution approach taking advantage of live-streaming social networks to deliver the media services from the cloud to both desktop and wireless end users in a large scale. For the video streaming over mobile devices, cloud computing can offer a natural solution to reduce the cost of deploying and operating mobile media networks [26]. Zakerinasab et al. [29] proposed an energy-efficient cloud-assisted streaming system for smartphones with a two-level scheme. Hu et al. [16] further proposed a public cloud assisted architecture to alleviate the traffic burden to the social service providers and further reduce the service latency of mobile users. For the encoding, decoding and transcoding of video streaming, the computation-intensive media processing tasks can be offloaded from the end devices to the cloud platform. In [10], a cloud-based real-time transcoding and transmission framework is presented to provide smooth video quality for mobile devices. In [32], Zhao et al. further explored a segment-based storage and transcoding trade-off strategy for multi-version VoD systems in the cloud. However, all above works only utilize the elastic resource provisioning to facilitate the video streaming service without considering the cooperation between the self-owned servers and cloud platform.

Generally, a geo-distributed cloud is ideal for supporting large scale media streaming applications by spanning multiple data centers at different geographical locations. There have been numerous studies on large scale video streaming in geo-distributed cloud environment. For example, Wang et al. [24] presented a generic framework that facilitates migrating live media streaming toward a cloud platform for a global service. Qiu et al. [23] investigated optimal migration of a content distribution service to a hybrid cloud consisting of private servers and public geo-distributed cloud services. Furthermore, cloud-based content delivery networks (Cloud CDN) cache and deliver contents from geo-distributed cloud data centers to end users. Zhang et al. [31] proposed an efficient online algorithm for dynamic content replication and request dispatching in cloud CDNs operating over a long time span, targeting overall cost minimization with performance guarantees. Lai et al. [19] developed a workload scheduling mechanism that aims at optimizing the tail latency while meeting the cost constraint given by application providers. In addition, as a new paradigm, fog computing refers to a platform for local computing, storage and distribution in edge devices rather than centralized data centers [22]. Some works are presented as cost-effective solutions with cooperation between the edge peers and remote cloud [11]. For example, Yan et al. [28] proposed a hybrid edge cloud and client adaptation framework for HTTP adaptive streaming to deal with inaccurate bandwidth estimation and unfair bitrate adaption under the highly dynamic cellular links. Even though all above strategies consider the cost minimization with performance guarantees, our work further analyzes the relationship between cloud provisioning cost and content replication cost under the time-varying video popularity. For the first time, we propose a cloud-assisted VoD architecture and formulate the online cloud migration problem in details. Using the traces from three large-scale video streaming systems, we extract many key characteristics of these systems which are relevant to the online implementation of cloud migration.

8 Conclusion

This paper considers the challenges, design and potential benefits of the hybrid cloud-assisted VoD deployment. We first develop a cloud-assisted VoD deployment model and formulate the cost. Then using a nine-month PPLive trace, a one-month Hulu trace, and a one-day Twitch.tv trace, we extract many key characteristics of large-scale VoD systems that are relevant to the hybrid cloud-assisted deployment and analyze exiting opportunities and challenges. Finally, we design three heuristic migration strategies and make extensive trace-driven performance evaluation. The simulation results show that our hybrid cloud-assisted deployment can save up to 30% bandwidth expense based on current data transfer price of content clouds and ISPs. For a large scale VoD systems, we propose a cloud-friendly peer replication strategy, which further reduces the migration cost by a factor of 4. Our simulation also shows that the cloud price and server bandwidth chosen play the most important roles in saving cost, while the cloud storage size and cloud content update strategy play the key roles in the user experience improvement.

Acknowledgments This research is supported by the following grants: “Multi-source heterogeneous big data management, analysis and mining for urban renewal” (National Natural Science Foundation of China, U1301253), “Application demonstration of big data for land resources management and service” (Science and Technology Planning Project of Guangdong Province, China, 2015B010110006), “Cooperative resource allocation optimization for software-defined

network based virtual CDN" (National Natural Science Foundation of China 61602214), Natural Science Foundation of Jiangsu Province in China (BK20160191), National Natural Science Foundation of China (61472212).

References

1. Amazon CloudFront. http://my.url.com/
2. Amazon S3. http://aws.amazon.com/s3
3. Azure CDN. http://www.microsoft.com/windowsazure/cdn/default.aspx
4. Azure Storage. http://www.microsoft.com/windowsazure/storage/default.aspx
5. Hulu. http://www.hulu.com/popular
6. ISP price compare. http://www.ispcompared.com/broadband.html
7. Joost. http://www.joost.com
8. PPLive. http://www.pplive.com
9. PPStream. http://www.ppstream.com
10. Baik E, Pande A, Zheng Z, Mohapatra P (2016) VSync: Cloud based video streaming service for mobile devices. In: Proceedings of IEEE INFOCOM
11. Bruin XM, Tordera EM, Tashakor G, Jukan A, Ren GJ (2016) Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems. IEEE Wirel Commun 23(5):120–128
12. Chen F, Zhang C, Wang F, Liu J, Wang X, Liu Y (2015) Cloud-assisted live streaming for crowdsourced multimedia content. IEEE Trans Multimedia 17(9):1471–1483
13. Hajjat M, Sun X, Sung E, Maltz D, Rao S, Sripanidkulchai K, Tawarmalani M (2010) Cloudward Bound: Planning for Beneficial Migration of Enterprise Applications to the Cloud. In: Proceedings of ACM SIGCOMM
14. Hou X, Li Y, Chen M, Wu D, Jin D, Chen S (2016) Vehicular fog computing: a viewpoint of vehicles as the infrastructures. IEEE Trans Veh Technol 65(6):3860–3873
15. Hu H, Wen Y, Chua TS, Huang J, Zhu W, Li X (2016) Joint content replication and request routing for social video distribution over cloud cdn: a community clustering method. IEEE Trans Circuits Syst Video Technol 26(7):1320–1333
16. Hu H, Wen Y, Niyato D (2017) Public Cloud Storage Assisted Mobile Social Video Sharing: A Supermodular Game Approach. IEEE Journal on Selected Areas in Communications. doi:10.1109/JSAC.2017.2659478
17. Huang Y, Fu TZ, Chiu D-M, Lui JC, Huang C (2008) Challenges, Design and Analysis of a Large-scale P2P-VoD Syste. In: Proceedings of ACM SIGCOMM
18. Jalali F, Hinton K, Ayre R, Alpcan T, Tucker RS (2016) Fog computing may help to save energy in cloud computing. IEEE J Sel Areas Commun 34(5):1728–1739
19. Lai Z, Cui Y, Li M, Li Z, Dai N, Chen Y (2016) TailCutter: Wisely Cutting Tail Latency in Cloud CDN under Cost Constraints. In: Proceedings of IEEE INFOCOM
20. Li B, Wang Z, Liu J, Zhu W (2013) Two decades of internet video streaming A retrospective view. ACM Trans Multimed Comput Commun Appl 9(1):2284–2294
21. Nan G, Mao Z, Yu M, Li M, Wang H, Zhang Y (2014) stackelberg game for bandwidth allocation in Cloud-Based wireless Live-Streaming social networks. IEEE Syst J 8(1):256 –267
22. Peng M, Yan S, Zhang K, Wang C (2016) Fog-computing-based radio access networks: issues and challenges. IEEE Netw 30(4):46–53
23. Qiu X, Li H, Wu C, Li Z, Lau FCM (2015) Cost-minimizing dynamic migration of content distribution services into hybrid clouds. IEEE Trans Parallel Distrib Syst 26(12):3330–3345
24. Wang F, Liu J, Chen M, Wang H (2016) Migration towards cloud-assisted live media streaming. IEEE/ACM Trans Networking 24(1):272–282
25. Wang X, Chen M, Kwon TT, Yang L, Leung VCM (2013) AMES-cloud A Framework of Adaptive Mobile Video Streaming and Efficient Social Video Sharing in the Clouds. IEEE Trans Multimedia 15(4):811–820
26. Wen Y, Zhu X, Rodrigues J, Chen C (2014) Cloud Mobile Media Reflections and Outlook. IEEE Trans Multimedia 16(4):885 –902
27. Wu W, Lui JC (2011) Exploring the Optimal Replication Strategy in P2P-VoD Systems: Characterization and Evaluation. In: Proceedings of IEEE INFOCOM
28. Yan Z, Xue J, Chen CW (2017) Prius: Hybrid edge cloud and client adaptation for http adaptive streaming in cellular networks. IEEE Trans Circuits Syst Video Technol 27(1):209–222
29. Zakerinasab MR, Wang M (2013) A Cloud-Assisted Energy-Efficient Video Streaming System for Smartphones. In: IEEE/ACM IWQoS
30. Zhang H, Xiao Y, Bu S, Niyato D, Yu R, Han Z (2016) Fog computing in multi-tier data center networks: A hierarchical game approach. In: ICC. IEEE
31. Zhang X, Wu C, Li Z, Lau FCM (2015) Online Cost Minimization for Operating Geo-distributed Cloud CDNs. In: Proceedings of IEEE IWQoS
32. Zhao H, Zheng Q, Zhang W, Du B, Li H (2017) A segment-based storage and transcoding trade-off strategy for multi-version vod systems in the cloud. IEEE Trans Multimedia 19(1):149–159
33. Zhou Y, Fu TZJ, Chiu DM (2011) Statistical Modeling and Analysis of P2P Replication to Support VoD Service. In: Proceedings of IEEE INFOCOM