Peer-to-Peer Netw. Appl. (2013) 6:247–256
DOI 10.1007/s12083-012-0148-4

Cost-effective capacity migration of Peer-to-Peer social media to clouds

Qian Zhang · Yusong Lin · Zongmin Wang

Received: 22 January 2012 / Accepted: 17 May 2012 / Published online: 30 May 2012

Abstract Social media streaming has become one of the most popular applications over the Internet. We have witnessed the successful deployment of commercial systems with CDN (Content Delivery Network)-based engines, but they suffer from excessive costs for deploying dedicated servers. And with the further expansions on network traffic of social media streaming, a cost-effective solution remains an illusive goal. The emergence of cloud computing sets out to meet the challenge by dynamically leasing cloud servers. This paper aims to realize the capacity migration of social media systems to clouds at the reduced cost. Firstly, by lowering the capacity requested from clouds to reduce the capacity migration cost. Based on the crawled data from YouTube which is the most representative online social media, we find that with larger than 90% probability, the YouTube user's all requested videos are within three hops of related videos. Then the three hops of related videos are regarded as a cluster and a user's request can be partly satisfied by other users who watch videos in the same cluster to lessen the capacity requested from clouds. Therefore the capacity migration for clusters is under the P2P (Peer-to-Peer) paradigm and a cloud-assisted P2P social media system is proposed. Secondly, given the diverse capacities, cost, limited lease size of cloud servers, we formulate an optimization problem about how to lease cloud servers to minimize the leasing cost and a heuristic solution is presented. The evaluation based on the crawled data from a cluster of YouTube videos shows the efficiency of the proposed schemes.

Keywords Social media · Capacity migration cost · Cloud · Peer-to-Peer · Cluster

Q. Zhang · Y. Lin (✉)
Information Engineering School,
Zhengzhou University, Zhengzhou,
People's Republic of China
e-mail: lys@zzu.edu.cn

Q. Zhang · Y. Lin · Z. Wang
Henan Provincial Key Lab on Information Networking,
Zhengzhou University, Zhengzhou,
People's Republic of China

Q. Zhang
e-mail: qzhang@zzu.edu.cn

Z. Wang
e-mail: zmwang@zzu.edu.cn

1 Introduction

The recent years have witnessed an explosion of social media streaming as a new killer Internet application. YouTube, the most representative online social media, enjoys more than 100 million videos being watched every day [1]. An April 2008 report estimated that YouTube consumed as much bandwidth as did the entire Internet in year 2000 [2], and is still enjoying nearly 20% growth rate per month [3]. Besides that, many other YouTube-like applications have emerged and been developing extremely fast. We have witnessed the successful deployment of these applications with CDN—based engines, but they suffer from excessive costs for deploying dedicated servers. And with the further expansions and rising expense on network traffic, a cost-effective solution should be proposed.

The emergence of cloud computing however sheds new lights into this dilemma. Cloud computing has recently emerged as a new computing paradigm for organizing a shared pool of servers in datacenters into a cloud infrastructure that can provide reliable, elastic and cost-effective resources to users. Dynamic resource provisioning via a cloud has been dramatically changing the way of enabling scalable and dynamic network services [4, 5]. There have been initial attempts for VoD and live streaming applications to migrate their partial capacity to clouds to mitigate the system deployment cost [7–9]. However, the distinct features of social media systems call for new solutions toward a successful cost-effective capacity migration to clouds. One of the distinct features in YouTube-like social media systems is that when a user finished one video and he/she is more likely to select the next video from the watched video's related video list [6]. Combined this feature with our crawled data from YouTube, we conclude that a user views videos within three hops of related videos with larger than 90% probability. This conclusion means that a user's all requested videos are mostly in three hops of related videos. We regard the three hops of related videos as a cluster and a user's request can be satisfied by other users who watched the videos that in the same cluster to lessen the capacity requested from clouds. Therefore, in this paper, the capacity migration is for clusters and the capacity migration for each cluster is in a P2P paradigm. A cloud-assisted P2P social media system is proposed. The capacity can't be satisfied by users will be supplemented by clouds. Given the capacity should be requested from clouds and realistic parameters of cloud servers, an optimization problem about how to lease cloud servers to minimize the leasing cost is formulated and a heuristic solution is presented.

The remainder of this paper is organized as follows. We discuss the related work in Section 2. In Section 3, we present the user's viewing behavior in YouTube-like social media systems based on the crawled data from YouTube. According to the conclusion presented in Section 3, we propose the model of a cloud-assisted P2P social media system in Section 4. We study the characteristics of one cluster in YouTube as an example in Section 5. A capacity prediction scheme to predict the capacity supplemented by clouds for a cluster is proposed in Section 6. In Section 7, we formulate an optimization problem about how to lease cloud servers to minimize the leasing cost and propose a heuristic solution for the optimization problem. Section 8 presents the trace-based evaluations. Finally, we conclude the paper in Section 9.

2 Related work

Recently there is an upsurge of interest in the research community in issues arising from running computation-intensive and data-intensive applications on clouds [14–19]. Many of these applications can now be satisfactorily supported by commercial cloud services [13, 20].

There have been initial attempts for VoD and live streaming applications to migrate their partial capacity to clouds to mitigate the system deployment cost. Wu et al. [7] introduced the paradigm of utilizing cloud services to support VoD applications. Based on a queueing network model, the viewing behaviors in multichannel VoD application can be characterized to derive the cloud server capacities needed to support smooth playback. Then a dynamic cloud resource provisioning algorithm considering the cloud utilization cost is proposed. Li et al. [9] extracted many key characteristics of large-scale VoD systems that are relevant to the hybrid cloud-assisted deployment and realized the cost-aware content migration of C/S-based VoD applications to clouds. CALMS (Cloud-Assisted Live Media Streaming) [8] is a generic framework that facilities a cost-effective live streaming migration to clouds. It well accommodates location diversity, mitigating the impact from user globalization and overall system deployment costs. As for the migration of social media systems to clouds, the study in [10] focused on the load balance of cloud servers and a scheme about the content migration of social media systems to clouds is presented. Different from these exiting works, our work aims to realize the capacity migration of social media systems to clouds at the reduced cost. Based on the distinct feature of YouTube-like social media systems, a cloud-assisted P2P social media system is presented. By lowering the capacity requested from clouds and minimizing the leasing cost of cloud servers to reduce the capacity migration cost.

3 Characteristics of the user's viewing behavior in YouTube-like social media systems

We study the user's viewing behavior in YouTube-like social media systems. In YouTube, We assume a user selects the next video from the related video list [6]. Figure 1 shows a concentric ring for video v. Each ring i represents the related videos of videos that in ring $i-1$. When a user finished watching video v, it can move on to the video that in any ring of video v. Suppose each video has r related videos, then the expected number

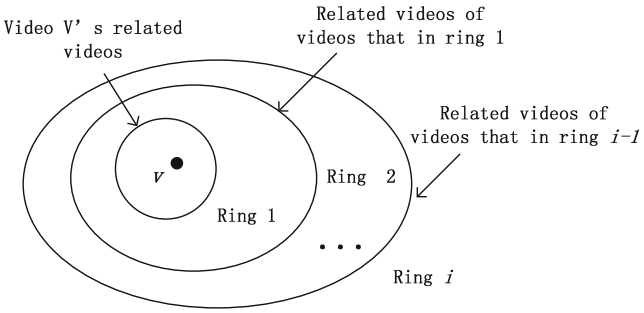


Fig. 1 Concentric ring for video v

of videos in ring i is r^i. For the correlations among videos, some videos in ring i may have already in ring j ($j < i$) and therefore the real number of videos in ring i is smaller than r^i. The real number of videos in ring i is n_i, then the probability of the real number occupies the expected number of videos in ring i is $P_i = n_i \setminus r^i$. The probability of a user through one source video to get videos in ring i is calculated as $P_1 \cdot P_2 \cdot ... \cdot P_i$. Suppose there are $j(j > i+1)$ rings for one video. Then the probability of a user through one source video to get the videos which are within ring i can be derived as $1 - \sum_{k=i+1}^{j} P_1 \cdot P_2 \cdot ... \cdot P_k$.

We study YouTube as an example and a video in YouTube generally has 20 related videos. We pick two source videos and crawl four rings of these two videos to see the probability of the real number occupies the expected number of videos in each ring. The two source videos have the same upload date. One of the source videos is very popular and the number of views is about 2,100,000, the other is an unpopular one and the number of views is only about 1,100. The probability of the real number occupies the expected number of videos in the crawled four rings of the two source videos is shown in Table 1. From Table 1, the probability of the real number occupies the expected number of videos in the popular video's rings is lower than that of the unpopular video. This phenomenon suggests that the videos in popular video's rings have higher probability to repeat. When a user selects an unpopular video, he/she watches the videos within ring three with the probability that is close to $1 - 1 \cdot 0.752 \cdot 0.498 \cdot 0.299 \approx 0.89$. And when a user selects a popular video, he/she watches the videos within ring three with the probability that is close to $1 - 1 \cdot 0.562 \cdot 0.209 \cdot 0.080 \approx 0.99$. Therefore we conclude that in YouTube-like systems, users view videos in three hops of related videos with the probability that is approximately larger than 90%. In other words, a user's all requested videos are mostly in three hops of related videos.

Table 1 Probability of the real number occupies the expected number of videos in four rings

Ring no.	Popular source video	Unpopular source video
1	1	1
2	0.562	0.752
3	0.209	0.498
4	0.080	0.299

4 The model of a cloud-assisted P2P social media system

In this section, the model of a cloud-assisted P2P social media system is presented. From Section 3, in YouTube-like social media systems, a user's all requested videos are mostly in three hops of related videos. Therefore a user's request can be partly satisfied by other users who watch the related videos that in three hops. To lessen the capacity migrated to clouds, the three hops of related videos are regarded as a cluster and the capacity requested from one cluster can be partly satisfied by the users in that cluster.

Figure 2 shows the model of a cloud-assisted P2P social media system. The social media system is partitioned into several clusters, the capacity requested from one cluster can't be satisfied by users will be supplemented by clouds. A broker is a communicating interface between the cloud provider and social media application provider, via which the application provider can submit requests to clouds. The track servers record the user's viewing information of each cluster which can be used to derive the capacity requested from clouds. The function of track servers will be detailed in Section 6. As there may be potential latencies in initiating leases in real world cloud platforms, e.g. 10–30 min in Amazon EC2, it is essential to make a leasing decision in advance. Therefore how to well predict the capacity supplemented by clouds for each cluster is also a key issue to realize the capacity migration. Then given that the cloud servers have different types, capacities, cost, and limited lease size, the other key problem is about leasing which type and how many cloud servers of that type to minimize the leasing cost.

5 Characteristics for clusters

We study one cluster of YouTube videos as an example to see the evolution of views and the popularity

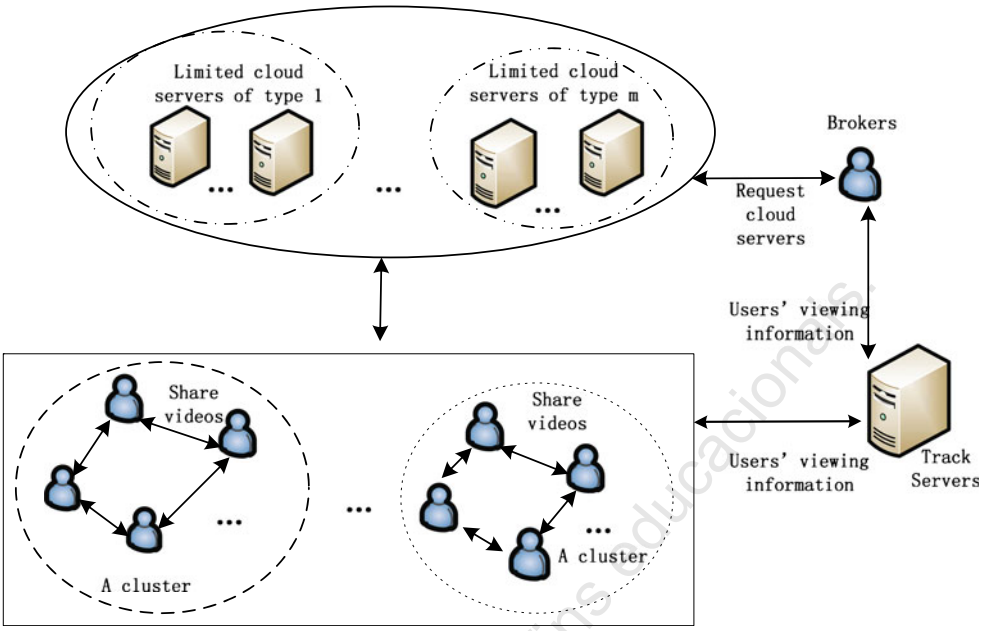


Fig. 2 A cloud-assisted P2P social media system

distribution of videos. Recall that the three hops of related videos are regarded as a cluster. The cluster has about 2000 related YouTube videos and the videos in a cluster are got by a crawl which start with one video and went to three depths. Then the number of views in the cluster is collected by a crawler which performed once a hour. As the lease duration of cloud servers is short, e.g. 1 h for EC2, the short-term evolution of views is investigated and we crawl 24 h for the cluster.

5.1 Evolution of views in short term

From Fig. 3, we can see that the number of views for the crawled cluster is increasing without large fluctuation and it is predictable. The reason for this may be the types of videos in a cluster are various and there are relative popular as well as unpopular videos, which can be seen from Fig. 5. Given the views time series $\{N_t\}$ of a cluster in the past few hours, we can make the fine-grained prediction into its future evolution. Due to the increasing trend of views exhibited in a cluster, $\{N_t\}$ is clearly non-stationary. The ARIMA(autoregressive integrated moving-average) model which is a generalization of an ARMA model(autoregressive moving average) is applied for non-stationary views prediction. Now we briefly outline the ARIMA model [11].

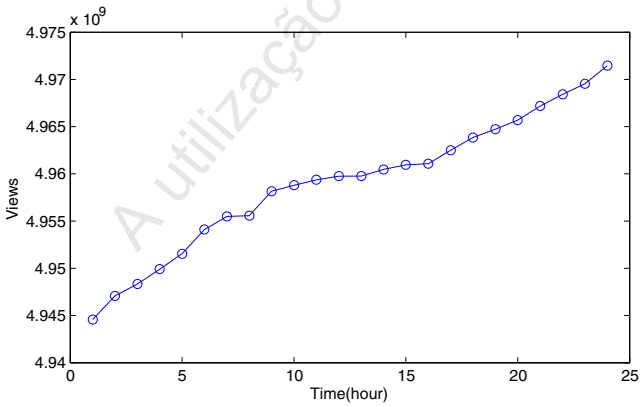


Fig. 3 The evolution of views

The ARIMA(p, d, q) series can be defined by the following equation:

$$A(B)\nabla^d y_t = C(B)\epsilon_t$$

where y_t is the time series, ϵ_t is a white noise process.

$$A(B) = 1 - \alpha_1 B - \alpha_2 B^2 - \ldots - \alpha_p B^p$$

$$C(B) = 1 - \beta_1 B - \beta_2 B^2 - \ldots - \beta_q B^q$$

B is the lag operator which gives the previous value of the series when placed in front of any variable with a time subscript: $Bx_t = x_{t-1}$, $(1 - B)x_t = x_t - x_{t-1}$. d is the number of differences required to give a stationary series. ∇^d is the dth power of difference operator.

For the given views time series N_t has a liner growth behavior, d is chosen to be 1 to make a stationary process. We make an one hour-ahead views prediction. The data of the first 10 h are chosen as the initial

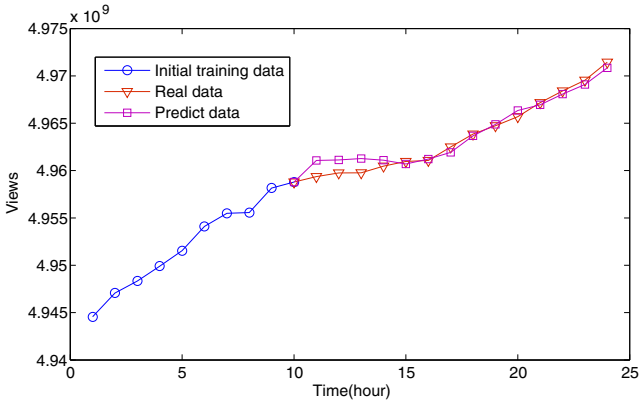


Fig. 4 Views prediction based on ARIMA model

training data, the parameter p and q are obtained through a maximum likelihood estimator [11]. We predict the number of views for the crawled cluster. Choose the first 10 h as the initial training data to predict the number of views in the 11th h, when the actual number of views of the 11th h is derived, it is added to the training data set to predict the views of the next hour. The process is dynamic for that when there has an actual value, it is added to the training data set to predict the next value. Figure 4 shows that one hour-ahead views prediction based on ARIMA model for the crawled cluster.

5.2 Video popularity distribution

Figure 5 shows the number of views against the rank of videos in log-log scale for the crawled cluster. We can see it begins with a fit like a Zipf's distribution with $a = 0.74$ and the tail decreases tremendously.

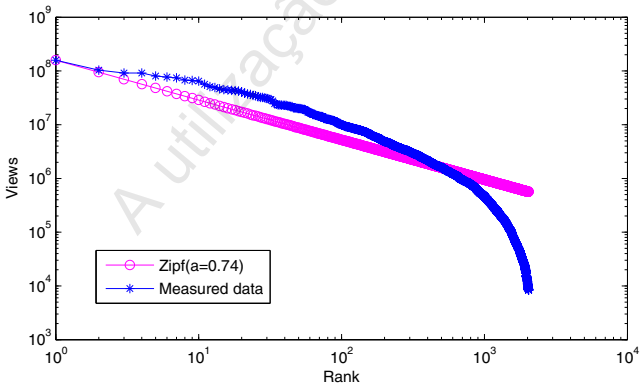


Fig. 5 The number of views against different rank of videos

6 Capacity prediction

As there may be potential latencies in initiating leases in real world cloud platforms, e.g. 10–30 min in Amazon EC2, it is essential to make a leasing decision in advance. Therefore the capacity prediction for a cluster to predict the capacity supplemented by clouds is also a key issue to successfully realize the capacity migration to clouds. In this section, we present the capacity prediction scheme for a cluster to derive the capacity supplemented by clouds. Firstly, the access probabilities among videos are given. Secondly, based on the access probabilities among videos, the user's contribution and the capacity supplemented by clouds are derived. We summarize important notations used in this paper in Table 2 for ease of reference.

6.1 Access probabilities among videos

The access probability between video i and j is the probability that a user finished video i then choose to watch video j. The track servers maintain a peer list for each video and peers in the peer list cache the watched videos. For each video, the access probabilities with other videos are calculated by the cache information of peers in its peer list. For example, for the peer list l_i of video i, suppose the total number of videos that peers in l_i cached is m and there are m_j peers in l_i who cached

Table 2 Notation table

Symbol	Definition
$N(t)$	Views for time period t.
$P_{ij}(c)$	The correlation probability between video i and j in cluster c.
m	The number of videos in a cluster.
u	The upload capacity of each peer.
b	The download bandwidth requested by one peer to ensure the smooth playback.
r	The ratio of the new comers.
$S_j(t)$	The number of peers who are not new comers for video j for time period t.
$E(V_j(t))$	Expected number of seeds for video j in time period t.
$E(V(t))$	Expected number of seeds for all videos in time period t.
R	Capacity supplemented by clouds.
C_i	The leasing cost of cloud servers of type i.
N_i	The maximal lese size of cloud servers of type i.
b_i	The bandwidth allocated for cloud servers of type i.
n_i	The number of cloud servers of type i to be leased

video j, then the access probability between video i and j is calculated as m_j/m.

6.2 Capacity provisioned by clouds with user-assistance

The capacity prediction is for one cluster. We suppose $N(t)$ is the predicted number of views for time period t and there are $r \cdot N(t)(0 \le r \le 1)$ new comers for the cluster, where r is the ratio of the new comers. As users in one cluster can share videos with each other, the $(1-r) \cdot N(t)$ users who cache the watched videos are potential seeds for videos in that cluster. Then the users' contributions to the requested capacity from one cluster can be derived.

The $(1-r) \cdot N(t)$ potential seeds' viewing behavior for different rank of videos in one cluster is supposed to follow a Zipf ($a = 0.74$) distribution. Then the number of users who cached video j can be derived and it is supposed to be $S_j(t)$. For the correlations among related videos, users who cached video j may also be the seeds for its related videos. The access probabilities among videos in cluster c can be demonstrated as a matrix $P(c)$. $P_{ij}(c)$ represents the access probability between video i and j. Suppose there are m videos in cluster c, the expected number of seeds for video j in time period t is $E(V_j(t))$ which can be derived as

$$E(V_j(t)) = \sum_{i=1, i \neq j}^{m} S_j(t) + S_i(t) \cdot P_{ij}(c) \quad (1)$$

Then the expected number of seeds for all videos in time period t is $E(V(t))$ which can be calculated as

$$E(V(t)) = \sum_{i=1}^{m} E(V_i(t)) \quad (2)$$

Suppose each user has the same upload bandwidth of u and the bandwidth requested for a user to ensure the smooth playback is b. Then the capacity supplemented by clouds for time period t can be derived as

$$R = b \cdot N(t) - u \cdot E(V(t)) \quad (3)$$

7 Cost-effective leasing cloud servers

Given the unit time for the duration of leasing a server is one hour, the provision algorithm presented below is periodically run every interval of 1 h. The capacity supplemented by clouds to satisfy the users' demand in each leasing duration is got by the capacity prediction scheme that is presented in Section 6. Given different types of cloud servers with different capacities, cost and maximal lease size, the objective is to find the optimal leasing types and numbers of cloud servers to minimize the leasing cost. In this section the problem about how to lease cloud servers is formulated and a heuristic solution for the problem is proposed.

7.1 Problem statement

Denote there are n types of cloud servers can be leased from the cloud providers. The maximal number of cloud servers can be leased from type i is N_i. The leasing cost for cloud servers of type i is C_i. Let b_i be the bandwidth capacity has been allocated for the cloud servers of type i. Recall that R is the predicted capacity supplemented by clouds, which can be derived from Eq. 3. Define a cloud service lease schedule as $S = \{(1, n_1), (2, n_2), ..., (j, n_j)\}$, $j \le n$, in which the item $(i, n_i)(1 \le i \le n)$ represents that there are n_i cloud servers of type i to be leased. The problem is to find a proper cloud lease schedule S, subjecting to the following constraints: (1) Streaming quality constraint: $\sum_{i=1}^{j} b_i \cdot n_i \ge R$, $j \le n$. (2) Maximal lease size constraint: $n_i \le N_i$. The streaming quality constraint asks that the requested capacity should be no less than the predicted capacity supplemented by clouds. As the lease size of cloud servers is limited, the maximal lease size constraint for each type of cloud servers is required. Under these constraints, the objective is to minimize the leasing cost: $C_{\text{lease}} = \sum_{i=1}^{j} C_i \cdot n_i$.

Then the optimal problem to decide the lease schedule is described as:

$$\min \sum_{i=1}^{j} C_i \cdot n_i$$

Constraints:

$$\sum_{i=1}^{j} b_i \cdot n_i \ge R,\ j \le n$$

$$n_i \le N_i, i = 1...n.$$

As the problem described above, the combinatorial optimization problem can be reduced to a Multiple Knapsack Problem [12].

7.2 Heuristic solution

The heuristic solution for the combinatorial optimization problem described in Section 7.1 is based on a greedy algorithm which is making the locally optimal choice at each stage with the hope of finding a global optimum.

The cloud server of type i is represented by a triple (i, C_i, N_i) $(1 \le i \le n)$, where N_i is the maximal lease size of cloud servers of type i. Set C is $\{(i, C_i, N_i)\}$ $(1 \le i \le n)$, it represents the set of all types of cloud servers. Recall that set S is the lease schedule. The element in S is (i, n_i) which represents there are n_i cloud servers of type i to be leased. The bandwidth allocated for cloud servers of type i is b_i and the capacity provided by the cloud servers in set S is R_0. The capacity needed to be supplemented by clouds is R. The description of the heuristic solution is as follows: (1) Pick a triple (i, C_i, N_i) from set C, the picked triple has the minimum value of C_i compared to other triples in set C. Then based on the value of $N_i * b_i + R_0$ to do the corresponding operations described as follows: (2) If $N_i * b_i + R_0 < R$, Put (i, N_i) in set S, recalculate R_0 and delete (i, C_i, N_i) from set C, then go to (1); (3) If $N_i * b_i + R_0 > R$, get the minimum integer value of n_i which makes the value of $n_i * b_i + R_0$ is no smaller than R, then put (i, n_i) in set S and S is the final set for the lease schedule; (4) If $N_i * b_i + R_0 = R$, put (i, N_i) in set S and S is the final set for the lease schedule. The pseudo-code of the process is presented in Algorithm 1.

Algorithm 1 Algorithm for leasing cloud servers

INPUT: The set of all types of cloud servers $C = (1, C_1, N_1), ..., (m, C_n, N_n)$
Pick a triple from set C with the minimum value of C_i, then put it in set S
while $(N_i * b_i + R_0) < R$ **do**
Pick a triple from set C with the minimum value of C_i, then put it in set S
Recalculate the value of R_0
end while
if $N_i * b_i + R_0 > R$ **then**
Get the minimum integer value of n_i which makes the value of $n_i * b_i + R_0$ is no smaller than R, then put (i, n_i) in set S
end if
if $N_i * b_i + R_0 = R$ **then**
Put (i, N_i) in set S
end if
OUTPUT: The leasing schedule set S

8 Simulation

8.1 Simulation parameters

The cost-effective capacity migration of social media system to clouds is for clusters in this paper. Then the views evolution in our simulation is based on the crawled cluster. The leasing cost of different types of cloud servers is based on Amazon EC2 [13] and the leasing cost of cloud servers is the price for hours. The meaning of parameters b, u, r is as described in Section 6. If there is no special explanation, we suppose $r = 0.6$, $b = u$. The scheme of leasing cloud servers is as presented in Section 7. The capacity allocated for different types of cloud servers is different and it is reflected by the number of peers that can be simultaneously satisfied by the cloud server. The supposed capacity and maximal lease size for each type of cloud servers are shown in Table 3.

Table 3 Cloud servers configurations

Type	Price per hour	Maximal lease size	Capacity
Small	\$0.12	75	9
Large	\$0.48	55	17
Extra large	\$0.96	40	29
Micro	\$0.03	90	5

8.2 Efficiency of the proposed schemes

We use the views of the crawled cluster in the first 10 h as the initial training data set and make the one-hour ahead prediction. The capacity provisioned by clouds is got from the capacity prediction scheme which is presented in Section 6.1. There are two metrics to evaluate it. One is the proportion of unmet users which is the number of unmet users divided by the number of total requested users. The other is the deviation rate between the real and predicted capacity which is calculated by the difference between the real and predicted capacity divided by the real capacity. From Fig. 6, the deviation is decreasing with time for that

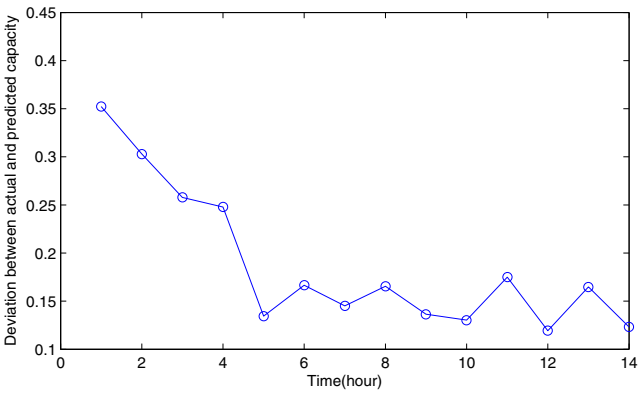


Fig. 6 Deviation between the actual and predicted capacity

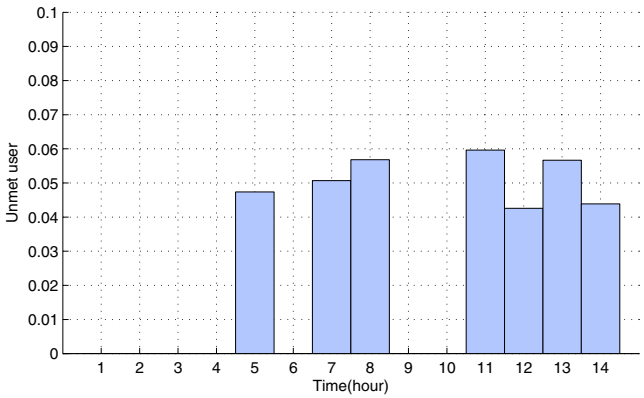


Fig. 7 Unmet user requests

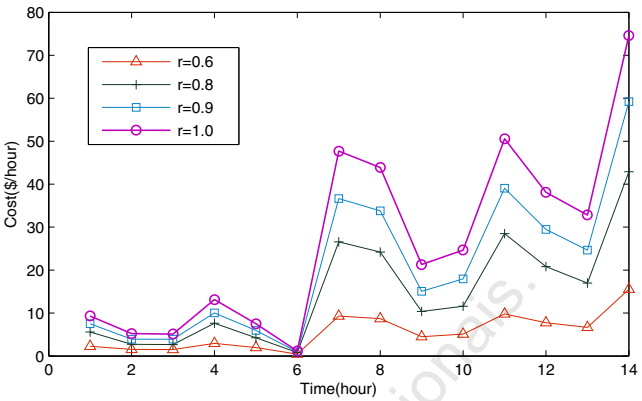


Fig. 9 Impact of ratio r on the cost

the dynamic prediction with the increasing number of training data makes the views prediction more accurate. Figure 7 shows the unmet user requests, we can see that the unmet user requests is zero in some time periods and the reason for this is the predicted capacity is larger than the real capacity. It is not to say that the efficiency of the capacity prediction is good when the unmet user requests is zero, for that it is uneconomic when the deviation between predicted and real capacity is high. As shown in Figs. 6 and 7, though the unmet user request is zero in the first 4 h, the corresponding deviation in these time periods are relatively high and there must be some wastage on leasing cloud servers.

The scheme about leasing cloud servers with diverse capacities, types, cost and limited lease size is based on a greedy algorithm. We compared it to a random based leasing scheme. In the random based leasing scheme, different from the greedy based algorithm, the first step of the leasing algorithm shown in Section 7.2 does not pick a triple with the minimum cost but randomly pick a triple. Figure 8 shows that the leasing cost of the random based scheme is higher than that of the proposed scheme.

8.3 Impact of ratio r on the leasing cost

Figure 9 shows that the impact of parameter r on the leasing cost. Recall that r is the ratio of new comers in each time period. It is clearly that the leasing cost is decreasing with r decreases. The reason for this phenomenon is that the decreasing r makes the capacity provided by users increasing and then lessen the capacity supplemented by clouds. $r = 1$ represents that the user's demand should be only satisfied by cloud servers without other users' assistance and under this circumstance, the cost is the highest. Figure 9 also suggests that users in one cluster cache the watched videos can be seen as seeds for the cached videos and the user's contribution decreases the demand requested

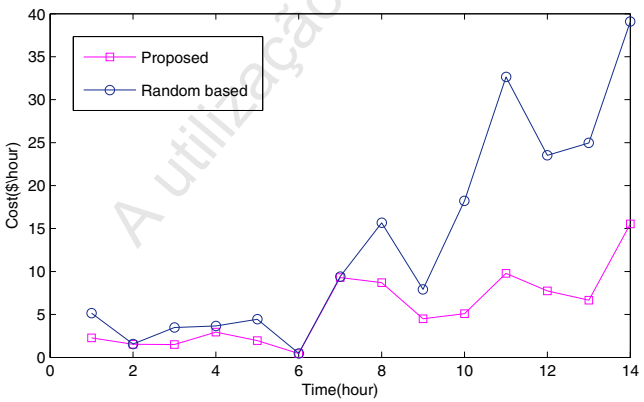


Fig. 8 Efficiency of the proposed leasing scheme

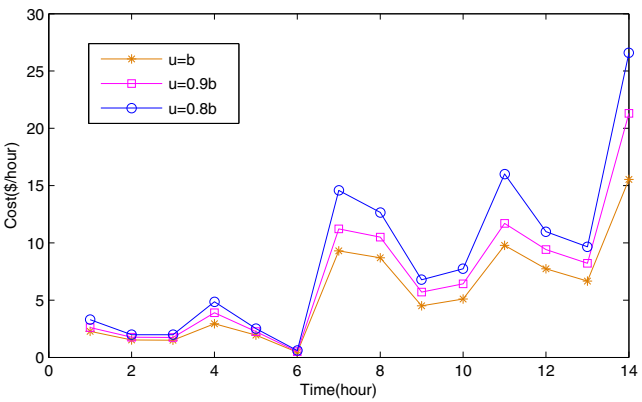


Fig. 10 Impact of user's upload capacity on the cost

from cloud servers as well as the leasing cost of cloud servers.

8.4 Impact of user's upload capacity u on the leasing cost

As users that in one cluster can share videos with each other to lessen the capacity migrated to clouds, we also evaluated the impact of users' upload bandwidth availability on the leasing cost of cloud servers. We vary the value of the user's upload bandwidth u to see the impact on the leasing cost. As expected, less cloud resource and leasing cost is needed when user's upload capacity is higher. As shown in Fig. 10, for each leasing duration, the higher upload capacity of users and the lower leasing cost of cloud servers.

9 Conclusion and future work

In this paper, we realized the capacity migration of social media systems to clouds at the reduced cost. Based on the crawled data from YouTube which is the most representative online social media, we find that a user views videos within three hops of related videos with the probability that is larger than 90%, in other words, a user's all requested videos are mostly in three hops of related videos. The three hops of related videos are regarded as a cluster and a user's request can be partly satisfied by other users in the same cluster to lessen the capacity requested from clouds. The capacity migration for clusters is in a P2P paradigm. A cloud-assisted P2P social media system is presented to reduce the capacity migration cost. The capacity cannot be satisfied by users in one cluster will be supplemented by clouds. As there may be potential latencies in initiating leases in real world cloud platforms, e.g. 10–30 min in Amazon EC2, it is essential for the successful migration to make a leasing decision in advance. And based on the characteristics of a cluster of You-tube videos, a capacity prediction scheme for one cluster is presented. Then given the capacity supplemented by clouds, and the diverse capacities, cost, limited lease size of cloud servers, we formulated an optimization problem about how to lease cloud servers to minimize the leasing cost and a heuristic solution is presented. The evaluation based on the real data from a cluster of YouTube videos showed the efficiency of the proposed schemes.

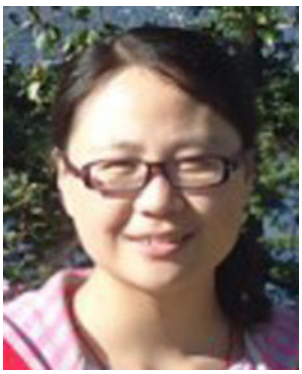
For each cluster, the clouds are supposed to have all videos and the content placement scheme is not considered. However, from the crawled data, we can see there have various types of videos in a cluster, such as popular and unpopular videos. Most unpopular videos may be served by users and it is obviously uneconomic when they are all placed on clouds. In the future work, we will focus on the cost-effective content placement scheme for the social media systems and take the cost of content placement into account. From the key issues for realizing the cost-effective capacity migration of social media systems to clouds, the further study for a capacity prediction model of social media systems is also needed.

Acknowledgements This paper is sponsored by Information Engineering Project of He Nan Province under Grant No. 2008xxh001 and Innovation Project of He Nan Province under Grant No. 2011HASTIT003 with Zhengzhou University.

References

1. YouTube serves up 100 million videos a day online. http://www.usatoday.com/tech/news/2006-07-16-youtube-views_x.htm. Online accessed 5 Dec 2011
2. Web could collapse as video demand soars. http://www.telegraph.co.uk/news/uknews/1584230/Web-could-collapse-as-video-demand-soars.html. Online accessed 5 Dec 2011
3. Corbett C (2006) Peering of video. http://www.nanog.org/mtg-0606/pdf/bill.norton.3.pdf. Accessed 5 Oct 2008
4. Armbrust M, Fox RGA, Joseph AD, Katz RH, Konwinski A, Lee G, Patterson DA, Rabkin A, Stoica I, Zaharia M (2007) Above the clouds: a berkeley view of cloud computing. University of California, Berkeley, Tech. Rep
5. Li A, Yang X, Kandula S, Zhang M (2010) CloudCmp: comparing public cloud providers. In: Proceedings of ACM IMC
6. Cheng X, Dale C, Liu J (2008) Statistics and social network of YouTube videos. In: Proceedings of IEEE IWQoS
7. Wu Y, Wu C, Li B, Qiu X, Lau FC (2011) CloudMedia: when cloud on demand meets video on demand. In: Proceedings of IEEE ICDCS
8. Wang F, Liu J, Chen M (2012) CALMS: migration towards cloud-assisted live media streaming. In: Proceedings of IEEE INFOCOM
9. Li H, Zhong L, Liu J, Li B, Xu K (2011) Cost-effective partial migration of vod services to content clouds. In: Proceedings of IEEE cloud
10. Cheng X, Liu J (2011) Load-balanced migration of social media to content clouds. In: Proceedings of ACM NOSSDAV
11. Box GE, Jenkins MG, Reinsel GC (2008) Time series analysis: forecasting and control. Wiley
12. Kellerer H, Pferschy U, Pisinger D (2004) Knapsack problems. Springer
13. Amazon Elastic Compute Cloud. http://aws.amazon.com/ec2/. Online accessed 15 Dec 2011
14. Pandey S, Wu L, Guru S, Buyya R (2010) A particle swarm optimization (PSO)-based heuristic for scheduling workflow applications in cloud comuputing environment. In: Proceedings of IEEE AINA
15. Xiao Y, Lin C, Jiang Y, Chu X, Shen S (2010) Reputation-based QoS provisioning in cloud computing via Dirichlet multinomial model. In: Proceedings of IEEE ICC

16. Peixoto MLM, Santana MJ, Estrella JC, Tavares TC, Kuehne BT, Santana RHC (2010) A metascheduler architecture to provide QoS on the cloud computing. In: Proceedings of IEEE ICT
17. Yu S, Wang C, Ren K, Lou W (2010) Achieving secure, scalable, and fine-grained data access control in cloud computing. In: Proceedings of IEEE INFOCOM
18. Wang C, Wang Q, Ren K, Lou W (2010) Privacy-preserving public auditing for data storage security in cloud computing. In: Proceedings of IEEE INFOCOM
19. Urgaonkar R, Kozat UC, Igarashi K, Neely MJ (2010) Dynamic resource allocation and power management in virtualized data centers. In: Proceedings of IEEE/IFIP NOMS
20. Amazon Simple Storage Service. http://aws.amazon.com/s3/. Online accessed 15 Dec 2011



Qian Zhang Ph.D. candidate in Information Engineering School, Zhengzhou University, China. She received the B.E. degree in Information Security from PLA Information Engineering University, China, in 2008. Her research interests are Peer-to-Peer network applications, social network and cloud computing.

Yusong Lin Associate professor of Zhengzhou University, China. He received his Ph.D. degree in Communication and Information Systems from PLA Information Engineering University, China, in 2005. His research interests are Peer-to-Peer network applications, mobile Internet technology and cloud computing.

Zongmin Wang Professor of Zhengzhou University, China. He received his Ph.D. degree from Tsinghua University, China, in 1995. He conducted his postdoctoral research in the University of Hong Kong from 1995 to 1996. His research interests are Peer-to-Peer network applications, multimedia applications, virtual reality technique and cloud computing.