On the Impact of Popularity Decays in Peer-to-Peer VoD Systems

Fei Chen
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
Email: feic@sfu.ca

Haitao Li
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
Email: haitaol@sfu.ca

Jiangchuan Liu
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
Email: jcliu@sfu.ca

***Abstract*—Today's peer-to-peer (P2P) Video-on-Demand (VoD) systems are known to be highly scalable in a steady state. For the dynamic scenario, much effort has been spent on accommodating sharply increasing requests (known as *flash crowd*) with effective solutions being developed. The high popularity upon a flash crowd however does not necessarily last long, and indeed often drops very fast after the peak. Compared to growth, a decay is seemingly less challenging or even beneficial given the less user demands. While this is true in a conventional client/server system, we find that it is not the case for peer-to-peer. A quick decay can easily de-stabilize an established overlay, and the resultant smaller overlay is generally less effective for content sharing. The replication of data segments, which is critical during flash crowd, will not promptly respond to a fast and globalized population decay, either. Many of the replicas can become redundant and, even worse, their spaces cannot be utilized for an extended period. In this paper, we seek to understand the impact of such decays and the key influential factors. To this end, we develop a mathematical model to trace the evolution of peer upload and replication during population churns, specifically during decays. Our model captures peer behaviors with common data replication and scheduling strategies in state-of-the-art peer-to-peer VoD systems. It quantitatively reveals the root causes toward escalating server load during a population decay. The model also facilitates the design of a flexible server provision to serve highly time-varying demands.**

I. Introduction

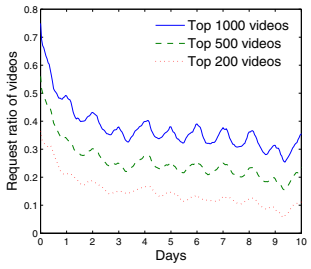
The Internet has witnessed a significant increase in the popularity of peer-to-peer (P2P) Video-on-Demand (VoD) applications, which takes advantage of peer upload bandwidth contributions to rapidly spread data among the users [1]. Ideally, a P2P VoD system is highly scalable and self-sustainable in a steady state [1] [7]. In practice, however, the server load saving can hardly be more than 95% [7] and is generally less, particularly in the presence of user population dynamics [11]. One of the most notable scenarios is *flash crowd*, in which hundreds of thousands of users joining the system within a short period of time, just after a new movie or drama series has been released [10]. Such a surge in user population can dramatically disturb the balance established in a steady peer-to-peer overlay, thus deteriorating the streaming quality [3]. There have been a series of works addressing the challenge of flash crowd [10], which often absorb the surge through deploying a potentially large number of servers (e.g., 60 dedicated servers in the Coolstreaming+ system [3]) or leveraging content delivery networks [5].

Experiences in commercial system deployment, e.g., PPLive [7], have shown these solutions for flash crowd work reasonably well, despite the extra server cost incurred. The other side of the coin however has not been well addressed. That is, in a VoD system, the high popularity upon a flash crowd does not necessarily last long, and indeed often drops very fast after the peak. For example, in YouTube, the top videos tend to experience significant bursts of popularity, receiving a large fraction of their views on a single peak day or week [8]; among the top-5000 most popular videos provided by Hulu, the popularity decays by 20% after the first day [9]. Our trace data from PPLive confirms that such a quick population decay exists in peer-to-peer VoD system as well.

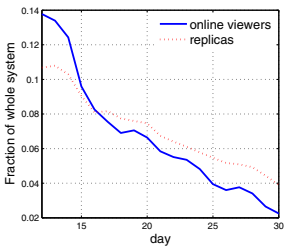
Compared to growth, the decay is seemingly less challenging and would be even beneficial given the less user demands. While this is true in the conventional client/server communication paradigm, we find that it is not the case for peer-to-peer. First, a decayed population means a smaller overlay for the video, which defeats the benefit of peer-to-peer sharing. This is particularly severe with a quick decay, which can easily de-stabilize an established overlay; Second, replication has been widely used to improve sharing efficiency for popular videos and to mitigate the impact of flash crowd [7]; yet the replica in individual peers' local storage will not promptly respond to a fast and globalized population decay. In other words, many of the replicas become redundant and, even worse, their spaces cannot be utilized for an extended period with state-of-the-art replication strategies. Lastly, such VoD systems as PPTV, Netflix, and YouTube now often release a group of popular videos (e.g. a TV drama series) together, and these videos will then experience similar user watching patterns (e.g., many users watched them one-by-one in a sequence). Their collective impact to the population (growth or decay) will be even more damaging.

In this paper, we seek to understand the impact of such decays and the key influential factors. To this end, we develop a mathematical model to trace the evolution of peer upload and replication during population churns, specifically during decays. Our model captures peer behaviors with common data replication and scheduling strategies in state-of-the-art peer-to-peer VoD systems. It reveals that, during a sharp population decay, the peers' local storage is not effectively utilized for upload, and the imperfect content replication with slow response inevitably results in an escalating server load. The model also facilitates the design of a flexible server provision to serve

978-1-4799-0590-4/13/$31.00 ©2013 IEEE



(a) Top-K Videos



(b) Replicas and Online Users

Fig. 1: Popularity Decays

highly time-varying demands. Numerical results demonstrate the influence of such key factors as upload ratio and eviction ratio during population decays as well as the superiority of our server provision design.

II. Existence and Challenges of Decayed Popularity

The myriad of different contents in today's VoD systems make user behavior and attention span highly variable with fast-changes [8]. Fig. 1(a) shows the popularity evolution of the TOP 1000, 500, and 200 popular videos in the Hulu web service [9], respectively. We can see that, despite a peak of access in the beginning, most of these popular videos suffer from a rapidly decayed popularity in the following days. For peer-to-peer delivery, each peak of access (i.e., flash crowd) needs considerable effort to accommodate. Yet the immediately followed popularity decay largely destroys the balance in an established peer-to-peer overlay and renders the effort to be useless. Consider a typical example comes from PPlive, one of the most successful P2P streaming systems with multi-million users. Our data traces from PPLive show that a very popular drama series containing 26 sets [1] quickly attracted nearly 20% online views among all the videos (over 10,000) in the whole system. The server load has increased to accommodate the flash crowd but then decreased after a large overlay has been established, which makes a number of data segments from the drama series be replicated among the peers. After two weeks, the popularity of the drama series declined sharply, and yet the server load increases sharply too. Fig. 1(b) shows the popularity decaying process of the drama series from the 12th day after they are released. We can see that its popularity has dramatically decreased from 14% to 2%. Meanwhile, the replication ratio decreases much slower, only from 11% to 4%. The replication strategy in PPLive is an online algorithm which needs sufficient time to adapt to the changing popularity. As such, many of the replicas for the drama series become temporally redundant in the decaying process, and even worse, prevent the peers' local storage spaces from being used for newer popular videos. This, together with the diminishing overlay size, contribute to the increased server load.

In the following section, we present a mathematic model to capture the inherent relationship between the video popularity decreasing process and the peer upload capacity evolution. Our model quantitatively explains the increase of the server load and identifies the key influential factor. It also facilitates the design of a flexible server provision strategy.

[1]The sets of the drama series are released together, and most of PPLive user viewed them one by one continuously.

TABLE I: Model Notation

Parameter	Definition
M	Number of videos in the back-end storage server
$N(t)$	Number of online users in the time slot t
$n_j(t)$	Number the online watching the video j in time slot t
$u(i)$	Upload bandwidth of peer i
$r(j)$	Playback rate of the video j
C	Storage capacity of the peer to store the video locally
$\alpha_{i,j}(t)$	The replication map of user i for video j at time slot t
$\beta_{i,j}(t)$	The upload scheduling map of user i for video j at time slot t
$S(t)$	Server support at time slot t
$D(t)$	Request demand at time slot t
$U(t)$	Upload capacity by peers at time slot t

III. System Model

In this section, we will present our basic model of the P2P VoD system, which assumes that there are M videos. Without loss generality, we assume that all the videos are of unit size and with the same playback rate $r_j = r$, for $j = 1, 2, ...M$ [6]. There is a server that stores all the videos and services as backup whenever a peer can not achieve the required download rate (equal to the playback rate) [5]. In each time slot t, the number of online users in the system is default $N(t)$. The total number of peers requests is $\sum_{j=1}^{M} n_j(t) = N(t)$ where $n_j(t)$ represents the population of online users viewing the video j. Each peer contributes the limited upload bandwidth $u(i)$ and the storage capacity to store C videos, where $C \ll M$.

The server support $S(t)$ is determined by two components, the current user request demand $D(t)$ and the user upload capacity $U(t)$. If the total bandwidth demand exceeds the user upload capacity, the server bandwidth has to be provisioned for the normal playback of all the peers in the system [3]. The user upload capacity $U(t)$ is determined by following three parameters, (a) the upload bandwidth u_i of each peer, (b) the replication distribution map $\alpha_{i,j}(t)$ (e.g., $\alpha_{a,b}(t) = 0$ means video b is not replicated by peer a at the time slot t), and (c) the upload scheduling map $\beta_{i,j}(t)$ (i.e. the bandwidth utilization of peer i for video j at time slot t). We assume that the download bandwidth of each peer is not the constraint in the system [10] [11]. With the global knowledge, we can have the lower bound of the server bandwidth support as follows:

$$Min. \quad S(t) = \sum_{j=1}^{M}\{rn_j(t) - \sum_{i=1}^{N(t)} u_i\alpha_{i,j}(t)\beta_{i,j}(t)\} \quad (1)$$

$$s.t. \quad 0 \le \sum_{j=1}^{M} n_j(t) \le N(t) \quad (2)$$

$$0 \le \sum_{j=1}^{M} \alpha_{i,j}(t)\beta_{i,j}(t) \le 1 \quad (3)$$

$$0 \le \sum_{j=1}^{M} \alpha_{i,j}(t) \le C \quad (4)$$

Eq. (2) shows the constraints of the online user request for the videos, and the constraints of the user upload ability with

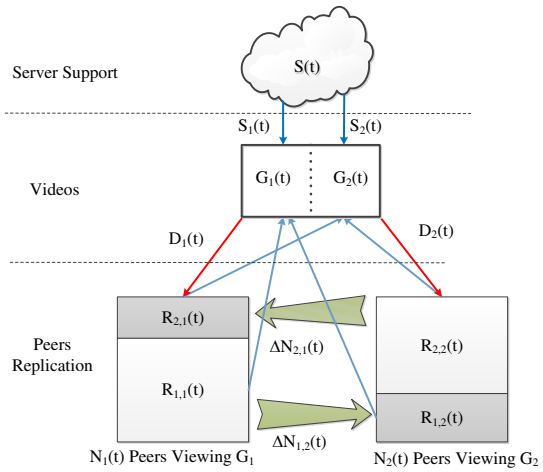


Fig. 2: Demand and Supply Relationship

the limited bandwidth and storage capacity are presented in Eq. (3) and Eq. (4) respectively.

Note that even the global information is given in the stationary scenario, the optimal solution is very hard to achieve [11]. Furthermore, with the dynamic peer churn and video popularity churn, it is too expensive to acquire the global information on time. Instead, in this paper, we focus on understanding how replication evolves in the local storage of peers and how much server resource should be provisioned during the popularity decay process. This facilitates the design of effective replication and scheduling strategies. We will simplify the proposed model on a general homogeneous case, as we are more interested in the asymptotic collective behavior of the system rather than the individual peer behavior.

To characterize the video popularity decay process, we assume that the M videos are divided into two groups, namely the popularity decaying group G_1 with size of K videos $m_1, m_2, m_3, ..., m_K$ and the popularity increasing group G_2 with the rest videos $m_{K+1}, m_{K+2}, ...m_M$. Thus, in our model the videos in G_1 can be considered as the members of the newly broadcasted drama series. They will experience an fast popularity decaying process, as there exists more viewers who have finished viewing these videos in the system. Accordingly, we define the number of peers viewing the videos in G_1 as $N_1(t)$, and the numbers of peers viewing the videos in G_2 as $N_2(t)$. The total number of peers in the system can be considered as constant [2], since the video popularity churn occurs on a much faster time scale than peer churn (i.e. peer entering or leaving the system) [6]. Therefore, we can assume a simple linear relationship between $N_1(t)$ and $N_2(t)$ in each time slot, $N_1(t) + N_2(t) = N$. We can have the total request demand in each time slot as follows:

$$D(t) = D_1 + D_2 = \sum_{j \in G_1, G_2} r n_j(t) = r N_1(t) + r N_2(t) \quad (5)$$

[2]We assume that in each time slot the number of the online peers is a constant, $N(t) = N$, for $t_0 \leq t \leq t_e$, where t_0 means the initial time slot and t_e means the final time slot.

In Fig. 2, we show the relationship between the upload capacity and the demand distribution. The blue lines indicate the upload capacity from the server support $S(t)$ and peer upload $U(t)$, which equals to the total demand capacity $D(t)$ indicated by the red lines. Since the total number of peers in time slot t is constant, the popularity churn is driven by the peers exchange between the different viewing groups (e.g. $\Delta N_{2,1}(t)$ implies the viewing peers flow from $G_2(t)$ to $G_1(t)$ in the time slot t). Define $N_2(t-1)$ as N_2' for short. Accordingly, we have:

$$\begin{cases} \Delta N_2(t) = N_2(t) - N_2' = \Delta N_{1,2}(t) - \Delta N_{2,1}(t) \\ \Delta N_1(t) = N_1(t) - N_1' = \Delta N_{2,1}(t) - \Delta N_{1,2}(t) \end{cases} \quad (6)$$

In each time slot t, the viewing population change (i.e. $\Delta N_1(t)$ and $\Delta N_2(t)$) directly influences the current request demand in the system. Define the relative popularity of the group members as $\rho_1(t) = \frac{N_1(t)}{NK}$ and $\rho_2(t) = \frac{N_2(t)}{N(M-K)}$. We further define that $\theta_1 = \frac{\Delta N_1(t)}{N}$ and $\theta_2 = \frac{\Delta N_2(t)}{N} = -\theta_1$ as the popularity churn ratio. We can then formulate the popularity churn as follows:

$$\begin{cases} \rho_1(t) - \rho_1(t-1) = \frac{\Delta N_1(t) - \Delta N_1(t-1)}{NK} = \frac{\theta_1}{K} \\ \rho_2(t) - \rho_2(t-1) = \frac{\Delta N_2(t) - \Delta N_2(t-1)}{NK} = \frac{\theta_2}{M-K} \end{cases} \quad (7)$$

In this paper we focus on the analysis of the popularity decay process, and assume that $\rho_1(t_0) \gg \rho_2(t_0)$.

Scheduling Strategy We assume that each peer has only partial knowledge of others and competes for the limited resources [10]. A random scheduling strategy is utilized for partner selection, and we consider $\frac{r}{\kappa}$ as the bit rate corresponding to a unit bandwidth of the connection. There should be at least κ partners for the normal playback of one peer, while the upload bandwidth of one peer is capable of supporting a maximum of $\frac{\bar{u}\kappa}{r}$ connections.

Replication Strategy The replication strategy implies how the video segments are replicated in the local storage of the peers after viewing process. In this paper, we will analyze one of the most popular cases, the least recently utilized (LRU) strategy, which is also the original choice of PPlive [7].

From the Fig. 2 we can see that when the user flows $\Delta N_{2,1}(t)$ transfer from $N_2(t)$ to $N_1(t)$, they continue to contribute their upload bandwidth for videos in $G_2(t)$ because there still exists replicas for videos of $G_2(t)$ in their local storage. Furthermore, according to [11], the system performance is indifferent to whether peers are homogeneous or heterogenous in bandwidth. Assuming an average upload bandwidth $u_i = \bar{u}$ for all the peers, we can have the upload capacity by peers $U(t)$ divided into 4 components as follows:

$$
\begin{aligned}
&U(t) \qquad (8)\\
&= \bar{u}\sum_{j=1}^{M}\sum_{i=1}^{N}\alpha_{i,j}(t)\beta_{i,j}(t)\\
&= \bar{u}(\sum_{j=1}^{K}\sum_{i=1}^{N}\alpha_{i,j}(t)\beta_{i,j}(t) + \sum_{j=K+1}^{M}\sum_{i=1}^{N}\alpha_{i,j}(t)\beta_{i,j}(t))\\
&= \bar{u}(\sum_{j=1}^{K}\sum_{i=1}^{N_1(t)}\alpha_{i,j}(t)\beta_{i,j}(t) + \sum_{j=1}^{K}\sum_{i=1}^{N_2(t)}\alpha_{i,j}(t)\beta_{i,j}(t) +\\
&\sum_{j=K+1}^{M}\sum_{i=1}^{N_1(t)}\alpha_{i,j}(t)\beta_{i,j}(t) + \sum_{j=K+1}^{M}\sum_{i=1}^{N_2(t)}\alpha_{i,j}(t)\beta_{i,j}(t))
\end{aligned}
$$

Generally the scheduling strategy and the replication strategy are uniform to all the peers in the distributed VoD streaming system. They collaborate to organize the dispersed upload bandwidth of peers in a distributed way. Thus, we can distinguish the peer upload according to the replication map $\alpha_{i,j}(t)$ of peers, namely $R_{1,1}(t)$, $R_{1,2}(t)$, $R_{2,1}(t)$ and $R_{2,2}(t)$ respectively. In which, $R_{a,b}(t)$ represents the replicas for videos in $G_a(t), a \in 1,2$ from peers in $N_b(t), b \in 1,2$. e.g. The $R_{1,2}(t)$ implies that the peers have joint the viewing group N_2, and in their local storage there still exists the replicas for the upload of videos in G_1.

To further analyze the replication evolution, we define two important concepts, namely the eviction ratio and the upload ratio as follows:

Definition 1: Let $\varepsilon_1(t)$ be the *eviction ratio* of the replicas for the videos in G_1 to be evicted by the replication strategy in the time slot t, and $\varepsilon_2(t)$ relates to the replicas for G_2. Since we assume that there are only two types of videos $\varepsilon_1(t)+\varepsilon_2(t) = 1$, we can also consider $\varepsilon_2(t)$ as the *reservation probability* for the replicas of G_1 to reside in the local storage and $\varepsilon_1(t)$ as the *reservation probability* for the replicas of G_2 to reside in the local storage.

Definition 2: Given the specific $R_1(t)$ and $R_2(t)$ in the whole system, we define the *upload ratio* $\eta_1(t)$ as a variable between 0 and 1 representing the upload bandwidth utilization for the videos of G_1, with $\eta_2(t)$ for G_2 accordingly.

Given that it is an closed queuing system, we assume that there are no new user flows and the local storage of each peer is fully cached with replicas for G_1 or G_2 [3]. From the Fig. 2, we can see that the replication evolution is mainly determined by two factors, namely, the eviction ratio (i.e. $\varepsilon_1(t)$ or $\varepsilon_2(t)$) and the viewing peers flows (i.e. $\Delta N_{1,2}(t)$ or $\Delta N_{2,1}(t)$). The former determines how many replicas should be replaced by the viewing videos through the replication strategy. The latter one refers to the peer flows exchange between the two viewing groups $N_1(t)$ and $N_2(t)$. Then we can specify the replication evolution process of the four parts as follows:

$$
\begin{cases}
R_{2,1}(t) = R'_{2,1}\varepsilon'_1 + \Delta N'_{2,1}C\\
R_{1,1}(t) = R'_{1,1}\varepsilon'_2 + N'_1 - \Delta N'_{1,2}C\\
R_{1,2}(t) = R'_{1,2}\varepsilon'_2 + \Delta N'_{1,2}C\\
R_{2,2}(t) = R'_{2,2}\varepsilon'_1 + N_2(t) - \Delta N'_{2,1}C
\end{cases} \qquad (9)
$$

where $\varepsilon(t)$ is the reservation probability for the replicas to reside in the local storage during time slot t. The replica $R_{2,1}$ is gradually replaced by $R_{1,1}$ when $N_1(t)$ peers are watching G_1, and the replica $R_{1,2}$ is gradually replaced by $R_{2,2}$ during $N_2(t)$ peers are watching G_2. The current replication for the videos of G_1 or G_2 in the system are as follows:

$$
\begin{cases}
R_1(t) = R_{1,1}(t) + R_{1,2}(t) = R'_1\varepsilon'_2 + N_1(t)\\
R_2(t) = R_{2,1}(t) + R_{2,2}(t) = R'_2\varepsilon'_1 + N_2(t)
\end{cases} \qquad (10)
$$

Furthermore, we have the peers' upload for G_1 and G_2 as follows:

$$
\begin{cases}
U_1(t) = \eta_1(R_1(t), R_2(t))N(t)\bar{u}\\
U_2(t) = \eta_2(R_1(t), R_2(t))N(t)\bar{u}
\end{cases} \qquad (11)
$$

Combining with user demand in Eq. (5), we can have the server support for G_1 and G_2 as:

$$
\begin{cases}
S_1(t) = rN_1(t) - \eta_1(R_1(t), R_2(t))N(t)\bar{u}\\
S_2(t) = rN_2(t) - \eta_2(R_1(t), R_2(t))N(t)\bar{u}
\end{cases} \qquad (12)
$$

We are more interested in the asymptotic collective behavior rather than the individual peer behavior. Instead of the single peer replication map $\alpha_{i,j}(t)$ and the upload scheduling map $\beta_{i,j}(t)$, the upload ratio $\eta(t)$ and the eviction ratio $\varepsilon(t)$ are introduced to analyze the change process of peer upload ability. We can estimate how much upload bandwidth resource from local replicas of peers can be utilized. The central server does not have to frequently query about the local replication map $\alpha_{i,j}(t)$ of each peer, while the global information of peer upload capacity can be evaluated if the video popularity is given. Therefore, once the sharp popularity decay results in the great amount of redundancy replicas in the local storage of peers, the extra provisioned server resource can be prepared for the normal playback of peers.

IV. Implementation and Discussion

A. Server Support Provisioning

In our mechanism of Fig. 3, all the video requests are sent to the central server from the distributed peers. The popularity information of each video in the system can be recorded by the central server. Without considering the new user flow, the viewer population change (i.e. ΔN_1 and ΔN_2) of the two groups of videos becomes the original cause to generate the server provision. On one hand, it directly leads to the change of the demand structure in the system. On the other hand, it leads to the change of video popularity in the two video groups by Eq. (7), which is also related to the sizes of the two groups (e.g. K or $M-K$). Further, the upload ratio $\eta(t)$ and the eviction ratio $\varepsilon(t)$ are influenced by the time-varying video popularity (i.e. $\rho_1(t)$ or $\rho_2(t)$) and the local storage capacity C. The replication evolution $R_1(t)$ and $R_2(t)$ are then

[3]To characterize the in sequence viewing pattern, we assume that the replicas of $\Delta N_{2,1}(t)$ or $\Delta N_{1,2}(t)$ are completely consisted of videos in G_2 or G_1. It makes sense that in reality the new joining users $\Delta N_{2,1}(t)$ from $N_2(t)$ have not watched the videos in G_1 ever before, as videos in G_1 are just released. Since the users continue to view the videos in G_1 one by one and $C \ll K$, it still makes sense that there only exist replicas for G_1 in the local storage of $\Delta N_{1,2}(t)$ who are leaving viewing group $N_1(t)$.

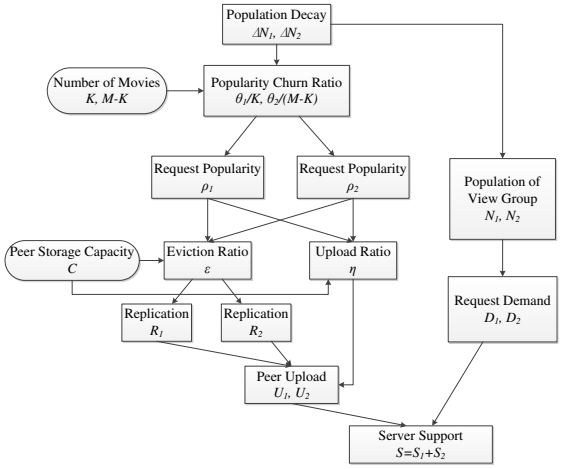


Fig. 3: Map of Process and Parameters

calculated by Eq. (10), and the upload evolution $U_1(t)$ and $U_2(t)$ are captured by Eq. (11). Finally, the server provision can be acquired by Eq. (12).

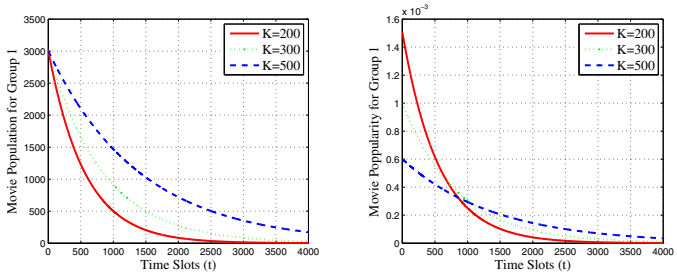
B. Practical Consideration

In the realistic implementation, the actual server support is always affected by the unpredictive user behavior, which is accompanied with the change of video popularity. Consider that one peer turns from online to offline. All replicas in its local storage are no longer available for other peers. It breaks out the established connections with these peers, who have to search for the new partners from the server. Before the new connections are rearranged for these peers, the extra server service should support them with the normal playback. As this peer becomes online active again, the local replicas are available for others to request. However, it is very common that the popularity of the replication has already dropped quickly after the several-days offline period, and seldom peers are still interested to the replicas of this peer. Therefore, the upload bandwidth of this peer can not be fully utilized until the replication is updated in the following viewing process.

And also the different viewing process will result in the distinct replication maps of the users, which further lead to the various upload ability. Consider 10 videos, with 5 popular ones and 5 unpopular ones. Peer A initially views the popular videos in sequential and then views the rest unpopular ones together. Oppositely, peer B watches the videos randomly. The replication maps would be generated differently. Generally, the type B behavior would be considered as the majority in the system. However, if a group of popular videos (e.g. TV drama series) are released together, the impact of type A viewing behavior could not be neglected.

V. Numerical Result And Insight

In this section, we will fist simulate the population decay environment. Then we further extend the analysis about the replication ratio, the upload ratio, and the server provision in the population decay environment.



(a) Viewer Population $N_1(t)$ (b) Video Popularity $\rho_1(t)$

Fig. 4: Time-varying Viewer Population and Video Popularity

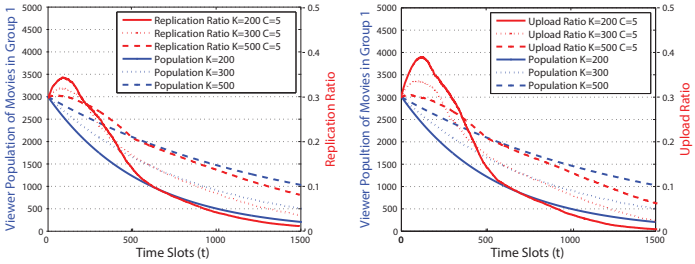
A. Time-varying Popularity

Like [10], we assume 10000 online peers and 2000 videos in the system. Initially, there are a maximum of 3000 online users (30%), watching K videos in G_1, and there are 7000 online users (70%) viewing $M_2 = 2000 - K$ videos in G_2. Considering that a user usually does not return to the video which has just been watched, the popularity of the videos in G_1 decreases over time as more users have completed watching the videos. This "fetch-at-most-once" behavior of users is very prevalent on current measurement of P2P system.

Fig. 4(a) shows the viewer population decay of G_1 and the popularity decay for each videos in G_1. We can see that both the video population and popularity decreases dramatically as the leaving rate is greater than the arrival rate. In our experiment, we keep the initial viewing population $N_1(t_0)$ as a constant, and change the number of videos in G_1 from 200 (K_1), 300 (K_2), to 500 (K_3). When $K_1 = 200$, the popularity of the videos experiences a very fast decreasing with the highest initial popularity 1.5×10^{-3}. As the number of videos K increases, the popularity will decrease more slowly, and the population tends to be relatively steady. It implies that the video group with the smaller size tend to experience a more dramatic popularity decaying process. With the same peak viewer population, they will suffer from higher density of joint request, and the users will leave fast after they complete the viewing.

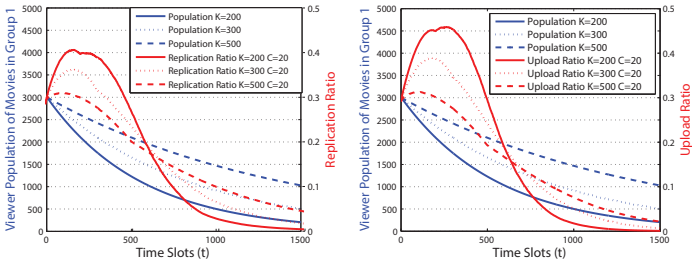
B. Upload Ratio and Replication Ratio

We examine the evolution process of the replication ratio and the upload ratio with different storage capacities ($C = 5$, $C = 20$) in the population decay environment mentioned above. From Fig. 4, we can see that, before the video reaches the average popularity, the replication ratio and the upload ratio still experience a short period increase, even though the video popularity and population have decreased already. Comparing with the scenarios $K = 300$ and $K = 500$, the high initial video popularity when $K = 200$ enables the replication aggregates greater than the viewing population fraction (given the total online user number is 10000), especially when the storage capacity is increased as $C = 20$. The server provision, which compensates the gap between upload ratio and the viewing population fraction, reaches the peak at about time slot 200. When the video popularity decreases to the average popularity at about time slot 500, both the replication ratio and



(a) Replication ratio when $C = 5$

(b) Upload ratio when $C = 5$



(c) Replication ratio when $C = 20$

(d) Upload ratio when $C = 20$

Fig. 5: Replication ratio and upload ratio evolution in the multiple peers scenario with time varying popularity

the upload ratio approach to the fraction of viewing population well. As the viewer population keeps decreasing, the videos in G_1 tend to be more unpopular comparing to the rest videos in the system. We can see the that both the replication ratio and the upload ratio experience an excessive decrease over the population change, especially when the local storage capacity of the peers is increased.

VI. Related Work

The benefits of utilizing peer-contributed resources to reduce server load have been verified for a long time through measurement studies [7]. Although it is estimated that 95% of the server bandwidth consumption can be saved through a P2P approach [7], the resource imbalance problem often breaks such an optimistic expectation. When the video popularity keeps steady, it can be considered as a static scenario. Great amount of existing models of caching strategy relied on the assumption of static user behavior [11]. Our work differentiates itself from these studies as we focus on the replication and upload evolution process of peers in the system rather than the stationary performance.

As argued by [11], the real world systems always experience a non-stationary video popularity churn. The server load is inevitable, since the peer local caching strategy can hardly respond fast to the temporarily sharping demand increase (e.g. flash crowd). The cloud-based server assisted strategy becomes a potential solution to the problem due to its flexibility to leveraging the elastic resource provisioning from the data center. Wang et al. [2] presented CALMS (Cloud-Assisted Live Media Streaming) to lease and adjust cloud server resources in a fine granularity to accommodate temporal and spatial dynamics of demands from online users. In [9], a hybrid cloud-assisted strategy was proposed through partial migration of VoD services to content clouds to deal with peak load of user demand. A predictive resource auto-scaling system was further developed that dynamically reserve the minimum bandwidth resources from multiple data centers for the VoD providers to match its short-term demand projections [4]. However, most of the former cloud based services are developed to accommodate the sharply increasing requests. To the best of our knowledge, modeling P2P VdD systems under drastic popularity decays has yet to be closely investigated.

VII. Conclusion

In this paper, we developed a mathematical model to trace the evolution of peer upload and replication during the population decays. Our model captured peer behaviors with common data replication and scheduling strategies in state-of-the-art peer-to-peer VoD systems. It reveals that, during a sharp population decay, the peers local storage is not effectively utilized for upload, and the imperfect content replication with slow response inevitably results in an escalating server load. The model also facilitated the design of a flexible server provision strategy to serve highly time-varying demands.

Acknowledgment

This work is supported in part by a Canada NSERC Discovery Grant and a China NSFC Major Program of International Cooperation Grant (61120106008).

References

[1] J. Liu, S. G. Rao, B. Li, and H. Zhang, "Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast," Proceedings of the IEEE, 96(1):11-24, 2008.

[2] F. Wang, J. Liu, and M. Chen, "CALMS: Migration towards Cloud-Assisted Live Media Streaming," In Proc. IEEE INFOCOM, 2012.

[3] B. Li, G. Y. Keung, S. Xie, F. Liu, Y. Sun, and H. Yin, "An Empirical Study of Flash Crowd Dynamics in a P2P-Based Live Video Streaming System," In Proc. IEEE GLOBECOM, 2008.

[4] D. Niu, H. Xu, B. Li, and S. Zhao, "Quality-Assured Cloud Bandwidth Auto-Scaling for Video-on-Demand Applications," In Proc. IEEE INFOCOM, 2012.

[5] H. Yin, X. Liu, T. Zhan, V. Sekar, F. Qiu, C. Lin, H. Zhang, and B. Li, "Design and Deployment of a Hybrid CDN-P2P System for Live Video Streaming: Experiences with LiveSky," In Proc. ACM Multimedia, 2009.

[6] D. Wu, Y. Liu, and K. W. Ross, "Queuing Network Models for Multi-Channel P2P Live Streaming Systems," In Proc. IEEE INFOCOM, 2009.

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

[8] F. Figueiredo, F. Benevenuto, and J. Almeida, "The Tube over Time: Characterizing Popularity Growth of YouTube Videos," In Proc. ACM WSDM, 2011.

[9] H. Li, L. Zhong, J. Liu, B. Li, and K. Xu, "Cost-Effective Partial Migration of VoD Services to Content Clouds," In Proc. IEEE CLOUD, 2011.

[10] F. Liu, B. Li, L. Zhong, B. Li, H. Jin, and X. Liao, "Flash Crowd in P2P Live Streaming Systems: Fundamental Characteristics and Design Implications," IEEE Transactions on Parallel and Distributed Systems, 23(7):1227-1239, 2012.

[11] Y. Zhou, T. Fu and DM. Chiu, "Statistical Modeling and Analysis of P2P Replication to Support VoD Service," In Proc. IEEE INFOCOM, 2011.