Dynamic Scaling of VoD Services into Hybrid Clouds with Cost Minimization and QoS Guarantee

Xuanjia Qiu*, Hongxing Li*, Chuan Wu*, Zongpeng Li† and Francis C.M. Lau*

*Department of Computer Science, The University of Hong Kong, Hong Kong, {xjqiu,hxli,cwu,fcmlau}@cs.hku.hk
†Department of Computer Science, University of Calgary, Canada, zongpeng@ucalgary.ca

***Abstract*—A large-scale video-on-demand (VoD) service demands huge server costs, to provision thousands of videos to millions of users with high streaming quality. As compared to the traditional practice of relying on large on-premise server clusters, the emerging platforms of geo-distributed public clouds promise a more economic solution: their on-demand resource provisioning can constitute ideal supplements of resources from on-premise servers, and effectively support dynamic scaling of the VoD service at different times. Promising though it is, significant technical challenges persist before it turns into reality: how shall the service provider dynamically replicate videos and dispatch user requests over the hybrid platform, such that the service quality and the minimization of overall cost can be guaranteed over the long run of the system? In this paper, we present a dynamic algorithm that optimally makes decisions on video replication and user request dispatching in a hybrid cloud of on-premise servers and geo-distributed cloud data centers, based on the Lyapunov optimization framework. We rigorously prove that this algorithm can nicely bound the streaming delays within the preset QoS target in cases of arbitrary request arrival patterns, and guarantee that the overall cost is within a small constant gap from the optimum achieved by a T-slot lookahead mechanism with known information into the future. We evaluate our algorithm with extensive simulations under realistic settings, and demonstrate that cost minimization and smooth playback can be achieved in cases of volatile user demands.**

I. Introduction

Video-on-demand (VoD) services have prospered on the Internet over the past decade. The prevailing large-scale VoD systems, which provide thousands of high-quality videos to millions of users, are based on either a server-client design [1][2] or the peer-to-peer paradigm [3][4]. In both cases, large server clusters are inevitable to provision all the storage and upload capacities (server-client), or an indispensable part to supplement insufficient peer supplies (peer-to-peer) [5]. The server capacities are typically implemented by dedicated servers of the VoD service provider or rented capacities from a content distribution network (CDN). In all these cases, large server costs are incurred.

The recent advance of cloud computing technologies has enabled rapid, on-demand server utility provisioning at much reduced costs. Especially, global-scale cloud platforms, such as Amazon CloudFront and Google App Engine, span multiple data centers in different geographic locations, and provide services close to the users. Such a geo-distributed cloud promises to be suitable for VoD applications with large user groups in different regions and volatile resource demands over time.

To utilize cloud resources for VoD service provisioning, videos can be replicated in storage servers in the cloud, and requests can be distributed to cloud-based web services, while the VoD provider maintains the original copies of the videos and serves some requests using its existing on-premise servers. The key challenges to deploy a VoD service, on the hybrid infrastructure consisting of on-premise servers and geo-distributed cloud data centers, are how to efficiently replicate videos and dispatch requests across the hybrid cloud, for the modest operational expenditure and good streaming delay guarantee at all times.

A few recent proposals have advocated migrating VoD services into a cloud platform. Li *et al.* [6] propose partial migration of VoD services to content clouds for cost saving, and design heuristic strategies to decide the update of cloud contents. Wu *et al.* [7] deploy a VoD application on an IaaS cloud containing a single data center. None of these work considers a hybrid geo-distributed cloud, and their migration algorithms do not provide guarantee of cost optimality over a long run of the system. Some previous studies on peer-to-peer based streaming systems also explore good content placement strategies and request routing policies [8][9], where it is assumed that resource of users is contributed for free. In contrast, we propose a comprehensive cost minimization framework to carefully tune the occupation amount of various types of resource. Placement of services to different sites has been investigated [10][11] based on the theories of Facility Location Problems [12], that focus on one-time optimization with fixed service demands, rather than online optimization over a long run of the system.

In this paper, we design a dynamic, joint video replication and request distribution algorithm, which minimizes overall operational cost of the VoD system while guaranteeing bounded streaming delays over time, based on the Lyapunov optimization framework [13][14]. Lyapunov optimization provides a framework for designing efficient algorithms that achieve arbitrarily close to optimal system performance over the long run, without a need for any information from the future. It has been extensively used in routing and channel allocation in wireless networks [13][15] and has recently been utilized for resource allocation in peer-to-peer networks [16] and CDNs [17]. We tailor Lyapunov optimization techniques in the hybrid cloud computing setting, to dynamically and jointly resolve the optimal video replication and request dispatching problems. Through rigorous analysis, we prove the optimality of our algorithm, showing that the overall operational cost is guaranteed to be within a small constant gap from the optimum achieved by a T-slot lookahead mechanism with information from the future. We also prove that our algorithm bounds the streaming delays within the preset quality of service (QoS) target in cases of arbitrary request arrivals. Guidelines on how to trade the operational cost for QoS and vice versa are also discussed. With extensive simulations, we show that our algorithm can achieve cost minimization and playback smoothness in large-scale VoD systems.

In the remainder of this paper, we formulate the problem in Sec. II, design a dynamic algorithm in Sec. III, analyze its optimality in Sec. IV, evaluate its performance in Sec. V, and conclude the paper in Sec. VI.

II. Problem Formulation

A. System Model

We study a VoD system that streams a collection of videos $\mathcal{V}$ to users in multiple geographical regions $\mathcal{Z}$. Each video v in $\mathcal{V}$ is divided to a set of media chunks $K^{(v)}$, each of size $b^{(v)}$ (in bytes). An on-premise server cluster is deployed by the VoD service provider, both as access portals to the VoD service and as the repository of the original video files. The maximum number of requests the on-premise servers can serve in a time slot is f.

There is a public cloud platform, containing multiple geo-distributed cloud data centers $\mathcal{C}$. There are two types of servers in each cloud data center, namely the back-end storage servers for storage and the front-end web-service servers to serve user requests. Let r_z denote the round-trip delay between region $z \in \mathcal{Z}$ and the on-premise server cluster, and e_{zc} be the round-trip delay between region z and data center $c \in \mathcal{C}$, reflecting geographic distances between the two regions.

We consider the following service charge model. The cost of uploading a byte from the on-premise servers is h. The charge of storage in data center c is d_c per byte. The cost to upload a byte from data center c is q_c. The cost to copy a byte of data from the on-premise servers to data center c is w_c. Removal of videos from a data center is cost free. These charges follow the typical charging models of leading commercial cloud providers such as Amazon EC2 [18] and S3[19].

B. Video Replication and Request Dispatching Problem

In our system, time is divided into equal-length time slots, numbered as $0, 1, 2, \ldots$. Each time slot is one unit time, which is enough for uploading any chunk of video $v \in \mathcal{V}$ of size $b^{(v)}$ at the unit bandwidth. In time slot t, $a_z^{(v,k)}(t)$ requests are generated for downloading chunk $k \in K^{(v)}$ of video $v \in \mathcal{V}$, from users residing in region z. We assume that the request generation is an arbitrary process over time, with A_{max} being the maximum number of requests arising from each region for all chunks of a video in each time slot.

A *control center* is responsible for collecting user requests, buffering them in request queues, and then dispatching them into the hybrid infrastructure of on-premise servers and cloud data centers. It also decides whether a video is to be replicated or removed from a cloud data center. Let $Q_z^{(v)}$ denote the request queue caching requests for chunks of video v from users in region z, $\forall z \in \mathcal{Z}, \forall v \in \mathcal{V}$, whose length (*i.e.*, the queue backlog) at time slot t is denoted by $Q_z^{(v)}(t)$.

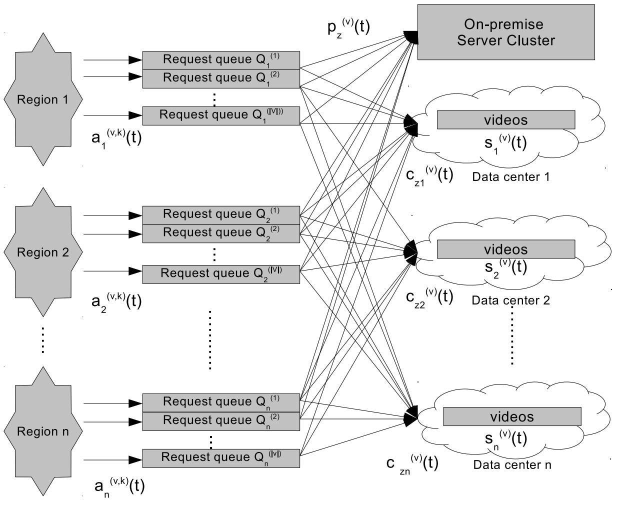

Fig. 1. The system model.

The decisions that the control center needs to make in each time slot of the dynamic system include: (1) Whether video v should be stored in data center c in time slot t or not, as indicated by binary decision variable $s_c^{(v)}(t)$, 1 for 'yes' and 0 for 'no', $\forall c \in \mathcal{C}, v \in \mathcal{V}$. (2) How many requests for chunks of video v from region z should be dispatched to the on-premise server cluster and how many to each data center c in time slot t, denoted by variables $p_z^{(v)}(t)$ and $c_{zc}^{(v)}(t)$, respectively, $\forall z \in \mathcal{Z}, v \in \mathcal{V}, c \in \mathcal{C}$. Note that requests for a chunk can only be dispatched to a data center where the corresponding video is stored, *i.e.*, $c_{zc}^{(v)}(t) > 0$ only if $s_c^{(v)}(t) = 1$.

The backlogs of request queues are updated as follows:

$$Q_z^{(v)}(t+1) = \max[Q_z^{(v)}(t) - p_z^{(v)}(t) - \sum_{c\in\mathcal{C}} c_{zc}^{(v)}(t), 0] + \sum_{k\in K^{(v)}} a_z^{(v,k)}(t) \quad \forall z \in \mathcal{Z}, \forall v \in \mathcal{V}, \forall t. \quad (1)$$

The system model is illustrated in Fig. 1. Important notations are summarized in Table I, for ease of reference.

Our objective is to design a dynamic algorithm for the control center to optimize video replication and request dispatching over time, such that the overall operational cost is minimized while the service quality is guaranteed. The operational cost in time slot t, $C(t)$, is modeled as follows:

$$\begin{aligned} C(t) \quad = \quad & \sum_{v\in\mathcal{V}}\sum_{z\in\mathcal{Z}}[(b^{(v)}p_z^{(v)}(t)h + \sum_{c\in\mathcal{C}} b^{(v)}c_{zc}^{(v)}(t)q_c] + \sum_{v\in\mathcal{V}}\sum_{c\in\mathcal{C}} b^{(v)}s_c^{(v)}(t)d_c \\ & + \sum_{v\in\mathcal{V}}\sum_{c\in\mathcal{C}} |K^{(v)}|b^{(v)}[s_c^{(v)}(t) - s_c^{(v)}(t-1)]^+ w_c, \end{aligned} \quad (2)$$

where the three items correspond to (1) the bandwidth charge for uploading chunks to users from the on-premise servers and the cloud data centers, (2) the storage cost for replicated videos at the data centers, (3) the migration cost for copying videos from the on-premise servers to the data centers, respectively. Here $[x]^+ = x$ if $x \geq 0$ and $[x]^+ = 0$ if $x < 0$.

TABLE I
IMPORTANT NOTATIONS

$\mathcal{V}$	Video set	$\mathcal{Z}$	User region set	$\mathcal{C}$	Cloud data center set
$K^{(v)}$	Set of chunks in video v				
$b^{(v)}$	Size of a chunk in video v in bytes				
$a_z^{(v,k)}(t)$	No. of requests for chunk k in video v from region z at time slot t				
$Q_z^{(v)}(t)$	Request queue for chunks of video v from region z				
A_{max}	Max no. of requests for chunks of a video from a region in a time slot.				
$p_z^{(v)}(t)$	No. of requests dispatched from $Q_z^{(v)}$ to on-premise servers at t				
$c_{zc}^{(v)}(t)$	No. of requests dispatched from $Q_z^{(v)}$ to data center c at t				
$s_c^{(v)}(t)$	Binary var: store video v on data center c at t (1) or not (0).				
f	Max. no. of requests on-premise servers can serve in a time slot				
μ_{max}	Max no. of requests dispatched from each request queue to a data center in a time slot				
d_c	Charge for storing a byte on data center c for one time slot				
q_c	Charge for uploading a byte from data center c				
h	Cost for uploading a byte from the on-premise servers				
w_c	Charge for copying a byte from on-premise servers to data center c.				
$W_z^{(v)}$	Upper bound of queueing delay of requests in queue $Q_z^{(v)}$				
$\epsilon_z^{(v)}$	Preset constant for controlling queueing delay in $Q_z^{(v)}$				
r_z	Round-trip delay between region z and on-premise server cluster				
e_{zc}	Round-trip delay between region z and data center c				
α	Upper bound of average round-trip delay				
R	Virtual queue for bounding average round-trip delay				
$D_z^{(v)}$	Virtual queue for bounding the queueing delay in $Q_z^{(v)}$				

On the other hand, the quality of a VoD service is evaluated by the streaming delays, that mainly consists of queueing delay in the respective request queue, and round-trip delay from when the request is dispatched from the queue to the time the first byte of the chunk is received. It is closely related to playback smoothness of a video in our system. We ignore processing delays inside a data center, due to the high interconnection bandwidth and CPU capacities inside a data center. Let α be the upper-bound of average round-trip delay per request, as set by the VoD service provider. We reasonably assume that for any region z, there exists a data center c' such that $\alpha > e_{zc'}$, *i.e.*, this bound should be at least larger than the round-trip delay between a user and its closest data center.

The optimization pursued by our dynamic algorithm is formulated as follows :

$$\min \overline{C(t)} \tag{3}$$

subject to:

$$\sum_{z\in\mathcal{Z}}\sum_{v\in\mathcal{V}} p_z^{(v)}(t) \le f, \forall t, \tag{4}$$

$$0 \le c_{zc}^{(v)}(t) \le \mu_{max} s_c^{(v)}(t), \forall z \in \mathcal{Z}, c \in \mathcal{C}, v \in \mathcal{V}, \forall t, \tag{5}$$

$$p_z^{(v)}(t) + \sum_{c\in\mathcal{C}} c_{zc}^{(v)}(t) \le Q_z^{(v)}(t), \forall z \in \mathcal{Z}, \forall v \in \mathcal{V}, \forall t, \tag{6}$$

$$\overline{\sum_{z\in\mathcal{Z}}\sum_{v\in\mathcal{V}}(p_z^{(v)}(t) r_z + \sum_{c\in\mathcal{C}} c_{zc}^{(v)}(t) e_{zc})} < \alpha \overline{\sum_{z\in\mathcal{Z}}\sum_{v\in\mathcal{V}}(p_z^{(v)}(t) + \sum_{c\in\mathcal{C}} c_{zc}^{(v)}(t))}, \tag{7}$$

$$p_z^{(v)}(t) \ge 0, \forall z \in \mathcal{Z}, \forall v \in \mathcal{V}, \forall t, \tag{8}$$

$$s_c^{(v)}(t) \in \{0,1\}, \forall c \in \mathcal{C}, \forall v \in \mathcal{V}, \forall t, \tag{9}$$

where $\overline{x(t)} = \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} x(t)$ represents the time-averaged value of $x(t)$. (4) are upload bandwidth constraints at the on-premise server cluster. Recall that each request is served by a unit bandwidth. (5) states that requests for a chunk are only dispatched to data centers storing the corresponding video at the time, and the number of requests dispatched from a request queue to a data center in a time slot stays within an upper bound μ_{max}. (6) states that the total number of requests dispatched from queue $Q_z^{(v)}$ cannot be larger than the current queue size. (7) guarantees that the average round-trip delay per request in the system is smaller than α. Although we only model round-trip delay bound in the constraints, we will show that our dynamic algorithm can guarantee a constant queueing delay bound in each request queue $Q_z^{(v)}$ as well.

We note that no constraints are formulated to represent storage capacity limit in the cloud data centers. This is due to the common observations that a cloud has enough storage capacity to serve more applications than a single VoD service.

III. Dynamic Video Replication and Request Dispatching Algorithm

We next design a dynamic algorithm to solve the optimization problem in (3). Based on Lyapunov optimization theory, the algorithm controls values of the decision variables in each time slot, to guarantee that all constraints in (3) are satisfied while the queueing delays in request queues are bounded.

Constraints (4)(5)(6)(8)(9) can be addressed in each time slot. Constraint (7) is on time-averaged variable values. (7) can be guaranteed and queueing delays can be bounded via virtual queue techniques in Lyapunov optimization theory [14].

A. Bounding Delays

To satisfy constraint (7), we introduce a virtual queue R, which is updated in each time slot as follows:

$$R(t+1) = \max[R(t) + \sum_{z\in\mathcal{Z}}\sum_{v\in\mathcal{V}}(p_z^{(v)}(t) r_z + \sum_{c\in\mathcal{C}} c_{zc}^{(v)}(t) e_{zc}) - \alpha \sum_{z\in\mathcal{Z}}\sum_{v\in\mathcal{V}}(p_z^{(v)}(t) + \sum_{c\in\mathcal{C}} c_{zc}^{(v)}(t)), 0], \forall t, \tag{10}$$

where the arrival rate $\sum_{z\in\mathcal{Z}}\sum_{v\in\mathcal{V}}(p_z^{(v)}(t) r_z + \sum_{c\in\mathcal{C}} c_{zc}^{(v)}(t) e_{zc})$ corresponds to the overall round-trip delay experienced by all requests in t, and departure rate $\alpha \sum_{z\in\mathcal{Z}}\sum_{v\in\mathcal{V}}(p_z^{(v)}(t) + \sum_{v\in\mathcal{V}} c_{zc}^{(v)}(t))$ is the product of the total number of requests in t and the pre-set upper bound of round-trip delay per request. Our algorithm should adjust $p_z^{(v)}(t)$'s and $c_{zc}^{(v)}(t)$'s to make sure that R is stable (*i.e.*, the length of this virtual queue will not grow unbounded). Then time-averaged arrival rate would not exceed time-averaged departure rate, and hence inequality (7) is guaranteed.

To bound queueing delays in the request queues $Q_z^{(v)}, \forall z \in \mathcal{Z}, v \in \mathcal{V}$, we apply the technique of *ϵ-persistent service* queue [20], and guarantee the stability of the following virtual queue $D_z^{(v)}$ associated with each request queue $Q_z^{(v)}$:

$$D_z^{(v)}(t+1) = \max[D_z^{(v)}(t) + 1_{\{Q_z^{(v)}(t)>0\}}(\epsilon_z^{(v)} - p_z^{(v)}(t) - \sum_{c\in\mathcal{C}} c_{zc}^{(v)}(t)) - 1_{\{Q_z^{(v)}(t)=0\}}\mu_{max}, 0], \forall v \in \mathcal{V}, \forall z \in \mathcal{Z}, \forall t, \quad (11)$$

where $\epsilon_z^{(v)} > 0$ is a constant that can be gauged to control the queueing delay bound. Intuitively, when the request queue $Q_z^{(v)}$ is not empty, $\epsilon_z^{(v)}$ is added to the virtual queue $D_z^{(v)}$ (arrivals), and the departure rate of the virtual queue, $p_z^{(v)}(t) - \sum_{c\in\mathcal{C}} c_{zc}^{(v)}(t)$, is the same as the departure rate from its corresponding request queue. If the request queue is empty, the length of the virtual queue decreases by μ_{max}. $p_z^{(v)}(t)$'s and $c_{zc}^{(v)}(t)$'s are strategically decided to keep the virtual queue stable; in this way, requests are timely dispatched from the request queue, resulting in limited queueing delay per request. The value of $\epsilon_z^{(v)}$ renders a tradeoff between the queueing delay bound and the optimality of operational cost achieved by our algorithm, which will be analyzed in Sec. IV.

B. Designing Dynamic Algorithm

Next we design a dynamic algorithm which stabilizes all kinds of queues modeled above, and solves optimization (3).

Let $\mathbf{\Theta}(t) = [\mathbf{Q}(t), \mathbf{R}(t), \mathbf{D}(t)]$ be the vector of all queues in the system. Define our Lyapunov function as

$$L(\mathbf{\Theta}(t)) = \frac{1}{2}\sum_{v\in\mathcal{V}}\sum_{z\in\mathcal{Z}}[(Q_z^{(v)}(t))^2 + (D_z^{(v)}(t))^2] + \frac{1}{2}(R(t))^2. \quad (12)$$

The one-slot conditional Lyapunov drift is

$$\Delta(\mathbf{\Theta}(t)) = \mathbb{E}\{L(\mathbf{\Theta}(t+1)) - L(\mathbf{\Theta}(t))|\mathbf{\Theta}(t)\}.$$

According to the *drift-plus-penalty* framework in Lyapunov optimization theory (Chapter 5 in [14]), simultaneously minimizing the upper bound of the "penalty" (*i.e.*, the time-averaged operational cost in (3) in our case) and stabilizing queues can be achieved by minimizing the upper bound of the following term in each time slot:

$$\Delta(\mathbf{\Theta}(t)) + VC(t),$$

where V is a non-negative parameter chosen by the VoD service provider, denoting a tradeoff between the operational cost and the streaming delays, which will be discussed in Sec. IV. We can derive the following inequality (detailed steps are included in our technical report [21]):

$$\begin{aligned}
&\Delta(\mathbf{\Theta}(t)) + VC(t)\\
&\le B - \sum_{v\in\mathcal{V}}\sum_{z\in\mathcal{Z}} p_z^{(v)}(t)[Q_z^{(v)}(t) + (\alpha - r_z)R(t)\\
&\quad +1_{\{Q_z^{(v)}(t)>0\}}D_z^{(v)}(t) - b^{(v)}Vh] - \sum_{v\in\mathcal{V}}\sum_{z\in\mathcal{Z}}\sum_{c\in\mathcal{C}} c_{zc}^{(v)}(t)\\
&\quad\quad [Q_z^{(v)}(t) + (\alpha - e_{zc})R(t) + 1_{\{Q_z^{(v)}(t)>0\}}D_z^{(v)}(t) - Vb^{(v)}q_c]\\
&\quad +V\sum_{v\in\mathcal{V}}\sum_{c\in\mathcal{C}} b^{(v)}[s_c^{(v)}(t)d_c + [s_c^{(v)}(t) - s_c^{(v)}(t-1)]^+|K^{(v)}|w_c]\\
&\quad + \sum_{v\in\mathcal{V}}\sum_{z\in\mathcal{Z}} Q_z^{(v)}(t)\sum_{k\in\mathcal{K}} a_z^{(v,k)}(t)\\
&\quad + \sum_{v\in\mathcal{V}}\sum_{z\in\mathcal{Z}} D_z^{(v)}(t)[1_{\{Q_z^{(v)}(t)>0\}}\epsilon_z^{(v)} - 1_{\{Q_z^{(v)}=0\}}\mu_{max}],
\end{aligned} \quad (13)$$

where

$$\begin{aligned}
B \;=\; &\frac{1}{2}|\mathcal{V}||\mathcal{Z}|[A_{max}^2 + \epsilon_{max}^2 + 2(f + |\mathcal{C}|\mu_{max})^2]\\
&+\frac{1}{2}(|\mathcal{V}||\mathcal{Z}||\mathcal{C}|\mu_{max}e_{max} + fr_{max})^2 + \frac{1}{2}\alpha^2(|\mathcal{V}||\mathcal{Z}||\mathcal{C}|\mu_{max} + f)^2
\end{aligned}$$

is a constant, $r_{max} = \max\{r_z|z \in \mathcal{Z}\}, e_{max} = \max\{e_{zc}|z \in \mathcal{Z}, c \in \mathcal{C}\}$, and $\epsilon_{max} = \max\{\epsilon_z^{(v)}|z \in \mathcal{Z}, v \in \mathcal{V}\}$.

By minimizing the right-hand-side of inequality (13), we are able to minimize the upper bound of $\Delta(\mathbf{\Theta}(t)) + VC(t)$, and thus the upper bound of $\overline{C(t)}$.

To do that, we first simplify the notation by defining

$$\begin{aligned}
&\gamma_z^{(v)}(t) = Q_z^{(v)}(t) + 1_{\{Q_z^{(v)}(t)>0\}}D_z^{(v)}(t) - Vb^{(v)}h + (\alpha - r_z)R(t),\\
&\eta_{zc}^{(v)}(t) = Q_z^{(v)}(t) + 1_{\{Q_z^{(v)}(t)>0\}}D_z^{(v)}(t) - Vb^{(v)}q_c + (\alpha - e_{zc})R(t),\\
&\phi_c^{(v)}(t) = Vb^{(v)}(d_c + 1_{\{s_c^{(v)}(t-1)=0\}}|K^{(v)}|w_c),
\end{aligned}$$

all of which are constants in time slot t.

Since $\sum_{v\in\mathcal{V}}\sum_{z\in\mathcal{Z}} Q_z^{(v)}(t)\sum_{k\in\mathcal{K}} a_z^{(v,k)}(t) + \sum_{v\in\mathcal{V}}\sum_{z\in\mathcal{Z}} D_z^{(v)}(t)[1_{\{Q_z^{(v)}(t)>0\}}\epsilon_z^{(v)} - 1_{\{Q_z^{(v)}=0\}}\mu_{max}]$ is constant in each time slot, minimizing the right-hand-side of (13) is equivalent to:

$$\begin{aligned}
\max \quad & F(t) = \sum_{v\in\mathcal{V}}\sum_{z\in\mathcal{Z}} p_z^{(v)}(t)\gamma_z^{(v)}(t) +\\
& \sum_{v\in\mathcal{V}}\sum_{z\in\mathcal{Z}}\sum_{c\in\mathcal{C}} c_{zc}^{(v)}(t)\eta_{zc}^{(v)}(t) - \sum_{v\in\mathcal{V}}\sum_{c\in\mathcal{C}} \phi_c^{(v)}(t)s_c^{(v)}(t) \quad (14)
\end{aligned}$$

subject to: constraints (4) (5) (6) (8) (9).

This problem is an integer linear program, which can be solved by optimization tool kits such as *GLPK* [22].

In summary, the flow of our dynamic algorithm, carried out by the control center, is as follows: The control center maintains a table of video replication information, with entries $s_c^{(v)}, \forall c \in \mathcal{C}, v \in \mathcal{V}$, which are initialized to be 0 at the beginning. In each time slot, it enqueues received requests for chunks of video v originated from region z to request queue $Q_z^{(v)}, \forall z \in \mathcal{Z}, v \in \mathcal{V}$. Virtual queues R and $D_z^{(v)}$'s are maintained simply as counters. By observing the queue states $\mathbf{\Theta}(t)$ and request arrival rates $a_z^{(v,k)}(t), \forall z \in \mathcal{Z}, \forall m \in \mathcal{V}, \forall k \in K^{(v)}$, the control center solves optimization (14) to calculate the optimal video replication strategies $s_c^{(v)}(t), \forall c \in \mathcal{C}, v \in \mathcal{V}$, and request dispatching strategies $p_z^{(v)}(t)$ and $c_{zc}^{(v)}(t), \forall z \in \mathcal{Z}, c \in \mathcal{C}, v \in \mathcal{V}$. The control center then signals the on-premise servers and data centers to replicate video and/or upload chunks accordingly.

IV. Analysis on Cost and Delays

We next analyze the queueing delay bound and the optimality of operational cost achieved by our dynamic algorithm, as well as the tradeoff between operational cost minimization and the delay bound. Due to the space limit, detailed proofs to all theorems are included in our technical report [21].

A. Bound of Queueing Delay

Theorem 1: (Bound of Queue Length) Define

$$Q_z^{(v)^{max}} = Vb^{(v)}(d_{\tilde{c}} + |K^{(v)}|w_{\tilde{c}} + q_{\tilde{c}}) + A_{max}, \qquad (15)$$

where

$$\tilde{c} = \text{argmin}_c\{b^{(v)}(d_c + |K^{(v)}|w_c + q_c)|\alpha - e_{zc} > 0, \forall c \in \mathcal{C}\}. \quad (16)$$

We have $Q_z^{(v)}(t) \le Q_z^{(v)^{max}}$, $\forall z \in \mathcal{Z}, \forall v \in \mathcal{V}, \forall t$.

Theorem 2: (Bounded Queueing Delay): The queueing delay in each request queue $Q_z^{(v)}$, $\forall z \in \mathcal{Z}, \forall v \in \mathcal{V}$, is bounded by

$$W_z^{(v)} = \lceil \frac{Vb^{(v)}(d_{\tilde{c}} + |K^{(v)}|w_{\tilde{c}} + q_{\tilde{c}}) + Q_z^{(v)^{max}}}{\epsilon_z^{(v)}} \rceil,$$

where $\tilde{c}$ is defined in (16).

B. Optimality against the T-Slot Lookahead Mechanism

Since request arrival rates are arbitrary in our system, it is difficult to find the global cost minimum, with which to compare the time-averaged cost achieved by our dynamic algorithm. Therefore, we compare with a local optimum, which is the optimal (objective function) value of a similar cost minimization problem within known information (e.g., request arrivals) for T time slots into the future, *i.e.*, a T-slot lookahead mechanism [14]. In the T-slot lookahead mechanism, time is divided into successive frames, each consisting of T time slots. Denote each frame as $F_l = \{lT, lT+1, \ldots, lT+T-1\}$, where $l = 0, 1, \ldots$. In each time frame, consider the following optimization problem over variables $c_{zc}^{(v)}(t), p_z^{(v)}(t), s_c^{(v)}(t)$, $\forall z \in \mathcal{Z}, \forall v \in \mathcal{V}, \forall c \in \mathcal{C}$, $t \in F_l$:

$$\min \frac{1}{T} \sum_{t=lT}^{lT+T-1} C(t) \qquad (17)$$

subject to:

$$\sum_{v \in \mathcal{V}} \sum_{z \in \mathcal{Z}} p_z^{(v)}(t) \le f, \forall t \in F_l,$$

$$0 \le c_{zc}^{(v)}(t) \le \mu_{max} s_c^{(v)}(t), \forall z \in \mathcal{Z}, c \in \mathcal{C}, v \in \mathcal{V}, t \in F_l,$$

$$p_z^{(v)}(t) + \sum_{c \in \mathcal{C}} c_{zc}^{(v)}(t) \le Q_z^{(v)}(t), \forall v \in \mathcal{V}, z \in \mathcal{Z}, t \in F_l,$$

$$\sum_{t=lT}^{lT+T-1} \sum_{z \in \mathcal{Z}} \sum_{v \in \mathcal{V}} (\sum_{c \in \mathcal{C}} c_{zc}^{(v)}(t) e_{zc} + p_z^{(v)}(t) r_z) < \alpha \sum_{t=lT}^{lT+T-1} \sum_{z \in \mathcal{Z}} \sum_{v \in \mathcal{V}} (\sum_{c \in \mathcal{C}} c_{zc}^{(v)}(t) + p_z^{(v)}(t)),$$

$$\sum_{t=lT}^{lT+T-1} [\sum_{k \in K^{(v)}} a_z^{(v,k)}(t) - p_z^{(v)}(t) - \sum_{c \in \mathcal{C}} c_{zc}^{(v)}(t)] \le 0, \quad \forall z \in \mathcal{Z}, v \in \mathcal{V},$$

$$p_z^{(v)}(t) \ge 0, \forall z \in \mathcal{Z}, v \in \mathcal{V}, t \in F_l,$$

$$s_c^{(v)}(t) \in \{0, 1\}, v \in \mathcal{V}, c \in \mathcal{C}, t \in F_l.$$

We then have the following result.

Theorem 3: (Optimality of Cost Minimization) Let $\widehat{C_l}$ denote the optimal objective function value in the T-slot Lookahead problem (17) in time frame F_l. The minimum operational cost in time slot t derived by our algorithm is $C(t)$. In the first LT time slots, where L is a constant. We have

$$\frac{1}{LT} \sum_{t=0}^{LT-1} C(t) \le \frac{1}{L} \sum_{l=0}^{L-1} \widehat{C_l} + \frac{BT}{V}, \qquad (18)$$

i.e., our algorithm achieves a time-averaged cost within constant gap $\frac{BT}{V}$ from that by assuming full knowledge of request arrivals in the T slots in the future.

C. Tradeoff between Operational Cost and QoS

Theorems 2 and 3 show that when V increases, worst-case queueing delay $W_z^{(v)}$ increases, while the gap between the operational cost achieved by our dynamic algorithm and that of the T-Slot lookahead mechanism shrinks. The pre-set constant $\epsilon_z^{(v)}$ has a similar effect: When $\epsilon_z^{(v)}$ increases, the worst-case queueing delay $W_z^{(v)}$ decreases, but B increases such that the gap to optimality increases.

V. Performance Evaluation

We evaluate our dynamic algorithm with extensive simulations under realistic settings. There are 100 regions and 1000 videos in the VoD system. The length of videos follows a uniform distribution within range $[1800, 3600]$ (seconds). The playback bitrate of each video is 400 Kbps. Each chunk corresponds to 60 seconds of video playback. Users in each region join the VoD system following Poission arrivals; the average inter-arrival times of Poission arrival processes in different regions are uniformly distributed within range $[1.2, 2.4]$ (seconds). User online time follows an exponential distribution with the mean of 30 minutes [23]. The probabilities that a user requests different videos are proportional to the popularity of the videos, which follows a Zipf-like distribution. Each user issues a random-seek VCR command periodically, with inter-command time following an exponential distribution with a 5-minute mean. The on-premise servers can serve at most 1000 requests in one time slot, and charges $\$1.2 \times 10^{-10}$ for uploading one byte.

There are 100 cloud data centers (each in one region). The charges by the cloud service are extracted from real-world settings [19]. The storage cost is $\$2 \times 10^{-13}$ per byte per hour in each data center. The cost of uploading from a data center follows a uniform distribution within range $[\$0.96 \times 10^{-10}, \$1.44 \times 10^{-10}]$ per byte. The cost of copying data from the on-premise servers to a data center is $\$1.2 \times 10^{-10}$ per byte. The maximum number of requests dispatched from a request queue to a data center in each time slot, *i.e.*, μ_{max}, is 24. The round-trip delay between users in a region and the on-premise servers, or between users in a region and a data center, follows a uniform distribution within range $[0.005, 0.025]$ (seconds). Especially, the round-trip delay within the same region is set to $e_{cc} = 0.005$ seconds for all $c \in \mathcal{C}$. The upper bound of round-trip delay is set as $\alpha = 0.015$ seconds. The length of a time slot in running our dynamic algorithm is 10 seconds.

Each $\epsilon_z^{(v)}$ is set proportional to $Vb^{(v)}(d_{\tilde{c}} + |K^{(v)}|w_{\tilde{c}} + q_{\tilde{c}}) + Q_z^{(v)^{max}}$ where $\tilde{c}$ and $Q_z^{(v)^{max}}$ are defined in (16) and (15), respectively. In this way, the queueing delay bounds $W_z^{(v)}$'s of the request queues are similar to each other. The mean of the parameter set $\{\epsilon_z^{(v)} | \forall z \in \mathcal{Z}, \forall v \in \mathcal{V}\}$, denoted by $\bar{\epsilon}$, has a default value of 1. The default value of V is 50000.

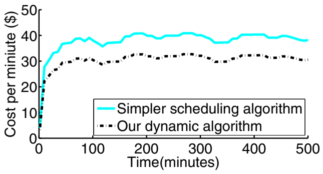

Fig. 2. Comparison of operational cost between our dynamic algorithm and a heuristic scheduling algorithm.

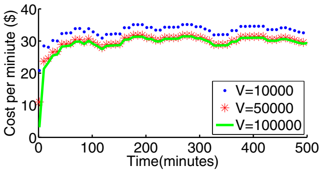

Fig. 3. Operational cost at different values of V.

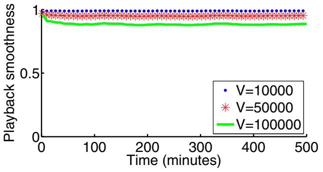

Fig. 4. Playback smoothness at different values of V.

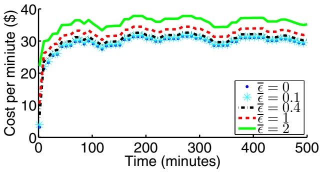

Fig. 5. Operational cost at different values of $\bar{\epsilon}$.

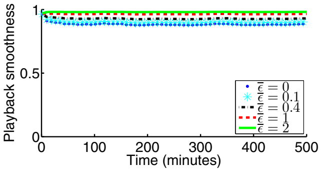

Fig. 6. Playback smoothness at different values of $\bar{\epsilon}$.

A. *Cost Optimality*

We compare our dynamic algorithm with a simpler scheduling algorithm, that processes all requests in the time slot when they arrive without buffering in queues, and decides video replication and request distribution by solving a one-time optimization similar to (3) in this time slot (instead of calculating over-time average of the involved quantities). Fig. 2 shows the costs incurred by both algorithms under the same settings. We observe that our dynamic algorithm outperforms the heuristic algorithm at all times.

B. *Tradeoff between Operational Cost and QoS*

1) Impact of V: Fig. 3 shows that when V increases, the operational cost becomes smaller. In Fig. 4, the playback smoothness is evaluated as the percentage of chunks downloaded before their respective playback deadlines in the entire system. When the streaming delay is smaller, more chunks can be downloaded by the users in time, so that the playback is more smooth. With the increase of V, the average streaming delay per request (queueing delay+round-trip delay) increases, and thus the playback is less smooth. These figures clearly show a tradeoff in V's setting. Setting $V = 50000$ can result in a good trade-off between cost optimality and the QoS.

2) Impact of $\epsilon_z^{(v)}$: In Fig. 5 and Fig. 6, we vary the value of $\bar{\epsilon}$, and observe that when its value increases, the operational cost increases while the playback smoothness improves. Therefore, $\bar{\epsilon}$ also renders a tradeoff between cost optimality and service quality. When $\bar{\epsilon}$ is larger than 1, the marginal increase of playback smoothness is small while the cost increase is significant. Therefore, $\bar{\epsilon} \in [0.8, 1.2]$ would be a good option in the algorithm.

VI. Conclusion

This paper investigates optimal deployment of VoD services on a hybrid cloud, consisting of on-premise servers and public geo-distributed cloud data centers. A dynamic algorithm is proposed based on Lyapunov optimization theory, to replicate videos in the hybrid cloud and to distribute user requests, which minimizes the long-run operational cost of the VoD service provider under service quality constraints. With rigorous theoretical analysis, we show that our algorithm approaches the optimality achieved by a mechanism with known information in the future T time slots by a small constant gap, no matter what the request arrival pattern is. Simulations further verify its performance under realistic settings. In our ongoing work, we are implementing a prototype VoD system on real-world cloud platforms based on the algorithm.

References

[1] K. C. Almeroth and M. H. Ammar, "On the Use of Multicast Delivery to Provide a Scalable and Interactive Video-on-Demand Service," *Journal of Selected Areas in Communications*, no. 6, pp. 1110–1122, 1996.

[2] A. Hu, "Video-on-Demand Broadcasting Protocols: A Comprehensive Study," in *Proc. of IEEE INFOCOM*, Apr. 2001.

[3] *PPLive*, http://www.pplive.com/.

[4] *UUSee*, http://www.uusee.com/.

[5] C. Wu, B. Li, and S. Zhao, "Multi-channel Live P2P Streaming: Refocusing on Servers," in *Proc. of IEEE INFOCOM*, Apr. 2008.

[6] H. Li, L. Zhong, J. Liu, B. Li, and K. Xu, "Cost-effective Partial Migration of VoD Services to Content Clouds," in *Proc. of IEEE CLOUD*, Jul. 2011.

[7] Y. Wu, C. Wu, B. Li, X. Qiu, and F. Lau, "CloudMedia: When Cloud on Demand Meets Video on Demand," in *Proc. of IEEE ICDCS*, Jun. 2011.

[8] B. Tan and L. Massoulie, "Optimal Content Placement for Peer-to-Peer Video-on-Demand Systems," in *Proc. of INFOCOM*, Apr. 2011.

[9] J. M. Almeida, D. L. Eager, M. K. Vernon, and S. J. Wrigh, "Minimizing delivery cost in scalable streaming content distribution systems," *IEEE Tran. on Multimedia*, no. 2, pp. 356–365, Apr. 2004.

[10] N. Laoutaris, G. Smaragdakis, K. Oikonomou, I. Stavrakakis, and A. Bestavros, "Distributed Placement of Service Facilities in Large-Scale Networks," in *Proc. of IEEE INFOCOM*, May 2007.

[11] J. Leblet, Z. Li, G. Simon, and D. Yuan, "Optimal Network Location in Distributed Virtualized Data-Centers," *Computer Communications*, no. 16, pp. 1968–1979, 2011.

[12] S. H. Owen and M. S. Daskin, "Strategic Facility Location: A Review," *Euro. Journal of Operational Research*, pp. 423–447, 1998.

[13] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource Allocation and Cross-layer Control in Wireless Networks," *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1–149, 2006.

[14] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.

[15] ——, "Energy Optimal Control for Time Varying Wireless Networks," *IEEE Tran. on Information Theory*, no. 7, pp. 2915–2934, Jul. 2006.

[16] M. J. Neely and L. Golubchik, "Utility Optimization for Dynamic Peer-to-Peer Networks with Tit-For-Tat Constraints," in *Proc. of the INFOCOM*, Apr. 2011.

[17] M. M. Amble, P. Parag, S. Shakkottai, and L. Ying, "Content-Aware Caching and Traffic Management in Content Distribution Networks," in *Proc. of the INFOCOM*, Apr. 2011.

[18] *Amazon Elastic Compute Cloud*, http://aws.amazon.com/ec2/.

[19] *Amazon Simple Storage Service*, http://aws.amazon.com/s3/.

[20] M. J. Neely, "Opportunistic Scheduling with Worst Case Delay Guarantees in Single and Multi-Hop Networks," in *Proc. of INFOCOM*, Apr. 2011.

[21] X. Qiu, H. Li, C. Wu, Z. Li, and F. C. M. Lau, "Dynamic Scaling of VoD Services into Hybrid Clouds with Cost Minimization and QoS Guarantee," http://www.cs.hku.hk/~xjqiu/papers/cloud-vod.pdf, The University of Hong Kong, Tech. Rep., Dec. 2011.

[22] *GLPK (GNU Linear Programming Kit)*, http://www.gnu.org/s/glpk/.

[23] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "A Measurement Study of a Large-Scale P2P IPTV System," *IEEE Tran. on Multimedia*, no. 8, pp. 1672–1687, 2007.