On the Cost-QoE Trade-off for Cloud-based Video Streaming under Amazon EC2's Pricing Models

Jian He, Yonggang Wen, *Member, IEEE,* Jianwei Huang, *Senior Member, IEEE,* Di Wu, *Member, IEEE*

***Abstract*—The emergence of cloud computing provides a cost-effective approach to deliver video streams to a large number of end users with the desired user quality-of-experience (QoE). Under such a paradigm, a video service provider (VSP) can launch its own video streaming services virtually, by renting the distribution infrastructure from one or more cloud service providers (CSPs). However, CSPs like Amazon EC2 normally offer multiple pricing options for virtual machine (VM) instances they can provide, such as on-demand instances, reserved instances, and spot instances. Such diverse pricing models make it challenging for a VSP to determine how to optimally procure the required number of VM instances in different types to satisfy dynamic user demands. Given the limited budget, a VSP needs to carefully balance the procurement cost and the achieved QoE for end users. In this paper, we investigate the trade-off between the cost incurred by VM instance procurement and the achieved QoE of end users under Amazon EC2's pricing models, and formulate the VM instance provisioning and procurement problem into a constrained stochastic optimization problem. By applying the Lyapunov optimization framework, we design an online procurement algorithm, which approaches the optimal solution with explicitly provable upper bounds. We also conduct extensive trace-driven simulations and our results show that our proposed algorithm (OPT-ORS) achieves a good balance between the procurement cost and the user QoE for cloud-based VSPs. In the achieved near-optimal situation, our algorithm guarantees that reserved VM instances are fully utilized to satisfy the baseline user demand, on-demand VM instances are only rent to handle flash crowds, while spot VM instances are rent more frequently than on-demand VM instances to serve user demand over the baseline due to their low prices.**

I. Introduction

Exponential growth of video traffic challenges the current paradigm to stream large amounts of video contents to end users. It has been reported in [1] that video traffic will grow with an annual rate of 34%, and contribute to 55 percent of all consumer Internet traffic by 2016. Such explosive growth of video traffic has started to and would continue to stress the global Internet, possibly resulting in poor Quality-of-Experience (QoE) for users. Specifically, this tussle demands new paradigms to distribute and stream video contents over the Internet in a cost-effective manner, while maintaining the required QoE for end users.

Existing solutions, mainly based on content delivery networks (CDNs) (e.g., Akamai[2]), are inefficient in resolving the aforementioned fundamental tussle. In a CDN-based architecture, video contents are pushed to the network edge, closer to the users. It serves well as a solution to improve the QoE and reduce the economical cost, compared to the traditional client-server architecture[3]. However, highly dynamic video traffic would result in low resource utilization due to semi-static resource allocation in CDNs. The utilization ratio of most CDNs today can be as low as 5%-10% [4]. Such a low utilization ratio would translate directly into high cost. The situation would be made even worse when dealing with a flash-crowd situation[5], in which numerous users are interested in the same content simultaneously. In such a case, system resources will be undersubscribed, resulting in deteriorated user experiences. Therefore, a more dynamic resource provisioning paradigm should be in place.

Cloud computing, owing to its elastic resource allocation capability, offers a natural solution for cost-effective video streaming with the desired QoE requirements. Specifically, system resources can dynamically scale up and down, matching the application demand. Adopting the above design principle, researchers have started to investigate the deployment of elastic video streaming services over the cloud infrastructure (e.g.,[6], [7], [8], [9]). However, previous work mostly focused on resource provisioning under a single pricing model.

Generally, cloud service providers (CSPs) offer multiple pricing options for VM instances to satisfy different user preferences. As a leading CSP in the world, Amazon EC2 offers three typical pricing models for VM instances, including: *1) On-demand pricing model*: users are allowed to pay by hours without a long-term commitment; *2) Reserved pricing model*: users can make a low, one-time, upfront payment for a long-term reservation (e.g., 1-3 years) of an instance; *3) Spot pricing model*: Amazon EC2 sets a spot price for instances dynamically. Users are allowed to set the maximum hourly price they are willing to pay and the instances are only allocated when the spot price is lower than the user-defined price. In spite that multiple pricing models provide

Manuscript received April 15, 2013; revised June 30, 2013; accepted September 6, 2013. Date of publication XXX, 2013; date of current version XXX, 2013. This work was supported in part by the NSFC under Grant 61003242, Grant 61272397, the Fundamental Research Funds for the Central Universities under Grant 12LGPY53, Guangdong Natural Science Funds for Distinguished Young Scholar under Grant S20120011187, the General Research Funds (Project Number CUHK 412710 and CUHK 412511) established under the University Grant Committee of the Hong Kong Special Administrative Region, China, MOE Tier-1 (RG 31/11), NTU Start-up Grant, a gift fund from MSRA and EIRP 02 from EMA in Singapore. This paper was recommended by Associate Editor S. Ci.

Jian He and Di Wu are with the Department of Computer Science, Sun Yat-sen University, China (e-mail: {hejian9@mail2, wudi27@mail}.sysu.edu.cn). Yonggang Wen is with the School of Computer Engineering , Nanyang Technological University, Singapore (e-mail: ygwen@ntu.edu.sg). Jianwei Huang is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: jwhuang@ie.cuhk.edu.hk). The corresponding author is Di Wu.

users with higher flexibility, however, it is also challenging for a cloud-based video service provider (VSP) to determine how to optimally procure the necessary number of VM instances in each type to meet dynamic user demands. There is a clear trade-off between the cost of VM instance procurement and the achieved QoE of end users. To be more competitive in the market, a VSP must utilize its budget economically while still guaranteeing the desired user QoE.

In this paper, our purpose is to provide guidelines for cloud-based VSPs on how to optimally procure VM instances to satisfy user demands under multiple pricing models. We consider a realistic scenario with Amazon EC2's pricing models, in which a VSP not only needs to minimize the consumption of cloud resources, but also needs to optimize the procurement plan in the presence of price diversity and volatility. Compared to the case with only a single pricing model, a cloud-based VSP can exploit the diversity among pricing models to further reduce its procurement cost.

We develop a theoretic model to explore the trade-off between the procurement cost and the achieved user QoE for cloud-based video streaming. We formulate the problem into a joint optimization problem of resource provisioning and procurement under price variety and demand dynamics. We aim at optimizing the time average of the weighted sum of the total procurement cost and the achieved QoE of end users. By using the Lyapunov optimization framework, we propose an approximate online algorithm, called **OPT-ORS**, with the explicitly provable performance upper bound. Our proposed algorithm also has low complexity by exploiting the structural properties of the optimal solution. Extensive trace-driven simulations have been conducted to verify the effectiveness of our proposed **OPT-ORS** algorithm in the practical settings. Our algorithm can guarantee that reserved instances are fully utilized to satisfy the baseline user demand, even without any information about the future fluctuation of user demands; spot instances are rent more frequently than on-demand instances to serve user demand over the baseline due to their low prices, and on-demand instances are rent only when flash crowds occur.

The rest of the paper is organized as follows. Section II reviews related work. The theoretic model is described in Section III. We formulate the optimization problem in Section IV. In Section V, we describe the design of our online strategy. The results obtained from trace-driven simulations are presented in Section VI. Section VII concludes the paper and discusses the future work.

II. Related Work

Cloud computing has become a promising approach to conduct large-scale content distribution over the Internet due to its capability of elastic resource allocation. Highly dynamic traffic demand drives researchers to investigate the migration of Internet video applications to the clouds (e.g., [10], [9], [8], [7], [11]) and thus reduce the unnecessary operating cost incurred by resource oversubscription. However, two major problems need to be resolved before the deployment of video applications over the cloud, namely, resource provisioning and resource procurement.

For cloud-based video applications, the problem of cloud resource provisioning has been extensively studied in [7], [8], [9], [12], [13], [14], and etc. Researchers have designed kinds of optimal resource provisioning strategies, which let VSPs scale up and down resources provisioned in clouds according to the demand dynamics. However, resource procurement, which is to strategically procure cloud resources from one or more CSPs, has not been well investigated yet.

Normally, cloud resources are rented in the unit of VM instances. Leading CSPs like Amazon EC2 provide multiple pricing models for their VM instances (see [15]). Amazon EC2 has three different pricing models, including on-demand pricing model, reservation pricing model, and spot pricing model. A cloud-based VSP can possibly exploit the price diversity to reduce its procurement cost. However, previous work on cloud-based video streaming mostly focused on the VM procurement under a single pricing model. For example, the on-demand pricing model is adopted in [9], [10], and the spot pricing model is analyzed in [16], [17], [18]. The problem of VM instance procurement under the spot pricing model for computation-intensive applications (e.g., MapReduce-based applications) has been studied in [19], [20], [21]. In addition, there has been some work (e.g., [22], [23], [24]) on maximizing the revenue of CSPs by designing dynamic cloud resource pricing strategies.

Our work differs from previous works in that we consider the joint problem of resource provisioning and procurement under multiple pricing models, which is more realistic and challenging for cloud-based VSPs.

III. System Model and architecture

For cloud-based video streaming, a VSP rents VM instances from one or more CSPs (e.g., Amazon EC2) in an elastic manner to provide video streaming services. Fig. 1 provides a simplified model of cloud-based video streaming systems. Generally, a CSP offers multiple types of standard VM instances (e.g., *small, medium, large* and *extra-large* instances), each of which has different hardware and bandwidth configuration. A VSP can rent a set of VM instances in different types to provide the required streaming capacity. Each user request will be served by a provisioned instance in the cloud. Due to its leading position, we take Amazon EC2 as a typical example of CSPs in the following sections.

A. Amazon EC2's Pricing Models

Based on the difference in pricing options, VM instances provided by Amazon EC2 can be classified into three major categories:

- *On-Demand Instances*: Users are allowed to pay for the rented VM instances by the hour with no long-term commitments or upfront payments. A user can increase or decrease the procured streaming capacity based on the demands of the application and only pay the specified hourly rate for used instances. Amazon EC2 always strives to provide enough available on-demand instances to meet user demands, but during the periods with very high demand, it is still possible that a user might not be

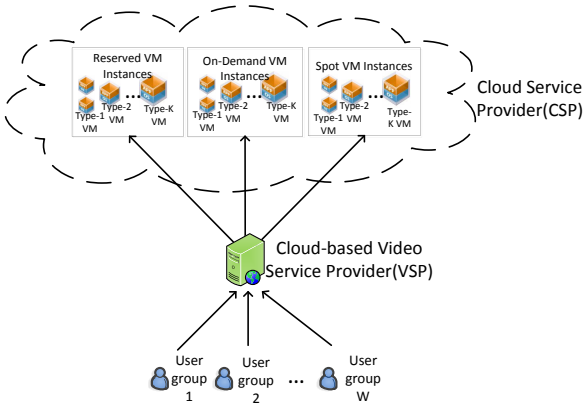


Fig. 1. Simplified Model of Cloud-based Video Streaming

able to launch specific on-demand instances for a short period of time.

- *Reserved Instances*: For reserved instances, users can make a low, one-time, upfront payment for a long-term reservation of an instance (e.g., one or three years) and pay a significantly low hourly rate for running an instance. For applications that have stable needs, by using reserved instances, it is able to achieve nearly 50% savings compared to using on-demand instances. From the functional perspective, reserved instances and on-demand instances perform identically.
- *Spot Instances*: Spot instances provide the ability for customers to purchase streaming capacity with no upfront commitment and at hourly rates usually lower than the on-demand rate. For spot instances, customers are allowed to specify the maximum hourly price that they are willing to pay to run a particular instance type. Amazon EC2 sets a *spot price* for each instance type dynamically, which is the price that all customers will pay to run a spot instance for that given period. The spot price fluctuates according to the supply and demand for VM instances, but customers will never pay more than the maximum price they have specified. If the spot price is higher than a customer's maximum price, the instance will be shut down by Amazon EC2 automatically. Other than the above differences, spot instances perform exactly the same as on-demand or reserved instances.

Assume that VM instances are classified into K types by their streaming capacity, and the streaming capacity of a VM instance takes one of the K values from the set $\{s_1, s_2, \cdots, s_K\}$. Abstractly, any VM instance can be characterized by a vector (s, p), where s is the streaming capacity, and p is the unit price per unit time (e.g., one hour). Based on the specific VM instance in use, Amazon EC2 provides the following pricing models:

- *On-demand pricing model*: the price to procure a type-k on-demand instance is p_k per unit time. The total cost of renting a type-k on-demand instance for t units of time is $p_k \cdot t$.
- *Reserved pricing model*: the price to procure a type-k reserved instance consists of two parts: (1) a one-time payment, q_k and (2) a usage-based payment, $\hat{p}_k$ per unit time. If being reserved, the total cost of using a type-k reserved instance for t units of time is $q_k + \hat{p}_k \cdot t$.
- *Spot pricing model*: the price to procure a type-k spot instance depends on the latest spot price of $\tilde{p}_k(t)$ at time t and the bidding price. The VSP can use the instance only if the bidding price is no less than $\tilde{p}_k(t)$.

B. User Request and QoE Model

Chunk-based video streaming is considered in this paper due to its popularity. To facilitate video streaming, the whole video is divided into multiple chunks, and a video chunk can be viewed immediately after being downloaded. The playback duration time of one video chunk lasts for a few seconds (e.g., 2-5 seconds). For each video, the streaming server will generate multiple versions of video copies encoded with different playback rates.

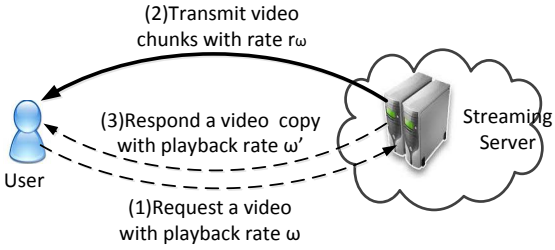


Fig. 2. Interaction between the user and the streaming server in the cloud

Fig. 2 describes the interaction between a user and a streaming server in the cloud. In the figure, the dashed lines represent the control message exchange and the solid line represents the real video data transmission between the user and the server. A user request specifies a desired playback rate ω, which indicates that the user wishes to download a video copy encoded with a playback rate ω. However, the downloading rate of a user may not be able to sustain the desired playback rate. Depending on the real downloading rate r_ω, the streaming server will respond with a video copy encoded with a playback rate ω'. The serving playback rate ω' is close to but no greater than the downloading rate r_ω.

Assume that the arrival of user requests follows a Poisson process in time slot t. Each user request is associated with a desired playback rate ω. The total number of user requests arrived in time slot t is a random variable, denoted by $\mu(t)$, with the mean as $\lambda(t)$.

Without loss of generality, we assume that the desired playback rate ω takes values from a finite and discrete set Ω with $|\Omega| = W$, and user requests are classified into W groups according to their desired playback rate. ω is a random variable with a p.m.f of $f(\omega)$ for $\omega > 0$.The actual total number of user requests with the desired playback rate ω in time slot t is denoted by $\mu(\omega, t)$. From the p.m.f. $f(\omega)$, we can know that the mean arrival rate of user requests with the desired playback rate ω in time slot t is $\lambda(\omega, t) = f(\omega)\lambda(t)$.

For a user with a downloading rate of r_ω and a desired playback rate of ω, a function $g(r_\omega)$ is defined to map the downloading rate r_ω to the serving playback rate ω', namely,

$g(r_\omega) = \arg_{\omega'} \min |r_\omega - \omega'|, \omega' \in \Omega$. The function $g(r_\omega)$ should satisfy that $g(r_\omega) \leq \omega$ because the surplus serving playback rate over the desired playback rate ω will not provide any QoE benefits. The QoE score of a user is defined by the function $Q(\omega, \omega') = Q(\omega, g(r_\omega))$, whose value falls in the range of $[0, 5]$ [1]. $Q(\omega, \omega')$ is a continuous differentiable, and increasing function of the playback rate ω' in the interval $[0, \omega]$. If $\omega' > \omega$, $Q(\omega, \omega')$ will be assigned the maximum value $Q_0 = 5$. As in [25], [27], the QoE score function of a user is defined as below:

$$Q(\omega, \omega') = \begin{cases} a_1 \ln \frac{a_2 \omega'}{\omega} & \omega' \leq \omega \\ Q_0 & \omega' > \omega \\ 0 & a_2 \omega' < \omega, \end{cases}$$

where a_1 and a_2 are two positive constant parameters, and $a_1 \ln a_2 = Q_0 = 5$.

IV. Problem Formulation

For a cloud-based VSP, there are two major objectives: *first*, the VSP should maximize the QoE of end users to improve its competitiveness in the market; *second*, the VSP should minimize the procurement cost of renting VM instances for cost-savings. However, the above two objectives are in conflict with each other. To enhance the QoE, the VSP might need to over-provision virtual machines, which in turn increase the procurement cost. Therefore, the VSP should first understand the trade-off between the QoE metrics and the procurement cost, and then identify an optimal operating point on the trade-off curve, so that it can determine the number of different VM instances to be procured under multiple pricing models.

In our problem formulation, users who have not finished downloading the required amount of video contents are defined as *active users*. For each desired playback rate ω, a queue $\mathbf{H}_\omega$ is defined for each user group, and $H_\omega(t)$ is the length of the queue at time slot t, which also represents the number of active users with the desired playback rate ω at time slot t. Denote $\Gamma_\omega(t)$ as the total upload bandwidth allocated by the provisioned VM instances to the group of active users with the desired playback rate ω.

Given $\Omega = \{\omega_1, \omega_2, \cdots, \omega_W\}$ and $\omega_1 < \omega_2 < \cdots < \omega_W$, we assume a demand-proportional resource allocation strategy. Therefore, the total upload bandwidth allocated to each user group satisfies the following relationship:

$$\Gamma_{\omega_1}(t) : \Gamma_{\omega_2}(t) : \cdots : \Gamma_{\omega_W}(t) = \omega_1 H_{\omega_1}(t) : \omega_2 H_{\omega_2}(t) : \cdots : \omega_W H_{\omega_W}(t).$$

The downloading rate of an active user in time slot t with the desired playback rate ω is denoted by $r_\omega(t)$. We assume uniform resource allocation within a group, that is,

$$r_\omega(t) = \frac{\Gamma_\omega(t)}{H_\omega(t)}, \forall \omega \in \Omega.$$

Under such a resource allocation strategy, the downloading rates of any two users in two different user groups, with the desired playback rate ω_i and ω_j respectively, satisfy $r_{\omega_i}(t) : r_{\omega_j}(t) = \omega_i : \omega_j$ due to $r_{\omega_i} : r_{\omega_j} = \frac{\Gamma_{\omega_i}(t)}{H_{\omega_i}(t)} : \frac{\Gamma_{\omega_j}(t)}{H_{\omega_j}(t)} = \omega_i : \omega_j$. Therefore, if $\frac{g(r_{\omega_i})}{g(r_{\omega_j})} = \frac{\omega_i}{\omega_j}$, we can obtain:

$$a_1 \ln a_2 \frac{g(r_{\omega_i})}{\omega_i} - a_1 \ln a_2 \frac{g(r_{\omega_j})}{\omega_j} = a_1 ln \frac{g(r_{\omega_i})\omega_j}{g(r_{\omega_j})\omega_i} = 0$$

Therefore, under a demand-proportional resource allocation strategy, any two users in two different user groups can experience the same QoE. Due to the uniform resource allocation within a user group, users within one group will experience the same QoE. Note that any other resource allocation strategy (e.g., fair allocation strategies in [28]), which can achieve a certain level of fairness among users, can also be applied to the above model directly, as long as the amount of upload bandwidth allocated to each user group can be known explicitly. Denote the total uploading rate of all provisioned VM instances in time slot t as $R(t)$. Given the demand-proportional resource allocation strategy, we have

$$\begin{aligned} \sum_{\omega \in \Omega} \Gamma_\omega(t) &= R(t), \\ \Gamma_\omega(t) &= \frac{\omega H_\omega(t)}{\sum_{\tilde{\omega} \in \Omega} \tilde{\omega} H_{\tilde{\omega}}(t)} R(t). \end{aligned} \quad (1)$$

Assume that the desired playback rate will not change during the downloading process of the whole video, while the serving playback rate will change dynamically according to the downloading rate. Then, the total QoE of all active users are given by:

$$\begin{aligned} Q(t) &= \sum_{\omega \in \Omega} Q(\omega, g(r_\omega)) \cdot H_\omega(t) \\ &= \sum_{\omega \in \Omega} a_1 \ln \frac{a_2 \min(g(r_\omega), \omega)}{\omega} \cdot H_{\omega(t)} \\ &= \sum_{\omega \in \Omega} a_1 \ln \frac{a_2 \min(\omega', \omega)}{\omega} \cdot H_{\omega(t)}. \end{aligned}$$

Let T_d be the playback duration time of a whole video and T_p be the duration time of one decision period of a VSP to adjust the procurement plan. We consider a time-slotted system, where the length of each time slot is the same as a decision period. Assume that the desired playback duration time τ of each user follows a uniform distribution within the interval $[0, T_d]$. The desired playback duration time is reflected by the playback duration time of video contents downloaded from the streaming server. A user will leave the system if the playback duration time of the downloaded video content has exceeded τ. For example, if the desired playback duration time is 60 seconds and each video chunk lasts for 5 seconds, then the user will leave the system after finishing downloading 12 video chunks. The impacts of user arrivals and departures within one decision period will be reflected by the variations of queue sizes (i.e., $H_\omega(t)$). Thus, the disparity of time scaling between user actions and the VSP decision can be mitigated via monitoring the queue sizes. The VSP only needs to keep track of the states of queue sizes to make procurement decision.

[1]Note that, the interval $[0, 5]$ is obtained through the curve fitting results based on the dataset in [25], [26].

For one user with the desired playback rate ω, the downloading rate during time slot t is $\frac{\Gamma_\omega(t)}{H_\omega(t)}$, then the amount of video content can be downloaded during time slot t is $\frac{\Gamma_\omega(t)}{H_\omega(t)}T_p$ when the user arrives the system at the beginning of the slot. If the user arrives at the middle of the time slot, then the amount of video content is less than $\frac{\Gamma_\omega(t)}{H_\omega(t)}T_p$. Given the serving playback rate $g(r(\omega))$, the playback duration time of the video content downloaded during time slot t is no longer than $\frac{\frac{\Gamma_\omega(t)}{H_\omega(t)}T_p}{g(r(\omega))}$. Thus, the probability that a user will leave the system is no more than $Pr(\tau \leq \frac{\frac{\Gamma_\omega(t)}{H_\omega(t)}T_p}{g(r(\omega))}) = \min\{\frac{\frac{\Gamma_\omega(t)}{H_\omega(t)}T_p}{g(r(\omega))} \cdot \frac{1}{T_d}, 1\}$. Note that, if $\frac{\frac{\Gamma_\omega(t)}{H_\omega(t)}T_p}{g(r(\omega))} \geq T_d$, the user will leave the system with a probability of 1.

The update of each queue can be expressed as follows:

$$\begin{aligned} H_\omega(t+1) &\leq \max[H_\omega(t) - \\ &\quad \min\{\frac{\frac{\Gamma_\omega(t)}{H_\omega(t)}T_p}{g(r_\omega)}\frac{1}{T_d}, 1\}H_\omega(t), 0] + \mu(\omega,t) \quad (2)\\ &= \max[H_\omega(t) - \frac{\Gamma_\omega(t)T_p}{g(r_\omega)T_d}, 0] + \mu(\omega,t) \quad (3)\\ &\leq \max[H_\omega(t) - \frac{\Gamma_\omega(t)T_p}{\omega T_d}, 0] + \mu(\omega,t), \quad (4) \end{aligned}$$

where $\mu(\omega,t)$ represents the actual number of request arrivals during time slot t. In the inequality (2), $\min\{\frac{\frac{\Gamma_\omega(t)}{H_\omega(t)}T_p}{g(r_\omega)} \cdot \frac{1}{T_d}, 1\}H_\omega(t)$ is the average number of users in queue H_ω who will leave the system given that $\min\{\frac{\frac{\Gamma_\omega(t)}{H_\omega(t)}T_p}{g(r_\omega)} \cdot \frac{1}{T_d}, 1\}$ represents the maximum probability that a user will leave the system. For the equality (3), if $\frac{\frac{\Gamma_\omega(t)}{H_\omega(t)}T_p}{g(r_\omega)} \cdot \frac{1}{T_d} < 1$, then $\max[H_\omega(t) - \min\{\frac{\frac{\Gamma_\omega(t)}{H_\omega(t)}T_p}{g(r_\omega)} \cdot \frac{1}{T_d}, 1\}H_\omega(t), 0] = \max[H_\omega(t) - \frac{\frac{\Gamma_\omega(t)}{H_\omega(t)}T_p}{g(r_\omega)} \cdot \frac{1}{T_d} \cdot H_\omega(t), 0]$; otherwise, $\max[H_\omega(t) - \min\{\frac{\frac{\Gamma_\omega(t)}{H_\omega(t)}T_p}{g(r_\omega)} \cdot \frac{1}{T_d}, 1\}H_\omega(t), 0] = \max[H_\omega(t) - \frac{\frac{\Gamma_\omega(t)}{H_\omega(t)}T_p}{g(r_\omega)} \cdot \frac{1}{T_d} \cdot H_\omega(t), 0] = 0$. The inequality (4) is obtained from $g(r_\omega) \leq \omega$.

Denote $P(t)$ as the total monetary cost to procure VM instances in time slot t. Let $P^{on}(t)$, $P^{re}(t)$ and $P^{sp}(t)$ be the monetary cost associated with the procurement of on-demand instances, reserved instances and spot instances respectively in time slot t, then $P(t) = P^{on}(t) + P^{re}(t) + P^{sp}(t)$. The objective can be defined as the following constrained stochastic optimization problem:

$$\begin{aligned} \boldsymbol{P1.1:} \quad \min \quad & \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}(-Q(t) + \alpha \cdot P(t)) \\ \text{s.t.} \quad & \lim_{T\to\infty}\sup\frac{1}{T}\sum_{t=1}^{T}\sum_{\omega\in\Omega}H_\omega(t) \leq \infty, \\ & \zeta(t) \in \boldsymbol{\chi(t)}, \end{aligned}$$

where the first constraint is to ensure the stability of all user queues. α is a tunable parameter which represents the tradeoff between the monetary cost and the QoE. Denote $A_k(t)$, $B_k(t)$, $N_k(t)$ as the maximum number of type-k on-demand instances, reserved instances, and spot instances that can be rented from the CSP in time slot t respectively. In spite that the cost function includes only the overall QoE of all users, the QoE of each individual user will be affected similarly under a demand-proportional resource allocation strategy. The impact of the solution on the QoE of each individual user is determined by the tradeoff parameter α. If the service provider cares more about the QoE experienced by users, it should choose a smaller α. A feasible procurement strategy is one strategy which does not violate these upper bounds, $A_k(t)$, $B_k(t)$ and $N_k(t)$, in each time slot t. Let $\zeta(t)$ denote the procurement strategy in time slot t which tells how many VM instances of each type to be rented, and $\boldsymbol{\chi(t)}$ denote the set of all feasible procurement strategies at time slot t.

If the mean arrival rate of user requests in time slot t is known beforehand, then Problem **P1.1** can be transformed into the following problem:

$$\begin{aligned} \boldsymbol{P1.2:} \min \quad & \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}E[-Q(t) + \alpha \cdot P(t)] \\ \text{s.t.} \quad & E[\sum_{\omega\in\Omega}\frac{\Gamma_\omega(t)T_p}{\omega T_d}] \geq \sum_{\omega\in\Omega}E[\mu(\omega,t)], \forall t \\ & \zeta(t) \in \boldsymbol{\chi(t)}, \forall t, \end{aligned}$$

where $\lambda(\omega,t) = E[\mu(\omega,t)]$ is the mean arrival rate of the Poisson process for users with the desired playback rate ω. The first constraint of Problem **P1.2** indicates that the stability of all queues can be ensured if the expected departure rate is larger than the expected arrival rate. Let T_c be the duration time of one charging period, which is on the order of hours (such as, one hour in EC2 pricing models) and assume that $T_c = mT_p, m \geq 1$. The feasible region of $\zeta(t)$ depends on the solution in the previous slots because the charging period of one VM lasts for multiple decision periods. However, we can design an optimal stationary randomized algorithm by solving the following optimization problem:

$$\begin{aligned} \boldsymbol{P1.3:} \quad \min \quad & \frac{1}{m}\sum_{t=t'}^{(m-1)T_p}E[-Q(t) + \alpha P(t)] \\ \text{s.t.} \quad & E[\sum_{\omega\in\Omega}\frac{\Gamma_\omega(t')mT_p}{\omega T_d}] \geq \sum_{t=t'}^{t'+(m-1)T_p}\sum_{\omega\in\Omega}E[\mu(\omega,t)] \\ & \zeta(t') \in \boldsymbol{\chi(t')}, \forall t', \end{aligned}$$

where $t' \mod T_c = 0$, which indicates that $\zeta(t), t \in [t', (m-1)T_p]$ is independent of $\zeta(t''), t'' < t'$. The transformation from Problem **P1.2** to Problem **P1.3** mitigates the disparity of time scaling between the decision period and the charging period. The optimal randomized algorithm makes decisions at the beginning of each charging period based on $\mu(\omega,t)$. Given the probability distribution of $\mu(\omega,t), \forall\omega \in \Omega, t$, the solution to Problem **P1.3** can be obtained through standard linear programming [29]. Note that, the optimal randomized algorithm is impractical as the probability distribution of $\mu(\omega,t)$ is not easy, if impossible, to be known a priori.

It has been proved in [30] that any online algorithm can not do better than the optimal stationary randomized algorithm solving Problem **P1.3**. Therefore, we consider the optimal stationary randomized algorithm as a baseline when analyzing

the performance of online algorithms. A cost function is defined as $c_t(\zeta(t)) = -Q(t) + \alpha \cdot P(t)$. Denote Δ as the capacity region of Problem **P1.3**, which means that for any $\mu(\omega, t) \in \Delta$, there exist feasible solutions to Problem **P1.3**. Denote $c_t^*(\epsilon)$ as the optimal results of Problem **P1.3** with $\mu(\omega, t) \in \Delta$ being replaced by $\mu(\omega, t) + \epsilon \in \Delta$, and c_t^* is the optimal value when $\epsilon = 0$. $\bar{c}^*(\epsilon)$ is the time average cost under the optimal stationary randomized algorithm, and $\bar{c}^*(\epsilon) = \frac{1}{T}\sum_{t=1}^{T} c_t^*(\epsilon)$.

V. Design of Online Provisioning and Procurement Strategy

In this section, we design an approximate online procurement algorithm to solve the optimization problem **P1.1**. By exploiting structural properties of the optimal solution, we can reduce the computational complexity of the online algorithm significantly.

A. Characteristics of Different VM Instances

We first describe the specific properties of VM instances under different pricing models. Starting from the time point of renting an instance, the charging time point of one VM instance is the time slot after a period of T_c. Define "*active VM instance*" as the VM instances whose charging time point hasn't exceeded the current time slot.

1) On-Demand Instances: Denote $a_k(t) \geq 0$ as the number of type-k on-demand VM instances provisioned in time slot t. Then $\sum_{i=1}^{m-1} a_k(t - iT_p)$ is the number of active on-demand VM instances provisioned in the previous slots before time slot t. Therefore, the number of active on-demand type-k VM instances in time slot t is $a_k(t) + \sum_{i=1}^{m-1} a_k(t - iT_p)$. Note that, $a_k(t - iT_p) = 0$ if $t - iT_p < 0$. The monetary cost in time slot t incurred by provisioning on-demand instances is given by $P^{on}(t) = \sum_{k=1}^{K} a_k(t) p_k$.

2) Reserved Instances: The reservation time of Amazon EC2 instances can be as long as one or three years. The reserved instances are more suitable to handle the stable portion of user demand. If the infimum of the mean arrival rate $\lambda_{min}(\omega)$ can be found, to guarantee queue stability, the optimal online provisioning strategy should ensure that $\sum_{\omega\in\Omega} \frac{\Gamma_\omega(t) T_p}{\omega \cdot T_d} \geq \sum_{\omega\in\Omega} \lambda_{min}(\omega)$ according to [30].

Let $\lambda_{min}(\omega)$ be the baseline demand of a user group, whose members have a desired playback rate of ω. Even without any prediction of user demand, we can utilize reserved instances to satisfy the baseline user demand. Denote b_k as the number of type-k reserved instances, then b_k can be derived by solving the following optimization problem:

$$\begin{aligned} \boldsymbol{P2.1}: \min_{b_k} \quad & \sum_{k=1}^{K} b_k \cdot q_k \\ \text{s.t.} \quad & \sum_{k=1}^{K} s_k b_k T_p \geq \sum_{\omega\in\Omega} \lambda_{min}(\omega) \cdot \omega \cdot T_d \\ & 0 \leq b_k \leq B_k. \end{aligned}$$

Denote $\hat{a}_k(t)$ as the number of type-k reserved VM instances provisioned in time slot t. The number of active type-k reserved instances in time slot t is $\hat{a}_k(t) + \sum_{i=1}^{m-1} \hat{a}_k(t - iT_p)$. Note that, $\hat{a}_k(t - iT_p) = 0$ if $t - iT_p < 0$. As the one-time fee does not affect the results of online algorithms, it can be ignored for simplicity. Then the usage-based monetary cost of reserved instances at time slot t is given by $P^{re}(t) = \sum_{k=1}^{K} \hat{a}_k(t) \hat{p}_k$, where K is the number of VM instance types.

3) Spot Instances: The price of a spot VM instance $\tilde{p}_k(t)$ can be described by a stochastic process. Due to the low spot price, the rental of spot instances can further reduce the monetary cost. For any arbitrary stochastic process of spot price, it is feasible to obtain the expected spot price $E[\tilde{p}_k(t)]$ of a type-k spot instance. As in [19], we can obtain the expected spot price by constructing a semi-Markov process for spot price evolution. $E[\tilde{p}_k(t)]$ can be derived from the stochastic kernel of the semi-Markov process, which is constructed by a maximum likelihood estimator based on the observed spot price history.

In addition, we assume that it is also feasible to obtain the *reliability insurance* ρ (see [16]), which is defined as the probability that the bidding price exceeds the actual spot price of a VM instance after declaring the bidding price. No specific bidding strategy is assumed. Intuitively, the reliability insurance is a function of the declared bidding price as a fraction of on-demand price.

Denote the number of type-k spot instances provisioned in time slot t as $n_k(t) \geq 0$, then the number of active type-k spot instances at slot t is given by:

$$n_k(t) + \sum_{i=1}^{m-1} n_k(t - iT_p).$$

Note that, $n_k(t - iT_p) = 0$ if $t - iT_p < 0$. The expected monetary cost of spot instances in time slot t is defined by

$$P^{sp}(t) = \sum_{k=1}^{K} n_k(t) E[\tilde{p}_k(t)] \rho.$$

B. Design of Approximate Online Algorithm

By combining three pricing models together, we can transform Problem **P1.1** into the following optimization problem:

$$\begin{aligned} \boldsymbol{P3.1}: \min_{\hat{a}_k(t), a_k(t), n_k(t)} \lim_{T\to\infty} & \frac{1}{T}\sum_{t=1}^{T} (-Q(t) + \alpha \cdot P(t)) \\ \text{s.t.} \quad \lim_{T\to\infty} \sup & \frac{1}{T}\sum_{t=1}^{T}\sum_{\omega\in\Omega} H_\omega(t) \leq \infty \\ & \sum_{k=1}^{K} s_k \cdot y(t) = R(t), \forall t \in [1, T] \quad \text{(a)} \\ & \hat{a}_k(t) + \sum_{i=1}^{m-1} \hat{a}_k(t - iT_p) \leq b_k, \\ & \qquad \forall t \in [1, T], k \in [1, K] \quad \text{(b)} \end{aligned}$$

$$\Gamma_\omega = \frac{\omega H_\omega(t) R(t)}{\sum_{\omega' \in \Omega} \omega' H_{\omega'}(t)}, \forall \omega \in \Omega, t \in [1,T] \quad \text{(c)}$$

$$a_k(t) \le A_k(t), \forall t \in [1,T], k \in [1,K] \quad \text{(d)}$$

$$n_k(t) \le N_k(t), \forall t \in [1,T], k \in [1,K]. \quad \text{(e)}$$

Note that, $n_k(t) \cdot \rho$ is the expected number of type-k spot VM instances that can be bidden successfully, and $y(t) = (\hat{a}_k(t) + \sum_{i=1}^{m-1} \hat{a}_k(t - iT_p)) + (a_k(t) + \sum_{i=1}^{m-1} a_k(t - iT_p)) + ((n_k(t) + \sum_{i=1}^{m-1} n_k(t - iT_p)) \cdot \rho)$ is the number of active instances in time slot t. The total cost incurred in time slot t is $P(t) = P^{on}(t) + P^{re}(t) + P^{sp}(t)$.

Due to the problem complexity, we transform Problem **P3.1** to Problem **P3.2** by applying the Lyapunov optimization framework [30]. Define the Lyapunov function as $L(\vec{H}(t)) = \sum_{\omega \in \Omega} H_\omega^2(t)$. The one-slot Lyapunov drift $E\{L(\vec{H}(t+1) - L(\vec{H}(t)))|\vec{H}(t)\}$ is upper bounded by $B - 2 \cdot E[\sum_{\omega \in \Omega} H_\omega(t) \cdot \frac{\Gamma_\omega(t) T_p}{\omega \cdot T_d} |\vec{H}(t)] + 2 \cdot E[\sum_{\omega \in \Omega} H_\omega(t) \cdot \mu(\omega, t) | \vec{H}(t)]$, where B is a constant (See Appendix B in our technical report [31] for details). According to the Lyapunov optimization framework, the original Problem **P3.1** can be transformed to a problem which aims to minimize a weighted sum of the upper bound of the Lyapunov drift at time t and $-Q(t) + \alpha P(t)$, with constant terms being deleted. We can design an online algorithm to achieve the approximately optimal solution for Problem **P3.1**. And the online algorithm can obtain the optimal solution to the following optimization problem at each decision period:

$$\boldsymbol{P3.2:} \quad \min_{a_k(t), \hat{a}_k(t), n_k(t)} - \sum_{j=1}^{W} H_{\omega_j}(t) \cdot \frac{\Gamma_{\omega_j}(t) T_p}{\omega_j \cdot T_d} + V \cdot [\sum_{j=1}^{W} -Q(\omega_j, g(\frac{\Gamma_{\omega_j(t)}}{H_{\omega_j}(t)})) \cdot H_{\omega_j}(t)] + \alpha V P(t)$$

$$\text{s.t.} \quad (a)(b)(c)(d)(e).$$

Denote the algorithm that can achieve the optimal solution to Problem **P3.2** as **OPT-ORS**, which is the abbreviation of *approximate OPTimal provisioning strategy with On-demand, Reserved and Spot instances*. Details of the **OPT-ORS** algorithm can be found in Algorithm 1.

In Algorithm 1, all the queue sizes $H_\omega(0)$ are initialized at the beginning of the algorithm, and then the online algorithm makes the procurement decision every decision period. Note that, the VSP should run the online algorithm to decide the amount of resources that should be procured from the CSP as long as the video streaming service is active. At the beginning of each decision period, the expected price of spot instances is calculated. Then, the resource allocation $\Gamma_\omega(t)$ is determined explicitly. $\Gamma_\omega(t)$ depends on the number of active instances procured in the previous decision period and the number of instances to be procured in the current decision period. After determining the resource allocation, the optimal procurement decision can be obtained by solving Problem **P3.2**. It can be easily verified that Problem **P3.2** is convex. Thus, we can solve Problem **P3.2** efficiently by exploiting standard convex optimization tools (e.g., cvx). At the end of each decision period, all queues are updated according to user arrivals and departures in the past decision period.

Algorithm 1 Online Algorithm OPT-ORS

Input:

Prices of on-demand VM instances and reserved VM instances;

Trade-off parameter α;

Decision period T_p, playback duration time of a whole video T_d, charging period T_c;

Parameters of QoE function a_1, a_2.

Output:

VM procurement decision

$$\widehat{a}_k(t), a_k(t), n_k(t), k = 1, \cdots, K.$$

1: Initialization step: Let $t = 0$, and set $H_\omega(0) = 0$, for all $\omega \in \Omega$.
2: **while** the video streaming service is active **do**
3: Calculate the expected price of spot instances $E[\widetilde{p}_k(t)]$ at the beginning the decision period according to the estimator in [19].
4: Determine the resource allocation $\Gamma_\omega(t)$ according to the demand proportional resource allocation strategy (1).
5: Calculate the optimal procurement decisions $(\widehat{a}_k^*(t), a_k^*(t), n_k^*(t))$ for each $k = 1, \cdots, K$ of Problem **P3.2**.
6: For each $\omega \in \Omega$, update the queues $H_\omega(t) = \max[H_\omega(t-1) - L_\omega(t), 0] + \mu(\omega, t)$, where $L_\omega(t)$ denotes the actual number of departed users from queue H_ω in decision period t.
7: Set $t \leftarrow t + 1$.
8: **end while**

From the structure of Problem **P3.2**, the properties of the optimal results obtained by **OPT-ORS** can be described in the following theorem.

Theorem 5.1: Denote $\hat{a}_k^*(t)$, $a_k^*(t)$ and $n_k^*(t)$ as the solution to Problem **P3.2**, then we have

1) Given $\hat{p}_k \le p_k, \forall k$, we have the following relationship:

(a) If $E[\tilde{p}_k(t)] \le \hat{p}_k$, then

$(i) n_k^*(t) < N_k(t) \Rightarrow \quad \hat{a}_k^*(t) = 0, a_k^*(t) = 0;$

$(ii) n_k^*(t) = N_k(t), \hat{a}_k^*(t) + \sum_{i=1}^{m-1} \hat{a}_k^*(t - iT_p) < b_k$

$\Rightarrow \quad a_k^*(t) = 0;$

(b) If $\hat{p}_k < E[\tilde{p}_k(t)] \le p_k$, then

$(i) \hat{a}_k^*(t) + \sum_{i=1}^{m-1} \hat{a}_k^*(t - iT_p) < b_k$

$\Rightarrow n_k^*(t) = 0, \quad a_k^*(t) = 0;$

$(ii) \hat{a}_k^*(t) + \sum_{i=1}^{m-1} \hat{a}_k^*(t - iT_p) = b_k, n_k^*(t) < N_k(t)$

$\Rightarrow \quad a_k^*(t) = 0;$

(c) If $\hat{p}_k < p_k < E[\tilde{p}_k(t)]$, then

$$(i)\hat{a}_k^*(t) + \sum_{i=1}^{m-1} \hat{a}_k^*(t - iT_p) < b_k$$
$$\Rightarrow a_k^*(t) = 0, n_k^*(t) = 0;$$
$$(ii)\hat{a}_k^*(t) + \sum_{i=1}^{m-1} \hat{a}_k^*(t - iT_p) = b_k, a_k^*(t) < A_k(t)$$
$$\Rightarrow \quad n_k^*(t) = 0.$$

2) If $\sum_{k=1}^{K} s_k(\hat{a}_k^*(t) + a_k^*(t) + n_k^*(t)) \geq \sum_{j=1}^{W} H_{\omega_j}(t)\omega_j$, which indicates that the total downloading rate is larger than total desired playback rate, then we have:

$$-s_k \sum_{j=1}^{W} \frac{H_{\omega_j}^2(t)T_p^2}{\sum_{\omega\in\Omega} \omega H_\omega(t) T_d} + V\alpha p_k \leq 0 \Rightarrow {a_k}^* = A_k(t),$$

$$-s_k \sum_{j=1}^{W} \frac{H_{\omega_j}^2(t)T_p^2}{\sum_{\omega\in\Omega} \omega H_\omega(t) T_d} + V\alpha \hat{p}_k \leq 0 \Rightarrow \hat{a}_k^* = b_k,$$

$$-s_k \sum_{j=1}^{W} \frac{H_{\omega_j}^2(t)T_p^2}{\sum_{\omega\in\Omega} \omega H_\omega(t) T_d} + V\alpha\rho E[\tilde{p}_k(t)] \leq 0 \Rightarrow$$
$$n_k^*(t) = N_k(t).$$

Proof: Please refer to Appendix A in our technical report [31] for the proof details. ∎

Due to the difficulty to make accurate user arrival and departure predictions in practice, we exploit Lyapunov optimization techniques to avoid the need of prediction. Therefore, instead of directly solving the optimization problem with complete information, our algorithm can approach the optimal solution with provable upper bounds by greedily minimizing the Lyapunov drift at each decision period.

Theorem 5.2: The online algorithm **OPT-ORS** can stabilize the system with an upper bounded average queue length and time-average cost. Moreover, for a given parameter $V > 0$ and any non-negative value ϵ, we have

$$\lim_{T\to\infty} \frac{1}{T} \cdot \sum_{t=1}^{T} \sum_{\omega\in\Omega} E[H_\omega(t)] \quad \leq \quad \frac{B}{2\epsilon} + \frac{V \cdot \bar{c}^*(\epsilon)}{2\epsilon}$$
$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} E[c^t(\zeta^t)] \quad \leq \quad \bar{c}^* + \frac{B}{V},$$

where $B = \frac{(\sum_{k=1}^{K}(A_k+b_k+N_k)s_k)^2 T_p^2}{{T_d}^2 \cdot \sum_{\omega\in\Omega}\omega^2} + \sum_{\omega\in\Omega} \mu_{\max}^2(\omega)$, $\mu_{\max}(\omega)$ is upper bound of the number of request arrivals $\mu(\omega, t)$ during one slot, and $A_k = \max\{A_k(t), \forall t\}, N_k = \max\{N_k(t), \forall t\}$. $\bar{c}^*$ is the optimal result achieved by the optimal stationary randomized algorithm which solves Problem **P1.3**.

Proof: Please refer to Appendix B in our technical report [31] for the proof details. ∎

Intuitively, given a type-k VM instance, it is of a higher priority to rent type-k instances under the pricing model that can provide a lower price. Part 1) of Theorem 5.1 reveals that the procurement decisions for different types of VMs are correlated. In one decision period, the number of VM instances of a certain type to be procured in the current decision period is determined by the number of active VM instances procured in the previous decision period and the price relationship across different types of VM instances. For example, the Property 1)(a) in Theorem 5.1, the expected price of the type-k spot VM instance $E[\tilde{p}_k(t)]$ is the lowest compared to other types of instances. If the optimal number of type-k spot instances procured has not reached the upper bound, namely, $n_k^*(t) < N_k(t)$, then no type-k on-demand or reserved instances will be rented in the optimal solution. Thus, for each instance type, the rentals under each pricing model are prioritized by the price under each pricing model. The priorities may be variant across different VM instance types. The variant prices of spot instances also affect the priorities. For type-k VM instances, they will not be procured until all other type-k' VM instances with lower prices have been procured. Furthermore, since the total surplus downloading rate over the total desired playback rate will result in excessive monetary cost without providing any QoE benefits, the total downloading rate provided by all active instances in time slot t is upper bounded by the total desired playback rate from all active users in time slot t. From the Property 2) in Theorem 5.1, though the surplus downloading rate can not provide QoE benefits, the leaving rate of users can increase with the surplus downloading rate. Therefore, to ensure queue stability, the surplus downloading rate may happen when the queue size is large enough.

From Theorem 5.1, it is able to further decrease the computation complexity of Algorithm **OPT-ORS**. The property 2) in Theorem 5.1 indicates that the size of feasible region can be reduced into a smaller set $\sum_{k=1}^{K} s_k(\hat{a}_k^*(t) + a_k^*(t) + n_k^*(t)) = \sum_{j=1}^{W} H_{\omega_j}(t)\omega_j$, then we can determine the optimal solution by applying the Property 2). From the Property 1) in Theorem 5.1, the further decrease of computation complexity can be achieved by determining the derivative of $\frac{d\Delta(\boldsymbol{H(t)})+V(-Q_t+\alpha P_t)}{dR(t)}$. For example, given $\hat{p}_k < E[\tilde{p}_k(t)] \leq p_k, \forall k$, then only reserved instances will be provisioned if the derivative is larger than 0 at the point $R(t) = \sum_{k=1}^{K} s_k b_k$. For each type of VM instances, the prices $\hat{p}_k$, $E[\tilde{p}_k(t)]$, and p_k divide the feasible solutions into three separate parts: $\{0 \leq \hat{a}_k(t) + \sum_{i=1}^{m-1} \hat{a}_k(t - iT_p) \leq b_k, n_k(t) = 0, a_k(t) = 0, \forall k\}$, $\{\hat{a}_k(t) + \sum_{i=1}^{m-1} \hat{a}_k(t - iT_p) = b_k, 0 \leq n_k(t) \leq N_k(t), a_k(t) = 0, \forall k\}$, and $\{\hat{a}_k(t) + \sum_{i=1}^{m-1} \hat{a}_k(t - iT_p) = b_k, n_k(t) = N_k(t), 0 \leq a_k(t) \leq A_k(t), \forall k\}$. The minimum value of $\Delta(\boldsymbol{H(t)}) + V(-Q_t + \alpha P_t)$ in each part can be obtained by calculating $\frac{d\Delta(\boldsymbol{H(t)})+V(-Q_t+\alpha P_t)}{dR(t)}$. The optimal solution to Problem **P3.2** is the smallest one.

VI. Simulation Experiments

In this section, we conduct trace-driven simulations to evaluate the effectiveness of our proposed algorithm.

A. Dataset Description and Experiment Settings

To make the simulation be more realistic, we use the video traffic dataset obtained from Youku [32], which is the largest VSP in China, to drive the simulation. The Youku dataset contains the video traffic logs from March 24, 2012 to March

30, 2012. Each traffic log includes the user ID, the user arrival time, and the requested video ID. By dividing the one week period into a series of 5-minute time slots and aggregating the number of user requests in each time slot, we can plot the evolution of user request pattern in Fig. 3.

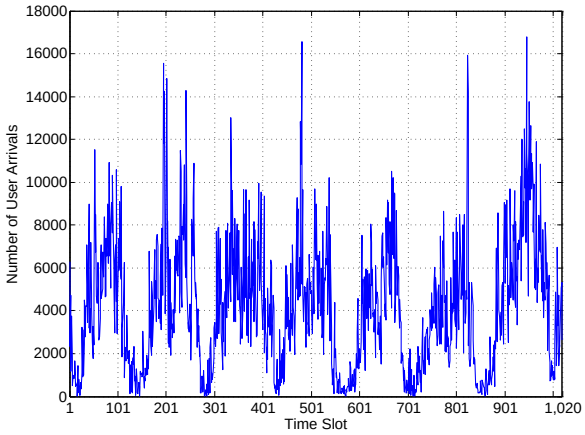


Fig. 3. Number of user request arrivals in each time slot

Define the set of available playback rates as $\Omega = \{200kbps, 400kbps, 800kbps\}$. The playback duration time of a whole video is $T_d = 600s$, and the length of one procurement adjustment slot is $T_p = 300s$. There are 4 types of VM instances, and their corresponding streaming capacity is $\{10Mbps, 20Mbps, 40Mbps, 80Mbps\}$. The number of VM instances in each type follows a uniform distribution $U(10, 30)$. $\alpha = 18000$,$V = 10000$, $\rho = 0.8$ and $b_k = 1, \forall k = 1, 2, 3$.

The prices of VM instances in each type are obtained directly from the Amazon EC2 pricing website [15]. There are four types of VM instances: *small, medium, large* and *extra large*. Fig. 4 shows the prices of VM instances (in the unit of US dollars) in each time slot. The price of on-demand instances are much higher than reserved and spot instances. The price of spot instances changes dynamically, and can be lower than the price of reserved instances.

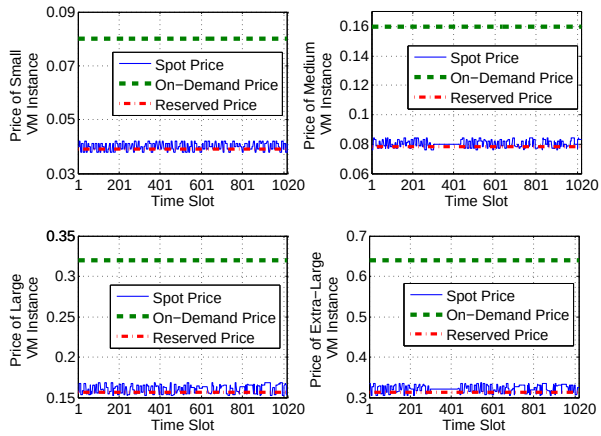


Fig. 4. Prices of VM instances in different types

For comparison, we select the algorithm that obtains optimal results of Problem **P1.3** as a baseline. Note that, the optimal algorithm needs to know all video traffic statistics a priori, which is impossible in reality.

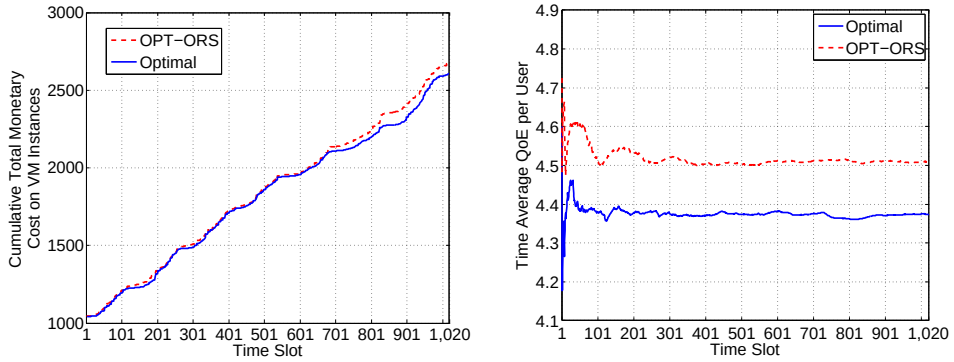
B. Simulation Results

We first analyze the total monetary cost of all VM instances. The total one-time fee for reserved instances is 1035 US dollars. Fig. 5(a) illustrates the cumulative total monetary cost of all VM instances. Algorithm **OPT-ORS** provisions more instances than the optimal algorithm, while **OPT-ORS** can ensure a higher time-average QoE, which is illustrated in Fig. 5(b). High demand will incur a higher monetary cost, which is consistent with the Property 2) in Theorem 5.1. It means that large queue size leads to provision more instances to ensure the stability of queues, which will result in a larger increasing rate of the monetary cost. From Fig. 5(a), we can see that the increasing rate of the monetary cost between time slot 100 and 200 is much smaller than that between time slot 300 and 400. From Fig. 3, we can see user demand between time slot 100 and 200 is much smaller than that between time slot 300 and 400. Fig. 5(b) illustrates the time average QoE per user. The algorithm **OPT-ORS** provides a higher expected QoE for each user by provisioning more VM instances. Because we calculate the time-average expected QoE, the fluctuation of QoE during the early slots will result in significant changes of the time-average value.

Fig. 6(a) illustrates the percentage of monetary cost on each type of VM instances. Note that, we do not include one-time fee of reserved instances since both **OPT-ORS** and the optimal algorithm incur the same one-time fee. In the figure, we can see that the percentage of monetary cost spent on reserved instances is quite stable because we only utilize reserved instances to satisfy the baseline demand. Flash crowds will lead to more monetary cost on on-demand instances. For example, the increase of monetary cost of on-demand instances at time slot 50, 200 and 950. Flash crowds result in large queue sizes, which further require more rentals of on-demand instances to ensure queue stability. Note that, the high price of on-demand VM instances also leads to a high increasing rate of monetary cost of on-demand instances. Spot instances are rent more frequently than on-demand instances to satisfy user demand over the baseline demand due to their low prices. Because of the high prices of on-demand instances, the percentage of monetary cost of spot instances will decrease significantly in time slot 50 and 200.

Fig. 6(b) illustrates the time-average weighted sum of the total monetary cost and the user QoE. In the figure, we can see that the difference between **OPT-ORS** and the optimal algorithm (around 1000) is smaller than the upper bound proved in Theorem 5.1 (> 3000).

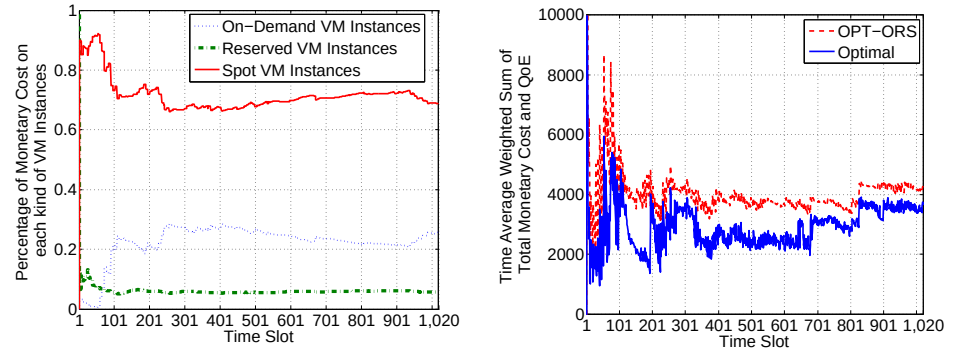
Fig. 7(a) shows the amount of bandwidth allocated to each user group. Fig. 7(b) illustrates the evolution of queue sizes. Queue $\mathbf{H_1}$, $\mathbf{H_2}$ and $\mathbf{H_3}$ are correspondent to user groups with the desired playback rate $200Kbps$, $400Kbps$, and $800Kbps$ respectively. From the two figures, we can see that the amount of allocated bandwidth is almost consistent with the evolution of queue sizes. The number of provisioned VM



(a) Total monetary cost on all VM instances

(b) Time Average QoE per User

Fig. 5. Monetary cost and QoE



(a) Percentage of the monetary cost on each type of VM instances

(b) Time average weighted sum of the total monetary cost and QoE

Fig. 6. Monetary cost distribution and time average of objective value

instances are highly correlated between two continuous time slots due to the mismatched timescales of decision period and charging period, while user arrivals and departures between two continuous time slots are independent. Therefore, we analyze the correlation between the bandwidth and the queue size in the time unit of one charging period. From the figures, we can see that large queue sizes require large bandwidth to be allocated. Large bandwidth will lead to high user departure rate, then the queue size will fast decrease. The correlation between the amount of allocated bandwidth and queue size are 0.79, 0.86 and 0.80 respectively.

Fig. 8 illustrates the time-average monetary cost and QoE when varying the value of α. We take 15 different values in the interval $[10000, 25000]$. Note that, we do not include the one-time fee in the monetary cost since the one-time fee has no effect on the comparison of **OPT-ORS**. In this figure, we can see that the monetary cost and the QoE exhibit concave relationship. Therefore, the objective problem is convex. What's more, when α is large (i.e., larger than 21000), the QoE will fast decrease and there is no significant reduction of monetary cost. If α is small (i.e., close to 10000), the QoE can hardly increase because it has approached the maximum value. $\alpha = 18000$ is a sweet spot for the real system configuration.

Fig. 9 shows the performance of **OPT-ORS** under various values of V. From this figure, we can see that a larger value

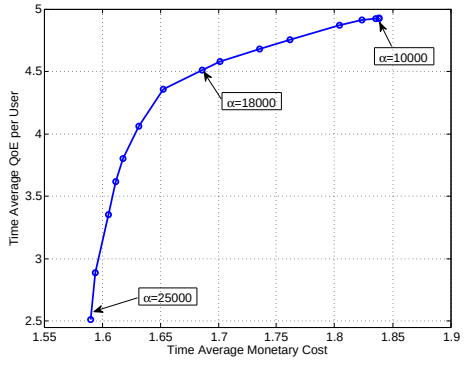


Fig. 8. Tradeoff between time average monetary cost and QoE with varying $\alpha \in [10000, 25000]$.

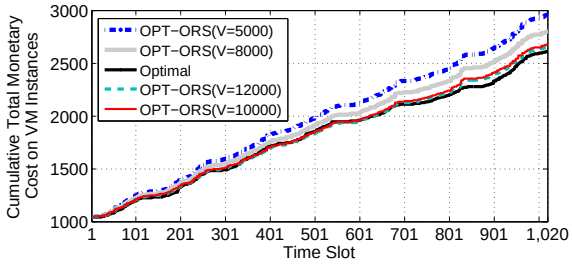
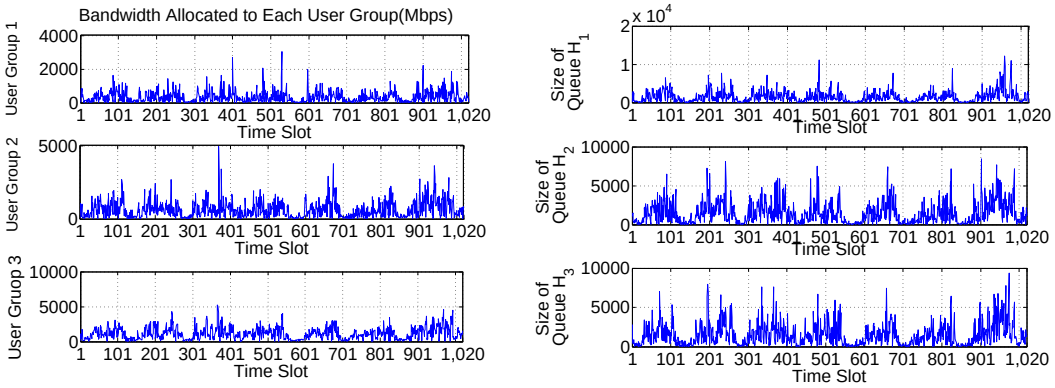


Fig. 9. Performance of **OPT-ORS** under various V



(a) Bandwidth allocated to each user group

(b) Size of each queue at each time slot

Fig. 7. Bandwidth allocation and queue size evolution

of V can approach the optimality more closely, which is consistent with the result in [30].

VII. Conclusion and Future Work

In this paper, we build a theoretic model to study the cost-QoE tradeoff for cloud-based video streaming providers, and consider a realistic scenario with Amazon's EC2 pricing models. We formulate the procurement of different VM instances under multiple pricing models as a constrained stochastic optimization problem, and design an online procurement algorithm to approach the optimal solution with explicitly provable bounds by applying the Lyapunov optimization techniques. By characterizing structural properties of our proposed algorithm, we can further reduce the computation complexity of the algorithm. Through extensive simulations, we evaluate the effectiveness of our proposed online algorithm. Our algorithm can result in an optimal tradeoff between the monetary cost incurred by VM instance procurement and the achieved user QoE efficiently. As the next step, we will investigate the case of adaptive video streaming, in which the desired playback rate of a video can be changed dynamically according to the bandwidth availability. We are also interested in the optimization of procurement plans under other pricing models.

References

[1] "Cisco visual networking index: Forecast and methodology, 2011-2016."

[2] "Akamai," http://www.akamai.com.

[3] C. Huang, J. Li, and K. W. Ross, "Can internet video-on-demand be profitable?" in *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4. ACM, 2007, pp. 133–144.

[4] F. Lo Presti, C. Petrioli, and C. Vicari, "Distributed dynamic replica placement and request redirection in content delivery networks," in *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2007. MASCOTS'07. 15th International Symposium on*. IEEE, 2007, pp. 366–373.

[5] 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 *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*. IEEE, 2008, pp. 1–5.

[6] Y. Jin, Y. Wen, G. Shi, G. Wang, and A. Vasilakos, "Codaas: An experimental cloud-centric content delivery platform for user-generated contents," in *Computing, Networking and Communications (ICNC), 2012 International Conference on*. IEEE, 2012, pp. 934–938.

[7] X. Qiu, H. Li, C. Wu, Z. Li, and F. C. Lau, "Dynamic scaling of VoD services into hybrid clouds with cost minimization and QoS guarantee," in *Packet Video Workshop (PV), 2012 19th International*. IEEE, 2012, pp. 137–142.

[8] F. Wang, J. Liu, and M. Chen, "Calms: Cloud-assisted live media streaming for globalized demands with time/region diversities," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 199–207.

[9] Y. Wu, C. Wu, B. Li, X. Qiu, and F. C. Lau, "Cloudmedia: When cloud on demand meets video on demand," in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*. IEEE, 2011, pp. 268–277.

[10] Z. Huang, C. Mei, L. E. Li, and T. Woo, "CloudStream: delivering high-quality streaming videos through a cloud-based SVC proxy," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 201–205.

[11] M. Hajjat, X. Sun, Y.-W. E. Sung, D. Maltz, S. Rao, K. Sripanidkulchai, and M. Tawarmalani, "Cloudward bound: planning for beneficial migration of enterprise applications to the cloud," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4, pp. 243–254, 2010.

[12] J. He, D. Wu, Y. Zeng, X. Hei, and Y. Wen, "Towards optimal deployment of cloud-assisted video distribution services," *IEEE Transactions on Circuits and Systems for Video Technology (IEEE TCSVT), Accepted*, 2013.

[13] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimization of resource provisioning cost in cloud computing," *Services Computing, IEEE Transactions on*, vol. 5, no. 2, pp. 164–177, 2012.

[14] S. Chaisiri, R. Kaewpuang, B.-S. Lee, and D. Niyato, "Cost minimization for provisioning virtual servers in amazon elastic compute cloud," in *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on*. IEEE, 2011, pp. 85–95.

[15] "Amazon EC2 pricing," http://aws.amazon.com/ec2/pricing/.

[16] O. Agmon Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafrir, "Deconstructing amazon EC2 spot instance pricing," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*. IEEE, 2011, pp. 304–311.

[17] B. Javadi, R. K. Thulasiramy, and R. Buyya, "Statistical modeling of spot instance prices in public cloud environments," in *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*. IEEE, 2011, pp. 219–228.

[18] M. Mattess, C. Vecchiola, and R. Buyya, "Managing peak loads by leasing cloud infrastructure services from a spot market," in *High Performance Computing and Communications (HPCC), 2010 12th IEEE International Conference on*. IEEE, 2010, pp. 180–188.

[19] Y. Song, M. Zafer, and K.-W. Lee, "Optimal bidding in spot instance market," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 190–198.

[20] S. Yi, D. Kondo, and A. Andrzejak, "Reducing costs of spot instances via checkpointing in the amazon elastic compute cloud," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*. Ieee, 2010, pp. 236–243.

[21] N. Chohan, C. Castillo, M. Spreitzer, M. Steinder, A. Tantawi, and C. Krintz, "See spot run: using spot instances for mapreduce workflows," in *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*. USENIX Association, 2010, pp. 7–7.

[22] H. Xu and B. Li, "Maximizing revenue with dynamic cloud pricing: The infinite horizon case," in *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2929–2933.
[23] Q. Wang, K. Ren, and X. Meng, "When cloud meets eBay: Towards effective pricing for cloud computing," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 936–944.
[24] D. Niu, C. Feng, and B. Li, "A theory of cloud bandwidth pricing for video-on-demand providers," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 711–719.
[25] W. Zhang, Y. Wen, Z. Chen, and A. Khisti, "QoE-driven cache management for HTTP adaptive bit rate streaming over wireless networks," *IEEE Transactions on Multimedia, in press*, 2013.
[26] M. Li, Z. Chen, and Y.-P. Tan, "QoE-aware resource allocation for scalable video transmission over multiuser MIMO-OFDM systems," in *Visual Communications and Image Processing (VCIP), 2012 IEEE*. IEEE, 2012, pp. 1–6.
[27] P. Reichl, B. Tuffin, and R. Schatz, "Logarithmic laws in service quality perception: where microeconomics meets psychophysics and quality of experience," *Telecommunication Systems*, pp. 1–14, 2011.
[28] T. Lan, D. Kao, M. Chiang, and A. Sabharwal, "An axiomatic theory of fairness in network resource allocation," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010.
[29] G. B. Dantzig, *Linear programming and extensions*. Princeton university press, 1998.
[30] M. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan and Claypool, 2010.
[31] J. He, Y. Wen, J. Huang, and D. Wu, "On the Cost-QoE Trade-off for Cloud-based Video Streaming under Amazon EC2's Pricing Models," CS Department, Sun Yat-sen University, Tech. Rep., 2013. [Online]. Available: http://netlab.sysu.edu.cn/~jhe/qoe-tr.pdf
[32] "Youku," http://www.youku.com/.

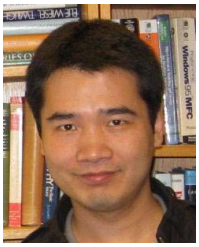
Jian He received the B.S. degree in Computer Science from Sun Yat-sen University in 2011. He is now a graduate student in the School of Information Science and Technology at Sun Yat-sen University. His research interests include content distribution networks, data center networking, green networking and network measurement.

Yonggang Wen (S'99-M'08) is an assistant professor with school of computer engineering at Nanyang Technological University, Singapore. He received his PhD degree in Electrical Engineering and Computer Science (minor in Western Literature) from Massachusetts Institute of Technology, Cambridge, USA. Previously he has worked in Cisco to lead product development in content delivery networks, which had a revenue impact of 3 Billion US dollars globally. Dr. Wen has published over 70 papers in top journals and prestigious conferences. His latest work in multi-screen cloud social TV has been featured by global media and has attracted much commercial attention. His research interests include cloud computing, green data center, big data analytics, multimedia network and mobile computing.

Jianwei Huang (S'01-M'06-SM'11) is an Associate Professor in the Department of Information Engineering at the Chinese University of Hong Kong. He received Ph.D. in Electrical and Computer Engineering from Northwestern University in 2005. He worked as a Postdoc Research Associate in the Department of Electrical Engineering at Princeton University during 2005-2007.

Dr. Huang currently leads the Network Communications and Economics Lab (ncel.ie.cuhk.edu.hk), with the main research focus on nonlinear optimization and game theoretical analysis of networks, especially on network economics, cognitive radio networks, and smart grid. He is the co-recipient of IEEE Marconi Prize Paper Award in Wireless Communications 2011, and a co-recipient of Best Paper Awards from IEEE WiOPT 2013, IEEE SmartGridComm 2012, WiCON 2011, IEEE GLOBECOM 2010, and APCC 2009. He received the IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award in 2009.

Dr. Huang has served as the Editor of IEEE Journal on Selected Areas in Communications - Cognitive Radio Series, Editor of IEEE Transactions on Wireless Communications, and Guest Editor of IEEE Journal on Selected Areas in Communications and IEEE Communications Magazine. He is the Chair of IEEE ComSoc Multimedia Communications Technical Committee, a Steering Committee Member of IEEE Transactions on Multimedia and IEEE ICME. He has served as the TPC Co-Chair of IEEE GLOBECOM Selected Areas of Communications Symposium 2013, IEEE WiOpt 2012, IEEE ICCC Communication Theory and Security Symposium 2012, IEEE GlOBECOM Wireless Communications Symposium 2010, IWCMC Mobile Computing Symposium 2010, and GameNets 2009.



Di Wu (M'06) received the B.S. degree from the University of Science and Technology of China in 2000, the M.S. degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2003, and the Ph.D. degree in Computer Science and Engineering from the Chinese University of Hong Kong in 2007. From 2007 to 2009, he was a postdoctoral researcher in the Department of Computer Science and Engineering, Polytechnic Institute of NYU, advised by Prof. Keith W. Ross. He has been an Associate Professor in the Department of Computer Science, Sun Yat-Sen University, China, since July 2009. He was the winner of IEEE INFOCOM 2009 Best Paper Award, and is a member of the IEEE, the IEEE Computer Society, the ACM, and the Sigma Xi.

His research interests include multimedia communication, cloud computing, peer-to-peer networking, Internet measurement, and network security.