On-Demand Bandwidth Pricing for Congestion Control in Core Switches in Cloud Networks

Abouzar Ghavami, Zhuozhao Li and Haiying Shen
Department of Electrical and Computer Engineering
Clemson University, Clemson, SC 29631
Email: ghavamip@gmail.com, {zhuozhl, shenh}@clemson.edu

***Abstract*—The cloud networks use switches to transfer inbound and outbound traffic through the data centers. Access of multiple tenants to the limited bandwidth capacity over the network switches increases the data traffic congestion in the network. The highly congested switches are vulnerable to get overloaded, and consequently slow down the flow of data traffic in the network. This paper proposes a nonlinear pricing policy for on-demand bandwidth allocation that jointly maximizes the total satisfaction of tenants and minimizes the congestion in the core switches. The optimal schedule is found through the best response strategy, in which each tenant updates its bandwidth allocation at each step based on the updated load-dependent predetermined nonlinear bandwidth pricing functions. The updated bandwidth allocations converge to the optimal bandwidth schedule that balances the load over the core switches. The performance of proposed pricing policy is evaluated under different scenarios.**

I. Introduction

Cloud computing is getting more popular on providing web services and delivery platform for IT infrastructures. Cloud computing refers to the on-demand delivery of IT infrastructures and applications via the internet based on pay-as-you-go pricing [1]. Due to the high purchasing cost of the servers, storage and network hardwares, and software licenses, it is more economic for the customers to rent the large-scale computation resources on demand from the popular cloud providers [2]. As an example, Netflix, the world's leading subscription media streaming service, is running on the Amazon web services (AWS) cloud [3].

In this paper we aim to control the congestion over the core switches in the cloud network. The core switches not only connect the tenants inside the data center, but they are the gateways of the datacenter to the outside world [4]. Therefore, it is essential to control congestion over the core switches to have a smooth data traffic transfer between the tenants inside and outside of the datacenter. To achieve this aim, we develop a bandwidth allocation mechanism that guarantees the QoS for bandwidth consumption of the tenants while minimizing congestion over the core switches.

Resource allocation in computer networks is studied extensively in the previous literature. The authors in [5], [6] and [7] maximize the total satisfaction of the users for bandwidth usage in a distributed network. As the problem is separable, the authors propose a synchronous update process with linear pricing mechanism to find the optimal solution in a decentralized framework. In this paper we jointly maximize the total satisfaction of the users and minimize the congestion in the cloud network. In this non-separable problem scenario, where the selfish and rational tenants do not necessarily reveal their private information, such as utility function and maximum bandwidth requirement, to the cloud provider, we propose an asynchronous update process with nonlinear pricing mechanism to find the optimal bandwidth schedule through a decentralized framework.

In order to have a fair, clear and predictable bandwidth allocation in the cloud, we consider that tenants contribute to the routing in the cloud. In our proposed method, the cloud provider predetermines load-dependent nonliear bandwidth pricing policies, and only monitors and reports the updated pricing policies to the tenants without interefering in the decentralized bandwidth allocation process among them. In this scenario that tenants are able to allocate their bandwidth over the core switches, the nonlinear pricing mechanism always results in the load balancing over the core switches; however, linear bandwidth pricing does not motivate the tenants to distribute their loads over the core switches, and therefore it does not guarantee load balancing over the core switches.

II. System Model

In this paper, we focus on the bandwidth allocation among tenants in core switches (denoted as $\mathcal{S} = \{1, ..., S\}$). The core switches are the root switches in a fat-tree topology of a data center [8]. The tenants are divided in two categories as in [9]: (i) reserved instance and (ii) on-demand instance tenants. We consider that the reserved instance tenants do not contribute in the bandwidth scheduling, and they are routed through the switches based on a predetermined routing schedule. We denote D_s as the total bandwidth of the reserved instance tenants passing through switch s, $s = 1, ..., S$. We consider a set of selfish and rational on-demand instance tenants, $\mathcal{K} = \{1, ..., K\}$. We define the maximum capacity of a switch as the total capacity of its outgoing links. We assume that all the switches have the same maximum capacity denoted as $B^{\max}$. Let $b_{k,s}$ denote the amount of bandwidth that tenant k sends through the outgoing links of switch s, and $B_s = D_s + \sum_{k \in \mathcal{K}} b_{k,s}$ as the total bandwidth of the outgoing links of switch s. We define $\frac{B_s}{B^{\max}}$ as the *congestion degree* of switch s, and we define $\frac{\sum_{s \in \mathcal{S}} B_s}{S \times B^{\max}}$ as the total congestion degree over the core switches.

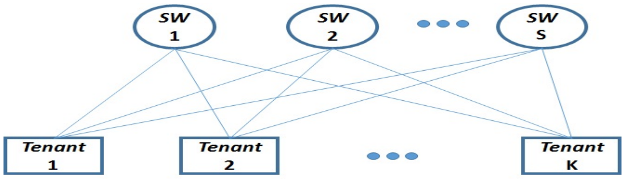

Fig. 1: The network topology of the tenants accessing the bandwidth resources of the core switches in the data center

Let $b_k = \sum_{s\in\mathcal{S}} b_{k,s}$ denote the total amount of bandwidth scheduled (reserved) by tenant k over all the core switches in $\mathcal{S}$. Let us denote $\mathbf{b}_k = (b_{k,1}, ..., b_{k,S})$ as the bandwidth schedule (reservation) of tenant k over all switches in $\mathcal{S}$. We assume that bandwidth schedules are determined for a specific period of time T, e.g. $T = 5$min, and the tenants update their bandwidth schedule after period T. Let $\mathbf{b} = (\mathbf{b}_1, ..., \mathbf{b}_K)$ denote the bandwidth schedule for the tenants in the network, and $\mathbf{b}_{-k}$ denote the bandwidth schedule for all the tenants excluding tenant k. We assume that each tenant k requires a maximum amount of bandwidth denoted by $b_k^{\max}$, i.e. $b_k \le b_k^{\max}$. We have $b_k^{\max} = \sum_{s\in\mathcal{S}} B^{\max}$ for a tenant without maximum bandwidth requirement. Let us denote $\mathcal{B}_k = \{\mathbf{b}_k | \sum_{s\in\mathcal{S}} b_{k,s} \le b_k^{\max}\}$ as the set of all feasible bandwidth allocation of tenant k, and denote $\mathcal{B} = \mathcal{B}_1 \times ... \times \mathcal{B}_K$ as the set of all feasible bandwidth allocations for all the tenants. The set $\mathcal{B}$ is compact and convex.

A feasible bandwidth schedule must satisfy each switch's capacity constraint:

$$B_s = D_s + \sum_{k\in\mathcal{K}} b_{k,s} \le \eta B^{\max}, \qquad \forall s. \tag{1}$$

where $\eta \in [0, 1]$ is the safety factor determined by the cloud provider to ensure that maximum capacity is not violated. In the next section, we formulate the bandwidth allocation problem, and introduce the social welfare and tenant's utility.

III. Problem Formulation

In this section, we formulate the problem to find the optimal bandwidth schedule that satisfies tenants' demands as much as possible and avoids congestion in the switches. To achieve this objective, the cloud provider specifies a nonlinear pricing policy that increases the unit price as the bandwidth usage increases. This disincentivizes the tenants to use bandwidth when the bandwidth usage (B_s) is high (i.e., the link is more likely to be congested). Therefore, the cloud provider uses a strictly convex function, denoted as $\mathcal{V}(\cdot)$, as the bandwidth payment function over a core switch.

TABLE I: Notations and definitions

Notation	Definition
$\mathcal{K}$	the set of tenants
$\mathcal{S}$	the set of switches
D_s	the total bandwidth of the reserved instance tenants passing through switch s
$b_{k,s}$	the amount of bandwidth that tenant k sends through the outgoing links of switch s
$b_k = \sum_{s\in\mathcal{S}} b_{k,s}$	the total amount of bandwidth scheduled (reserved) by tenant k over all the core switches
$B^{\max}$	the maximum capacity of switches
B_s	the bandwidth usage of switch s
$\mathcal{U}_k(\cdot)$	the satisfaction function of tenant k for bandwidth schedule $\mathbf{b}_k$
$\mathcal{V}(\cdot)$	the bandwidth payment function over a core switch

Let $\mathcal{U}_k(\sum_{s\in\mathcal{S}} b_{k,s})$ deote the satisfaction function of tenant k for bandwidth schedule $\mathbf{b}_k$. The satisfaction function, $\mathcal{U}_k(\cdot)$, is considered to be non-decreasing as a higher bandwidth provision makes a tenant more satisfied. Also, the marginal satisfaction of a user is non-increasing because a tenant's level of satisfaction gradually gets saturated when the provisioned bandwidth increases [10]–[12]. Thus, we consider that $\mathcal{U}_k(\cdot)$ is strictly increasing and strictly concave, and its second derivative is continuous in $\mathcal{B}_k$. The cloud provider needs to provide bandwidth to meet the tenants' satisfaction and un-congested network support for all the tenants. Since these two factors affect the QoS to the tenants' applications, which represents their welfare, we define *social welfare* of tenants as a joint consideration of these two factors in the following:

$$\begin{gathered} \mathcal{S}_w(\mathbf{b}) = \sum_{k=1}^{K} \mathcal{U}_k(\sum_{s\in\mathcal{S}} b_{k,s}) - \sum_{s\in\mathcal{S}} (\mathcal{V}(B_s) - \mathcal{V}(D_s)) \\ \text{s.t.} \quad B_s - \eta B^{\max} \le 0, \quad \forall s. \\ \mathbf{b} \in \mathcal{B}. \end{gathered} \tag{2}$$

The fixed term $\mathcal{V}(D_s)$, $\forall s$, is added to make the social welfare an unbiased function with respect to $\mathbf{b}$, i.e. $\mathcal{S}_w(\mathbf{0}) = 0$. When the total bandwidth demands $B_s > \eta B^{\max}$, switch s is considered as overloaded. We denote $\mathcal{C}(x)$ as the overload cost function associated with switch s. As the overload cost function needs to increase more at higher bandwidth usage to penaltize tenants more at overloading, we define $\mathcal{C}(x)$ as a strictly convex function with $\mathcal{C}(x) = 0$ for $x < 0$ [13].

Let us define $\mathcal{Z}(x) \doteq \mathcal{V}(x) - \mathcal{V}(D_s) + \mathcal{C}(x - \eta B^{\max})$ as the total congestion and overload cost of the bandwidth reservation $\mathbf{b}$ on each switch s. As $\mathcal{V}(\cdot)$ and $\mathcal{C}(\cdot)$ are strictly convex functions, $\mathcal{Z}(\cdot)$ is also a strictly convex function. As $\mathcal{U}_k(\cdot)$ is strictly concave for each tenant k, and $\mathcal{Z}(\cdot)$ is strictly convex, $\mathcal{W}(\cdot)$ is a strictly concave function in $\mathcal{B}$. By transferring the maximum capacity constraint as the overload cost function, the social welfare of the tenants in (2) is rewritten as in the following:

$$\begin{gathered} \mathcal{W}(\mathbf{b}) = \sum_{k=1}^{K} \mathcal{U}_k(\sum_{s\in\mathcal{S}} b_{k,s}) - \sum_{s\in\mathcal{S}} (\mathcal{V}(B_s) - \mathcal{V}(D_s)) \\ - \sum_{s\in\mathcal{S}} \mathcal{C}(B_s - \eta B^{\max}) \doteq \sum_{k=1}^{K} \mathcal{U}_k(b_k) - \sum_{s\in\mathcal{S}} \mathcal{Z}(B_s), \\ \mathbf{b} \in \mathcal{B}. \end{gathered} \tag{3}$$

We define the *socially optimal* bandwidth schedule as a feasible bandwidth schedule that maximizes the social welfare of the tenants as in (3). In order to incentivize tenants to avoid congestion in the switches, the cloud provider determines a pricing policy that charges each tenant a fee based on its scheduled bandwidth over the switches.

IV. Proposed Bandwidth Pricing Policy

In this section, we propose the bandwidth pricing policy to transfer the congestion and overload cost of the cloud provider to the tenants. Let us define $\mathcal{Y}_{k,s}(\mathbf{b}, \mathbf{b}_{-k}) = \mathcal{Z}\left(D_s + \sum_{j\in\mathcal{K}/\{k\}} b_{j,s} + b_{k,s}\right)$ as the congestion and overload cost that tenant k impose on the core switch s with its reserved bandwidth schedule $b_{k,s}$ while the other tenants have scheduled $\mathbf{b}_{-k}$ bandwidth schedule. We derive the bandwidth payment function $\xi_k(\mathbf{b}_{-k}, \mathbf{b}_k)$ for tenant k to pay for bandwidth schedule $\mathbf{b}_k$ over the core switches in $\mathcal{S}$, as in the following:

$$\xi_k(\mathbf{b}_{-k}, \mathbf{b}_k) = \sum_{s\in\mathcal{S}} \left[\mathcal{Y}_{k,s}\left(\mathbf{b}_{-k}, \mathbf{b}_k\right) - \mathcal{Y}_{k,s}\left(\mathbf{b}_{-k}, \mathbf{0}\right)\right]. \quad (4)$$

We have added the terms, $\mathcal{Y}_{k,s}(\mathbf{b}_{-k}, \mathbf{0})$, to achieve an unbiased cost function for the tenants, i.e. $\xi_k(\mathbf{b}_{-k}, \mathbf{0}) = 0, \forall k$. The utility function of tenant k, $\mathcal{F}_k(\mathbf{b}_k, \mathbf{b}_{-k})$, for scheduling $\mathbf{b}_k$ is calculated as in the following:

$$\mathcal{F}_k\left(\mathbf{b}_{-k}, \mathbf{b}_k\right) = \mathcal{U}_k\left(\sum_s b_{k,s}\right) - \xi_k\left(\mathbf{b}_{-k}, \mathbf{b}_k\right). \quad (5)$$

Let us define $[x]^+ = \min\{0, x\}$. The following Lemma shows that each tenant k balances the total load over the core switches in order to maximize its own utility.

Lemma 1. *For the total bandwidth allocation of* $\hat{b}_k \in [0, b_k^{\max}]$*, there exist a* unique *constant level* $\lambda^*(\hat{b}_k) > 0$*, such that the utility maximizing bandwidth schedule* $\hat{\mathbf{b}}_k(b_k) : \sum_{s\in\mathcal{S}} \hat{b}_{(k,s)} = \hat{b}_k$*, is* uniquely *derived as in the following:*

$$\hat{b}_{k,s}(b_k) = [\lambda^*(b_k) - (D_s + \sum_{j\in\mathcal{K}/\{k\}} b_{j,s})]^+, \qquad \forall s \in \mathcal{S}. \quad (6)$$

V. Best Response Update Process

In this section, we find the socially optimal bandwidth schedule through a distributed update process. We consider selfish and rational tenants that do not necessarily reveal their private information, such as satisfaction function, $\mathcal{U}_k(\cdot)$, and $b_k^{\max}$, to the cloud provider. Without knowing these information, the cloud provider is not able to find the socially optimal bandwidth schedule in a centralized manner. To find the optimal bandwidth schedule, we propose a decentralized bandwidth allocation framework, in which the cloud provider uses an asynchronous-based best response strategy process [14] to allocate the bandwidth for the tenants.

Let $\mathbf{b}^m$ denote the updated bandwidth schedules of the tenants over the core switches at iteration step m. The cloud provider reports the updated bandwidth payment function of tenant k at step $m+1$, as $\xi_k^{m+1}(\mathbf{b}_k^m, \mathbf{b}_k)$, $\forall k \in \mathcal{K}$. Tenant k updates its bandwidth schedule, $\mathbf{b}_k^{m+1}$, to maximize its individual utility:

$$\mathbf{b}_k^{m+1} = \arg\max_{\mathbf{b}_k\in\mathcal{B}_k} \mathcal{F}_k(\mathbf{b}_{-k}^m, \mathbf{b}_k) \quad (7)$$

The optimal bandwidth schedule in (7) is calculated based on its load balancing property in Lemma 1.

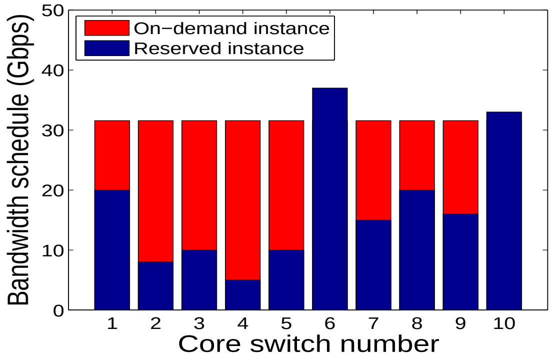

Fig. 2: Total load for (a) on-demand and (b) reserved instance tenants over the core switches.

For convergence of the best response strategy, each tenant must updates its bandwidth, at least once, in a limited number of $N+K$ successive updates. After N updates, if a tenant fails to update its bandwidth allocation, the cloud provider requests for its bandwidth update. If the tenant does not respond to the cloud provider's request, the cloud provider will drop it out of the update process.

From the strictly concavity of the social welfare function and the utility functions of the tenants, the proposed best response strategy update process converges to a socially optimal bandwidth schedule.

Theorem 1. *The best response update process converges to a socially optimal bandwidth schedule.*

VI. Numerical Results

In this section, we evaluate and verify the performance of our proposed pricing policy for the on-demand bandwidth allocation in the cloud network through the simulation results. We consider a cloud network with $S = 10$ core switches each with the maximum bandwidth capacity of $B^{\max} = 40$Gbps. We assign the utility function $\mathcal{U}_k = \log(1+b_k)$ for consuming b_k amount of bandwidth for tenant k. The cloud provider sets the bandwidth cost for each tenant to minimize the congestion cost over the core switches. We consider the nonlinear pricing policy $\mathcal{V}(x) = \beta(1 + \frac{x}{B^{\max}})^2$ to determine the bandwidth payment function for the tenants. We assume that the total load for reserved instance customers, D_s, $s = 1, ..., 10$, is reserved over the core switches.

Figure 2 shows the total reserved bandwidth for reserved and on-demand instance tenants. The simulation is resulted from running the best response strategy for 10000 number of updates with $K = 40$ homogeneous on-demand instance tenants. We have set the congestion cost factor $\beta = 0.36$ to achieve 80 percent congestion degree over the core switches. We have assumed that the Reserved instance tenants bandwidth schedule, D_s, is predetermined as shown in Fig. 2. The On-demand instance tenants compete for the remaining bandwidth over the switches. This has resulted in the balanced load over the core switches as mentioned in Lemma 1.

In order to encourage tenants to demand more bandwidth over the core switches, the cloud provider decreases the

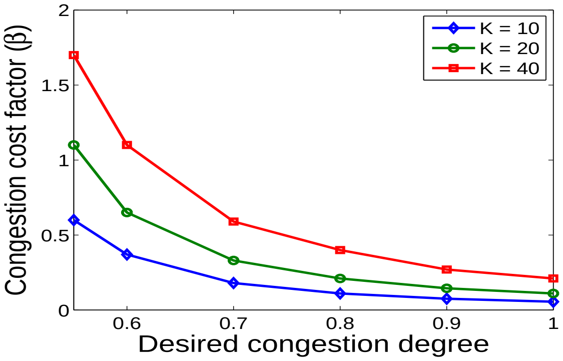

Fig. 3: The variations of the congestion cost factor (β) to achieve the total desired congestion degree over the core switches for $K = 10, 20$ and 40 number of the tenants.

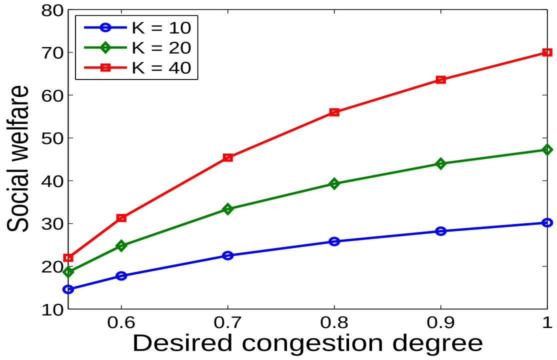

Fig. 4: The social welfare of the on-demand instance tenants vs total congestion degree over the core switches for $K = 10, 20$ and 40 number of the tenants.

congstion cost factor β. Decreasing the congestion cost factor (β) decreases the bandwidth payment cost of the tenants, and consequently increases the total bandwidth consumption and the total congestion degree over the core switches that is defined as $C_d = \frac{\sum_{s \in S} B_s}{S \times B^{max}}$. Figure 3 shows how the cloud provider decreases the cost factor (β) in order to achieve its desired congestion degree over the core switches.

Figure 4 demonstrates the social welfare of the tenants increases as the cloud provider decreases the cost factor β to achieve the desired congestion degree over the core switches. This is mainly due to the fact that decreasing the congestion cost factor β decreases the bandwidth payment cost for the tenants, and consequently increases users' satisfaction. Figure 4 also shows that for a fixed congestion degree, the social welfare of the tenants increases due to decrease of the cost factor β and increase of the total satisfaction of the tenants.

Figure 5 shows the convergence ratio of the total bandwidth of the tenants as the bandwidth update steps increases. As this figure shows, we achieve a convergence ratio less than of 10^{-3} for $K = 10$, 20 and 40 number of tenants with running the best response strategy for $M = 60$, 130 and 265 number of bandwidth updates. This figure shows that the convergence ratio reduces through a exponentially decaying function as the number of bandwidth updates increases. Furthermore, the number of bandwidth updates to achieve the convergence ratio of 10^{-3} increases proportional to the number of the tenants updating their bandwidth schedules.

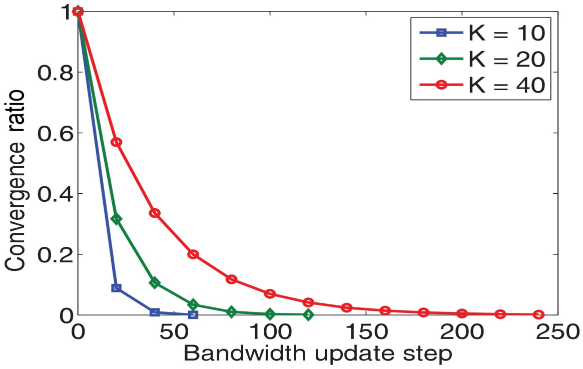

Fig. 5: The convergence ratio of the total bandwidth of the tenants vs bandwidth update steps for $K = 10, 20$ and 40 number of the tenants.

VII. Concluding Remarks

In this paper we proposed load-dependent bandwidth pricing for congestion control over the core switches in the cloud networks. The aim of the paper is to maximize the total satisfaction functions of the tenants while avoiding the congestion and overloading over the core switches. We utilized a nonlinear pricing mechanism that results in a load-balancing bandwidth schedule over the switches. In order to find the optimal solution, we performed a decentralized best response strategy with asynchronous update process. We verified our results with numerical simulations for different number of the tenants.

Acknowledgements

This research was supported in part by U.S. NSF grants NSF-1404981, IIS-1354123, CNS-1254006, IBM Faculty Award 5501145 and Microsoft Research Faculty Fellowship 8300751.

References

[1] Amazon Web Services, "What is cloud computing," http://aws.amazon.com/what-is-cloud-computing/, [Accessed in July 2015].

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010.

[3] Amazon Web Services, "Aws case study: Netflix," http://aws.amazon.com/solutions/case-studies/netflix, [Accessed in July 2015].

[4] A. Greenberg and et al, "Towards a next generation data center architecture: Scalability and commoditization," in *Proc. ACM PRESTO '08*, New York, NY, USA, 2008, pp. 57–62.

[5] F. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research society*, pp. 237–252, March 1998.

[6] S. Yang and B. Hajek, "VCG-Kelly Mechanisms for Allocation of Divisible Goods: Adapting VCG Mechanisms to One-Dimensional Signals," *IEEE J. Selected Areas Comm.*, vol. 25, no. 6, pp. 1237–1243, August 2007.

[7] Y. Haikel, R. Mazumdar, and C. Rosenberg, “A game theoretic framework for bandwidth allocation and pricing in broadband networks,” *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 667–678, Oct 2000.

[8] e. a. Chen, “Survey on routing in data centers: Insights and future directions,” *IEEE Network*, vol. 25, no. 4, pp. 6–10, July 2011.

[9] “Amazon ec2 pricing,” http://aws.amazon.com/ec2/pricing/, [Accessed in June 2015].

[10] H. Jin, X. Wang, S. Wu, S. Di, and X. Shi, “Towards optimized fine-grained pricing of IaaS cloud platform,” *IEEE Transactions on cloud Computing*, 2014.

[11] E. Behnam and A. Smith, “The amordad database engine for metagenomics,” *Oxford J. Bioinformatics*, vol. 30, no. 20, pp. 2949–2955, June 2014.

[12] D. Niu, C. Feng, and B. Li, “Pricing cloud bandwidth reservations under demand uncertainty,” in *Proc. of SIGMETRICS*, 2012.

[13] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.

[14] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*, 1st ed. Cambridge, MA: The MIT Press, 1994.