Multimedia Processing Pricing Strategy in GPU-accelerated Cloud Computing

著者	LI He, OTA Kaoru, DONG Mianxiong, V. VASILAKOS Athanasios, NAGANO Koji
journal or publication title	IEEE Transactions on Cloud Computing
page range	1-1
year	2017-02-22
URL	http://hdl.handle.net/10258/00010159

doi: info:doi/10.1109/TCC.2017.2672554

Multimedia Processing Pricing Strategy in GPU-accelerated Cloud Computing

He Li, *Member, IEEE,* Kaoru Ota, *Member, IEEE,* Mianxiong Dong, *Member, IEEE,* Athanasios V. Vasilakos, *Senior Member, IEEE,* Koji Nagano

Abstract—Graphics processing unit (GPU) accelerated processing performs significant efficiency in many multimedia applications. With the development of GPU cloud computing, more and more cloud providers focus on GPU-accelerated services. Since the high maintenance cost and different speedups for various applications, GPU-accelerated services still need a different pricing strategy. Thus, in this paper, we propose an optimal GPU-accelerated multimedia processing service pricing strategy for maximize the profits of both cloud provider and users. We first analyze the revenues and costs of the cloud provider and users when users adopt GPU-accelerated multimedia processing services then state the profit functions of both the cloud provider and users. With a game theory based method, we find the optimal solutions of both the cloud provider's and users' profit functions. Finally, through large scale simulations, our pricing strategy brings higher profit to the cloud provider and users compared to the original pricing strategy of GPU cloud services.

Index Terms—Multimedia, GPU-accelerated, Cloud Computing, Pricing

1 Introduction

Recent years, with significantly improved efficiency, the graphic processing unit (GPU) plays more and more important role in multimedia processing applications, such as GPU-accelerated video encoding and image processing [1][2]. Meanwhile, some GPU-equipped cloud providers begin to provide GPU-accelerated cloud computing services [3]. Thus, as GPU devices bring high cost and energy consumption, deploying GPU-accelerated multimedia processing services in clouds is a scalable and flexible solution [4][5].

In GPU-accelerated cloud computing, a fundamental technology is GPU virtualization [6]. Early GPU virtualization technologies are based on the remote procedure call technology which sends GPU related system calls to special virtual machines with GPU devices [7]. It is hard to isolate different tasks with negligible performance degradation [8][9]. As the later I/O virtualization seems a solution that can support full GPU utilization in virtualized environment, GPU devices are not shared between different virtual machines with simple device mapping [10]. GPU cloud computing services become realistic due to GPU virtualization developed by GPU vendors to support full isolation and sharing between virtual machines [11].

As GPU cloud computing service brings new opportunity to the commercial cloud market, it still needs a new strategy for pricing the new cloud resource [12]. Usually, high performance GPU devices bring a much higher cost than general processors mainly including the additional rack space and energy consumption [13]. Straightforwardly, The cloud provider has to use a higher price of GPU resources than general services to cover the additional cost [14].

However, users will not choose GPU-accelerated services with expensive prices as the performance of GPU acceleration is not always higher than general computing. As general cloud computing resources are much more than GPU resources, users prefer to use more general computing resources rather than use expensive GPU services [15]. The cloud provider needs to use reasonable prices of GPU-accelerated services to motivate users.

Meanwhile, as speedup ratios between applications are different, it needs an on-demand pricing strategy with different workloads [16]. Therefore, GPU-accelerated services need a new pricing strategy instead of existing strategies which only consider homogeneous resources such as processors, memories and storage space in the cloud environment.

In this paper, we first analyze the main scenario of multimedia processing services in GPU-accelerated cloud computing. With this scenario, we discuss the main motivations of the pricing strategy from both the cloud provider and users. We consider the cloud provider should use a varying prices for users with different applications and users can arrange their tasks with different speedup ratios and the prices. Then, we model the payoffs of the cloud provider and users and state interaction between the cloud provider and users as a leader-follower (Stackel) game. In the first stage, the cloud provider decides the prices of GPU and general resources for each user. Accordingly, in the second stage, every user decides how many tasks should be executed

- *H. Li, K. Ota, M. Dong and K. Nagano are with Department of Information and Electronic Engineering, Muroran Institute of Technology, Muroran, Hokkaido, Japan. E-mail: {heli, ota, mxdong, nagano}@mmm.muroran-it.ac.jp*
- *Athanasios V. Vasilakos is with Lulea University of Technology, Sweden. E-mail: vasilako@ath.forthnet.gr*

with GPU-acceleration, and finds the game equilibrium. The game model with equilibrium analysis can apply different system settings, including the scale of the cloud provider, the speedup ratios of applications, the user's utility, etc. As a result, it is possible to apply the derivation of the optimal decisions to other heterogeneous cloud resources.

To evaluate our work, we add the GPU cloud instance into cloudsim [17], a popular cloud simulation framework, to simulate GPU-accelerated cloud computing. We use the speedup ratio data from the GPU vendor with different applications. In simulations, we compare the payoffs of the cloud provider between our pricing strategy and prices from commercial cloud providers. From the simulation results, we find our pricing strategy brings better payoffs to the cloud providers.

The main contributions of this paper are summarized as follows.

- We first study the pricing problem to maximize the payoff of GPU-accelerated services. Since GPU-accelerated cloud computing is a prospective technology, our work is the first work to optimize the payoff of the cloud provider.
- We then design the optimal pricing strategy to balance the maintenance cost of GPU devices and the speedup ratios of GPU-acceleration. It is a challenging problem which needs to understand thoroughly the impact of pricing strategy in GPU-accelerated cloud computing.
- We model the interaction of the cloud provider and users as a two-stage Stackelberg game, and analyze the game equilibrium. The analysis is generic and use variable system settings, which is applicable to different GPU-accelerated cloud computing scenarios.
- We take the performance evaluation of the strategy with extensive simulations with settings from realistic GPU cloud providers. We also compare our pricing strategy with some other pricing methods and the results show our strategy performs better than others.

The rest of this paper is summarized as follows. Section 2 reviews the related work. Our network scenario and motivation are introduced in Section 3. Section 4 presents the problem formulation. An optimal pricing strategy is proposed in Section 5. Section 6 gives the simulation results. Finally, Section 7 concludes this paper and give the future work.

2 Related Work

In this section, we first introduce some main technologies of GPU-accelerated services in cloud computing. Then, we discuss some pricing strategies in cloud computing.

2.1 GPU-accelerated cloud computing

With the rapid development of general-purpose computing on graphics processing units (GPGPU), GPU-acceleration can improve the performance of many general computing applications [18][19]. As the closed structure and the difficulty of I/O virtualization, GPUs are still considered as scarce resources [20][21]. However, with its high performance, many works focused on GPU-accelerated cloud computing [22][23].

In the past decade, there are two solutions to provide GPU-acceleration in cloud computing, including general I/O virtualization and GPU virtualization [24]. General I/O virtualization means the GPU and other I/O devices are virtualized and each virtual machine can access virtual devices. Unlike simple block or character devices such as disks and network interface cards [25], it is very hard to divide complex GPU devices [26].

Some hardware companies, such as Intel and AMD, propose the I/O virtualization to support assign I/O devices to virtual machines [27][28]. As devices are considered as PCI-express (PCIe) devices, it is possible to bind the PCIe devices to virtual machines[29]. With bound GPU devices, virtual machines have GPU computing resources as the same as general physical machines.

However, the binding between virtual machines and GPU devices needs several devices in a single physical server to support multiple virtual machines. Meanwhile, GPU resources are not flexible for different applications. Therefore, some previous work proposed GPU specific virtualization technology including GPU library virtualization [30] and virtualization in GPU devices.

GPU library virtualization is a technology that modifying GPU graphic or general computing library in the virtual machines and moving workloads from general virtual machines to the virtual machines with GPU device [21]. GPU library virtualization is a very flexible solution that GPU resources can be divided with the requirement of computing tasks. However, additional overheads in workload moving and isolation risks stop the cloud provider considering GPU devices as the computing resources in cloud computing.

Virtualization in GPU devices is a final solution for GPU-accelerated cloud computing [16]. The GPU vendors implement virtualization in their products that a single GPU device is able to be virtualized into multiple virtual GPU devices that have same functions with the physical one. Therefore, virtual machines can use virtual GPU to support different applications, especially the GPGPU tasks. With GPU virtualization, several cloud companies begin to provide GPU-accelerated cloud instances. In this paper, we focus on the pricing strategy of the GPU-accelerated services.

2.2 Cloud pricing strategy

As the cloud computing is a very important commercial model, the pricing strategy is a very important issue for both academies and companies [31][32][33][34].

The pricing strategy of commercial cloud services is usually considered as sensitive intelligence. With different discount and varying prices, the final cost is not completely consistent with the initial open prices. Agmon Ben-Yehuda et al. [35] analyzed the instance spot price histories of Amazon EC2 which is one of most successful cloud services. From the analysis results, the prices usually seem not to be market-driven and the Amazon company generates prices from within a tight price interval via a dynamic hidden reserve price.

Some researchers also proposed some optimal pricing strategies as suggestions to cloud providers. Hadji et al. [36] proposed an optimal suggested pricing strategy by the providers and the optimal user demands. According to the demands and the updated price requests, the model provides different prices for the cloud provider. Moreover, with the Stackelberg game theoretical model, the strategy consists in finding the game equilibrium. However, as the pricing strategy only considered the IaaS environment with instance pricing, it is not appropriate to the multimedia processing services.

Furthermore, there are some other useful theoretical works focused on pricing strategies in cloud computing. Sharma et al. [37] applied a financial economic model for a statement of the pricing problem in cloud computing. With the financial economic model and Moore's law, the pricing strategy found the lower and upper boundaries of prices for the customers. On these two boundaries, the pricing strategy is considered beneficial for both customers and cloud providers. However, this pricing strategy only considers homogeneous resources and services.

For heterogeneous resources and services, the pricing strategy becomes more complex with different cost and revenues. Mihailescu and Teo [38] proposed a dynamical pricing strategy in federated clouds in which resources are shared among many cloud service providers. Moreover, the dynamic pricing strategy also supports heterogeneous sources and users in federated clouds and brings better buyer welfare and more successful request than fixed pricing strategies.

As the dynamic pricing strategy seems better for the cloud environment, the difference between fixed and dynamic prices is studied deeply. Yeo et al [39] considered fixed prices could not be fair to different users with different request even though fixed prices were more straightforward for customers. Therefore, they proposed a strategy to charge variable prices with reservation which lets users understand the exact costs that are calculated when users take the reservation. Thus, pricing with reservation becomes a better strategy for users rent resources from cloud providers.

As a result, even though there is no direct pricing strategy for GPU-accelerated cloud services, the dynamic and reservation based pricing strategies seem appropriate from previous works. Therefore, we try to design a strategy that the cloud providers can dynamically price user resources based on the required workloads.

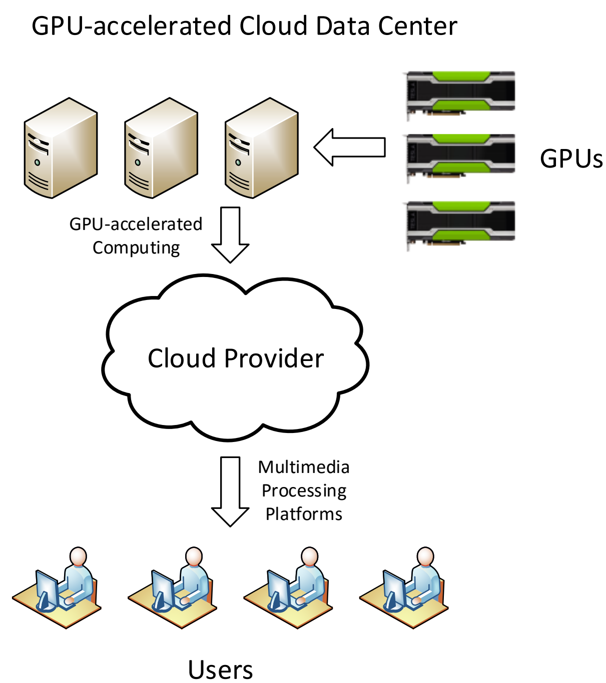


Fig. 1. Cloud provider encapsulates GPU-accelerated computing to the users for multimedia processing

3 Background and Motivation

In this section, we first introduce the scenario of multimedia processing services from GPU-accelerated Cloud Providers. Then, we discuss the motivations on the pricing strategy of the GPU-accelerated services.

3.1 Multimedia processing services from GPU-accelerated cloud providers

As GPU-accelerated computing becomes popular, some cloud providers such as nVidea and Amazon begin to provide GPU-accelerated services. As shown in Fig. 1, the cloud provider encapsulates the GPU-accelerated computing resources and provides multimedia processing services to users. Usually, since GPUs bring much higher energy consumption and heat than the general processors, the cloud provider only equips a part of servers with GPUs. Thus, we consider this type of cloud data center as GPU-accelerated cloud instead of pure GPGPU cloud.

Then, as cloud computing adopts virtualized machines as service units, an important problem is the solution to assign GPUs to instances. In general virtualization, as the GPU vendors only provide closed device drivers, it is very hard to virtualize GPU hardware to multiple GPU instances for equipment in different virtual machines. Thus, there are several methods focus on this problem including GPU virtualization and I/O virtualization discussed in the related work section. As the GPU virtualization provides more dynamical and flexible virtualization and we focus on the GPU-accelerated general purpose computing rather than GPU-accelerated computer vision, we consider the GPU-accelerated service is based on the GPU virtualization in the scenario.

Furthermore, another problem is the way to encapsulate computing resources such as processors, memory and GPU. In traditional cloud computing services, there are several levels including Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Service as a Service (SaaS). A flexible way is choosing virtual machines/instances as the encapsulation of computing resources which is adopted in IaaS level. In the GPU-accelerated cloud computing, as we choose GPU virtualization as a major way to share GPUs between users, we consider the PaaS level encapsulation that providers provide several multimedia processing platforms equipped with GPU-accelerated libraries is a more appropriate solution.

Therefore, with the multimedia processing platforms, users can deploy their tasks equipped with GPU-accelerated libraries for GPU-accelerated processing. Meanwhile, when the speedup of some applications by GPU-acceleration is not obvious, we assume that users only choose general computing for task processing.

3.2 Motivation

The pricing strategy in the mentioned scenario is very important with various user requirements and the GPU computing resources. First, as the cost of GPU computing resources is higher than general resources, it needs a higher price to cover the additional cost. For example, a typical GPU card such as nVidea Tesla k40 has a thermal design power (TDP) of 235 watts while the 18 cores Intel Xeon E7-8895 v3 has 175 watts. As the general processors can always work and the GPUs only work in the GPU-acceleration, it needs a different price to cover the GPU maintenance cost.

Second, as the GPU-accelerated performance is different between applications, high priced GPU resources will lead to potential users choose other solutions instead of GPU acceleration. From the evaluation results of existing works, GPU-accelerated applications have different speedup ratios compared to the original versions. From the report of NVidia, CAFFE, a machine learning application, can have a speedup ratio of 14 times than the CPU version while Quantum espresso in materials science has nearly the same performance. If the additional cost from GPU is higher than the utility, most users will not choose the GPU-acceleration services.

Third, as there are a few of cloud providers have GPU-accelerated services, it is hard to consider the GPU resources as unlimited. Thus, the relationship between users and cloud providers is unequal with the scarce GPU resources. Actually, until 2016, there are only six providers have GPU cloud services in the market. Thus, we consider the status of the providers is higher than users in the pricing strategy.

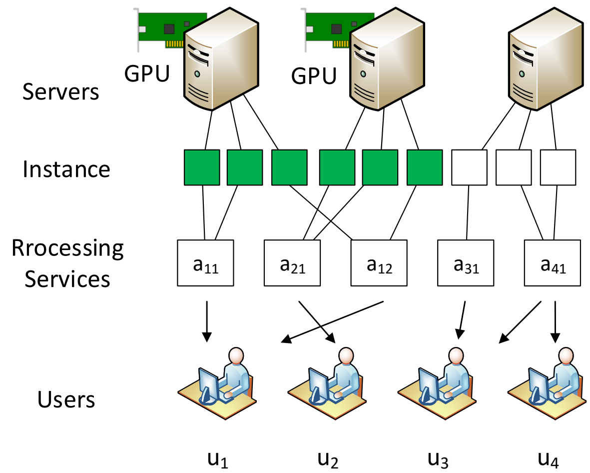


Fig. 2. Illustration of the GPU-accelerated and general services in GPU-accelerated cloud computing

4 Problem Statement

In this section, we first model the cloud provider provides multimedia processing services in GPU-accelerated cloud computing then state the problem of pricing strategy of the services to users.

As shown in Fig. 2, users purchase multimedia processing services from the cloud providers with cloud instances. In cloud instances, there are two types including GPU equipped and general instances. Users can choose GPU-accelerated and general services for different multimedia processing tasks. Thus, we use set $U = \{u_1, u_2, u_3, ..., u_{|U|}\}$ to denote the users who want to use the GPU-accelerated services. In the pricing problem, each user has different tasks. Thus, we use a_{ij} to denote one task of user i and a set $A_i = \{a_{i1}, a_{i2}, a_{i3}, ..., a_{i|A_i|}\}$ to denote the user i's all tasks. As the GPU-accelerated speedup ratio of each task is different, we use s_{ij} to denote the speedup ratio of task a_{ij}.

Then, we discuss the GPU-accelerated service from the cloud provider. As the user needs to choose instances with or without GPU-acceleration, we define a value $x_{ij} \in [0, 1])$ to denote the ratio of workload with GPU-acceleration in task a_{ij}.

Thus, we define a set $X_i = \{x_{i1}, x_{i2}, x_{i3}, x_{i|A|}\}$to denote the decision of user i to arrange GPU-acceleration to task set A_i. To describe the time cost of each task, we consider a workload unit that a task can be finished in a time unit with a unit of general computing resource. Therefore, we can define a value l_{ij} to denote the number of workload units of task a_{ij} and a set $L_i = l_{i1}, l_{i2}, l_{i3}, ..., l_{i|A_i|}$ to denote the workload unit numbers of tasks of user u_i. With the definition of workload units, we use t_{ij} to denote the entire time to serially finish task a_{ij} as

$$t_{ij} = \frac{l_{ij}}{s_{ij}} \cdot x_{ij} + l_{ij} \cdot (1 - x_{ij}), \tag{1}$$

while $u_i \in U$ and $a_{ij} \in A_i$ and a set $T_i = \{t_{i1}, t_{i2}, t_{i3}, ..., t_{i|A_i|}\}$ to denote the executing time of all tasks of user u_i.

Now, we discuss the price in the GPU-accelerated cloud. As the workload of each user is different, the pricing strategy of the cloud provider sets a different price for each user. Meanwhile, as discussed in Section 3.2 that the cost of GPU computing resource is higher the general computing resource, to user u_i, we use p_i^c and p_i^g to denote the one time unit price for general computing resources and GPU-accelerated computing resources, respectively. Meanwhile, as the general computing services are not unique, the price p_i^c is not higher than the other providers for similar services. We define a value p^m where $p_i^c \leq p^m$ to denote the maximum price of p_i^c. With the different price of computing resource, we use C_i to denote the cost for user u_i to finished its tasks as

$$C_i = \sum_{j=1}^{|A_i|} \frac{l_{ij}}{s_{ij}} \cdot x_{ij} \cdot p_i^g + l_{ij} \cdot (1 - x_{ij}) \cdot p_i^c \tag{2}$$

while $p_i^c \leq p_i^g$ and $p_i^c \leq p_i^m$.

Thus, as the results of task processing bring utility to the users, we use a Utility $U_i(L_i, X_i)$ function to denote the utility that user u_i can receive from processing all tasks. As we seek an elastic model of the pricing strategy, the user utility function is compatible with multiple previous models [40] [41]. Therefore, we use a function $O_i^U(X_i; p_i^c, p_i^g)$ to denote the payoff of user u_i when choosing a strategy (X_i) as

$$O_i^U(X_i; p_i^c, p_i^g) = U_i(L_i, X_i) - C_i. \tag{3}$$

To the cloud provider, we first discuss the cost for each unit of computing resources including general and GPU-accelerated computing resources. We use e_c and e_g to denote the cost of one unit of general and GPU-accelerated computing resource, respectively. Thus, we can find E_i to denote the cost for providing service to user u_i as

$$E_i = \sum_{j=1}^{|A_i|} \frac{l_{ij}}{s_{ij}} \cdot x_{ij} \cdot e_g + l_{ij} \cdot (1 - x_{ij}) \cdot e_c. \tag{4}$$

With the service cost for user u_i, it is easy to find a function $O_i^P(p_i^c, p_i^g; X_i)$ to denote the payoff of the cloud provider with the pricing strategy (p_i^c, p_i^g) as

$$O_i^P(p_i^c, p_i^g; X_i) = C_i - E_i. \tag{5}$$

We list all notations in the pricing strategy of the GPU-accelerated multimedia processing service in Table 1. As there are limited cloud providers have GPU-accelerated services, we consider the cloud provider as the leader in the game with the cloud users. Thus, the pricing problem of the GPU-accelerated multimedia processing services can be considered as a two level Stackelberg game between the users and the cloud provider. A Stackelberg game is a leadership model in economics in which the leader firm moves before the follower. In game terms, game players are a leader and a follower and they compete on quantity. Thus, in our model, the game players are the cloud provider and users. In the first stage, the cloud provider (leader) decides the price of general and GPU-accelerated computing resources for maximizing its payoff. The object of the cloud provider is to maximize its payoff, which consists of revenue from the user paid for cloud services, and the cost of maintaining the general and GPU-accelerated computing resources. In the second stage, under the decisions from the leader, user u_i decides whether tasks need GPU-acceleration. The payoff of each user u_i depends on the utility U_i from the finished task workloads and the payment on general and GPU-accelerated computing resources.

TABLE 1
Notations in the pricing problem of the GPU-accelerated multimedia services

U	Set of cloud users
u_i	One user in U
A_i	Task set of user u_i
a_{ij}	One task in A_i
s_{ij}	Speedup ratio of a_{ij} with GPU-acceleration
X_i	All decisions for GPU-accelerated tasks of u_i
x_{ij}	Ratio of workload in a_{ij} is accelerated with GPU
L_i	All task workloads of u_i
l_{ij}	Number of workload units of a_{ij}
t_{ij}	Time for executing a_{ij}
p_i^c	General computing resource for u_i
p_i^g	GPU-accelerated computing resource for u_i
C_i	Cost for executing all tasks of u_i
$U_i(\cdot)$	Utility of executing all tasks of u_i
$O_i^U(\cdot)$	Payoff function of u_i
e_c	Cost of one unit general computing resource
e_g	Cost of one unit GPU-accelerated computing resource
E_i	Cost for executing all tasks of u_i
O_i^P	Payoff function of the cloud provider from u_i

5 OPTIMAL PRICING STRATEGY

In this section, we study the provider-user game under complete information, where both the cloud provider and the users know all system parameters mentioned above. We solve the game by backward induction. First, we solve the user's best GPU-acceleration decision strategy in the second stage. Then, we study the provider's best pricing strategy in the first stage.

5.1 Best decision of users in the second stage

We assume that the number of user tasks is elastic that the analysis can be easily extended to other scenarios. Specifically, give the provider's pricing strategy (p_i^{c*}, p_i^{g*}), user u_i can derive the optimal assignment strategy (X_i) by solving the problem as

$$\begin{aligned} &\max_{X_i} \quad O_i^U(X_i; p_i^{c*}, p_i^{g*}) \\ &s.t., \qquad x_{ij} \in [0,1], \qquad i \in [1, |U|], j \in [1, |A_i|]. \end{aligned} \tag{6}$$

It is easy to check that (6) is a convex optimization. We first study the optimal strategy (x_{ij}^*) with a particular task a_{ij} (fixed the scheduling decisions in other $|A_i| - 1$ tasks), and then study the optimal strategy $(X_i^*) = (x_{ij}^*)_{a_{ij} \in A_i}$ of all $|A_i|$ tasks jointly which is the solution of (6).

Now we consider the strategy of a single task a_{ij}. We first use a strategy way that converges to the optimal single-task strategy. Then, we characterize the optimal scheduling step by step.

We use o_{ij} denote the first-order derivatives of payoff $O_i^U(\cdot)$ for user u_i with respect to a_{ij} as

$$o_{ij}(x_{ij}) \triangleq \frac{dO_i^U(x_{ij})}{dx_{ij}} = U_i'(l_{ij}, x_{ij}) - \frac{l_{ij}}{s_{ij}} \cdot p_i^{g*} + l_{ij} \cdot p_i^{c*}. \tag{7}$$

As we assume the utility function is derivable, the value of $O_i^U(x_{ij})$ when $o_{ij}(x_{ij}) = 0$ should be the maximum. We use x_{ij}^d to denote the value when $o_{ij}(x_{ij}) = 0$ as

$$x_{ij}^d = \arg_{x_{ij}}(o_{ij}(x_{ij}) = 0). \tag{8}$$

With the value of x_{ij}^d, we can get the maximum payoffs of the users. While the solution will exceed the range of ratio x_{ij}, we study the value of $o_{ij}(x_{ij}^d)$ to find the solution in $[0, 1]$. Obviously, as the payoff of user u_i is monotonic increasing when $o_{ij}(1) > 0$ and $x_{ij}^d \notin [0, 1]$, the optimal solution is $x_{ij} = 1$. Otherwise, the optimal solution is $x_{ij} = 0$ when $o_{ij}(1) < 0$ and $x_{ij}^d \notin [0, 1]$. We use a value x_{ij}^* to denote the solution for maximizing the payoff of user u_i's decision on task a_{ij} as

$$x_{ij}^* = \begin{cases} 1, & x_{ij}^d \notin [0,1], o_{ij}(1) > 0, \\ 0, & x_{ij}^d \notin [0,1], o_{ij}(1) < 0, \\ x_{ij}^d, & x_{ij}^d \in [0,1]. \end{cases} \tag{9}$$

Since the workloads of each task are not divided infinitely, the unit number of workloads is integer. With solution in (9), we design an algorithm to decide the optimal ratio of the workloads processed by GPU-acceleration in task a_{ij}. As shown in Algorithm 1, the strategy adds one unit workload for GPU-acceleration in each loop until the ratio exceeds the optimal value.

Algorithm 1 Single Task Strategy

$x_{ij}^* \leftarrow 0$;
$l_{ij}' \leftarrow 0$;
while $x_{ij}^* \leq 1$ **do** and $o_{ij}(x_{ij}^*) > 0$
  $l_{ij}' \leftarrow l_{ij} + 1$;
  $x_{ij}^* \leftarrow \frac{l_{ij}'}{l_{ij}}$;
end while

Now we study the optimal strategy $(X_{ij}^*) = (x_{ij}^*)_{a_j \in A_j}$ with the task set A_i of user u_i. As we assume tasks are isolated in the cloud environment, the resource assignment of each task is independent. Thus, we can get the optimal solution of X_i as

$$X_i^* = \bigcup_{j=1}^{|A_i|} x_{ij}^* \tag{10}$$

while s_{ij}^* is calculated in (9).

Thus, for less time complicity, we choose an optimization learning from the binary search algorithm and propose an algorithm for solute the GPU-accelerated ratio X_i of user u_i as Algorithm 2.

Algorithm 2 Strategy for task set A_i

$X_i^* \leftarrow \varnothing$;
for $j \leftarrow 1$ to $|A_i|$ **do**
  $x_{ij}^* \leftarrow 0$;
  $l_{ij}' \leftarrow 0$;
  $l_b \leftarrow l_i$;
  $l_e \leftarrow 0$;
  while $l_b > l_e$ and $o_{ij}(x_{ij}^*) > 0$ **do**
    if $o_{ij}(x_{ij}^*) > 0$ **then**
      $l_{ij}' \leftarrow l_{ij}' + \frac{l_{ij}' + l_b}{2}$;
      $l_e \leftarrow l_{ij}'$;
      $x_{ij}^* \leftarrow \frac{l_{ij}'}{l_{ij}}$;
    else if $o_{ij}(x_{ij}^*) < 0$ **then**
      $l_{ij}' \leftarrow \frac{l_{ij}' + l_e}{2}$;
      $l_b \leftarrow l_{ij}'$;
      $x_{ij}^* \leftarrow \frac{l_{ij}'}{l_{ij}}$;
    else if $o_{ij}(x_{ij}^*) = 0$ **then**
      break;
    end if
  end while
  $X_i^* \leftarrow X_i^* \cup \{x_{ij}^*\}$;
end for

First, the algorithm sets X_i^* as an empty set and calculate each x_{ij}^* with different task a_{ij} of user u_i. For each x_{ij}^* of task a_{ij}, the algorithm uses a loop to find the optimal value by binary searching. Before the searching loop, the algorithm defines three temporary variables, l_{ij}', l_b and l_e, to denote the corresponding task workloads with x_{ij}^*, the highest and lowest inclusive workload boundaries that are searched. The binary search is not complex that the value of l_{ij}' is set to the middle value between the former value and the boundary value. If the former value is more than the solution, new l_{ij}' is set to the middle value between the former value and the highest boundary value otherwise the lowest if the former value is less than the solution. After search loop, the algorithm adds the solution of x_{ij}^* to the set X_i^*. The optimal solution of user u_i is generated after all tasks in set A_i are processed.

5.2 Best Decision of the Controller in the First Stage

Now we begin to study the problem to find the best decision of the cloud provider to maximize its payoff. The provider can derive the optimal price p_i^c and p_i^g with the u_i's decision of the task ratio X_i^* for GPU-acceleration by solving the problem as

$$\begin{aligned} \max_{p_i^c, p_i^g} \quad & O_i^P(p_i^c, p_i^g; X_i^*) \\ s.t., \quad & p^m \geq p_i^c \geq 0, \\ & p_i^g \geq 0, \\ & X_i^* \text{ is solved in (6)}, \\ & i \in [1, |U|], j \in [1, |A_i|]. \end{aligned} \tag{11}$$

To simplify the problem, we only consider the ratio x_{ij}^d of user u_i's optimal strategy is in the range $[0, 1]$. Therefore, it is easy to check that (11) is also a convex optimization. Hence, it admits an optional solution that can be characterized by the method of Lagrange multipliers.

Let $h_i^c(\cdot)$ and $h_i^g(\cdot)$ denote the first-order derivatives of the provider's payoff functions from user u_i with respect to p_i^c and p_i^g as

$$\begin{aligned} h_i^c(p_i^c, p_i^g) = \; & \frac{\partial O_i^P}{\partial p_i^c} = \frac{\partial C_i}{\partial p_i^c} - \frac{\partial E_i}{\partial p_i^c} = \\ & \textstyle\sum_{j=1}^{|A_i|} \frac{l_{ij}}{s_{ij}} \cdot \frac{\partial x_{ij}^*}{\partial p_i^c} \cdot (p_i^g - e_g) + \\ & l_{ij} \cdot (1 - \frac{\partial x_{ij}^*}{\partial p_i^c}) \cdot (1 - e_c) \end{aligned} \tag{12}$$

and

$$\begin{aligned} h_i^g(p_i^c, p_i^g) \; & = \frac{\partial O_i^P}{\partial p_i^g} = \frac{\partial C_i}{\partial p_i^g} - \frac{\partial E_i}{\partial p_i^g} = \\ & \textstyle\sum_{j=1}^{|A_i|} \frac{l_{ij}}{s_{ij}} \cdot \frac{\partial x_{ij}^*}{\partial p_i^g} (1 - e_g) + \\ & -l_{ij} \cdot (1 - \frac{\partial x_{ij}^*}{\partial p_i^g}) \cdot (p_i^c - e_c). \end{aligned} \tag{13}$$

Further, we can get the constraint function set from (8). We use p_i^{c*} and p_i^{g*} to denote the solutions of (5). Therefore, as the value of $o_{ij}(l_{ij}, x_{ij}^*) \equiv 0$, we can find the optimal solutions with the equation set as

$$\begin{cases} h_i^g(p_i^{c*}, p_i^{g*}) = 0, \\ h_i^c(p_i^{c*}, p_i^{g*}) = 0. \end{cases} \tag{14}$$

We apply Newton's method to solve the equation set. Then, we need to define the functions in iterations with (14). We use $H_i^c(\cdot)$ and $H_i^g(\cdot)$ to denote the iterative functions for calculating p_i^{c*} and p_i^{g*} as

$$\begin{cases} H_i^c(p_i^c, p_i^g) = \\ \frac{h_i^c(p_i^c, p_i^g) \cdot h_{ig}'^g(p_i^c, p_i^g) - h_i^g(p_i^c, p_i^g) h_{ig}'^c(p_i^c, p_i^g)}{h_{ic}'^g(p_i^c, p_i^g) \cdot h_{ig}'^c(p_i^c, p_i^g) - h_{ic}'^c(p_i^c, p_i^g) h_{ig}'^g(p_i^c, p_i^g)}, \\ H_i^g(p_i^c, p_i^g) = \\ \frac{h_i^g(p_i^c, p_i^g) \cdot h_{ic}'^c(p_i^c, p_i^g) - h_i^c(p_i^c, p_i^g) h_{ic}'^g(p_i^c, p_i^g)}{h_{ic}'^g(p_i^c, p_i^g) \cdot h_{ig}'^c(p_i^c, p_i^g) - h_{ic}'^c(p_i^c, p_i^g) h_{ig}'^g(p_i^c, p_i^g)}. \end{cases} \tag{15}$$

where $h_{ic}'^c(p_i^c, p_i^g) = \frac{\partial h_i^c(p_i^c, p_i^g)}{\partial p_i^c}$, $h_{ig}'^c(p_i^c, p_i^g) = \frac{\partial h_i^c(p_i^c, p_i^g)}{\partial p_i^g}$, $h_{ic}'^g(p_i^c, p_i^g) = \frac{\partial h_i^g(p_i^c, p_i^g)}{\partial p_i^c}$, and $h_{ig}'^g(p_i^c, p_i^g) = \frac{\partial h_i^g(p_i^c, p_i^g)}{\partial p_i^g}$.

As shown in Algorithm 3, we first define two temporary values $p_i'^c$ and $p_i'^g$ to denote the former calculated solutions of p_i^c and p_i^g. We first guess the values of p_i^c and p_i^g that $h_i^c(p_i^{c0}, p_i^{g0}) < 0$ and $h_i^g(p_i^{c0}, p_i^{g0}) < 0$. Then, the algorithm begins iterations to find solutions. We set a value ϵ to denote the expected distance between the iterated and the optimal solutions and the algorithm continue iterating until the values of $h_i^c(p_i^c, p_i^g)$ and $h_i^g(p_i^c, p_i^g)$ are no more than ϵ. After iterations, the algorithm can find the solutions for the cloud provider's strategy.

Algorithm 3 Strategy for the cloud provider

Find $h_i^c(p_i^{c0}, p_i^{g0}) < 0$ and $h_i^g(p_i^{c0}, p_i^{g0}) < 0$ as a given guess;
$p_i^c \leftarrow p_i^{c0}$;
$p_i^g \leftarrow p_i^{g0}$;
$p_i'^c \leftarrow 0$;
$p_i'^g \leftarrow 0$;
while $h_i^c(p_i^c, p_i^g) > \epsilon$ and $h_i^g(p_i^c, p_i^g) > \epsilon$ **do**
    $p_i^c \leftarrow p_i'^c + H_i^c(p_i'^c, p_i'^g)$;
    $p_i^g \leftarrow p_i'^g + H_i^g(p_i'^c, p_i'^g)$;
    $p_i'^c \leftarrow p_i^c$;
    $p_i'^g \leftarrow p_i^g$;
end while
$p_i^{c*} \leftarrow p_i^c$;
$p_i^{g*} \leftarrow p_i^g$;

6 Evaluation

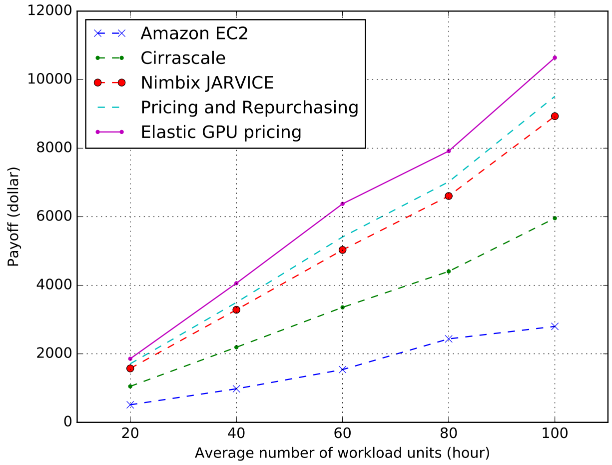
In this section, we execute extensive simulations for the pricing strategy evaluation. We first describe the settings of the simulations then discuss the results of the performance evaluation.

We use a workstation computer as the simulation platform which equips a Core™ i7 4770 (8M Cache, up to 3.90GHz) CPU, 16GByte RAM and 2TByte HDD. We use cloudsim 3.0.3 as the major simulator. We test each simulation 20 times and record the average result.

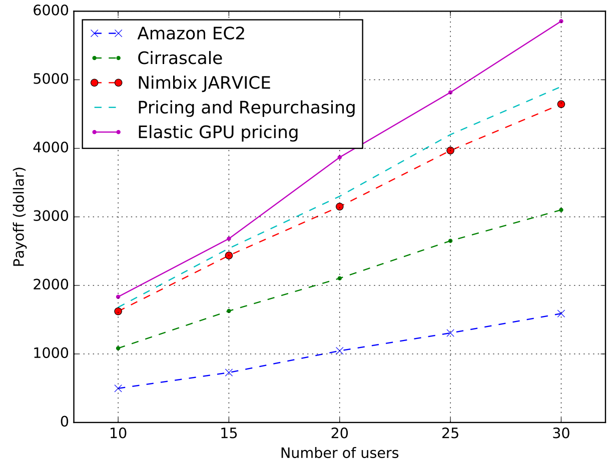
For comparison, we use the static prices from Amazone Elastic Compute Cloud (EC2)[14], Cirrascale [42] and NIMBX JARVICE [43]. As these three companies provide different hardware plans, we first choose the instance g2.2xlarge as the standard instance with 1536 CUDA units from EC2. Then, we calculate the prices of instances from other two providers with the same CUDA units. As tasks have different speedup ratios, we set the ratios according to the benchmark results of Tesla K40 from the Nvidia company. For comparing our solution with other dynamic algorithm, we also perform the pricing and repurchasing algorithm in all experiments, which focuses on the streaming processing service pricing in cloud computing [44].

We first evaluate the payoff with different workloads per each task. We set the number of workload units per each workload is uniform distributed in [10, 30], [30, 50], [50, 70] and [70, 100] in each step. And we set the number of users is 10 and the number of tasks is uniformly distributed in [10, 90]. Further, we set the cost of one unit of the GPU resource is 0.4 dollar per hour. From the results in Fig. 3(a), the payoff of the cloud provider increases with more workloads per task. Obviously, our pricing strategy brings higher payoff than other static price strategies. Meanwhile, the payoff increases with more unit prices. When the average task units become to 250 hours, the payoff with our pricing strategy becomes nearly 3 times of the payoff with the default GPU price.

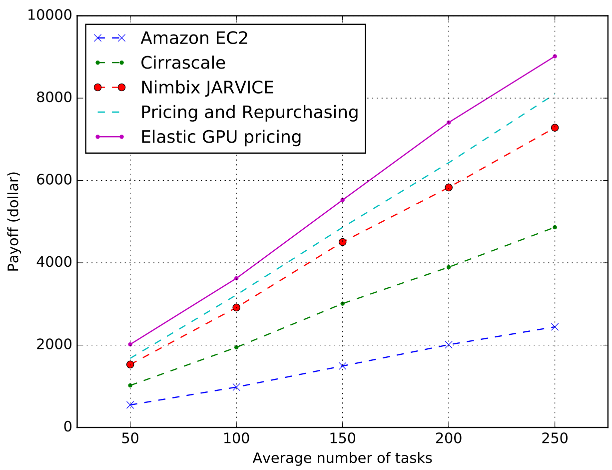
Then, we evaluate the payoff with different number of users. We set the number of users from 10 to 30 and



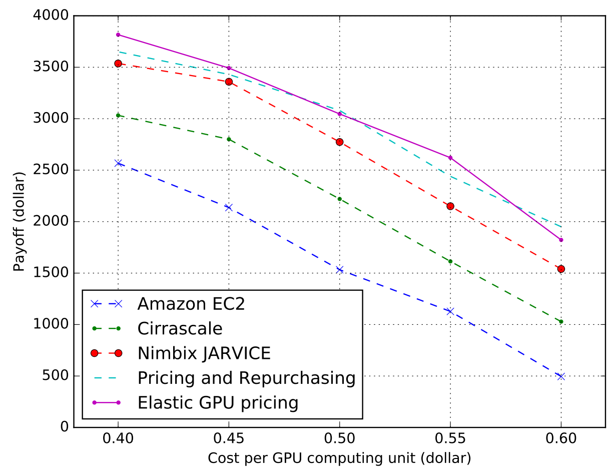
(a) Payoff with different average workloads of tasks



(b) Payoff with different number of users



(c) Payoff with different average number of tasks



(d) Payoff with different cost per GPU computing unit

Fig. 3. Payoff results with different settings of workloads, users, tasks and GPU cost

increase 5 users per each step. And the number of workload units per task and tasks is uniformly distributed in [10, 30] and [10, 90], respectively. The cost of one unit of the GPU resource stays the same. From the results show in Fig. 3(b), we can find the payoff increases with more users rent the cloud services. Our pricing strategy still performs better than the static prices. When the number of users is less than 15, the difference between our pricing strategy and the price of the Nimbix instance is similar while the gap becomes larger with more users in the cloud services.

As the task scale of users is an important issue to the cloud provider's payoff, we also evaluate the payoff with different task scales per each user. We set the number of tasks per user is uniformly distributed in [10, 90], [60, 140], [110, 190], [160, 240] and [210, 290] in each step. The number of users is set 10. Other settings stay the same with the previous simulation. From the results in Fig. 3(c), we can find the payoff of the cloud provider increases with the number of tasks. The difference between the price of our strategy and other prices is larger than other simulations. With the average task number of 50, the payoff of our pricing strategy is nearly 2000 dollars while the default price in Amazon EC2 brings no more than 600 dollars. With the average task number of 250, the payoff of our strategy is near 4 times more than the default price.

Furthermore, we evaluate the payoff with different cost of GPU resources since the GPU cost is also important to the payoff of the cloud provider. We set the cost of one unit GPU resource from 0.40 to 0.6 dollar and increase the 0.05 dollar in each step of the simulation. The number of tasks is uniformly distributed in [10, 90] and other settings stay the same with the previous experiment. From the result shown in Fig. 3(d), the payoff decreases obviously with the cost increasing. With the price of 0.65 dollar per hour, the payoff of the cloud provider is less than 500 dollars when the cost

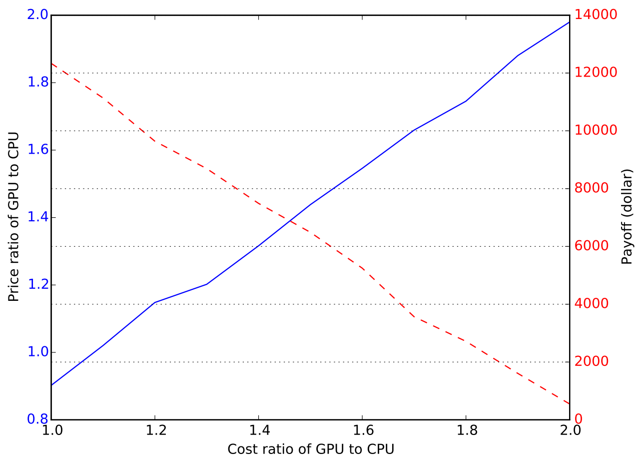


Fig. 4. Price ratio of GPU to CPU resources and payoff with different cost ratio of GPU to CPU resources

increases to 0.6 dollar per hour, which is only one fifth of the payoff with 0.4 dollar per hour. Our strategy still performs better than other prices while the difference between our strategy.

Since the GPU and CPU prices are very important to our pricing strategy, we test the price ratio of GPU to CPU resources and payoff with different cost ratio of GPU to CPU resources. We set the cost ratio of GPU to CPU from 1 to 2 and increase it by 0.1 in each step of the simulation. As shown in Fig. 4, we find the price ratio and payoff is relevant to the cost ratio between CPU and GPU. When the GPU cost is controlled to the same level of CPU cost, the payoff is maximized and the price of GPU resources is lower than CPU resources. While the cost of GPU resources increase, the payoff is decreased with high GPU price. Therefore, if the cloud provider wants to increase the revenue from GPU-accelerated cloud services, it is very important to control the cost of GPU resources with an attractive price.

Finally, we find that our pricing strategy can bring better payoff than static prices of existing cloud providers even with increased prices. As a result, the user requirement driven elastic price strategy is a better choice than the general static pricing strategy. Furthermore, even though we choose the prices from Amazon EC2, we consider our pricing strategy is also appropriate for the model that provides processing services directly.

7 Conclusion

In this paper, we propose an elastic pricing strategy for multimedia processing services in GPU-accelerated cloud environment. Unlike the static prices from existing cloud providers, the pricing strategy will provide varying prices of GPU computing resources according to the user's requirement. To maximize the payoff of both the cloud provider and users, we formulate the elastic pricing strategy as a two-stage leader-follower (Stackelberg) game, and analyze the game equilibrium. We also evaluate our pricing strategy with extensive simulations and compare the payoff with other pricing strategies. From the result of performance evaluation, the elastic pricing strategy brings more payoff to the cloud provider than other methods.

In the future, we will plan to design and implement a cloud framework to support multimedia processing services with GPU-acceleration. Meanwhile, it is significant to find the combined optimization on the resource scheduling and pricing since the virtual GPU is a very different resource from the general virtual processors. A deeper experiment with more real world testbed is also needed to evaluate the efficiency of GPU-accelerated cloud computing.

Acknowledgments

This work is sponsored by JSPS KAKENHI Grant Number 15K15976, 26730056, 16K00117, JSPS A3 Foresight Program, and Research Fund for Postdoctoral Program of Muroran Institute of Technology.

References

[1] Z. Xia, X. Wang, X. Sun, Q. Liu, and N. Xiong, "Steganalysis of lsb matching using differences between nonadjacent pixels," *Multimedia Tools and Applications*, vol. 75, no. 4, pp. 1947–1962, 2016.

[2] J. Owens, M. Houston, D. Luebke, S. Green, J. Stone, and J. Phillips, "Gpu computing," *Proceedings of the IEEE*, vol. 96, no. 5, pp. 879–899, May 2008.

[3] V. T. Ravi, M. Becchi, G. Agrawal, and S. Chakradhar, "Supporting gpu sharing in cloud environments with a transparent runtime consolidation framework," in *Proceedings of the 20th International Symposium on High Performance Distributed Computing*, ser. HPDC '11. New York, NY, USA: ACM, 2011, pp. 217–228.

[4] W. Zhu, C. Luo, J. Wang, and S. Li, "Multimedia cloud computing," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 59–69, May 2011.

[5] J. Yin, X. Lu, C. Pu, Z. Wu, and H. Chen, "Jtangcsb: A cloud service bus for cloud and enterprise application integration," *IEEE Internet Computing*, vol. 19, no. 1, pp. 35–43, Jan 2015.

[6] M. Dowty and J. Sugerman, "Gpu virtualization on vmware's hosted i/o architecture," *SIGOPS Oper. Syst. Rev.*, vol. 43, no. 3, pp. 73–82, Jul. 2009.

[7] H. A. Lagar-Cavilla, N. Tolia, M. Satyanarayanan, and E. de Lara, "Vmm-independent graphics acceleration," in *Proceedings of the 3rd International Conference on Virtual Execution Environments*, ser. VEE '07. New York, NY, USA: ACM, 2007, pp. 33–43.

[8] J. N. Matthews, W. Hu, M. Hapuarachchi, T. Deshane, D. Dimatos, G. Hamilton, M. McCabe, and J. Owens, "Quantifying the performance isolation properties of virtualization systems," in *Proceedings of the 2007 Workshop on Experimental Computer Science*, ser. ExpCS '07. New York, NY, USA: ACM, 2007.

[9] J. Yin, X. Lu, X. Zhao, H. Chen, and X. Liu, "Burse: A bursty and self-similar workload generator for cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 3, pp. 668–680, March 2015.

[10] X. Zhang and Y. Dong, "Optimizing xen VMM based on intel virtualization technology," in *Proceedings of 2008 International Conference on Internet Computing in Science and Engineering (ICICSE '08)*, Jan 2008, pp. 367–374.

[11] A. Herrera, "Nvidia grid: Graphics accelerated vdi with the visual performance of a workstation," *Nvidia Corp*, 2014.

[12] B. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *Proceedings of the Fifth International Joint Conference on INC, IMS and IDC, 2009. NCM '09.*, Aug 2009, pp. 44–51.

[13] S. Huang, S. Xiao, and W. Feng, “On the energy efficiency of graphics processing units for scientific computing,” in *Proceedings of the 2009 IEEE International Symposium on Parallel Distributed Processing (IPDPS 2009)*, May 2009, pp. 1–8.

[14] I. Amazon Web Services, “Ec2 instance pricing amazon web services (aws),” https://aws.amazon.com/ec2/pricing/, accessed Janurary, 2016.

[15] M. Dong, H. Li, K. Ota, L. T. Yang, and H. Zhu, “Multicloud-based evacuation services for emergency management,” *IEEE Cloud Computing*, vol. 1, no. 4, pp. 50–59, Nov 2014.

[16] C. NVIDIA, “Tesla gpu accelerators for servers—nvidia,” http://www.nvidia.com/object/tesla-servers.html, accessed Janurary, 2016.

[17] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, “Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[18] S. Lee, S.-J. Min, and R. Eigenmann, “Openmp to gpgpu: A compiler framework for automatic translation and optimization,” in *Proceedings of the 14th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, ser. PPoPP ’09. New York, NY, USA: ACM, 2009, pp. 101–110.

[19] D. G. Merrill and A. S. Grimshaw, “Revisiting sorting for gpgpu stream architectures,” in *Proceedings of the 19th International Conference on Parallel Architectures and Compilation Techniques*, ser. PACT ’10. New York, NY, USA: ACM, 2010, pp. 545–546.

[20] P. Karger and D. Safford, “I/o for virtual machine monitors: Security and performance issues,” *IEEE Security Privacy*, vol. 6, no. 5, pp. 16–23, Sept 2008.

[21] L. Shi, H. Chen, J. Sun, and K. Li, “vcuda: Gpu-accelerated high-performance computing in virtual machines,” *IEEE Transactions on Computers*, vol. 61, no. 6, pp. 804–816, June 2012.

[22] K. Suttisirikul and P. Uthayopas, “Accelerating the cloud backup using gpu based data deduplication,” in *Proceedings of the IEEE 18th International Conference on Parallel and Distributed Systems (ICPADS 2012)*, Dec 2012, pp. 766–769.

[23] M. Oikawa, A. Kawai, K. Nomura, K. Yasuoka, K. Yoshikawa, and T. Narumi, “Ds-cuda: A middleware to use many gpus in the cloud environment,” in *Proceedings of the 2012 SC Companion:High Performance Computing, Networking, Storage and Analysis (SCC ’12)*, Nov 2012, pp. 1207–1214.

[24] J. Duato, F. D. Igual, R. Mayo, A. J. Peña, E. S. Quintana-Ortí, and F. Silla, “An efficient implementation of gpu virtualization in high performance clusters,” in *Proceedings of Euro-Par 2009–Parallel Processing Workshops*. Springer, 2010, pp. 385–394.

[25] D. Li, X. Liao, H. Jin, B. Zhou, and Q. Zhang, “A new disk i/o model of virtualized cloud environment,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1129–1138, June 2013.

[26] V. Gupta, A. Gavrilovska, K. Schwan, H. Kharche, N. Tolia, V. Talwar, and P. Ranganathan, “Gvim: Gpu-accelerated virtual machines,” in *Proceedings of the 3rd ACM Workshop on System-level Virtualization for High Performance Computing*, ser. HPCVirt ’09, New York, NY, USA, 2009, pp. 17–24.

[27] I. AMD, “O virtualization technology (iommu) specification,” *AMD Pub*, vol. 34434, 2007.

[28] R. Hiremane, “Intel virtualization technology for directed i/o (intel vt-d),” *Technology@ Intel Magazine*, vol. 4, no. 10, 2007.

[29] J. Duato, A. Pena, F. Silla, R. Mayo, and E. Quintana-Orti, “rcuda: Reducing the number of gpu-based accelerators in high performance clusters,” in *Proceedings of 2010 International Conference on High Performance Computing and Simulation (HPCS 2010)*, June 2010, pp. 224–231.

[30] H. Li, H. Jin, and X. Liao, “Graphic acceleration mechanism for multiple desktop system based on virtualization technology,” in *Proceedings of the IEEE 14th International Conference on Computational Science and Engineering (CSE 2011)*, Aug 2011, pp. 447–452.

[31] C. Microsoft, “Pricing - cloud services — microsoft azure,” https://azure.microsoft.com/en-us/pricing/details/cloud-services/, accessed Janurary, 2016.

[32] I. Google, “Google compute engine pricing — google cloud platform,” https://cloud.google.com/compute/pricing, accessed Janurary, 2016.

[33] M. Dong, X. Liu, Z. Qian, A. Liu, and T. Wang, “Qoe-ensured price competition model for emerging mobile networks,” *IEEE Wireless Communications*, vol. 22, no. 4, pp. 50–57, August 2015.

[34] X. Liu, M. Dong, K. Ota, P. Hung, and A. Liu, “Service pricing decision in cyber-physical systems: Insights from game theory,” *IEEE Transactions on Services Computing*, vol. 9, no. 2, pp. 186–198, March 2016.

[35] O. Agmon Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafrir, “Deconstructing amazon ec2 spot instance pricing,” *ACM Trans. Econ. Comput.*, vol. 1, no. 3, pp. 16:1–16:20, Sep. 2013.

[36] M. Hadji, W. Louati, and D. Zeghlache, “Constrained pricing for cloud resource allocation,” in *Proceedings of the 10th IEEE International Symposium on Network Computing and Applications (NCA 2011)*, Aug 2011, pp. 359–365.

[37] B. Sharma, R. K. Thulasiram, P. Thulasiraman, S. K. Garg, and R. Buyya, “Pricing cloud compute commodities: A novel financial economic model,” in *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (Ccgrid 2012)*, ser. CCGRID ’12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 451–457.

[38] M. Mihailescu and Y. M. Teo, “Dynamic resource pricing on federated clouds,” in *Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid 2010)*, May 2010, pp. 513–517.

[39] C. S. Yeo, S. Venugopal, X. Chu, and R. Buyya, “Autonomic metered pricing for a utility computing service,” *Future Generation Computer Systems*, vol. 26, no. 8, pp. 1368–1380, 2010.

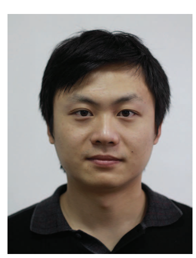
[40] J. Kephart and R. Das, “Achieving self-management via utility functions,” *IEEE Internet Computing*, vol. 11, no. 1, pp. 40–48, Jan 2007.

[41] N. Paton, M. A. De Aragão, K. Lee, A. A. Fernandes, and R. Sakellariou, “Optimizing utility in cloud computing through autonomic workload execution,” *Bulletin of the Technical Committee on Data Engineering*, vol. 32, no. 1, pp. 51–58, 2009.

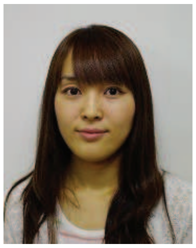
[42] C. Cirrascale, “Plans - cirrascale cloud services,” http://www.cirrascale.com/cloud/plans.aspx, accesed January, 2016.

[43] I. Nimbix, “Jarvice is the cloud platform for big data,” https://www.nimbix.net/jarvice/, accessed January, 2016.

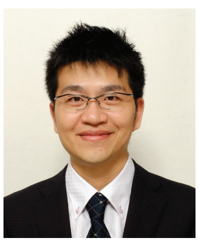
[44] H. Li, M. Dong, K. Ota, and M. Guo, “Pricing and repurchasing for big data processing in multi-clouds,” *IEEE Transactions on Emerging Topics in Computing*, vol. PP, no. 99, pp. 1–1, 2016.



He Li received the B.S., M.S. degrees in Computer Science and Engineering from Huazhong University of Science and Technology in 2007 and 2009, respectively, and Ph.D. degree in Computer Science and Engineering from The University of Aizu in 2015. He is currently a Postdoctoral Fellow with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan. His research interests include cloud computing and software defined networking. He has received the best paper award from IEEE VTC2016-Fall. Dr. Li is a guest associate editor of IEICE Transactions on Information and Systems. He is the recipient of IEEE TCSC Outstanding Dissertation Award 2016.



Kaoru Ota was born in Aizu-Wakamatsu, Japan. She received M.S. degree in Computer Science from Oklahoma State University, USA in 2008, B.S. and Ph.D. degrees in Computer Science and Engineering from The University of Aizu, Japan in 2006, 2012, respectively. She is currently an Assistant Professor with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan. From March 2010 to March 2011, she was a visiting scholar at University of Waterloo, Canada. Also she was a Japan Society of the Promotion of Science (JSPS) research fellow with Kato-Nishiyama Lab at Graduate School of Information Sciences at Tohoku University, Japan from April 2012 to April 2013. Her research interests include Wireless Networks, Cloud Computing, and Cyber-physical Systems. Dr. Ota has received best paper awards from ICA3PP 2014, GPC 2015, IEEE DASC 2015, and IEEE VTC 2016-Fall. She is an editor of IEEE Communications Letters, Peer-to-Peer Networking and Applications (Springer), Ad Hoc & Sensor Wireless Networks, International Journal of Embedded Systems (Inderscience) and Smart Technologies for Emergency Response & Disaster Management (IGI Global), as well as a guest editor of ACM Transactions on Multimedia Computing, Communications and Applications (leading), IEEE Communications Magazine, and IEEE Wireless Communications.



Mianxiong Dong received B.S., M.S. and Ph.D. in Computer Science and Engineering from The University of Aizu, Japan. He is currently an Associate Professor in the Department of Information and Electronic Engineering at the Muroran Institute of Technology, Japan. Prior to joining Muroran-IT, he was a Researcher at the National Institute of Information and Communications Technology (NICT), Japan. He was a JSPS Research Fellow with School of Computer Science and Engineering, The University of Aizu, Japan and was a visiting scholar with BBCR group at University of Waterloo, Canada supported by JSPS Excellent Young Researcher Overseas Visit Program from April 2010 to August 2011. Dr. Dong was selected as a Foreigner Research Fellow (a total of 3 recipients all over Japan) by NEC C&C Foundation in 2011. His research interests include Wireless Networks, Cloud Computing, and Cyber-physical Systems. He has received best paper awards from IEEE HPCC 2008, IEEE ICESS 2008, ICA3PP 2014, GPC 2015, IEEE DASC 2015 and IEEE VTC 2016-Fall. Dr. Dong serves as an Editor for IEEE Communications Surveys and Tutorials, IEEE Network, IEEE Wireless Communications Letters, IEEE Cloud Computing, IEEE Access, and Cyber-Physical Systems (Taylor & Francis), as well as a leading guest editor for ACM Transactions on Multimedia Computing, Communications and Applications (TOMM), IEEE Transactions on Emerging Topics in Computing (TETC), IEEE Transactions on Computational Social Systems (TCSS). He is the recipient of IEEE TCSC Early CareerAward 2016.

Athanasios V. Vasilakos is recently Professor with the Lule University of Technology. He served or is serving as an Editor for many technical journals, such as the IEEE Transactions on Network and Service management; IEEE Transactions on Cloud Computing, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Cybernetics; IEEE Transactions on Nanobioscience; IEEE Transactions on Information Technology in Biomedicine; ACM Transactions on Autonomous and Adaptive Systems; the IEEE Journal on Selected Areas in Communications. He is also General Chair of the European Alliances for Innovation (http://www.eai.eu).

Koji Nagano is a professor of Department of Information and Electronic Engineering, Faculty of Engineering, Muroran Institute of Technology since 2010. He received a Dr. degree from Tohoku University in 1989. From 1989-91 he was a fellow of the Japan Society for the Promotion of Science for Japanese Junior Scientists. His recent research activities are on signal processing, acoustic wave propagation, and methodologies for environmental footprints.