Profit-Maximizing Virtual Machine Provisioning Based on Workload Prediction in Computing Cloud

Qing Li[1], Qinghai Yang[1] Qingsu He[2] and Kyung Sup Kwak[3]
[1] State Key Laboratory on ISN, School of Telecommunications Engineering, Xidian University
No.2 Taibainan-lu, Xi'an, 710071, Shaanxi,China.
[e-mail: qhyang@xidian.edu.cn]
[2] State Grid Information and Telecommunication Group
[e-mail: heqingsu@sgitg.sgcc.com.cn]
[3]Graduate School of Information Technology and Telecommunications, Inha University,
253 Yonghyun-dong, Nam-gu, Incheon, 402-751, Korea.
[e-mail: kskwak@inha.ac.kr]
*Corresponding author: Qinghai Yang

Received July 8, 2015; revised September 8, 2015; accepted October 2, 2015; published December 31, 2015

Abstract

Cloud providers now face the problem of estimating the amount of computing resources required to satisfy a future workload. In this paper, a virtual machine provisioning (VMP) mechanism is designed to adapt workload fluctuation. The arrival rate of forthcoming jobs is predicted for acquiring the proper service rate by adopting an exponential smoothing (ES) method. The proper service rate is estimated to guarantee the service level agreement (SLA) constraints by using a diffusion approximation statistical model. The VMP problem is formulated as a facility location problem. Furthermore, it is characterized as the maximization of submodular function subject to the matroid constraints. A greedy-based VMP algorithm is designed to obtain the optimal virtual machine provision pattern. Simulation results illustrate that the proposed mechanism could increase the average profit efficiently without incurring significant quality of service (QoS) violations.

Keywords: Cloud computing, virtual machine provision, service level agreement, workload prediction, profit maximization

This research was supported in part by NRF of Korea (MSIP-)-NRF-(2014K1A3A1A20034987), NSF of China (61471287) and National Key Project (2013ZX03003011-003)

http://dx.doi.org/10.3837/tiis.2015.12.012 **ISSN : 1976-7277**

1. Introduction

Recent years we have witnessed the popularity of cloud computing services and the drastic growth of data centers' deployment for provisioning computing resources. Meanwhile, the advances in virtualization technologies make it possible to create virtual computing clusters with high performance [1]. Naturally, a cloud in the real world is selfish, and would try any means to maximize its own profit [2], -i.e., the income from processing tasks minus the operational costs and expenses in electricity. And the customers in cloud are willing to experience services with quality guaranteed. Computing resource scheduling has been expected as a key solution for maximizing the providers' profit, and as well for protecting customers' quality of service (QoS) in cloud.

In this paper, we consider the problem of scheduling randomly-arriving jobs onto different VMs in order to maximize the revenue whilst guaranteeing the service level agreement (SLA) constraints. We adopt the exponential smoothing (ES) prediction to forecast the arrival rate of forthcoming jobs. Then we adopt a diffusion approximation (DA) model to estimate the proper service rate for satisfying the demand for SLA. The virtual machine provisioning (VMP) problem is formulated as a facility location problem which is shown to be an NP-hard problem. Further, we make an equivalent analysis by introducing the concepts of submoduar function and matroid. Based on the analysis, we propose a VMP method to maximize the profit for a data center in cloud computing.

The rest of the paper is organized as follows. In Section 2, we discuss the related works. In Section 3, we present the system model. In Section 4, we formulate the VMP problem and give some theoretical analysis of workload prediction and service rate acquisition. In Section 5, we make the equivalent analysis of the VMP problem and propose the VMP algorithm. Section 6 shows the numerical results for the proposed mechanism. Finally, we conclude this paper in Section 7.

2. Related Work

As a large number of customers access the cloud service, there are many studies to solve the virtual machine provisioning (resource allocation) problem in cloud computing. Basic resource scheduling methods always consider two aspects : one is characteristics of workload, the other is characteristics of the resource in data center. The VMs can be provisioned in many different ways.

Some algorithms have been investigated for improving the utilization ratio of resource, such as First Come First Service (FCFS) [3] [4], Shortest Job First (SJF) [5] [6], Max-Min scheduling [7]. An Adaptive First Come First Service (AFCFS) algorithm was developed in [3], where jobs were executed according to the order of job arriving time. The factor such as varying workloads and different workload patterns were not taken into account. Whereas, varying workloads and different workload patterns were taken into account in [4] to provide a comprehensive performance-cost analysis of the task scheduling. A comparative study for the SJF algorithm and an integrated grouping based scheduling with both priority-aware features and SJF was investigated in [5]. A queue based hybrid algorithm in conjunction with SJF was proposed in [6] to show the optimal performance in terms of waiting time and average respond time. An improved Max-Min task-scheduling algorithm was studied in [7] for improving the resource utilization as well as for reducing the respond time of tasks. It

first selected the task with the longest execution time (Max), calculated the estimated time of the tasks in each VM, then selected the VM with the shortest completion time (Min), and finally allocated the task to the VM.

Some schemes intended to reduce the operational cost including electricity bills (the energy consumption). Two energy-aware virtual machine (VM) allocation algorithms were proposed in [8] to reduce the energy consumption of physical servers, nevertheless the delay metric was not considered. A modified best fit descending algorithm was proposed in [9] for minimizing the incremental power caused by a new VM placing, while it can not providing strict SLA for ensuring trivial performance degradation. A Modified Breadth First Search (MBFS) algorithm was designed in [10] to find the optimal VM for each task where tasks were prioritized and a VM tree was constructed before the execution of MBFS algorithm.

Some other approaches focused on the SLA requirement of jobs, e.g., delay constraint [11] [12] [13]. A conservative backfilling algorithm was modified in [11] by utilizing the earliest deadline first (EDF) algorithm and the largest weight first (LWF) algorithm to guarantee the deadline while improving the resource utilization. However it did not consider the various types of VMs. The deadline information from the SLAs was used in [12] to make decisions for task assignment and deferral, while the heterogeneity of computing resources was neglected. The problem of virtual machines' consolidating while protecting the SLA of each virtual machine was investigated in [13].

There are less researches in terms of profit maximization. Closely combined with an auction mechanism, a dynamic VM trading and scheduling algorithm was designed [2] to maximize the providers' profit. It could optimally schedule randomly arriving jobs with different resource requirements and SLAs onto different data centers, and judiciously turn on and off servers in the clouds based on the current electricity prices. In contrast to the algorithm in [2], even with the same objective, we considered the profit of one cloud service provider, whereas the work of [2] studied on a federation of clouds.

These proposed mechanisms only took the current loading status of VMs and the requirement of the requesting jobs into consideration, but the forthcoming jobs were not included. Although the information of the forthcoming jobs was unknown, an accurate prediction method could greatly help the resource allocation to make adequate decisions to maximize the profit. In [14], the author proposed prediction-based distributed capacity allocation and load redirect algorithms for IaaS cloud systems by minimizing the cost of running VMs. This prediction model was expected to be useful in context characterized by time series with non-stationary behaviour, which may not be suitable for our workload scenario.

Although many researchers proposed scheduling method in cloud computing environment to improve the performance of the system and guarantee the delay constraints, there is till potential to improve the profit gained by the cloud providers. Hence, we design a greedy-based VMP mechanism integrated with ES prediction to increase the average profit efficiently without incurring significant QoS violations.

3. System Model

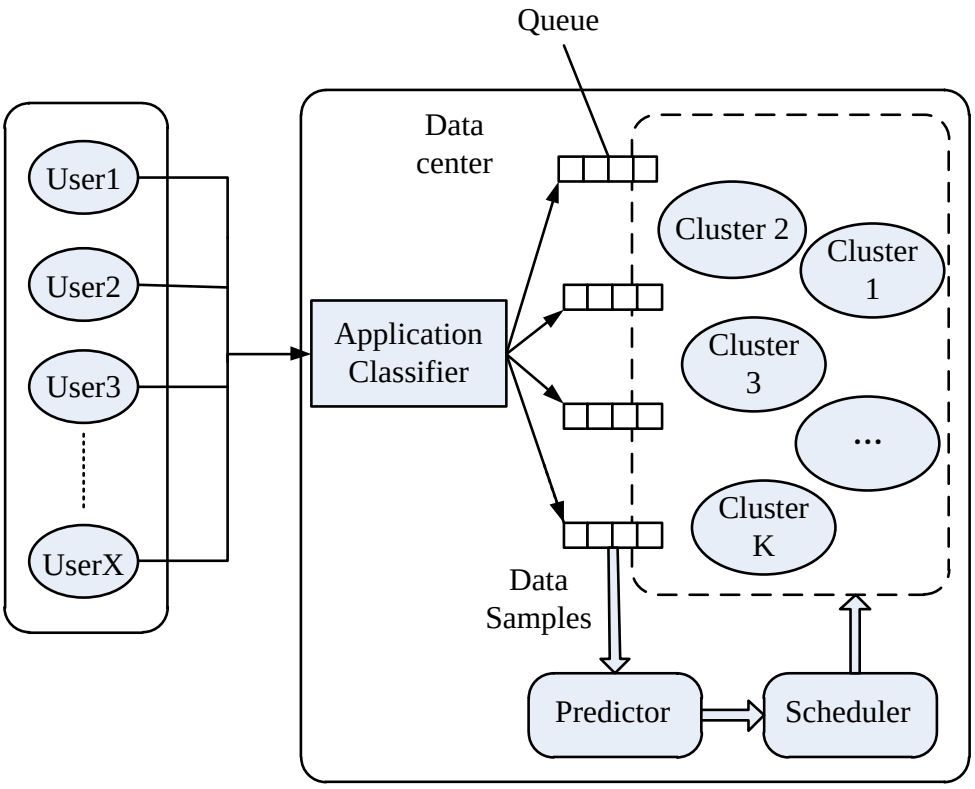

Fig. 1. System model of VM provisioning

A system model of a data center in cloud computing is illustrated in **Fig. 1**. It consists of an application classifier, a predictor, a scheduler, *num* requests' queues, and *K* virtual machines clusters.The application classifier classifies the requests into different types. The predictor is capable of forecasting the future workload according to a historic record of past offered ones. The scheduler determines the amount of VMs serving the incoming applications over a period of time, and the *num* queues are the buffers of requests waiting to be processed.

The system runs in a time-slotted fashion, that is, the time horizon is partitioned into time-slot indexed by t_s. A time frame t_m is defined as the length of VMP period of N_s time-slots. At the beginning of each time-slot t_s, the application classifier accepts and classifies various requests from customers without delay. Then job requests enter their dedicated queues. At the same time, the predictor predicts the arrival rate according to the current and historical data, and reports to the scheduler. During the time frame, the scheduler collects the data of requests' arrival rates at each time-slot and computes the average value, then estimates the proper serving rate using the DA model. Subsequently, it generates the virtual machine provisioning pattern. At the beginning of the next frame, the scheduler reallocates the virtual machines accordingly.

Assume that the system serves *num* types of applications. The arrival process of each type of applications into the data center is expressed as:

$$A_n = \{A_n(t_s), t_s \geq 0\}, n \in \mathrm{N} = \{1, 2, \cdots, num\}, \quad (1)$$

where A_n is a non-homogeneous Poisson process and N is the set of all types of applications. We suppose that $R_n(t_s)$ users of type n come at t_s, and the average arrival rate of the workload during a frame is denoted as $\lambda_n(t_m)$, i.e.,

$$\lambda_n(t_m) = \frac{1}{N_s}\sum_{t_s=1}^{N_s} R_n(t_s). \tag{2}$$

Let $D_n(t_s)$ denote the queuing and processing delay experienced by the applications of type n arriving at t_s and $D_n^{\max}$ denote the respond delay bound for the application of type n. Associated with each type n, an SLA contract specifies the QoS requirements agreed between the service provider and the service user. We use the delay-bound violation probability to statistically characterize the SLA [15]. Accordingly the delay-bound violation probability cannot exceed the target probability, denoted by p^{SLA}, given by:

$$\sup_t \Pr\left(D_n(t_s) \ge D_n^{\max}\right) \le p^{SLA}. \tag{3}$$

We assume that each VM hosts a single request and multiple VMs hosting the same application can run in parallel at different physical locations. We also assume that K types of VMs develop K virtual clusters, corresponding to various sets of configurations of CPU, storage and memory.

As the income and cost vary on a time frame basis. Hence, the provider has to face the VM scheduling problem to determine the optimal provision set of each application type in every frame according to the predicted workload, while guaranteeing SLA constraints. For type n, the predicted average arrival rate is denoted by $\hat{\lambda}_n(t_m)$ and the best serving rate is denoted by $\mu_n^*(t_m)$ during time-frame t_m. As we do VMP periodically, the current virtual machine provisioning is uncorrelated with the past ones. For the convenience of notation, we omit the time subscript in the formulation of VMP problem.

4. VMP Problem Formulation

4.1 Formulation of the VMP Problem

The objective of the VMP problem is to determine both the type and the number of VMs able to serve requests with an arrival rate of $\hat{\lambda}_n(t_m)$ for maximizing the profit while guaranteeing the delay constraint. We will study the VMP problem by assuming that we have known the best serving rate μ_n^* currently. We suppose that VMs are homogeneous in terms of processing capability for type n and $\mu_n^{\max}$ denotes the maximum service rate that a VM can reach while processing type n application, then the number of VMs for processing type n application is

$$m_n = \frac{\mu_n^*}{\mu_n^{\max}}. \tag{4}$$

The required VMs for serving type n applications are denoted by $\mathsf{M}_n = \{1, 2, \cdots, m_n\}$. As mentioned above, there are K VM clusters, which constitute a set denoted by V. And capacities of VM clusters are $C_j, \forall j \in \mathsf{V}$. We say that the cluster is active if there are jobs being processed in it, thus we need a binary variable $\gamma_j, \forall j \in \mathsf{V}$ to illustrate the state of the

cluster. And $\gamma_j = 1$ means that the cluster j is active. For VM clusters, we consider the electricity cost of running and cooling the servers as the main component of the operational cost in a data center, and simply model the operation cost as a constant, denoted by $z_j, \forall j \in \mathsf{V}$. For purpose of clarity, we have another binary variable $\theta_{ij}^n, \forall n \in \mathsf{N}, i \in \mathsf{M}_n, \forall j \in \mathsf{V}$, where θ_{ij}^n=1 means that the i_{th} VM serving the n_{th} type application is hosted in cluster j. To be profitable, providers charge the customer for the certain service. Let $p_{nj}, \forall n \in \mathsf{N}, \forall j \in \mathsf{V}$ be the service charge for accepting a job of type n hosted in a VM of type j, which remains fixed within a time-frame, but may vary across different frames. The VMP problem for maximizing the profit can be formulated as:

$$\begin{aligned} \max_{\theta} \quad & \sum_{n\in\mathsf{N}} \sum_{i\in\mathsf{M}_n} \max_{j\in\mathsf{V}} p_{nj}\theta_{ij}^n - \sum_{j\in\mathsf{V}} z_j\gamma_j \\ \text{s.t} \quad & \sum_{n\in\mathsf{N}} \sum_{i\in\mathsf{M}_n} \theta_{ij}^n \le C_j, \forall j \in \mathsf{V}, \qquad \text{(C1)} \\ & \sum_{j\in\mathsf{V}} \theta_{ij}^n = 1, \forall n \in \mathsf{N}, \forall i \in \mathsf{M}_n, \quad \text{(C2)} \end{aligned} \tag{5}$$

where $\gamma_j = \max(\theta_{ij}^n), \forall n \in \mathsf{N}, \forall i \in \mathsf{M}_n$. Constraint (C1) means that the number of VMs allocated to process the jobs should not exceed its capacity for each cluster, and constraint (C2) ensures that each VM cannot be hosted in two or more than two clusters. Given the all the types of applications and the profit function (namely the object function), finding the optimal VM set to maximize the profit is in conformity with the form of the facility location problem.

Remark: *Guaranteeing of the SLA requirement is not stated in the constraints. Since we have taken the QoS into consideration in the process of solving the proper service rate, the guarantee of delay is preemptive before we formulate the VMP problem.*

4.2 Workload Prediction

Before we solve the VMP problem, we need to acquire the serving rate of the time frame t_m according to the prediction $\hat{\lambda}_n(t_m)$. In the following we will discuss the workload prediction and serving rate acquisition respectively.

To predict the average arrival rate $\hat{\lambda}_n(t_m)$, we adopt a simple and efficient model, namely, the ES prediction model. In ES prediction, part of the historical data is weighted and averaged, in line with the principle of weighting more on fresh data. The predicted value is within the historical maximum and minimum. It can counter the effect of abnormal data and demonstrate the regular statistics information in data processing. In this work, we choose a version of ES prediction model [16] considering a certain time frame. At moment t, the sample of arrival rate is $y(t)$, the ES model predicts the arrival rate t_m steps ahead $\tilde{y}(t+t_m)$. The static smoothing coefficient is $\alpha(0<\alpha<1)$, and we adopt a dynamic smoothing coefficient

$$\varphi_t = \frac{\alpha}{(1-\alpha)^t}. \tag{6}$$

When $t>1, 0<\varphi_t<1$, and $\lim_{t\to 0}\varphi_t = \alpha$. Thus, the first order smooth value and the second

order smooth value at time t are :

$$\begin{cases} s_t^{(1)} = \varphi_t y(t) + (1-\varphi_t)^t s_{t-1}^{(1)}, \\ s_t^{(2)} = \varphi_t s_t^{(1)} + (1-\varphi_t)^t s_{t-1}^{(2)}. \end{cases} \tag{7}$$

The prediction equation is in a simple linear form, expressed as

$$\tilde{y}(t+t_m) = a_t + b_t t_m, \tag{8}$$

where the prediction coefficients a_t and b_t are the linear combination of the first order and the second order smooth values, i.e.

$$a_t = 2s_t^{(1)} - s_t^{(2)}, \tag{9}$$

$$b_t = \frac{\varphi_t}{1-\varphi_t}\left(s_t^{(1)} - s_t^{(2)}\right). \tag{10}$$

The basic smoothing formulas with the initials, which are the weighted average of workload samples and the first order smooth values respectively, do not change the basic characteristics of exponential smoothing, wherein

$$s_0^{(1)} = \varphi_t \sum_{i=1}^{t} (1-a)^{t-i} y_i, \tag{11}$$

$$s_0^{(2)} = \varphi_t \sum_{i=1}^{t} (1-a)^{t-i} s_i^{(1)}. \tag{12}$$

Smoothing coefficient α has a great influence on the smoothing accuracy of prediction. The smaller α is, the stronger ability of smoothing the model has, contrarily the larger α is, the more flexibility the model has to adapt rapid change. We define the prediction error as $e_t, e_t = \tilde{y}(t) - y(t)$. We search the optimal α to minimize the sum of the square of prediction error, that is,

$$\text{minMSE} = \frac{1}{N_s}\sum_{t=1}^{N_s} e_t^2 \cdot \rho^{n-t}, \tag{13}$$

where ρ $(0 < \rho < 1)$ is a weighing factor to weight more on fresh data. With the method proposed in [16], we divide α into 100 aliquots in [0,1] and develop a global search to find the the optimal α^*. Then we apply the α^* to the prediction.

4.3 Serving Rate Acquisition

Each incoming application to cloud enters a first in first out (FIFO) data buffer, which is modeled as an M/G/1 queue of each type n containing unscheduled jobs, with $Q_n(t_s)$ as its length in t_s. Let $\mu_n(t_m)$ denote the service rate of type n, which remains constant over the time frame, then we have the length of queue n in $t_s + 1$

$$Q_n(t_s+1) = \max\left\{0, Q_n(t_s) + \tau\left(R_n(t_s) - \mu_n(t_m)\right)\right\}. \tag{14}$$

In reality, the length of buffer is finite, and we have $Q_n^{\max}$ as the length of the data buffer. According to the Little theorem [17], the relationship between $Q_n^{\max}$ and $D_n^{\max}$ is

$$Q_n^{\max} = \lambda_n(t_m) D_n^{\max}. \tag{15}$$

Based on Eq. (3), the delay constraints can be converted into queue length constraints, i.e.,

$$\sup_t \Pr\left(Q_n\left(t_s\right) \geq Q_n^{\max}\right) \leq P^{SLA}. \tag{16}$$

We apply diffusion approximation to acquire the effective bandwidth [15] of the random data arrival process A_n related to QoS requirement. Due to the space limit, the theoretical background of diffusion approximation is not covered here. Details of the diffusion approximation can be found in [18].

$$\beta_n\left(t_m\right) = \hat{\lambda}_n\left(t_m\right) - \mu_n\left(t_m\right), \beta_n < 0, \tag{17}$$

where $\beta_n\left(t_m\right)$ is drift coefficient [18]. The probability density function of queue length q is

$$p(q) = -\frac{2\beta_n\left(t_m\right)}{\lambda_n} \exp\left(\frac{2\beta_n\left(t_m\right)}{\sigma_n^2\left(t_m\right)} q\right), \tag{18}$$

where $\sigma_n^2\left(t_m\right)$ is the variance of data arrival process A_n. The probability that the queue length exceeds the maximal length at t_s is

$$\Pr\left(\mathrm{Q_n}\left(\mathrm{t_s}\right) \geq \mathrm{Q_n^{max}}\right) = \exp\left(\frac{2\beta_{\mathrm{n}}\left(\mathrm{t_m}\right)}{\sigma_{\mathrm{n}}^2\left(\mathrm{t_m}\right)} \mathrm{Q_n^{max}}\right). \tag{19}$$

Intuitively, the smaller the violation probability is, the larger the serving rate is required to satisfy the SLA demand. We have

$$\sup_t Pr\left(Q_n\left(t_s\right) \geq Q_n^{\max}\right) = P^{SLA}. \tag{20}$$

Then by solving Eq. (20), we can acquire the proper serving rate

$$\mu_n^*\left(t_m\right) = \hat{\lambda}_n\left(t_m\right) - \frac{\sigma_n^2\left(t_m\right)}{2Q_n^{\max}} \log\left(P^{SLA}\right), \tag{21}$$

which guarantees the SLA and lays the foundation of VMP formulation.

5. Equivalent Analysis of VMP Problem

So far, we have elaborated the VMP problem which is a facility location problem. It has been proved that exact solution of facility location problem is NP-hard [19]. Many researches use some heuristic algorithms or alter the problem by relaxation to solve the equivalent problem. In this paper, we study the problem by adopting the submodular function and matroid theory. We will start with some definitions. Subsequently we prove that the VMP problem can be rewritten as the maximization of submodular function subject to matroid constraints [20].

5.1 Matroids and Submodular function

Linear independence is a well-known and useful concept. Matroids are structures that generalize this concept of independence for general sets. Informally, we need a finite ground set E that matroid is a way to label some subsets of E as independent. In vector spaces, the ground set is a set of vectors, and subsets are called independent of each other if their vectors are linearly independent in the usual linear algebraic sense. Formally, we have the following definitions which can be found in [21].

Definition 1 (Matroid) : A (finite) ground set E and a set of subsets of E, $\varnothing \neq \mathsf{I} \subseteq 2^{\mathsf{E}}$ are called a set system, notating (E,I). The set system (E,I) is called a matroid if

i1) $\varnothing \in \mathsf{I}$,

i2) $\forall \mathsf{X} \in \mathsf{I}, \mathsf{Y} \in \mathsf{I} \Rightarrow \mathsf{Y} \in \mathsf{I}$ (called "down monotone" or "down closed"),

i3) $\forall \mathsf{X}, \mathsf{Y} \in \mathsf{I}$, with $|\mathsf{X}| = |\mathsf{Y}| + 1$, then there exist $x \in \mathsf{X} \setminus \mathsf{Y}$ such that $\mathsf{Y} \cup \{x\} \in \mathsf{I}$.

$\mathsf{M} = (\mathsf{E},\mathsf{I})$ is an example of partition matroid if we have $\mathsf{E} = \mathsf{E}_1 \cup \mathsf{E}_2 \cup \cdots \mathsf{E}_l$, partitioning E into disjoint sets. Define a set of E as

$$\mathsf{I} = \left\{ \mathsf{X} \subseteq \mathsf{E} : \left| \mathsf{X} \cup \mathsf{E}_i \right| \leq k_i, i = 1, \cdots l \right\}, \tag{22}$$

where $k_1, \cdots, k_l$ are fixed parameters.

Definition 2 (Submodular function): Given a ground set E, a function $f : 2^{\mathsf{E}} \to R$ is submodular if for any $\mathsf{A}, \mathsf{B} \in \mathsf{E}$, we have that:

$$f(\mathsf{A}) + f(\mathsf{B}) \geq f(\mathsf{A} \cup \mathsf{B}) + f(\mathsf{A} \cap \mathsf{B}). \tag{23}$$

An alternative and equivalent definition is: a function $f : 2^{\mathsf{E}} \to R$ is a submodular if for any $\mathsf{A} \subseteq \mathsf{B} \subset \mathsf{E}$, and $e \in \mathsf{E} \setminus \mathsf{B}$, we have that:

$$f(\mathsf{A} \cup \{e\}) - f(\mathsf{A}) \geq f(\mathsf{B} \cup \{e\}) - f(\mathsf{B}), \tag{24}$$

which means that the incremental value, gain, or cost of e decreases as the context in which e is considered grows from A to B.

5.2 Equivalent Analysis of the VMP Problem

In this part we prove that our objective function and our constraints are independent sets of submodular function and matroid respectively. Firstly we define a ground set E. Denoting the i_{th} VM serving the n_{th} type application hosted in cluster j by b_{ni}^{j}, the ground set is:

$$\mathsf{E} = \{b_{ni}^{j}, \forall n \in \mathsf{N}, i \in \mathsf{M}_n, j \in \mathsf{E}\} \tag{25}$$

The ground set can be partitioned into K disjoint sets $\mathsf{E}_1, \mathsf{E}_2, \cdots \mathsf{E}_K$, where $\mathsf{E}_j = \{b_{ni}^{j}, \forall n \in \mathsf{N}, i \in \mathsf{M}_n\}$ means that all the applications are hosted in the cluster j.

Theorem 1 The constraints (C1) and (C2) in Eq. (5) can be written as partition matroid on the ground set defined in (22).

Proof: In the VMP problem, we want to find the optimal provisioning scheme of the VMs serving different types of applications. Each scheme can be expressed by a set $\mathsf{X} \subseteq \mathsf{E}$, called the provision set, for example if $b_{ni}^{j} \in \mathsf{X}$, the i_{th} VM serving the n_{th} type application is hosted in j. A set of elements which are hosted in the cluster j are equal to X_j, that $\mathsf{X}_j \square \mathsf{X} \cap \mathsf{E}_j, \forall j \in V$ is a subset of the ground set E associated to the cluster j. Therefore, the constraint on the capacity of clusters can be expressed as $\mathsf{X} \subseteq \mathsf{I}$ where

$$\mathsf{I} = \{\mathsf{X} \subseteq \mathsf{E} : \left| \mathsf{X} \cap \mathsf{E}_j \right| \leq C, \forall j = 1, 2, \cdots, K\}. \tag{26}$$

Comparing I in Eq. (26) and the definition of partition matriod in Eq. (22), we observe that the constraints form a partition matroid with $K = l$ and $C = k_j$ for $j = 1, 2, \cdots, K$. The partition matroid $\mathsf{M} = (\mathsf{E},\mathsf{I})$ is called C -uniform matroid. And the same VM cannot be located in two clusters, namely

$$\mathsf{X}_j \cap \mathsf{X}_k = \varnothing, \forall j, k \in \mathsf{V} \tag{27}$$

Theorem 2 The VMP objective function in Eq. (5) is a submodular function.

Proof: A provision set $X \subseteq E$, the objective function can be written as $f(X) = \sum_{n \in N} \sum_{i \in M_n} \max_{j \in X} p_{ij} - m(X)$, and $m(X)$ is a modular function corresponding to $\sum_{j \in V} z_j \gamma_j$.

Let $f(X) = \overline{f}(X) - m(X)$, where

$$\overline{f}(X) = \sum_{n \in N} \sum_{i \in M_n} \overline{f}_i(X) \tag{28}$$

and

$$\overline{f}_i(X) = \max_{j \in X} p_{ij} \tag{29}$$

Through a simple analysis, we know $\overline{f}_i(X)$ is a submodular function. And according to the property of submodular function we can prove that the sum of submodular functions $\overline{f}(X) = \sum_{n \in N} \sum_{i \in M_n} \overline{f}_i(X)$ is submodular. Namely, for any $A, B \subset E$, $\overline{f}(A) + \overline{f}(B) \geq \overline{f}(A \cup B) + \overline{f}(A \cap B)$ holds. Moreover, $m(X)$ is a modular function with $m(A)+m(B)=m(A \cup B)+m(A \cap B)$, which never destroys the inequality. Note of course that if $m(X)$ is modular, so is $-m(X)$. Finally, it is proved that $f(X) = \overline{f}(X) - m(X)$ is submodular.

■

Having proven that the constraints form a matroid and the objective function is submodular,
we can restate the optimization problem as following:

$$\begin{aligned} \max_{X} \quad & f(X) \\ \text{s.t.} \quad & X \subseteq I, \quad \text{(C3)} \\ & X_j \cap X_k = \varnothing, \forall j,k \in V, \quad \text{(C4)} \end{aligned} \tag{30}$$

where the constraints (C3) and (C4) have the same meaning as (C1) and (C2) in Eq. (5).

5.3 The VMP Algorithm

A greedy algorithm is a quite natural way for maximizing a submodular function subject to a matroid constraint which starts with an empty set. In each iteration, it adds an element that maximally improves the current solution (according to $f(X)$) while maintaining independence of the solution. Classical results on approximations of submodular function claim that the greedy algorithm achieves 1/2 of the optimal value [22]. If the matroid is uniform, the greedy algorithm yields a $(1 - 1/e)$-approximation. What's more, it is optimal in the special case of our model [23]. Based on the results obtained by rewriting the VMP problem as submodular function subject to a matroid, an adaptive VMP algorithm that dynamically adjust virtual machine provisioning pattern with the variation of the amount of workload is designed. In addition, as we have neglected the prediction error, our result is not optimal but may be close to the optimal value. The global mechanism of VM provision is stated in Algorithm 1, where $f_X(d) = f(X + d) - f(X)$.

Algorithm 1

Input: the predicted average arrival rate $\hat{\lambda}_n(t_m)$, the objective function $f(\mathsf{X})$.

Output: VM provisioning set X.

Step 1: Initialize $\mathsf{X} \leftarrow \varnothing, \mathsf{X}_j \leftarrow \varnothing, \forall j \in \mathsf{V}$.

Step 2: Calculate the proper serving rate of any application type n $\mu_n^*(t_m)$ using Eq. (21).

Step 3: Acquire the the number of VMs needed m_n using Eq. (4).

Step 4: Develop the ground set E using the information in step 3, let $\mathsf{D} \leftarrow \mathsf{E}$, $\mathsf{D}_j \leftarrow \mathsf{E}_j, \forall j \in \mathsf{E}$.

Step 5: Select $b_{\chi\gamma}^{\beta} = \arg max_{d \in \mathsf{D}} f_{\mathsf{X}}(d)$, and update the set D and X, $\mathsf{D} \leftarrow \mathsf{D} \setminus b_{\chi\gamma}^{\beta}$. $\mathsf{X} \leftarrow \mathsf{X} + b_{\chi\gamma}^{\beta}$.

If $|\mathsf{X}_\beta| = C_\beta$, update $\mathsf{D} \leftarrow \mathsf{D} \setminus \mathsf{D}_\beta$.

Step 6: Repeat step 5 till $b_{\chi\gamma}^{\beta} = 0$.

Step 7: Return X.

6 Simulation Results

6.1 Parameter Setting

The VM provisioning problem for a cloud data center with 40 VM clusters is considered, whose capacities are set to 400 uniformly. The benefit is randomly drown from a Gaussian distribution as well as the operational cost. The SLA violation probability threshold is set to 10^{-3}. The delay bounds are set to 5, 10, 15 respectively. The maximal serving rates of one VM are drew from [1 1/2 1/3 1/4 1/5 1/6 1/7 1/8 1/9 1/10] with equal probability. If we take 1 minute as a time slot that the simulation lasts for 1440 consecutive time-slots representing a whole day of 24 hours. And the VMs are reallocated every 60 time-slots (hourly). As we know, the data traffic in cloud usually has obvious periodicity, e.g., traffic is higher in daytime than that in deep night. Three typical kinds of workload have been randomly generated based on the method used in [14]. In our experiments, the following daily workload has been considered with 1 minute sample time interval:

- Normal day scenario: It describes the baseline workload where the number of application requests changes following the law described in Eq. (31). The pattern of workload described by this formula has peaks and valleys, which represent the variation of workload during a day.

$$\lambda(t) = A\sin\left[\frac{2\pi}{T}(t-t')\right] + B\sin\left[\frac{4\pi}{T}(t-t'')\right] + C, \tag{31}$$

where T denotes the period of application arrival, while t', t'' are constant.

- Heavy day scenario: It exhibits a 30% increment in the number of the application requests with respect to the baseline workload.
- Noisy day scenario: It is characterized by the same request workload belonging to the heavy day scenario with an additional noise component (we added a white Guassian noise with zero mean and standard deviation equal to 15% of the heavy day peak).

In this way, we increase the system variability in order to prove the accuracy of the prediction model and the robustness of our overall scheme in highly variable contexts. Simulation parameters are listed in Table 1.

Table 1. Simulation Parameters

Parameters	Values
Simulation cycle T	1440 time-slots
Resource allocation cycle	60 time-slots
Numbers of cluster	40
Capacity of cluster	400
Numbers of applications types	8
The violation probability	10^{-3}
The delay bounds D^{max}	5,10,15
The time parameter t'	144
The time parameter t''	84

6.2 Simulation Results

In this section, the workload adaption performance of the proposed scheme is examined. In the following quantitative analysis, we take Fig. 2 as an example to show the variation of VMs' number over the 24 hours for the normal day, heavy day and noisy day scenarios. We also plot the variation of workload and the predicted average, and put the three curves in one figure. To be intuitive, the three data sets are normalized respectively by the way described in Eq. (32)

$$S_i^{Norm} = \frac{S_i - S_{\min}}{S_{\max} - S_{\min}}, i = 1, 2, \cdots, \tag{32}$$

where S_i is any data in the corresponding data sets, $S_{\min}$ is smallest value, $S_{\max}$ is the largest value, S_i^{Norm} is the normalized value. As we normalize the real-time workload, S_i is the real-time workload sample, $S_{\max}$ is the largest sample and $S_{\min}$ the smallest sample over the three workload traces. Similarly, we perform the same normalization process for both the predicted average and the number of VMs.

In the scheme, we predict the workload one hour ahead from which we can get the average value that determines the amount of VMs running. Since the prediction model considered in this paper is able to provide an accurate prediction quality that, in terms of mean square error is lower than 10%, the predicted average is corresponding with the real-time workload. We also observe that the number of running VMs each hour is highly correlated to the workload arriving over that period. It implies that the proposed scheme can adjust the number of running VMs according to the variation of the amount of workload in real-time.

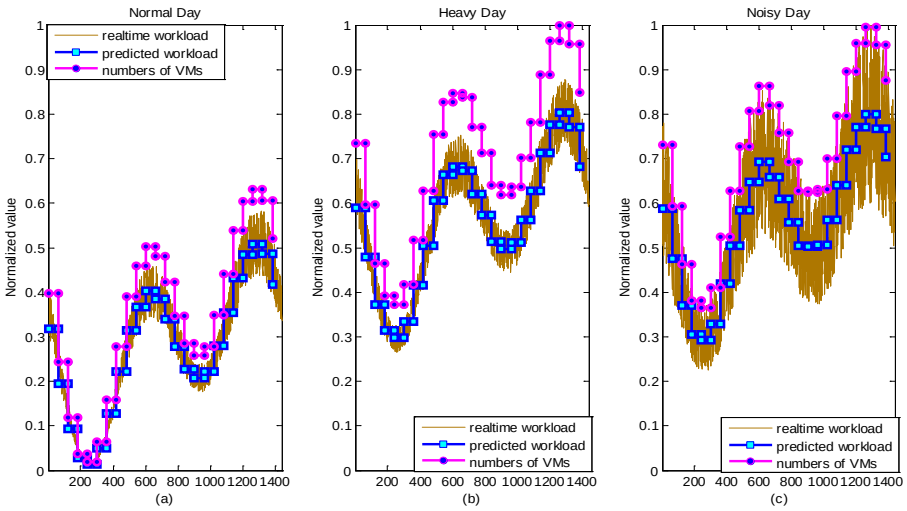

Fig. 2. Variation of VMs' number versus time

In order to verify the theoretical analysis about the properties of submodular function in Eq. (24), we take the variation of profit in three allocations among the total 24 ones as examples shown in Fig. 3. We observe that the profit obtained from starting one VM is non-increasing in each allocation. Note that the allocations done hourly are independent with each other. What we want to illustrate in this figure is that the trend of each curve is descending. Though the three curves have meeting points, it just means that the profit gained by increasing the VM is equal. As the incremental value of each element decreases as the set size grows, we can use Algorithm 1 to solve the problem.

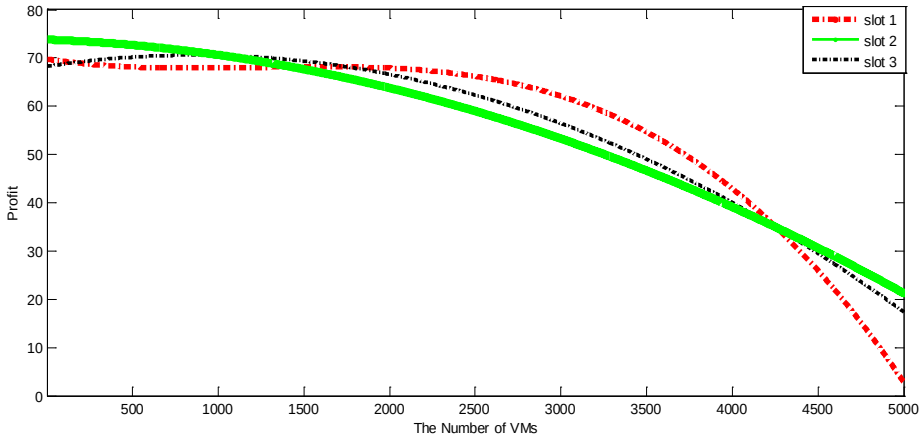

Fig. 3. The non-increasing profit in three slots

To illustrate scheduling quality of the proposed method, Fig. 4 compares the performance of the proposed VMP algorithm with two other algorithms (FCFS algorithm [3], MBFS algorithm [10]) in the three workload scenarios mentioned above. In this experiment, we divide the day into four time intervals (1-6, 7-12, 13-18, 19-24), and then compare the average ratio of profit in each time interval. As we do the normalization in Fig. 2, we also normalize the profit gained by scheduling with different algorithms in different workload

scenarios. Compare the three subfigures, as the workload increase, the profits gained in the heavy day and noisy day are more than that gained in the normal day, which is reasonable. In Fig. 4 (a), compared with FCFS and MBFS, the profit gained using the VMP algorithm increases by 23.7% and 23.0% respectively. In Fig. 4 (b), compared with FCFS and MBFS, the profit gained using the VMP algorithm increases by 21.8% and 11.0%, respectively. In Fig. 4 (c), compared with FCFS and MBFS, the profit gained using the VMP algorithm increases by 33.2% and 10.3%, respectively. From the simulation results, we can conclude that, our proposed VMP algorithm is always more efficient than FCFS and MBFS. Because when the requests come, FCFS algorithm allocates the VM avavible, which is lack of a selecting and matching process. MBFS prioritizes the requests, constructs a VM tree, and selects the appropriate VM (with the maximal profit in our paper) to execute the task. Though MBFS is a priority based algorithm, the request with higher priority may occupy the VM which is best fit for the request with lower priority, that may result in a drop in the performance. The idea of a VMP is to select best match among the tasks and VMs without any priority, which leads to the optimal performance.

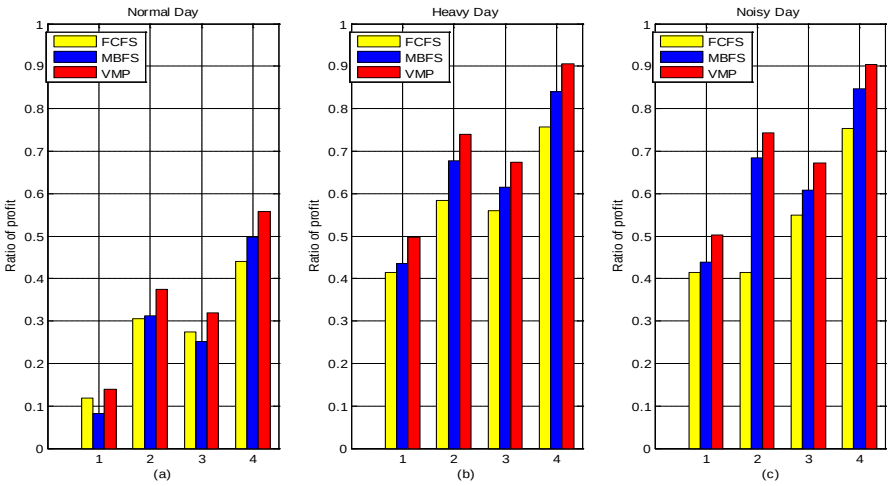

Fig. 4. Ratios of profit gained by different methods

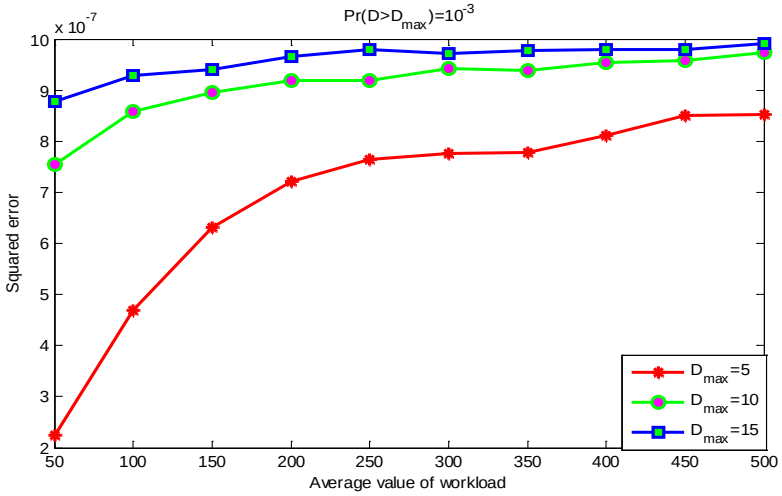

Fig. 5. QoS guarantee under different delay constraints

Fig. 5 provides the QoS guarantee of the proposed method under different delay bound constraints. Note that, the squared error in the figure is derived from the equation

$$\text{squared\_error}=(\text{p\_simulation-p\_theory})^2, \tag{33}$$

where p_theory is the given value of the violation probability and p_simulation is obtained from the simulation when the jobs are executed at the rate derived from DA model utilizing p_theory. Since DA model is a statistical model with little error, which is acceptable as long as the error is not sufficient to affect the system performance. In the simulation, we assume the violation probability of exceeding the delay bound is 10^{-3}. And delay bounds are set to be 5, 10, 15 respectively. The order of magnitude of the squared error is 10^{-7}, which means the QoS requirement can be satisfied under different delay constraints.

7. Conclusions

In this paper, our VM provisioning method applies ES prediction and statistical model to infer the future resource consumption patterns of VMs. Based on the forecasting results, the optimal VMP pattern are estimated by solving the VMP problem and the VMs are efficiently allocated accordingly. Especially, the optimization problem can be expressed as a submodular function under matroid constraints, which can help us to solve the VMP problem on the basis of a greedy algorithm. Simulation results verify that the proposed scheme has efficiently improved the profit of the cloud provider.

References

[1] C. Wang, W. Hung and C. Yang, "A prediction based energy conserving resources allocation scheme for cloud computing," in *proc. of IEEE GrC*, pp. 320-324, Oct. 2014. Article (CrossRef Link)

[2] Li, C. Wu and Z. Li, "Virtual machine trading in a federation of clouds: individual profit and social welfare maximization," *IEEE/ACM Transactions on Networking*, 2014. Article (CrossRef Link)

[3] I. Moschakis and H. Karatza, "Evaluation of gang scheduling performance and cost in a cloud computing system," *Journal of Supercomputing*, vol. 59, pp. 975-992, 2012. Article (CrossRef Link)

[4] D. Villegas, A. Antoniou, S. M. Sadjadi and A. Iosup, "An analysis of provisioning and allocation policies for infrastructure-as-a-service clouds," *12th IEEE/ACM International Symposium on cluster, cloud and grid computing*, pp. 612-619, 2012. Article (CrossRef Link)

[5] J. Ru and J. Keung, "An empirical investigation on the simulation of priority and shortest job first scheduling for cloud-based software systems," *22nd Australian Conference on Software Engineering*, pp. 78-87, 2013. Article (CrossRef Link)

[6] S. Behzad, R. Fotohi and M. Effatparvar, "Queue based job scheduling algorithm for cloud computing," *International Research Journal of Applied and Basic Sciences*, Vol. 4(11), pp. 3785-3790, 2011. Article (CrossRef Link)

[7] S. Behzad, R. Fotohi and M. Effatparvar, "An improved Max-Min task-scheduling algorithm for elastic cloud," in P*roc. of IEEE IS3C*, pp. 340 - 343, 2014. Article (CrossRef Link)

[8] N. Hung, N. Thoai and N. Son, "Performance constraint and power-aware allocation for user requests in virtual computing lab," *Journal of Science and Technology, Special on International Conference on Advanced Computing and Applications(Vietnam)*, vol. 49, no. 4A, pp. 383-392, 2011. Article (CrossRef Link)

[9] A. Beloglazov, J. Abawajy and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755-768, 2012. Article (CrossRef Link)

[10] R. Yadav and V. Kushwaha, "An energy preserving and fault tolerant task scheduler in cloud computing," in *Proc. of IEEE ICAETR*, pp.1-5, 2014. Article (CrossRef Link)
[11] S. Shin, Y. Kim and S. Lee, "Deadline-guaranteed scheduling algorithm with improved resource utilization for cloud computing," in *Proc. of IEEE CCNC*, pp.814-819, 2015. Article (CrossRef Link)
[12] M. Adnan, R. Sugihara. Hung and R. Gupata, "Energy efficient geographical load balancing via dynamic deferral of workload," in *Proc. of IEEE CLOUD*, pp. 188-195, June 2012. Article (CrossRef Link)
[13] Z. Huang and D. H. K. Tsang, "SLA guaranteed virtual machine consolidation for computing clouds," in *Proc. of IEEE ICC*, pp. 1314-1319, June 2012. Article (CrossRef Link)
[14] D. Ardagna, S. Casolari and B. Panicucci, "Flexible distributed capacity allocation and load redirect algorithms for cloud systems," in *Proc. of IEEE CLOUD*, pp. 163-170, July 2011. Article (CrossRef Link)
[15] Q. Du and X. Zhang, "Statistical QoS provisionings for wireless unicast/multicast of multi-layer video streams," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 3, pp. 420-433, April 2010. Article (CrossRef Link)
[16] P. Ji, D. Xiong and P. Wang, "A study on exponential smoothing model for load forecasting," *APPEEC*, pp. 1-4, March 2012. Article (CrossRef Link)
[17] D. Bertsekas and R. Gallager, "Data Networks," Prentice-Hall, 1987. Article (CrossRef Link)
[18] H. Kobayashi, "Application of the Diffusion approximation to queueing networks I: equilibrium queue distributions," *Journal of the Association for Computing Machinery*, vol. 21, no. 2, pp. 316-328, April 1974. Article (CrossRef Link)
[19] N. Megiddo, A. Tamir, "On the complexity of locating linear facilities in the plane," *Operations Research Letters*, vol. 1, no. 51, pp. 194–197, 1982. Article (CrossRef Link)
[20] N. Golrezaei, K. Shanmugam and A. Dimakis, "Femto caching: wireless video content delivery through distributed caching helpers," in *Proc. of IEEE INFOCOM*, pp. 1107-1115, March 2012. Article (CrossRef Link)
[21] S. Fujishige, "Submodular Functions and Optimization," 2005. Article (CrossRef Link)
[22] G. Calinescu, C. Chekuri and M. Pal, "Maximizing a monotone submodular function subject to a matroid xonstraint," *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1740-1766, Dec. 2010. Article (CrossRef Link)
[23] G. Nemhauser, L. Wholsey and M. Fisher, "An analysis of approximations for maximizing submodular set functions," *Mathematical Programming*, vol. 14, no. 1, pp. 265-294, 1978. Article (CrossRef Link)

Qing Li received the B.S. degree in Communication Engineering from Hebei University, China in 2014.Now she is currently working towards the M. S. degree in Communication and Information Systems at Xidian University. Her research interests in resource scheduling in cloud computing environment, information fusion and autonomic communication.

Qinghai Yang received his B.S. degree in Communication Engineering from Shandong University of Technology, China in 1998, M.S. degree in Information and Communication Systems from Xidian University, China in 2001, and Ph. D. in Communication Engineering from Inha University, Korea in 2007 with university-president award. From 2007 to 2008, he was a research fellow at UWB-ITRC, Korea. Since 2008, he is with Xidian University, China . His current research interest lies in the fields of autonomic communication, content delivery networks and LTE-A techniques.

Qingsu He is a member of Chinese Society for Electrical Engineering, the senior engineer of power system automation. He is with State Grid Information & Telecommunication Group Co., Ltd, serving on its subsidiary development planning department director and general manager of business innovation. His research focus lies in the area of power system automation, new technology and product development and application of promotion, intelligent information communication technology. He has published over 12 technical papers, more than 30 patents, 10 software copyrights and as well the book of IOT and Smart Grid.

Kyung Sup Kwak received the B.S. degree from the Inha University, Inchon, Korea in 1977, and the M.S. degree from the University of Southern California in 1981 and the Ph.D. degree from the University of California at San Diego in 1988, under the Inha University Fellowship and the Korea Electric Association Abroad Scholarship Grants, respectively. From 1988 to 1989 he was a Member of Technical Staff at Hughes Network Systems, San Diego, California. From 1989 to 1990 he was with the IBM Network Analysis Center at Research Triangle Park, North Carolina. Since then he has been with the School of Information and Communication, Inha University, Korea as a professor. He had been the chairman of the School of Electrical and Computer Engineering from 1999 to 2000 and the dean of the Graduate School of Information Technology and Telecommunications from 2001 to 2002 at the Inha University, Inchon, Korea. He is the current directors of Advanced IT Research Center of Inha University, and UWB Wireless Communications Research Center, a key government IT research center, Korea. Since 1994 he had been serving as a member of Board of Directors, and during the term of 2002-2000 year, he had been the vice president for Korean Institute of Communication Sciences (KICS). He has been the KISC's president of 2006 year term. His research interests include multiple access communication systems, mobile communication systems, UWB radio systems and ad-hoc networks, high-performance wireless Internet. Mr. Kwak is members of IEEE, IEICE, KICS and KIEE.