Optimal Cloud Resource Provisioning in On-Demand plan

V. Mamatha, T.V. Sampath kumar and CH. Muni Koteswara Rao

***Abstract*--In cloud computing, cloud providers can offer cloud consumers two provisioning plans for computing resources, namely reservation and on-demand plans. In general, cost of utilizing computing resources provisioned by reservation plan is cheaper than that provisioned by on-demand plan, since cloud consumer has to pay to provider in advance. With the reservation plan, the consumer can reduce the total resource provisioning cost. However, the best advance reservation of resources is difficult to be achieved due to uncertainty of consumer's future demand and providers' resource prices. To address this problem, an optimal cloud resource provisioning (OCRP) algorithm is proposed by formulating a stochastic programming model. The OCRP algorithm can provision computing resources for being used in multiple provisioning stages as well as a long-term plan, e.g., four stages in a quarter plan and twelve stages in a yearly plan. The demand and price uncertainty is considered in OCRP. In this paper, different approaches to obtain the solution of the OCRP algorithm are considered including deterministic equivalent formulation, sample-average approximation, and Benders decomposition. Numerical studies are extensively performed in which the results clearly show that with the OCRP algorithm, cloud consumer can successfully minimize total cost of resource provisioning in cloud computing environments.**

Keywords – Cloud computing, resource provisioning, virtualization, virtual machine placement, stochastic programming.

I. INTRODUCTION

Cloud computing is a large-scale distributed computing paradigm in which a pool of computing resources is available to users (called cloud consumers) via the Internet . Computing resources, e.g., processing power, storage, software, and network bandwidth, are represented to cloud consumers as the accessible public utility services. Infra-structure-as-a-Service (IaaS) is a computational service model widely applied in the cloud computing paradigm. In this model, virtualization technologies can be used to provide resources to cloud consumers. The consumers can specify the required software stack, e.g., operating systems and applications; then package them all together into virtual machines (VMs). The hardware requirement of VMs can also be adjusted by the consumers. Finally, those VMs will be outsourced to host in computing environments operated by third-party sites owned by cloud providers.

V.Mamatha is studying M.Tech in C.S.E, and T.V. Sampath kumar is working as Assoc. Prof in CSE,AVSCET, Venkatachalam, Nellore (Dt), A.P, and CH. Muni Koteswara Rao working as ASST .Prof in CSE, NBKRIST, Vidyanagar, SPSR Nellore (Dt), A.P, Emails: mamata.vadlapudi@gmail.com, tanjavuru.sampath@gmail.com, kamalc102@gmail.com

A cloud provider is responsible for guaranteeing the Quality of Services (QoS) for running the VMs. Since the computing resources are maintained by the provider, the total cost of ownership to the consumers can be reduced.

In cloud computing, a resource provisioning mechanism is required to supply cloud consumers a set of computing resources for processing the jobs and storing the data. Cloud providers can offer cloud consumers two resource provisioning plans, namely short-term on-demand and long-term reservation plans. Amazon EC2 and GoGrid are, for instances, cloud providers which offer IaaS services with both plans. In general, pricing in on-demand plan is charged by pay-per-use basis (e.g., 1 day). Therefore, purchasing this on-demand plan, the consumers can dynamically provision resources at the moment when the resources are needed to fit the fluctuated and unpredictable demands. For reservation plan, pricing is charged by a one-time fee (e.g., 1 year) typically before the computing resource will be utilized by cloud consumer. With the reservation plan, the price to utilize resources is cheaper than that of the on-demand plan. In this way, the consumer can reduce the cost of computing resource provisioning by using the reservation plan. For example, the reservation plan offered by Amazon EC2 can reduce the total provisioning cost up to 49 percent when the reserved resource is fully utilized (i.e, steady-state usage).

With the reservation plan, the cloud consumers a priori reserve the resources in advance. As a result, the under-provisioning problem can occur when the reserved resources are unable to fully meet the demand due to its uncertainty. Although this problem can be solved by provisioning more resources with on-demand plan to fit the extra demand, the high cost will be incurred due to more expensive price of resource provisioning with on-demand plan. On the other hand, the over provisioning problem can occur if the reserved resources are more than the actual demand in which part of a resource pool will be underutilized. It is important for the cloud consumer to minimize the total cost of resource provisioning by reducing the on-demand cost and over-subscribed cost of underprovisioning and overprovisioning. To achieve this goal, the optimal computing resource management is the critical issue.

In this paper, minimizing both underprovisioning and overprovisioning problems under the demand and price uncertainty in cloud computing environments is our motivation to explore a resource provisioning strategy for cloud consumers. In particular, an optimal cloud resource provisioning (OCRP) algorithm is proposed to minimize the total cost for provisioning resources in a certain time period. To make an optimal decision, the demand uncertainty from cloud consumer side and price uncertainty from cloud providers are taken into account to adjust the tradeoff between on-demand and oversubscribed costs. This optimal decision is obtained by formulating and solving a stochastic integer programming problem with multistage recourse.

Benders decomposition and sample-average approximation are also discussed as the possible techniques to solve the OCRP algorithm. Extensive numerical studies and simulations are performed, and the results show that OCRP can minimize the total cost under uncertainty.

The major contributions of this paper lie in the mathematical analysis which can be summarized as follows:

1. The optimal cloud resource provisioning algorithm is proposed for the virtual machine management. The optimization formulation of stochastic integer programming is proposed to obtain the decision of the OCRP algorithm as such the total cost of resource provisioning in cloud computing environments is minimized. The formulation considers multiple provisioning stages with demand and price uncertainties.
2. The solution methods based on Benders decomposition and sample-average approximation algorithms are used to solve the optimization formulation in an efficient way.
3. The performance evaluation is performed which can reveal the importance of optimal computing resource provisioning. The performance comparison among the OCRP algorithm and the other approaches is also presented.

The proposed mathematical analysis will be useful to the cloud consumers (e.g., organization and company) for the management of virtual machines in cloud computing environment. The proposed OCRP algorithm will facilitate the adoption of cloud computing of the users as it can reduce the cost of using computing resource significantly.

II. RELATEDWORK

Available resource provisioning options were discussed in. The resource provisioning strategies in distributed systems were addressed in.

In an architectural design of on-demand service for grid computing was proposed. In a profile-based approach to capture expert's knowledge of scaling applications was proposed in which extra demanded resources can be more efficiently provisioned. In the concept of resource slot was proposed. The objective is to address uncertainty of resources availability. In a binary integer program to maximize revenues and utilization of resource providers was formulated. In an optimization framework for resource provisioning was developed. This framework considered multiple client QoS classes under uncertainty of workloads (e.g., demands of computing resources). The arrival pattern of workloads is estimated by using online forecasting techniques. In heuristic method for service reservation was proposed. Prediction of demand was performed to define reservation prices. In K-nearest neighbors algorithm was applied to predict the demand of resources. In contrast, our work specifies that demands are given as probability distributions. In addition, the price difference between reservation and on-demand plans was not taken into account in all works in the literature.

As virtualization is a core technology of cloud computing, the problem of virtual machine placement (VM placement) becomes crucial the broker-based architecture and algorithm for assigning VMs to physical servers were developed. In a resource management consisting of resource provisioning and VM placement was proposed. In techniques of VM placement and consolidation which leverage min max and shares features provided by hypervisors were explored. In a dynamic consolidation mechanism based on constraint programming was developed. This consolidation mechanism was origin-ally designed for homogeneous clusters. However, heterogeneity which is common in a multiple cloud provider environment was ignored. In a dynamic VM placement was proposed. However, the placement in is heuristic-based which cannot guarantee the optimal solution. Stochastic programming has been developed to solve resource planning under uncertainty in various fields, e.g., production planning, financial management, and capacity planning. For example, in the authors applied the stochastic programming approach for planning of electrical power generation and transmission line expansion while some uncertainties affecting to the planning are taken into account. It is shown that stochastic programming is the promising mathematical tool which is able to address the optimal decision making in the stochastic environment.

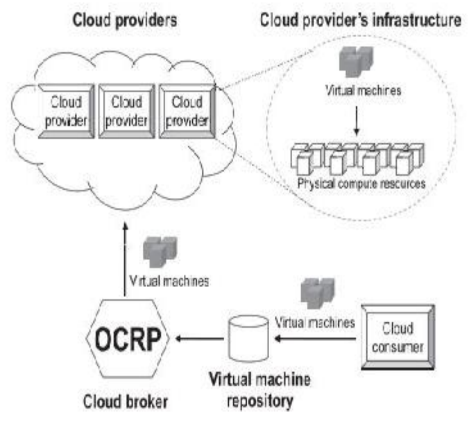

Fig. 1. System model of cloud computing environment.

III. SYSTEMMODEL AND ASSUMPTION

A. Cloud Computing Environment

As shown in Fig. 1, the system model of cloud computing environment consists of four main components, namely cloud consumer, virtual machine (VM) repository, cloud providers, and cloud broker. The cloud consumer has demand to execute jobs. Before the jobs are executed, computing resources has to be provisioned from cloud providers. To obtain such resources, the consumer firstly creates VMs integrated with software required by the jobs. The created VMs are stored in the VM repository. Then, the

VMs can be hosted on cloud providers' infrastructures whose resources can be utilized by the VMs. In Fig. 1, the cloud broker is located in the cloud consumer's site and is responsible on behalf of the cloud consumer for provision resources for hosting the VMs. In addition, the broker can allocate the VMs originally stored in the VM repository to appropriate cloud providers. The broker implements the OCRP algorithm to make an optimal decision of resource provisioning.

In OCRP, there are multiple VM classes used to classify different types of VM. Let $T C$ IN_1 denote the set of VM classes. It is assumed that one VM class represents a distinct type of jobs (e.g., one class for web application and the other for database application). A certain amount of resources is required for running the VM, and this required amount of resources can be different for VM in different classes. With this resource requirement, the cloud broker can reserve computing resources from cloud providers to be used in the future according to the actual demand. This demand can be determined as the number of created VMs. In this case, it is possible that additional resources can be provisioned instantly from cloud providers if the reserved resources are not enough to accommodate the actual demand.

LetJ C IN_1 denote the set of cloud providers. Each cloud provider supplies a pool of resources to the consumer. Let R denote the set of resource types which can be provided by cloud providers. Resource types can be computing power (in unit of CPU-hours), storage (in unit of GBs/ month), and network bandwidth for Internet data transfer (in unit of GBs/month). Each VM class specifies the amount of resources in each resource type. Let b_{ir} be the amount of resource type r required by the VM in class iƐI. 1. IN_1={1,2,3,...}. It is assumed that every cloud provider prepares facilities, e.g., virtualization management software, network facility, and load balancer, to support the consumer's hosting VM. Note that the key notations used in the paper are listed in Table 1.

B. Provisioning Plans

A cloud provider can offer the consumer two provisioning plans, i.e., reservation and/or on-demand plans. For planning, the cloud broker considers the reservation plan as medium- to long-term planning, since the plan has to be subscribed in advance (e.g., 1 or 3 years [2]) and the plan can significantly reduce the total provisioning cost [4]. In contrast, the broker considers the on-demand plan as short-term planning, since the on-demand plan can be purchased anytime for short period of time (e.g., one week) when the resources reserved by the reservation-plan are insufficient (e.g., during peak load).

C. Provisioning Phases

The cloud broker considers both reservation and on-demand plans for provisioning resources. These resources are used in different time intervals, also called provisioning phases. There are three provisioning phases: reservation, expending, and on-demand phases. As shown in Fig. 2, these phases with their actions perform in different points of time (or events) as follows. First in the reservation phase, without knowing the consumer's actual demand, the cloud broker provisions resources with reservation plan in advance. In the expending phase, the price and demand are realized, and the reserved resources can be utilized. As a result, the reserved resources could be observed to be either over-provisioned or under provisioned. If the demand exceeds the amount of reserved resources (i.e., underprovisioned), the broker can pay for additional resources with on-demand plan, and then the on-demand phase starts.

D. Provisioning Stages

A provisioning stage is the time epoch when the cloud broker makes a decision to provision resources by purchasing reservation and/or on-demand plans, and also allocates VMs to cloud providers for utilizing the provisioned resources. Therefore, each provisioning stage can consist of one or more provisioning phases. The number of provisioning stages is based on the number of planning epoches considered by the cloud broker, e.g., a yearly plan consists of 12 provisioning stages (i.e., 12 months). LetT C IN_1 denote the set of all provisioning stages where $|T| \geq 2$. For resource provisioning under uncertainty, the broker is assumed to be able to reserve the resources in the first provisioning stage. Also, the broker obtains a solution, called recourse action, for provisioning resources against uncertainty parameters (i.e., demand and price) in every stage. These uncertainty parameters in each stage will be observed by the broker after the resource reservation has been made. The observed uncertainty parameters are called realization (e.g., the actual number of created VMs after the jobs are submitted by the consumers). Then, the broker will take the recourse action according to the realization, e.g., utilizing the reserved resource and/or provisioning more resource with on-demand plan.

Symbol	Definition
$\mathcal{I}$	Set of virtual machine (VM) classes while $i \in \mathcal{I}$ denotes the VM class index
$\mathcal{J}$	Set of cloud providers while $j \in \mathcal{J}$ denotes the cloud provider index
$\mathcal{K}$	Set of reservation contracts while $k \in \mathcal{K}$ denotes the reservation contract index
$\mathcal{T}$	Set of provisioning stages while $t \in \mathcal{T}$ denotes the provisioning stage index
$\mathcal{R}$	Set of resource types while $r \in \mathcal{R}$ denotes the resource type index
Ω	Set of scenarios while $\omega \in \Omega$ denotes the scenario index
$c^{(R)}_{ijk}$	Reservation cost subscribed to reservation contract k charged by cloud provider j to cloud consumer's VM class i in the first provisioning stage
$c^{(r)}_{ijkt}(\omega)$	Reservation cost subscribed to reservation contract k charged by cloud provider j to cloud consumer's VM class i in provisioning stage t and scenario ω
$c^{(e)}_{ijkt}(\omega)$	Expending cost subscribed to reservation contract k charged by cloud provider j to cloud consumer's VM class i in provisioning stage t and scenario ω
$c^{(o)}_{ijt}(\omega)$	On-demand cost charged by cloud provider j to cloud consumer's VM class i in provisioning stage t and scenario ω
b_{ir}	Amount of resource type r required by VM class i
$d_{it}(\omega)$	Number of VMs (or demand) required to execute class i in provisioning stage t and scenario ω
$a_{jrt}(\omega)$	Maximum capacity of resource type r that cloud provider j can offer to cloud consumer in provisioning stage t and scenario ω
$x^{(R)}_{ijk}$	Decision variable representing the number of VMs in class i provisioned in reservation phase subscribed to reservation contract k offered by cloud provider j in the first provisioning stage
$x^{(r)}_{ijkt}(\omega)$	Decision variable representing the number of VMs in class i provisioned in reservation phase subscribed to reservation contract k offered by cloud provider j in provisioning stage t and scenario ω
$x^{(e)}_{ijkt}(\omega)$	Decision variable representing the number of VMs in class i run in expending phase subscribed to reservation contract k offered by cloud provider j in provisioning stage t and scenario ω
$x^{(o)}_{ijt}(\omega)$	Decision variable representing the number of VMs in class i provisioned in on-demand phase offered by cloud provider j in provisioning stage t and scenario ω

Table 1 : List of Key Notations

Fig. 3 shows the relationship between provisioning phases and provisioning stages, where a yearly plan with 12 provisioning stages, namely T= {T_1, T_2, …. , T_{12}} ,is considered. Fig. 3a shows the example of all three provisioning phases existing in each stage. In Fig. 3b, each stage may not consist of all provisioning phases. For example, the reservation phase is performed inT1 and the number of resources are reserved. FromT1-T3, the expending phase starts in some points of time when some resources reserved inT1 are utilized. Then, the on-demand phase starts inT3due to the insufficient reserved resources, and some resources are provisioned with on-demand plan. In T4, the reservation phase starts again and so on.

E. Reservation Contracts

A cloud provider can offer the consumer multiple reservation plans with different reservation contracts. Each reservation contract refers to the advance reservation of resources with the specific time duration of usage. For example, the reservation plan offered by Amazon EC2 has two reservation contracts, namely 1-year contract and 3-year contract. The certain amount of resources are reserved for 1 year in the one-year contract and 3 years in the three-year contract starting from the time when they are provisioned.

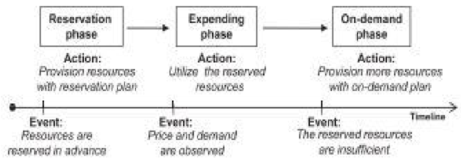

Fig. 2. Transition of provisioning phases.

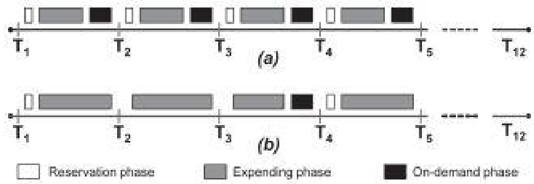

Fig. 3. Relationship between provisioning phases and provisioning stages.

Let K C IN_1 denote the set of all reservation contracts which are offered by cloud providers. Let L_k denote the time duration (in unit of provisioning stages) specified in reservation contract kƐK. Let T_k denote the set of stages at which the cloud broker can provision resources by contract k. Let F_{kt} be the set of stages at which some resources reserved by contract k could be utilized at stage tƐT. Given the total number of stages |T|, both T_k and F_{kt} are expressed as follows:

$$\mathcal{T}_k = \{1, \dots, |\mathcal{T}| - L_k + 1\}, \quad (1)$$

$$\mathcal{F}_{kt} = \{\max(1, t - L_k + 1), \dots, \min(t, |\mathcal{T}| - L_k + 1)\}. \quad (2)$$

In Fig. 4, the example of advance reservation for the yearly plan with 3-month (K1) and 6-month (K2) reservation contracts is shown (i.e., L_{K1} = 3 and L_{K2} = 6). The boxes above the timeline represent the time coverage of some reserved contracts. As shown in, T_{K1} = { T_1, T_2, . . . T_{10} } and T_{K2} = { T_1, T_2, . . .T_7 }are the sets of stages at which resources can be provisioned by K_1 and K_2, respectively. Contract K_1 is subscribed three times (e.g., T_7 - T_9), while contract K_2 is subscribed twice (e.g., T_5-T_{10}). Fig. 4 also shows that some stages can be covered by two (or more) subscribed contracts, e.g.,T_1-T_3 are covered by contracts K_1 and K_2. In Fig. 4, any set F_{kt} in can be obtained, e.g., F_{K1T4} = { T_2, T_3, T_4} , F_{k1T11}={T_9 , T_{10}}, F_{k2T12}={T_7}.

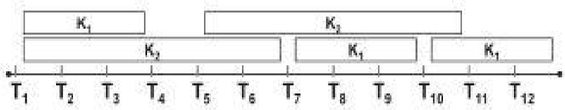

Fig. 4. Example of advance reservations with 3-month (K1) and 6-month (K2) contracts.

F. *Uncertainty of Parameters*

The optimal solution used by the cloud broker is obtained from the OCRP algorithm based on stochastic integer programming. Stochastic programming takes a set of uncertainty parameters (called scenarios), described by a probability distribution into account. Let Ω denote the set of all scenarios in every provisioning stage and Ω_t denote the set of all scenarios in provisioning stage t. Set Ω is defined as the Cartesian product of all Ω_t, namely

$$\Omega = \prod_{t \in \mathcal{T}} \Omega_t = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_{|\mathcal{T}|}. \quad (3)$$

It is assumed that the probability distribution of Ω has finite support, i.e., set Ω has a finite number of scenarios with respective probabilities p(w)Ɛ[0,1] where w is a composite variable defined as $w=(w_1,....w_{|t|})$ Ɛ Ω. In this paper, demand and price are considered as scenarios inwhose probability distribution is assumed to be available. The actual scenario of uncertainty parameter after it is observed by the broker is called realization.

G. *Provisioning Costs*

With three aforementioned provisioning phases, there are three corresponding provisioning costs incurred in these phases, namely reservation, expending, and on-demand costs. The main objective of the OCRP algorithm is to minimize all of these costs while the consumer's demand is met, given the uncertainty of demand and price.

For cloud provider, the price is defined in dollars ($) per resource unit. Let $c^{(R)}_{jkr}$ denote the unit price (i.e., costs to the consumer) of resource type r subscribed to reservation contract k provided by cloud provider j in reservation phase of the first provisioning stage. It is assumed that the price of reservation plan in the first stage is charged by a fixed one-time fee. The reservation cost $c^{(R)}_{jkr}$ is the cost for provisioning every resource type defined as follows:

$$c^{(R)}_{ijk} = \sum_{r \in \mathcal{R}} b_{ir} c^{(R)}_{jkr}. \quad (4)$$

The prices in reservation and expending phases could be adjusted by cloud providers without informing the consumer in advance, except the price of the reservation plan in the first provisioning stage. For instance, the cost of electric power to supply a cloud provider's data center could be increased by power plants in the next few months, and the cloud provider will be able to increase the costs of computing resources in the future as well. For the prices in provisioning stage t given scenario w in both reservation and expending phases, $c_{jkrt}^{(r)}(w)$ and $c_{jkrt}^{(e)}(w)$ denote the unit prices of resource type r with reservation contract k provided by cloud provider j, respectively. Let $c_{ijkt}^{(r)}(w)$ and $c_{ijkt}^{(e)}(w)$, defined with the similar way as , denote the reservation and expending costs for provisioning every resource type, respectively.

IV. STOCHASTIC PROGRAMMING MODEL

In this section, the stochastic programming with multistage recourse is presented as the core formulation of the OCRP algorithm. First, the original form of stochastic integer programming formulation is derived. Then, the formulation is transformed into the deterministic equivalent formulation (DEF) which can be solved by traditional optimization solver software.

A. *Stochastic Integer Programming for OCRP*

Minimize:

$$z = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} c^{(R)}_{ijk} x^{(R)}_{ijk} + \mathbb{E}_{\Omega}\left[\mathcal{Q}(x^{(R)}_{ijk}, \omega)\right], \quad (5)$$

subject to:

$$x^{(R)}_{ijk} \in \mathbb{N}_0, \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall k \in \mathcal{K}. \quad (6)$$

The general form of stochastic integer program of the OCRP algorithm is formulated in (5) and (6). The objective function (5) is to minimize the cloud consumer's total provisioning cost. Decision variable $x_{ijk}^{(R)}$ denotes the number of VMs provisioned in the first provisioning stage. In other words, this number refers to as the total amount of reserved resources. The expected cost under the uncertainty Ω is defined as $IE_\Omega[Q(x_{ijk}^{(R)},w)]$ where $Q(x_{ijk}^{(R)},w)$ is expressed as follows:

$$\mathcal{Q}(x^{(R)}_{ijk}, \omega) = \min_{Y=(x^{(r)}_{ijkt}(\omega), x^{(e)}_{ijkt}(\omega), x^{(o)}_{ijt}(\omega))} \mathcal{C}(Y), \quad Y \in \Upsilon(x^{(R)}_{ijk}, \omega). \quad (7)$$

$$\mathcal{Q}(x_{ijk}^{(\mathrm{R})},\omega) = \min \mathcal{C}(Y), \quad (8)$$

where

$$\mathcal{C}(Y) = \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}_k} c_{ijkt}^{(\mathrm{r})}(\omega)x_{ijkt}^{(\mathrm{r})}(\omega) + \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}\left(\sum_{k\in\mathcal{K}} c_{ijkt}^{(\mathrm{e})}(\omega)x_{ijkt}^{(\mathrm{e})}(\omega) + c_{ijt}^{(\mathrm{o})}(\omega)x_{ijt}^{(\mathrm{o})}(\omega)\right), \quad (9)$$

subject to:

$$x_{ijkt}^{(\mathrm{e})}(\omega) \le \sum_{\tilde{t}\in\mathcal{F}_{kt}} x_{ijk\tilde{t}}^{(\mathrm{r})}(\omega), \quad \forall i\in\mathcal{I},\forall j\in\mathcal{J},\forall k\in\mathcal{K},\forall t\in\mathcal{T}, \quad (10)$$

$$x_{ijk}^{(\mathrm{R})} = x_{ijk\bar{t}}^{(\mathrm{r})}(\omega), \quad \bar{t}=1,\forall i\in\mathcal{I},\forall j\in\mathcal{J},\forall k\in\mathcal{K}, \quad (11)$$

$$\sum_{j\in\mathcal{J}}\left(\sum_{k\in\mathcal{K}} x_{ijkt}^{(\mathrm{e})}(\omega)+x_{ijt}^{(\mathrm{o})}(\omega)\right) \ge d_{it}(\omega), \quad \forall i\in\mathcal{I},\forall t\in\mathcal{T}, \quad (12)$$

$$\sum_{i\in\mathcal{I}} b_{ir}\left(\sum_{k\in\mathcal{K}} x_{ijkt}^{(\mathrm{e})}(\omega)+x_{ijt}^{(\mathrm{o})}(\omega)\right) \le a_{jrt}(\omega), \quad \forall j\in\mathcal{J},\forall r\in\mathcal{R},\forall t\in\mathcal{T}, \quad (13)$$

$$x_{ijkt}^{(\mathrm{r})}(\omega)\in\mathbb{N}_0, \quad \forall i\in\mathcal{I},\forall j\in\mathcal{J},\forall k\in\mathcal{K},\forall t\in\mathcal{T}_k, \quad (14)$$

$$x_{ijkt}^{(\mathrm{e})}(\omega)\in\mathbb{N}_0, x_{ijt}^{(\mathrm{o})}(\omega)\in\mathbb{N}_0, \quad \forall i\in\mathcal{I},\forall j\in\mathcal{J},\forall k\in\mathcal{K},\forall t\in\mathcal{T}. \quad (15)$$

B. Deterministic Equivalent Formulation

Given a probability distribution of all scenarios in set Ω, the formulation in (5)-(15) can be transformed into the deterministic integer programming called deterministic equivalent formulation as expressed in (16)-(23). To solve this DEF, probability distributions of both price and demand must be available, i.e.,p(w) in (16). Then, the DEF can be solved by using traditional optimization solver software. For example, the formulation is implemented using MathProg script, and then the script is solved by GNU Linear Programming Kit (GLPK) [23].

Minimize:

$$\hat{z}_\Omega = \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} c_{ijk}^{(\mathrm{R})}x_{ijk}^{(\mathrm{R})} + \sum_{\omega\in\Omega}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}_k} p(\omega)c_{ijkt}^{(\mathrm{r})}(\omega)x_{ijkt}^{(\mathrm{r})}(\omega) + \sum_{\omega\in\Omega}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} p(\omega)\left(\sum_{k\in\mathcal{K}} c_{ijkt}^{(\mathrm{e})}(\omega)x_{ijkt}^{(\mathrm{e})}(\omega)+c_{ijt}^{(\mathrm{o})}(\omega)x_{ijt}^{(\mathrm{o})}(\omega)\right), \quad (16)$$

subject to: (6)

$$x_{ijkt}^{(\mathrm{e})}(\omega) \le \sum_{\tilde{t}\in\mathcal{F}_{kt}} x_{ijk\tilde{t}}^{(\mathrm{r})}(\omega), \quad \forall i\in\mathcal{I},\forall j\in\mathcal{J},\forall k\in\mathcal{K},\forall t\in\mathcal{T},\forall\omega\in\Omega, \quad (17)$$

$$x_{ijk}^{(\mathrm{R})} = x_{ijk\bar{t}}^{(\mathrm{r})}(\omega), \quad \bar{t}=1,\forall i\in\mathcal{I},\forall j\in\mathcal{J},\forall k\in\mathcal{K},\forall\omega\in\Omega, \quad (18)$$

$$\sum_{j\in\mathcal{J}}\left(\sum_{k\in\mathcal{K}} x_{ijkt}^{(\mathrm{e})}(\omega)+x_{ijt}^{(\mathrm{o})}(\omega)\right) \ge d_{it}(\omega), \quad \forall i\in\mathcal{I},\forall t\in\mathcal{T},\forall\omega\in\Omega, \quad (19)$$

$$\sum_{i\in\mathcal{I}} b_{ir}\left(\sum_{k\in\mathcal{K}} x_{ijkt}^{(\mathrm{e})}(\omega)+x_{ijt}^{(\mathrm{o})}(\omega)\right) \le a_{jrt}(\omega), \quad \forall j\in\mathcal{J},\forall r\in\mathcal{R},\forall t\in\mathcal{T},\forall\omega\in\Omega, \quad (20)$$

$$x_{ijkt}^{(\mathrm{r})}(\omega)\in\mathbb{N}_0, \quad \forall i\in\mathcal{I},\forall j\in\mathcal{J},\forall k\in\mathcal{K},\forall t\in\mathcal{T}_k,\forall\omega\in\Omega, \quad (21)$$

$$x_{ijkt}^{(\mathrm{e})}(\omega)\in\mathbb{N}_0, \quad \forall i\in\mathcal{I},\forall j\in\mathcal{J},\forall k\in\mathcal{K},\forall t\in\mathcal{T},\forall\omega\in\Omega, \quad (22)$$

$$x_{ijt}^{(\mathrm{o})}(\omega)\in\mathbb{N}_0, \quad \forall i\in\mathcal{I},\forall j\in\mathcal{J},\forall t\in\mathcal{T},\forall\omega\in\Omega. \quad (23)$$

V. BENDERS DECOMPOSITION

In this section, the Benders decomposition algorithm is applied to solve the stochastic programming problem formulated. The goal of this algorithm is to break down the optimization problem into multiple smaller problems which can be solved independently and parallelly. As a result, the time to obtain the solution of the OCRP algorithm can be reduced. The Benders decomposition algorithm can decompose integer programming problems with complicating variables into two major problems: master problem and subproblem.

*Property 1.*The DEF derived in (16)-(23) is the problem whose structure has multiple complicating variables.

Proof. Variables $x_{ijkt}^{(e)}(w)$ from the DEF defined in (16)-(23) are considered as complicating variables [6]. Since variables $x_{ijkt}^{(e)}(w)$ exist in constraints (17), (19), and (20), the variables prevent the decomposability of the DEF. If variables $x_{ijkt}^{(e)}(w)$ are given the fixed values are denoted by $x_{ijkt}^{(fix)}(w)$, the DEF can be decomposed into two types of independent optimization sub problems, namely S_1 and S_2 (w) presented as follows:

$$z_v^{(r)} = \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} c_{ijk}^{(R)} x_{ijk}^{(R)} + \sum_{\omega\in\Omega}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}_k} p(\omega) c_{ijkt}^{(r)}(\omega) x_{ijkt}^{(r)}(\omega), \quad (24)$$

subject to: (6), (17), (18), (21)

$$x_{ijkt}^{(e)}(\omega) = x_{ijkt}^{(\mathrm{fix})}(\omega), \ \forall i\in\mathcal{I}, \forall j\in\mathcal{J}, \forall k\in\mathcal{K}, \forall t\in\mathcal{T}, \forall\omega\in\Omega, \quad (25)$$

$[S_2(\omega)]$ Minimize:

$$z_v^{(o)}(\omega) = \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} p(\omega) c_{ijt}^{(o)}(\omega) x_{ijt}^{(o)}(\omega), \quad (26)$$

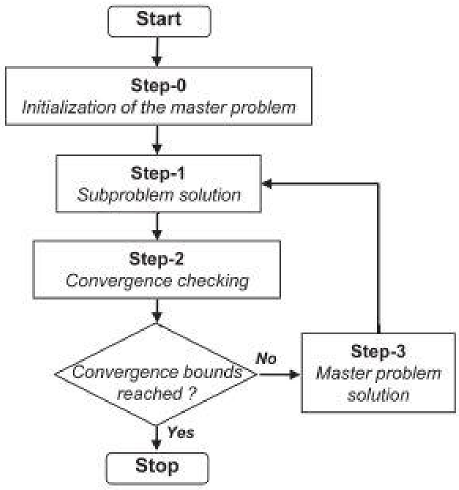

Fig. 5. Flowchart of Benders decomposition algorithm.

subject to: (19), (20), (23)

$$x_{ijkt}^{(e)}(\omega) = x_{ijkt}^{(\mathrm{fix})}(\omega), \ \forall i\in\mathcal{I}, \forall j\in\mathcal{J}, \forall k\in\mathcal{K}, \forall t\in\mathcal{T}. \quad (27)$$

From this decomposition, we conclude that the DEF has the structure with multiple complicating variables. From Property 1, the problem can be solved by Benders decomposition algorithm. The algorithm consists of steps which are performed iteratively. At each iteration, the master problem constituted by the complicating variables and subproblems constituted by the other decision variables are solved, then lower and upper bounds are calculated. The algorithm stops when optimal solution converges, i.e., the lower and upper bounds are satisfactorily close to each other.

In Fig. 5, the flowchart of Benders decomposition algorithm is shown. The algorithm for solving OCRP is presented in four steps (i.e., Step-0 to Step-3) as follows.

Step-0: Initialization of the master problem. In Step-0, the step is the initialization of the master problem. This Step-0 is performed only once, while Step-1 to Step-3 are repeatable in the algorithm. Let v denote the iteration counter and initially set v= 1. The master problem as expressed in (28)-(32) is an alternative form of the formulation DEF shown in (16)-(23).

Minimize:

$$z_v^{(e)} = \sum_{\omega\in\Omega}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}} p(\omega) c_{ijkt}^{(e)}(\omega) x_{ijktv}^{(e)}(\omega) - \alpha_v, \quad (28)$$

subject to:

$$\alpha_v \geq \alpha^{(\mathrm{lb})}, \quad (29)$$

$$\sum_{j\in\mathcal{J}} x_{ijktv}^{(e)}(\omega) \leq d_{it}(\omega), \ \forall i\in\mathcal{I}, \forall t\in\mathcal{T}, \forall\omega\in\Omega, \quad (30)$$

$$\sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}} b_{ir} x_{ijkt}^{(e)}(\omega) \leq a_{jrt}(\omega), \ \forall j\in\mathcal{J}, \forall r\in\mathcal{R}, \forall t\in\mathcal{T}, \forall\omega\in\Omega, \quad (31)$$

$$x_{ijktv}^{(e)}(\omega) \in \mathbb{N}_0, \ \forall i\in\mathcal{I}, \forall j\in\mathcal{J}, \forall k\in\mathcal{K}, \forall t\in\mathcal{T}, \forall\omega\in\Omega, \quad (32)$$

The objective function (28) is directly derived from that in(16). $X_{ijktv}^{(e)}(w)$ represents variable $X_{ijkt}^{(e)}(w)$ in iteration v of master problem, while variable α_v provides the minimum cost given reservation and on-demand costs. This α_v will be improved in consequent iterations. Initially, α_v can be fixed by constant $\alpha^{(lb)}$ as shown in constraint (29). This $\alpha^{(lb)}$ can be estimated from an economical analysis or historical data of prior solutions [6]. Constraints (30)-(32) define the boundary $X_{ijktv}^{(e)}(w)$. After solving the master problem, the algorithm proceeds to the Step-1.

Step-1: Subproblem solution. In Step-1, multiple subproblems are formulated and solved. Lets assign solution $X_{ijktv}^{(e)}(w)$ obtained from the master problem to variables

$$X_{ijkt}^{(fix)}(w)$$

Given the fixed solution $X_{ijkt}^{(fix)}(w)$, two aforementioned subproblems, namely, S_1 and $S_2(w)$ can be solved concurrently.

The subproblem S_1 is presented in which the objective function is to minimize the reservation cost. Let variable $\lambda_{ijktv}^{(r)}(w)$ denote the optimal solution of the dual problem of S_1 in iteration v associated with constraint. The solution of $\lambda_{ijktv}^{(r)}(w)$ will be used in Step-3.

The subproblem $S_2(w)$ is presented in which the objective function is to minimize the on-demand cost when the realization is set to w. $S_{2(w)}$ associates with the number of scenarios $|\Omega|$, and hence $|\Omega|$ subproblems are generated. Note that $|\Omega|$ is the Cardinality of set Ω. Let variable $\lambda_{ijktv}^{(o)}(w)$ denote the optimal value of the dual problem of $S_2(w)$ in iteration v associated with constraint (27). The solution of $\lambda_{ijktv}^{(0)}(w)$ will be used in Step-3.

Step-2: Convergence checking. In Step-2, the convergence of lower and upper bounds of the solutions obtained from master problem and subproblems is checked. Both bounds are adjusted in each iteration. The lower bound in iteration v denoted as $z_v^{(lb)}$ can be obtained from the objective function of the master problem, i.e., $z_v^{(lb)} = z_v^{*(e)}$. The upper bound in iteration v denoted by $z_v^{(ub)}$ can be obtained from

$$z_{\nu}^{(\mathrm{ub})} = z_{\nu}^{*(\mathrm{e})} - \alpha_{\nu} + z_{\nu}^{*(\mathrm{r})} + \sum_{\omega\in\Omega} z_{\nu}^{*(\mathrm{o})}(\omega). \quad (33)$$

Let ε denote a small tolerance value to verify the convergence of both lower and upper bounds. The Benders decomposition algorithm stops when $z_v^{(ub)}$-$z_v^{(lb)} < \varepsilon$, which means both bounds are acceptably close to each other and the optimal solution can be found in iteration v. Otherwise, the algorithm proceeds to the next iteration in which Step-3 will perform.

Step-3: Master problem solution.

$$\alpha_{\nu} \geq \sum_{\omega\in\Omega}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} \left((\lambda_{ijkt\overline{\nu}}^{(\mathrm{r})}(\omega) + \lambda_{ijkt\overline{\nu}}^{(\mathrm{o})}(\omega))\right.$$
$$\left.(x_{ijkt\nu}^{(\mathrm{e})}(\omega) - x_{ijkt\overline{\nu}}^{(\mathrm{e})}(\omega))\right) \quad (34)$$
$$+ z_{\overline{\nu}}^{*(\mathrm{r})} + \sum_{\omega\in\Omega} z_{\overline{\nu}}^{*(\mathrm{o})}(\omega), \quad \overline{\nu}\in\{1,\ldots,\nu-1\}.$$

Let the iteration counter be increased by v←v+1. Then, the master problem in (28)-(32) can be further relaxed by additional constraints called Benders cuts. In addition, the solution of the master problem will adjust the cost α_v and also the expending cost according to solution of $x_{ijktv}^{(e)}(w)$. As shown in the Benders cuts are constructed from the optimal costs obtained from master problem and subproblems in the prior iterations. After solving this master problem, Step-1 is repeated and the same iterative process continues.

VI. SAMPLE AVERAGE PPROXIMATION

In the case that the number of scenarios is numerous, it may not be efficient to obtain the solution of the OCRP algorithm by solving the stochastic programming formulation defined in (16)-(23) directly if all scenarios in the problem are considered. To address this complexity issue, the sample-average approximation (SAA) approach is applied. This approach selects a set of scenarios, e.g., N scenarios, where N is smaller than the total number of scenarios |Ω|. Then, these N scenarios can be solved in a deterministic equivalent formulation. The optimal solution can be obtained if N is large enough which can be verified numerically.

In this section, the SAA approach is applied to approximate the expected cost in every considered provisioning stage, i.e., $Q(x_{ijk}^{(R)}, w)$. A sampling method (e.g., Monte Carlo and Latin hypercube), is used to generate scenarios $\Omega_N = \{ w_1, \ldots, w_n\}$ where N denotes the sample size. Let N = { 1, . . . , N } be the set of indices of samples. Then, the expected cost can be redefined. The function $Q(x_{ijk}^{(R)}, N)$ is the SAA to the objective function. Then, the problem can be transformed into a deterministic equivalent formulation, as called approximation problem (AP) formulation.

Minimize:

$$\hat{z}_N = \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} c_{ijk}^{(\mathrm{R})} x_{ijk}^{(\mathrm{R})} + \frac{1}{N}\sum_{n\in\mathcal{N}}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}_k} c_{ijkt}^{(\mathrm{r})}(\omega_n) x_{ijkt}^{(\mathrm{r})}(\omega_n)$$
$$+\frac{1}{N}\sum_{n\in\mathcal{N}}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}\left(\sum_{k\in\mathcal{K}} c_{ijk}^{(\mathrm{e})}(\omega_n) x_{ijkt}^{(\mathrm{e})}(\omega_n) + c_{ijt}^{(\mathrm{o})}(\omega_n) x_{ijt}^{(\mathrm{o})}(\omega_n)\right) \quad (35)$$

subject to: (6)

$$x_{ijkt}^{(\mathrm{e})}(\omega_n) \leq \sum_{\bar{t}\in\mathcal{F}_{kt}} x_{ijk\bar{t}}^{(\mathrm{r})}, \quad \forall i\in\mathcal{I}, \forall j\in\mathcal{J}, \forall k\in\mathcal{K}, \forall t\in\mathcal{T}, \forall n\in\mathcal{N}, \quad (36)$$

$$x_{ijk}^{(\mathrm{R})} = x_{ijk\bar{t}}^{(\mathrm{r})}(\omega_n), \quad \bar{t}=1, \forall i\in\mathcal{I}, \forall j\in\mathcal{J}, \forall k\in\mathcal{K}, \forall n\in\mathcal{N}, \quad (37)$$

$$\sum_{j\in\mathcal{J}}\left(\sum_{k\in\mathcal{K}} x_{ijkt}^{(\mathrm{e})}(\omega_n) + x_{ijt}^{(\mathrm{o})}(\omega_n)\right) \geq d_{it}(\omega_n), \quad \forall i\in\mathcal{I}, \forall t\in\mathcal{T}, \forall n\in\mathcal{N}, \quad (38)$$

$$\sum_{i\in\mathcal{I}} b_{ir}\left(\sum_{k\in\mathcal{K}} x_{ijkt}^{(\mathrm{e})}(\omega_n) + x_{ijt}^{(\mathrm{o})}(\omega_n)\right) \leq a_{jrt}(\omega_n), \quad (39)$$
$$\forall j\in\mathcal{J}, \forall r\in\mathcal{R}, \forall t\in\mathcal{T}, \forall n\in\mathcal{N},$$

$$x_{ijkt}^{(\mathrm{r})}(\omega_n) \in \mathbb{N}_0, \quad \forall i\in\mathcal{I}, \forall j\in\mathcal{J}, \forall k\in\mathcal{K}, \forall t\in\mathcal{T}_k, \forall n\in\mathcal{N}, \quad (40)$$

$$x_{ijkt}^{(\mathrm{e})}(\omega_n), x_{ijt}^{(\mathrm{o})}(\omega_n) \in \mathbb{N}_0, \quad \forall i\in\mathcal{I}, \forall j\in\mathcal{J}, \forall k\in\mathcal{K}, \forall t\in\mathcal{T}, \forall n\in\mathcal{N}. \quad (41)$$

$$\hat{\mathcal{Q}}(x_{ijk}^{(\mathrm{R})}, N) = \frac{1}{N}\sum_{n=1}^{N} \mathcal{Q}(x_{ijk}^{(\mathrm{R})}, \omega_n). \quad (42)$$

Let z* and x* denote the optimal objective function value and optimal solution of the original formulation, respectively. Let $z^{\wedge *}_n$ and $x^{\wedge *}_N$ denote the optimal objective function value and optimal solution of the AP formulation, respectively. Note that although $z^{\wedge *}_N$ becomes closer to z* when N is large, value z^*N naturally varies according to set of samples Ω_N. Therefore, an estimation method is required to achieve the SAA lower and upper bounds of the optimal solution. Obviously, $z^{\wedge *}_N$ forms the SAA upper bound of z* as follows:

$$z^* \leq \hat{z}_N^*. \quad (43)$$

In addition, the SAA lower bound of z* is formed by the following unbiased property

$$\mathbb{E}[\hat{z}_N^*] \leq z^*. \quad (44)$$

For the properties in (43) and (44), bounding method is required to obtain the estimates of both SAA upper and lower bounds on z* with a certain confidence interval. The next two following sections present the estimation of the

SAA bounds by applying the similar method to that in [7], [21], [24].

The optimal solution based on the SAA approach can be obtained when N is sufficiently large. However, we can select an optimal solution by solving different SAA problems with different size of N. Until the same solution is found in these problems, we can choose the solution as the desired solution. The detail and result of this solution method is presented in the next section.

Provisioning Algorithms

i. Bender Decomposition

Fig. 9 shows the bound convergence obtained by solving Bender decomposition algorithm. The adjustment of lower and upper bounds is performed in each iteration. At iteration v = 42, algorithm converges. The optimal solution obtained from the decomposition is the same as one obtained by solving DEF without decomposition. We observe that the subproblems can be solved efficiently due to their smaller number of variables and parallelization. However, solving the master problem requires substantial amount of time since more Benders cuts have to be added.

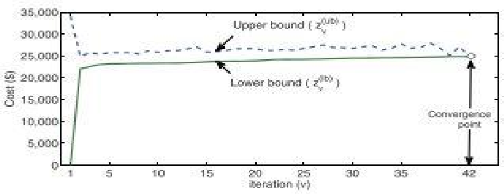

Fig. 9. Convergence of the upper and lower bounds by applying the Benders decomposition.

Discussion

i. Experimental Results

1. Balance of costs. We observe that the cloud broker with OCRP will minimize on-demand cost rather than the oversubscribed cost. Since resource pricing in the on-demand plan is higher and possibly increased by cloud providers, the reservation plan is more attractive by the cloud broker. However, reserving too many VMs may not be optimal (e.g., as that of MaxRes from Table 4). Therefore, the tradeoff between on-demand and oversubscribed costs needs to be adjusted in which OCRP can optimally perform.

2. Virtual machine outsourcing. The VM outsourcing from a private cloud to a public cloud provider (or public cloud) shows interesting result. In the experiment, the private cloud fully utilizes its own resources. Then, extra VMs can be spilled over to public clouds. Purchasing and deploying new hardware to a private cloud may not be an optimal solution, since the total cost of ownership (TCO) must be considered. To reduce this TCO, workload outsourcing is the attractive choice which is shown from our evaluation.

ii. Implementation Issues

1. Multiple provisioning stages planning issue. The OCRP algorithm can be applied to multiple provisioning stages representing long-term planning. Since the optimal solution of the first provisioning stage depends on multiple probability distributions describing the uncertainty occurring in consequent time epochs, multiple stages planning is needed. For example, the workload of some online souvenir shopping websites could be dramatically increased in the high-season consisting of many time periods in a year (e.g., Christmas Day, Valentine's Day, etc.). As a result, the websites should provision resources by considering multiple time epochs (i.e., provisioning stages) in advance, while reservation contracts offered by cloud providers can be taken into account to reduce the provisioning cost.

2. Use of decomposition method. The use of decomposition method for OCRP has to be carefully considered, since the formulation of the OCRP algorithm is a pure integer program which is the NP-hard problem. Although the subproblems can be solved in parallel, the master problem with the additional Benders cuts requires considerable computational time. The performance improvement of the decomposition algorithm will be considered in the future work.

3. Benefit of SAA. Sample-average approximation method can overcome the provisioning problems with a large set of scenarios which are impossible to solve with deterministic equivalent formulation directly. In contrast, the approximation algorithm with estimation of SAA lower and upper bounds can yield tolerably solutions while the problems can be practically solved in timely manner.

4. Limitation of stochastic programming. Stochastic programming does not address the method to obtain appropriate probability distributions describing un-certainty (i.e., distributions of scenarios Ω). How-ever, this limitation can be alleviated by applying variance reduction techniques (e.g., importance sampling) to increase the precision of the estimates of uncertainty.

iii. Future Research Direction

For the future work, scenario reduction techniques will be applied to reduce the number of scenarios. In addition, the optimal pricing scheme for cloud providers with the consideration of competition in the market will be investigated.

V. CONCLUSION

In this paper, we have proposed an optimal cloud resource provisioning (OCRP) algorithm to provision resources offered by multiple cloud providers. The optimal solution obtained from OCRP is obtained by formulating and solving stochastic integer programming with multistage recourse. We have also applied Benders decomposition approach to divide an OCRP problem into subproblems which can be solved parallelly. Furthermore, we have applied the SAA approach for solving the OCRP problem with a large set of scenarios. The SAA approach can effectively achieve an estimated optimal solution even the problem size is greatly large. The performance evaluation of the OCRP algorithm has been performed by numerical studies and simulations. From the results, the algorithm can optimally adjust the tradeoff between reservation of resources and allocation of on-demand resources. The OCRP algorithm can be used as a resource provisioning tool for the emerging cloud computing market in which the tool can effectively save the total cost.

REFERENCES

[1] I. Foster, Y. Zhao, and S. Lu, “Cloud Computing and Grid Computing 360-Degree Compared,”Proc. Grid Computing Envir-onments Workshop (GCE ’08),2008.

[2] F.V. Louveaux, “Stochastic Integer Programming,”Handbooks in OR & MS,vol. 10, pp. 213-266, 2003.

[3] A.J. Conejo, E. Castillo, and R. Garcı´a-Bertrand, “Linear Program-ming: Complicating Variables,”Decomposition Techniques in Mathematical Programming,chapter 3, pp. 107-139, Springer, 2006.

[4] J. Linderoth, A. Shapiro, and S. Wright, “The Empirical Behavior of Sampling Methods for Stochastic Programming,”Ann. Opera-tional Research,vol. 142, no. 1, pp. 215-241, 2006.

[5] G. Juve and E. Deelman, “Resource Provisioning Options for Large-Scale Scientific Workflows,”Proc. IEEE Fourth Int’l Conf. e-Science,2008.

[6] Z. Huang, C. He, and J. Wu, “On-Demand Service in Grid:Architecture Design, and Implementation,”Proc. 11th Int’l Conf. Parallel and Distributed Systems (ICPADS ’05),2005.

[7] Y. Jie, Q. Jie, and L. Ying, “A Profile-Based Approach to Just-in-Time Scalability for Cloud Applications,”Proc. IEEE Int’l Conf. Cloud Computing (CLOUD ’09),2009.

[8] Y. Kee and C. Kesselman, “Grid Resource Abstraction, Virtualiza-tion, and Provisioning for Time-Target Applications,” Proc. IEEE Int’l Symp. Cluster Computing and the Grid,2008.

[9] A. Filali, A.S. Hafid, and M. Gendreau, “Adaptive Resources Provisioning for Grid Applications and Services,”Proc. IEEE Int’l Conf. Comm.,2008.

[10] D. Kusic and N. Kandasamy, “Risk-Aware Limited Lookahead Control for Dynamic Resource Provisioning in Enterprise Com-puting Systems,”Proc. IEEE Int’l Conf. Autonomic Computing,2006.

[11] K. Miyashita, K. Masuda, and F. Higashitani, “Coordinating Service Allocation through Flexible Reservation,” IEEE Trans. Services Computing,vol. 1, no. 2, pp. 117-128, Apr.-June 2008.

[12] J. Chen, G. Soundararajan, and C. Amza, “Autonomic Provision-ing of Backend Databases in Dynamic Content Web Servers,”Proc. IEEE Int’l Conf. Autonomic Computing,2006.