SLA Based Dynamic Virtualized Resources Provisioning for Shared Cloud Data Centers

Zhiliang Zhu[1,2], Jing Bi[1], Haitao Yuan[2], Ying Chen[3]
[1]*School of Information Science and Engineering*, [2]*College of Software*,
Northeastern University, *Shenyang 110004*, *P.R. China*
[3]*IBM Research-China*, *Beijing 100193*, *P.R. China*
zzl@mail.neu.edu.cn, *neubijing@gmail.com*, *neuhaitao@gmail.com*, *yingch@cn.ibm.com*

***Abstract*—Cloud computing focuses on delivery of reliable, secure, sustainable, dynamic and scalable resources provisioning for hosting virtualized application services in shared cloud data centers. For an appropriate provisioning mechanism, we developed a novel cloud data center architecture based on virtualization mechanisms for multi-tier applications, so as to reduce provisioning overheads. Meanwhile, we proposed a novel dynamic provisioning technique and employed a flexible hybrid queueing model to determine the virtualized resources to provision to each tier of the virtualized application services. We further developed meta-heuristic solutions, which is according to different performance requirements of users from different levels. Simulation experiment results show that these proposed approaches can provide appropriate way to judiciously provision cloud data center resources, especially for improving the overall performance while effectively reducing the resource usage extra cost and maximizing the global profit of cloud infrastructure providers.**

Keywords-cloud computing; provisioning; Service level agreement (SLA); performance modeling

I. INTRODUCTION

Cloud computing delivers infrastructure, platform, and software (application) as services, which are made available as subscription-based services in a pay-as-you-go model to consumers. These cloud services in industry are respectively referred to as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Nowadays, large-scale and diverse cloud services are running in cloud data center. The main purpose of management of cloud data center is to ensure the quality and cost-effectiveness of cloud services so as to achieve much economic profits. The online clients can get services by sending their requests with corresponding parameters and invoking automatic execution of flows from cloud data centers. Meanwhile, the cloud services should provide effective Service Level Agreements (SLAs) to ensure and differentiate services quality. SLA is agreement about quality of cloud services signed between cloud service providers and consumers. SLA specifies concretely expected performance metric and charging model, which include response time, throughput, availability, reward and penalty, and so forth. Furthermore, cloud computing relies on virtualization technologies. Virtualization can enable server consolidation, namely multiple virtual machines served on the same physical machine. Each virtual machine supports the operating system and application environments of a server being consolidated in the virtualized environment. Virtualization can also enable dynamic or automatic resource provisioning to adapt to fluctuating computing demands. Therefore, virtualization is important for establishing cost-effective and dynamic IT infrastructure. However, users are willing to focus more on adapting applications for their business demands and outsourcing the IT infrastructure of hosting their application services to cloud environment. Current cloud computing paradigms are not easily to meet users' purpose, especially facing the requirement of diverse applications from different users.

In order to meet the constraint of SLA and provisioning existing virtualized resources optimally, this paper focuses on the problem of virtualized resources provisioning for existing cloud data center and aims to satisfy the requirement of users' business with virtualization technologies, and maximize the overall profit of IaaS providers when SLA guarantees are satisfied or violated. Firstly, we present the overall architecture based on the cloud IaaS. The architecture ensures that virtualized application services can execute effectively, and meet SLA requirement of user. Furthermore, we can dynamically provision the virtualized resources and establish performance optimization model for cloud environment-oriented multi-tier virtualized application services. Then, the optimal control strategies are designed for on-demand dispatcher. The on-demand dispatcher can decide to turn ON or OFF virtual machines according to the dynamic variations of workload. We further develop meta-heuristic solutions based on the mixed tabu-search optimization algorithm to solve the optimization problem, which accords to different performance requirements of users from different levels. The algorithm can ensure the maximal profit of cloud IaaS. Experiment results are presented to show the benefits of our approach.

The main contributions of the paper include: (1) We develop a novel cloud data center architecture based on virtualization mechanisms for application services; (2) We propose a flexible hybrid queueing model to determine the virtualized resources to provision to each

978-0-7695-4460-1/11 $26.00 © 2011 IEEE
DOI 10.1109/CLOUD.2011.91

tier of the virtualized application services; (3) Based on the proposed hybrid model, we develop meta-heuristic solu- tions to ensure to maximize the global profit of cloud IaaS providers; (4) The proposed optimization methodology is verified with the three-tier virtualized application services of handling dynamic workloads through simulation.

The remainder of the paper is organized as follows. Section 2 gives related works. Section 3 describes the overall architecture design of cloud IaaS. Section 4 formalizes the allocation problem of virtualized resources and establishes resolution performance model to solve the problem. Section 5 describes several different resources provisioning methods. Section 6 gives performance evaluation results and corresponding analysis. Section 7 demonstrates the quality and efficiency of our solutions.

II. RELATED WORK

In recent years, some researches have focused on the problem of resource management and performance control in data center [1, 2]. However, nowadays, most of those methods can not sufficiently adapt to complex cloud environment. These research efforts usually assume the system as the equilibrium state, and employ the method of average value analysis which is not sufficiently precise. E.g., aiming at increasingly complex computing system, in [1], IBM propose the conception of autonomic computing, whose definition is a technology which can realize self-management of system with the least manual intervention. The core ideology is self-management, i.e., resources are allocated on demand to multiple running services environments in cluster systems. Based on this conception, the authors in [2] propose that utility function is employed as objective function of self-management resources, which can be used to decide appropriate behavior of every component according to utility value. However, cloud computing focus on the infrastructure resource provided as well as the service itself provided. Meanwhile, IT infrastructure resources can be provided in the form of service. Utility computing method can not meet dynamic requirement of virtualized service resources in cloud environment.

In [3], the authors propose a dynamic capacity allocation resolving model which is focused on multi-tier network applications, which can decide the amount of resources should be allocated to every tier of application services, and judge the proper time at which these resources are allocated with the combination of prediction method. In [4], the authors propose resources allocation controller which can be used in multi-tier data center, and can maximize profit according to a particular charging model. The heuristic solving method is also designed. However, its limitation is that this model employ physical resources allocation mode, and its assumed is that available resources are sufficient. In addition, different application tiers are not distinguished.

Based on the resource reallocating scheme provided by VMMs (Virtual Machine Monitor, i.e. Xen [5] and VMware [6]), many researchers [7, 8] focus on improving resource utilization as well as guaranteeing quality of the hosted services via on-demand local resource scheduling models or algorithms within a physical server. However, most of them could not be good solutions to tradeoff between resource utilization and SLA. For example, the authors in [7] present a novel system-level application resource demand phase analysis and prediction prototype to support on-demand resource provisioning. The process takes into consideration application's resource consumption patterns, pricing schedules defined by the resource provider, and penalties associated with SLA violations. The authors in [8] improve resource utilization and performance of some services by hugely reducing performance of others. How to improve resource utilization, as well as guarantee SLA, is a challenge in a VM-based cloud data center. In [9], the authors also propose a set of local resource scheduling algorithms, which improve the resource utilization as well as improve performance of some critical services with small performance degradation of others. Yet, local optimization could not always lead to global optimization [10]. It is necessary to provide a global resource scheduling in a shared cloud computing environment. The authors in [11] present a flexible two-level resource management system that is able to provide high quality of service with much lower resource allocation costs than worst-case provisioning. Its fuzzy model guarantees for the maximization of global profit. All the above methods are the performance model of single tier application.

However, our methods in this paper can ensure to maximize the global profit of cloud IaaS providers according to different requirement of resources under different workload. Meanwhile, in order to meet the requirement of SLA signed with users, virtualized resources are employed as provisioning unit to make self-adaptive and dynamic adjustment. In addition, virtualized resources can be required and released on demand, so the flexibility and scalability of control problem in SLA can be greatly improved. Furthermore, probabilistic analysis can be made about all arrival requests so these requests can be processed with the hypotheses that demand in SLA agreement must be satisfied.

III. VIRTUAL MACHINE BASED CLOUD IAAS ARCHITECTURE

The description of the architecture on cloud IaaS is showed in Figure 1, which is a multi-tier architecture, including servers cluster and virtual machines (VMs) cluster. BPaaS or SaaS providers (IT consumers and users) sign SLA agreement with a cloud IaaS provider. In order to make a profit, BPaaS or SaaS providers require an effective approach to meet their users' specific requirement about quality of services. Furthermore, the goal of cloud IaaS providers is to design the most effective and economic strategy to manage their available resources and share vast application services.

Nowadays, typical online Web application services are usually designed as multi-tier model. Every tier runs in different servers' clusters, on which VM clusters are running in cloud IaaS architecture. In order to make virtualized resources allocated reasonably to meet the SLA requirements of users, an on-demand dispatcher (ODD) is designed to dispatch strategies in cloud IaaS. At the same time, to avoid making allocator become the bottle-neck of cloud IaaS, multiple VM dispatchers which employ self-adaptive load balancing algorithms are running in allocator. The dispatchers can distribute requests from different business in different levels of virtualized application services (VASs) to be executed on one specific VM (virtualized service resources) in the back-end. What's more, the dispatchers give full consideration to factors like current load and requests allocation situation of VMs in the back-end. Here VAS refers to the application that is published together with VM into which applications and operating systems are packaged. The load information (CPU, memory, disk, I/O, and so on) are provided by VMs manager in virtualized application service environment (VASE), which reports regularly the latest load information to ODD. ODD can decide to start or stop VMs according to the latest load situation of system, whose goal is to meet services requirements of final users and meanwhile, maximize global profit of cloud IaaS providers.

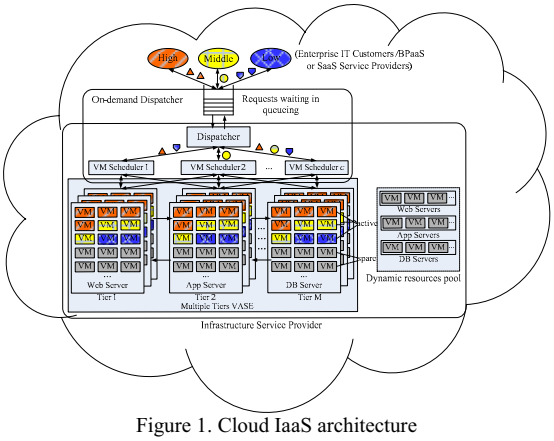

Figure 1. Cloud IaaS architecture

The dynamic flexibleness of different VASs leads to frequent occupation and release of resources from corresponding tier. In order to ensure that VAS can get required resources when required and improve utilization rate of virtualized services resources to reduce operational maintenance cost of cloud IaaS, one of the important guarantee of design of cloud IaaS architecture is high-efficiency reuse of virtualized services resources. It is common that cloud IaaS maintain virtualized services resources in dynamic resources pool. When VAS requires resources, it will get corresponding services resources from dynamic resources pool. What's more, corresponding services resources which can be used by other are recycled into resources pool. In this way, cloud IaaS can make use of virtualized services resources efficiently and adjust resources occupied by different VAS in a dynamic and real time way according to dynamic variations of workload.

IV. THE SYSTEM PERFORMANCE MODEL

In this section, the constrained non-linear optimization problem is defined for dynamic virtualized resources optimization and the hybrid optimization performance model is also established.

A. Optimization Problem Description

The section formulizes virtualized resources allocation problem for cloud IaaS in Figure 1. Assume that N M-tier VASEs run in cloud IaaS, which include multiple different user classs K. Physical server from each tier is shared by multiple VMs serving different virtualized applications. Furthermore, a VASE may include multiple VMs that are distributed on physical servers from several tiers. Assume the number of clients classs in VASE i is K_i, and there are $n_{i,j}$ VMs in the jth tier. So, the crucial variable of the problem is defined as a $N\times(M+1)$ matrix, *ConfigMAT*, which refers to the allocation plan of VMs on physical servers in each tier, formally:

$$ConfigMAT=\begin{bmatrix} c_{1,0} & c_{1,1} & \cdots & c_{1,M} \\ c_{2,0} & \ddots & c_{i,j} & \vdots \\ \vdots & & \ddots & \\ c_{N,0} & \cdots & & c_{N,M} \end{bmatrix}$$

Let $c_{i,j}$ represent the number of active VMs allocated in the jth tier of VAS i. If $c_{i,j}$ is 0, it means that no active VMs exist in the jth tier of VAS i. In order to control the granularity of VMs allocation, the upper limit of $c_{i,j}$ is set as C_i, which refers to the maximal number of VMs of VAS i, in all cloud IaaS. $n_{i,j}$ refers to maximal number of VMs occupied in the jth tier of VAS i. The allocation matrix *ConfigMAT* is considered as valid if following constraint is met.

$$\begin{cases} 0<\sum_{j=0}^{M} c_{i,j} \le C_i, & \forall i\in[1,N] \\ 0\le c_{i,j}\le n_{i,j}, & \forall i\in[1,N], \forall j\in[0,M] \end{cases} \quad (1)$$

The constraint (1) restricts that the total number of VMs occupied in all cloud infrastructure and the number of VMs occupied in the same tier can not exceed the total number of available virtualized resources. Each VASE i has its corresponding local profit function definition, denoted by:

$$P_i = f\left(\lambda_i, c_{i,0}, c_{i,1}, ..., c_{i,M}, SLA_i\right)$$

The global profit value P_g is function of every local profit value of VASE, so the whole optimization problem can be formulized as following problem (P_1):

$$\max\left\{P_g = g\left(P_1, P_2, ..., P_N\right)\right\} \quad (2)$$

In order to maximize the profit of cloud IaaS providers based on SLA, on the condition that equation (1) is met, the global profit value in equation (2) is optimized. Furthermore, virtualized resources of cloud data center can be used effectively. The concrete form of problem (P_1) will be presented in latter part of this section.

The profit function used is described as follows. Here, the analysis is focused on multi-tier VASs which include multiple classes of online businesses in virtualized applications services environment. The arrival rate of the request class k in the jth tier of VASE i is represented as $\lambda_{i,k,j}$, and response time $R_{i,k}$ is considered as a performance metric. Assume the SLA agreement has been signed between cloud IaaS and clients before the system runs, where the specific performance requirements and charging model are defined as follows:

- $\overline{R}_{i,k}$ - the expected SLA target response time of request level k in VASE i. If a request is served in target response time, the positive profit is contributed for cloud IaaS providers, i.e., if $R_{i,k} \le \overline{R}_{i,k}$, SLA_i is the profit type. Otherwise, the case that a request is served beyond target response time will bring cloud IaaS providers penalty, i.e., if $R_{i,k} > \overline{R}_{i,k}$, SLA_i is the penalty type.
- $\overline{AV}_{i,k}$ - VM expected value of SLA goal availability in request level k of VASE i, that is, client can accept the minimal value of availability. If $AV_{i,k} \ge \overline{AV}_{i,k}$, client can accept cloud IaaS provider. Otherwise, client can not accept.
- C_i - maximal VMs number of VASE i in all cloud IaaS. If $\sum_{j=0}^{M_{i,k}} c_{i,j} \le C_i$, the refusal of clients' requests will lead to the penalty of $d_{i,k}$. i.e., when actual the number of VMs exceeds the concerted upper limit value, the refused clients' requests will not be counted into penalty. This makes clients must estimate actual requirements of applications services carefully and make an appropriate plan of expense before deployment of applications services.
- $c_{i,k,j,w}^{active}$ - average price of active VM w in the jth tier of request level k in VASE i.
- $c_{i,k,j,w}^{spare}$ - average price of spare VM w in the jth tier of request level k in VASE i.

Our goal is to maximize profit value of cloud IaaS providers, furthermore, the difference between revenue, and penalty, loss and cost of VMs from SLA can be maximized. The global profit function can be formulized as:

$$\begin{aligned} Profit(E) = \sum_{i=1}^{N}\sum_{k=1}^{K_i}\Big\{\Lambda_{i,k}\cdot\left(\left(-m_{i,k}\right)\cdot R_{i,k} + u_{i,k}\right) - \left(d_{i,k}\cdot x_{i,k}\right) - \\ \left(LV_{i,k}\cdot\left(1 - AV_{i,k}\right)\right)\Big\} - \sum_{i=1}^{N}\sum_{k=1}^{K_i}\sum_{j=0}^{M}\left(\sum_{w=1}^{c_{i,j}} c_{i,k,j,w}^{active} + \sum_{w=1}^{n_{i,j}-c_{i,j}} c_{i,k,j,w}^{spare}\right) \end{aligned} \quad (3)$$

where

- $\Lambda_{i,k}$ is total arrival rate of request class k in VASE i.
- $R_{i,k}$ is end-to-end response time of request class k in VASE i, formulized as:

$$R_{i,k} = \frac{1}{\Lambda_{i,k}}\left(\lambda_{i,k,0}\cdot R_{i,k,0} + \sum_{j=1}^{M}\sum_{w=1}^{c_{i,j}}\lambda_{i,k,j,w}\cdot R_{i,k,j,w}\right)$$

- $\lambda_{i,k,j,w}$ is arrival rate of VM w in requests class k in the jth tier in VASE i.
- $m_{i,k} = u_{i,k} \Big/ R_{i,k} > 0$, $-m_{i,k}$ refers to slope of utility function $u_{i,k}$.
- $u_{i,k}(x) = \dfrac{bestVal - x}{bestVal - worstVal} \in [0...1]$, here, x equals to $R_{i,k}$, $bestVal$ is 0, $worstVal$ is $\overline{R}_{i,k}$.
- $AV_{i,k}$ is the availability of VMs for request level k in VASE i.

  $$AV_{i,k} = \prod_{j=0}^{M}\left(1 - FV_{i,k,j}\right) = \prod_{j=0}^{M} AV_{i,k,j}$$

  where $FV_{i,k,j} = \dfrac{DT_{i,k,j}}{UT_{i,k,j} + DT_{i,k,j}}$ is the jth tier failure probability; $AV_{i,k,j} = \dfrac{UT_{i,k,j}}{UT_{i,k,j} + DT_{i,k,j}}$ is the jth tier availability; $UT_{i,k,j}$ is the jth tier available state time of VMs; $DT_{i,k,j}$ is the jth tier failure state time of VMs.
- $LV_{i,k}$ is the loss value of failure for request class k in VASE i.
- To request class k, $x_{i,k}$ is the number of refused requests which can lead to penalty. $d_{i,k}$ is each unit penalty of requests class k in VASE i.

So the global optimal problem (P_1) can be formalized as:

$$\begin{aligned} &\max_{\lambda_{i,k,j,w}, \mu_{i,k,j,w}, c_{i,j}} \left\{P_g = g\left(P_1, P_2, ..., P_N\right)\right\} \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K_i}\Big\{\left(-m_{i,k}\right)\cdot\left(\lambda_{i,k,0}\cdot R_{i,k,0} + \sum_{j=1}^{M}\sum_{w=1}^{c_{i,j}}\lambda_{i,k,j,w}\cdot R_{i,k,j,w}\right) - \left(d_{i,k}\cdot x_{i,k}\right) \\ &\quad - \left(LV_{i,k}\cdot\left(1 - AV_{i,k}\right)\right)\Big\} - \sum_{i=1}^{N}\sum_{k=1}^{K_i}\sum_{j=0}^{M}\left(\sum_{w=1}^{c_{i,j}} c_{i,k,j,w}^{active} + \sum_{w=1}^{n_{i,j}-c_{i,j}} c_{i,k,j,w}^{spare}\right) \end{aligned}$$

subject to

$$\Lambda_{i,k} = \lambda_{i,k,0} = \sum_{w=1}^{c_{i,1}}\lambda_{i,k,1,w}\ , \forall i,k\ , \quad (4)$$

$$\sum_{w=1}^{c_{i,j}} \lambda_{i,k,j,w} = \lambda_{i,k,j}\ , \forall i,k,j\ , \tag{5}$$

$$AV_{i,k} \geq \overline{AV}_{i,k}\ , \tag{6}$$

$$\lambda_{i,k,j} < \sum_{w=1}^{c_{i,j}} \mu_{i,k,j,w}\ , 0 \leq j \leq M \tag{7}$$

$$\lambda_{i,k,j,w} \geq 0\ , \forall i,k,j,w\ .$$

$\sum_{i=1}^{N}\sum_{k=1}^{K_i} \Lambda_{i,k} u_{i,k}$ has been omitted in above objective function, because it does not depend on decision-making variable. The constraint (4) shows that all load are estimated as the requests arrival rate in the *0*th tier of request level *k* in VASE *i*, i.e., the sum of requests allocated on VMs arrival rate of the *1*th tier of request level *k* in VASE *i*. The constraint (5) shows that the arrival rate in the *j*th tier of of request level *k* in VASE *i* is equal to the sum of requests arrival rate of VM *w* in the *j*th tier of request level *k* in VASE *i*. The constraint (6) restrict the availability $AV_{i,k}$ of VMs no less than goal availability $\overline{AV}_{i,k}$. The constraint (7) allows request level *k* in VASE *i* to be executed on VM *w* only when the *j*th tier of request level *k* in VASE *i* has been allocated to VM *w*.

B. Analysis of Optimization Problem

The section mainly aims on online virtualized applications services, so response time is viewed as main performance metric to measure quality of services in VASE. The request arrival rate is adopted to represent density of load. The proposed model can effectively support the dynamic provisioning of cloud IaaS resources, and satisfy requirements of SLA signed with users, as showed in Figure 2.

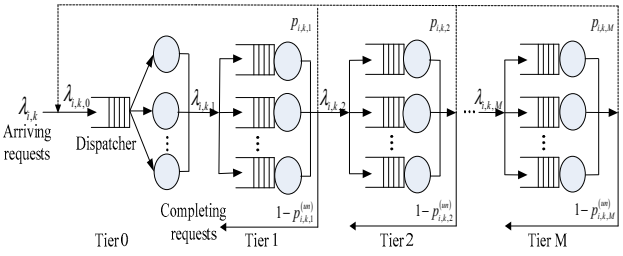

Figure 2. Network queueing model

In cloud computing environment, a large amount of clients issue requests for resources in cloud IaaS. The requests are either waiting in queueing in a certain tier or being served in a certain tier before they leave systems. What's more, a part of clients may leave the system at once after having visited services resources of a certain tier, or return to initial state of the system and revisit after having visited services resources of a certain tier. So the hybrid queueing network is adopted to establish performance resolution model for our system. This model can trace actions of every tier in VASE, e.g. HTTP, J2EE, and Database tier. The service principle of FCFS (First Come First Served) is adopted by queue, i.e., clients' requests are served according to the corresponding arrival sequence. In the manner of request class *k*, clients' requests arrive in cloud data center and visit services in VASE *i*, and the requests rate is $\lambda_{i,k}$. The locus analysis of actual network business website [12] has shown that network workload conforms to Poisson distribution. So suppose exterior requests arrival stream conforms to Poisson distribution, and the interval of arrival time conforms to exponential distribution. Let $\Lambda_{i,k} = \lambda_{i,k,0}$, where, $p_{i,k,j}$ refers to probability of request class *k* which finishes serving requests of the *j*th tier and return to initial state to reserve requests. $p_{i,k,j}^{(un)}$ represents probability of request class *k* which finishes serving requests of the *j*th tier and arrive in the *j*+1 tier in VASE *i*, meanwhile, the probability of $1-p_{i,k,j}^{(un)}$ of clients in the *j*th tier in VASE *i* finish the process of request class *k* and return. $\lambda_{i,k}$ refers to the arriving requests rate of request class *k* in VASE *i*. As showed in Figure 2,

$$\lambda_{i,k,0} = \lambda_{i,k} + \lambda_{i,k,1}p_{i,k,1} + \lambda_{i,k,2}p_{i,k,2} + \cdots + \lambda_{i,k,M}p_{i,k,M} \tag{8}$$

Let $M_{i,k} = M$ and $j = 0$, then $\lambda_{i,k,1} = p_{i,k,0}^{(un)}\lambda_{i,k,0}$, $\lambda_{i,k,2} = \left(p_{i,k,1}^{(un)} - p_{i,k,1}\right)\cdot\lambda_{i,k,1}$, $\lambda_{i,k,3} = \left(p_{i,k,2}^{(un)} - p_{i,k,2}\right)\cdot\lambda_{i,k,2}$, ... , $\lambda_{i,k,M} = \left(p_{i,k,M-1}^{(un)} - p_{i,k,M-1}\right)\cdot\lambda_{i,k,M-1}$, i.e., $\lambda_{i,k,j} = \left(p_{i,k,j-1}^{(un)} - p_{i,k,j-1}\right)\cdot\lambda_{i,k,j-1}$, and $p_{i,k,0}^{(un)} = 1$, $0 \leq p_{i,k,j-1}^{(un)} \leq 1$, $p_{i,k,M} = p_{i,k,M}^{(un)}$, $(\forall j \in [1, M])$.

Then

$$\lambda_{i,k,0} = \lambda_{i,k} \Big/ \left(1 - p_{i,k,1} - \sum_{j=2}^{M_{i,k}} (p_{i,k,j} \cdot \prod_{q=1}^{j-1} (p_{i,k,q}^{(un)} - p_{i,k,q}))\right) \tag{9}$$

Here, on-demand dispatcher (ODD) ($j = 0$) is modeled as an *M/M/c* system model, in which, there are *c* schedulers for VMs all together. The effective utilization rate of ODD is ensured as 60%~80%. According to Little's law [13], we can compute the average end to end response time of ODD in VASE *i*, namely $R_{i,k,0}$.

Then establish multiple *M/G/*1 performance resolution models for other tier in multi-tier VASE *i*. The common distribution requirement is solved by the approach of embedding Markov chain [14]. It is assumed that clients' requests are scheduled arrive in VM *w* at the rate of $\lambda_{i,k,j,w}$, $1 \leq w \leq c_{i,j}$. We can compute the value of average response time of every tier in VASE *i*, $1 \leq j \leq M$, namely $R_{i,k,j,w}$. What's more, $\rho_{i,k,j} = \lambda_{i,k,j} \Big/ \sum_{w=1}^{c_{i,j}} \mu_{i,k,j,w} < 1$ is the utilization rate of resource (e.g., CPU) allocated to VMs in every tier of VASE *i*. We further developed meta-heuristic solutions based on the hybrid stochastic optimization algorithm.

V. PROVISIONING METHOD

The real workload varies in a continuous and dynamic way in cloud data center. In order to make that cloud IaaS can provide the guarantee of SLA for multi-tier virtualized applications services, the virtualized resources must be allocated on demand in a dynamic way. Thus, the changing trend of workload of multiple VASs can be monitored in time. The concrete distributed method of virtualized resources will influence the control efficiency and optimization effect to a great extent. According to the cloud IaaS architecture, which proposes management based on virtualized mechanism in section 4, the dynamic provisioning of virtualized resources is designed and optimized, whose goal is to ensure to maximize the global profit of cloud IaaS providers.

A. Dynamic Provisioning Strategy based on Virtual Machine

The goal of problem (P_1) is to find optimal configuretion variable matrix which makes global profit of next period maximum with some constraints. Because the constraint optimization problem is a NP-hard problem, a local domain search method can be adopted, i.e., gradient-climbing method [15], where resources of every unit are allocated to the services whose marginal revenue are the highest, i.e., proceed in the direction of slope of current point and move up hill until the goal converges to a particular fixed point. Although gradient-climbing method is commonly used and easily understood, its search performance depends completely on domain structure and initial solution and it converges very slowly around the optimal solution. What's more, the process only can guarantee to converge to local optimal solution, and it can not guarantee to converge to global optimal solution.

To explore for the global optimal value, we have to use this method in conjunction with particular stochastic optimization methods. Here, we take a modified Tabu Search (TS) [16] for example to show how the optimization is performed. We would not discuss the details and results of the hybrid tabu-search algorithm in this paper, due to the limitation of paper page. A high-quality initial allocation plan of VMs is adopted as initial solution of the whole search process by hybrid tabu-search algorithm, In every loop, current matrix is disturbed and a new allocation is generated as initial solution of gradient descent. After reaching a particular fixed point, the variation of profit is calculated. Then the best configuration so far is recorded to guarantee the convergence of all the search process. To evaluate the effectiveness of dynamic provisioning strategy based on virtual machine (DVM-Pro), another two resource allocation strategies are presented, which are described briefly as follows.

B. Static Resources Allocation Strategy

Static resources allocation strategy is called Stat-RA. Stat-RA method is simple resources allocation method. Before applications services run, services resources of the whole cloud data center are allocated to each VASE according to different priority levels of applications services and the allocation state of resources remain unchanged all the time. So, this allocation method is not flexible when workload varies inevitably. The problem of how to allocate resources capacity requirement is considered carefully before the system is put into operation. However, it will no doubt lead to low utilization rate of resources if resources are prepared according to workload of the worst case. The resources of cloud data center will be wasted in a great deal and the profit will be reduced. When resources are prepared according to workload of common situation, the SLA penalty may occur when workload peaks unexpectedly.

C. Dynamic Resources Allocation Strategy based on Physical Machine

Dynamic resources allocation strategy based on physical machine is called DPM-RA. DPM-RA which is based on physical resources makes that every node runs in a certain tier of applications services, multiple applications services which runs in different nodes can be isolated effectively. The allocation model can measure and estimate states of system in every period. The physical resources can be adjusted accordingly on demand according to changes of workload. One hand, when the requirement of a particular application service is cancelled, the corresponding node which provides this service can be used to meet the new requirement of application services. On the other hand, when the requirement of a particular application service comes to peak of workload, resources can be gathered together by the system to cope with the requirement of the application service. And when the requirement reduces, the corresponding nodes can be reallocated to other applications services in a dynamic way.

VI. EXPERIMENTAL EVALUATION

The performance of virtualized resources optimization method based on SLA in cloud data center is evaluated by concrete experiments in this section. The total profit of cloud IaaS providers can be maximized. What's more, the virtualization technologies are employed to isolate different applications and add availability rate, and the performance of cloud IaaS can be further improved.

A. Experimental Setup

It is common that typical multi-tier VASs tend to distribute computing tasks to business logic tier and distribute data processing tasks to database tier in the back end. So nodes with higher abilities are distributed to the last two tiers. Two VASEs are included in cloud data center, in each of which runs corresponding class of business that can be visited by users of different levels. In order to evaluate the applicability of proposed multi-tier model method in complex cases, diverse different values of parameters are set to describe VASs, described in detail in Table 1, Table 2, and Table 3.

TABLE I. CHARACTERISTICS FOR RUBIS

Parameter	ODD tier	Web tier	App tier	DB tier
Single VM capacity (req/sec)	300	90	150	180
VMs number	7	16	10	8
$p_{i,k,j-1}^{(un)}$	-	-	0.82	0.89
$p_{i,k,j-1}$	-	-	0.11	0.16
Max throughput (%)	0.85	0.85	0.85	0.85

TABLE II. CHARACTERISTICS FOR TPC-W

Parameter	ODD tier	Web tier	App tier	DB tier
Single VM capacity (req/sec)	300	90	150	180
VMs number	6	12	8	7
$p_{i,k,j-1}^{(un)}$	-	-	0.81	0.85
$p_{i,k,j-1}$	-	-	0.15	0.17
Max throughput (%)	0.85	0.85	0.85	0.85

TABLE III. SLA CHARACTERISTICS FOR RUBIS AND TPC-W

Parameter	Level 1	Level 2	Level 3
Goal response time (sec)	0.2	0.4	0.5
Availability (%)	0.996	0.981	0.962
Average revenue ($)	2.4	1.8	1.0
Average penalty ($)	8.0	6.5	5.0

B. *Simulation Experiments*

In this section, the same setting of experiment is adopted to evaluate action and efficiency of several resources allocation methods described in section 5. Figure 3 shows the variation of accumulative net earnings of resources with time. It can be clearly found that when DVM-Pro method is used, though requests workload present obvious concussion, the profit keeps on increasing in a stable way all the time. When DPM-RA method is used, the rising trend is not effectively capture in the time of 960min~1140min, which causes a lot of punishment, so net earnings dropped significantly. Similarly, in accordance with the expected, the result of Stat-RA method is the worst. The reason is that its provisioning amount of resources never varies, so the changing trend of requests workload can not be captured completely. So the profit value presents the obvious tend of declining in this time of 960min~1140min while the method of DVM-Pro still keeps higher profit value, i.e., the adoption of DVM-Pro ensures to maximize global profit of cloud IaaS providers.

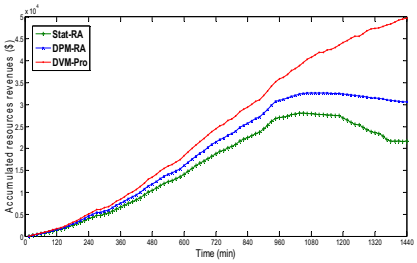

Figure 3. Accumulated resources revenues

In order to make a further and careful observation to the self-adaptive situation of resources provisioning, DVM-Pro method can capture the trend of workload variations precisely to a great extent. And the presented availability rate is relatively high, and the excessive requests are restrained only when the arrival workload exceeds the limit of total provisioning amount of virtualized resources, thus some penalties are generated. E.g., in cloud data center, suppose the upper limit of maximal amount of virtualized resources in Web tier is set to 28. As illustrated in Figure 4, the total provisioning amount of virtualized resources for two VASEs are insufficient to satisfy arrival workload requests in the time of 960min~1140min, so excessive requests requirement can be retrained to keep response time under the expected value of SLA. Furthermore, the global profit can be guaranteed to be higher.

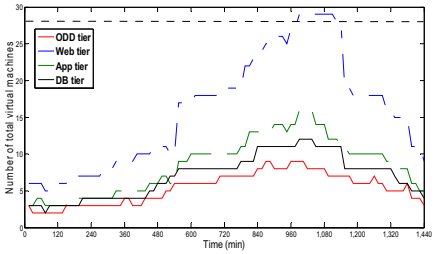

Figure 4. Total VMs of two VASEs

Due to the difference of charging strategies of three levels, in order to maximize global profit of cloud IaaS providers, clients of Level 1 (TPC-W) expect to get the best quality of services because they pay the most. When the total amount of resources in cloud data center is sufficient to cope with arrival workload requests, ODD will adjust strategies about provisioning of virtualized resources as reasonably as possible to meet SLA requirement of all clients' requests. However, when the amount of virtualized resources is relatively limited and insufficient to cope with the outburst of arrival workload requests, ODD will be sensitive to this and decide the way to allocate virtualized resources and refuse excessive requests. As illustrated in Figure 4, when total amount of virtualized resources is insufficient to cope with arrival requests and requests of every level compete for limited resources, DVM-Pro chooses to refuse part of requests of Level 2 (RUBiS), thus, the expected SLA requirement of Level 1 (TPC-W) can be ensured.

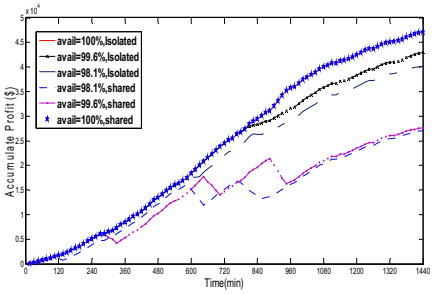

Figure 5. Accumulated profit

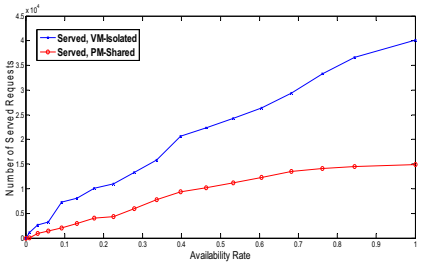

Figure 6. Number of served requests

In order to evaluate the performance isolation effect and efficiency when VMs are used as servers, a series of experiments are conducted to evaluate the impact that availability of system makes on performance when sharing and isolating services environment. Availability is defined as the availability probability of a tier of an application in unit time. In the experiments, assume availabilities of an application instance in every tier are equal. The length of time is set to 15 minutes. In order to compare variations of profit, the experiment for comparison is conducted between isolation and share of environments. The result is illustrated in Figure 5 and 6. Figure 5 shows the accumulated global profit in RUBiS environment. Figure 6 shows total amount of completed requests when different availabilities are set respectively. It can be found that the accumulative global profit of shared environment is equal to that of isolated environment when failures do not happen. However, with the decrease of availability, the performance of shared environment declines more quickly than that of isolated environment. It can also be found that with the increase of availability, isolated environment can serve more requests than shared environment, so superiority of isolated environment is more obvious. The results show that virtualized infrastructure resources can provide the ability of performance isolation. Furthermore, the availability and performance of the whole system can be improved.

VII. CONCLUSION

In this paper, a cloud IaaS architecture is proposed which ensures virtualized resources to be used reasonably. Furthermore, a multi-tier architecture oriented virtualized application performance resolving model is developed according to the dynamic characteristics of VMs and SLA restriction of clients. According to this model, virtualized resources can be allocated intelligently for various requests from users of different levels. Therefore, the global profit of cloud IaaS providers can be maximized. Simulation experiments show that the proposed method can provide appropriate services for users of different levels, especially for improving the profit of cloud IaaS provider while effectively reducing the resource usage extra cost. In future work, we are planning to incorporate new provisioning policies to cloud data centers, in order to offer a built-in support to allocate the currently available cloud data centers.

ACKNOWLEDGMENT

This work was supported in part by the IBM Ph.D. Fellowship Award, and the National Natural Science Foundation of China under Grant 60872040.

REFERENCES

[1] E. Kalyvianaki, T. Charalambous, and S. Hand, “Self-adaptive and self-configured CPU resource provisioning for virtualized servers using kalman filters”, Proceedings of the 6th international conference on Autonomic computing, Barcelona, Spain, June 15-19, 2009.

[2] W. E. Walsh, G. Tesauro, and J. O. Kephart, “Utility functions in autonomic systems”, Proceedings of the First IEEE International Conference on Autonomic Computing, New York, NY, USA, May 17-18, 2004.

[3] B. Urgaonkar, P. Shenoy, and A. Chandra, “Agile dynamic allocation of multi-tier Internet application”, ACM Trans. on Autonomous and Adaptive Systems, 2008, vol. 3, pp. 1-39.

[4] D. Ardagna, M. Trubian, and L. Zhang, “SLA based profit optimization in multi-tier systems”, Proceedings of the 4th IEEE International Symposium on Network Computing and Applications, Cambridge, Massachusetts, USA, July 27-29, 2005.

[5] P. Barham, B. Dragovic, and K. Fraser, et al, “Xen and the art of virtualization”, Proceedings of the nineteenth ACM symposium on Operating systems principles, NY, USA, 2003.

[6] VMware, Inc. VMware ESX Server User's Manual Version 1.5, Palo Alto, CA, April 2002.

[7] J. Zhang, M. Yousif, and R. Carpenter, et al, “Application resource demand phase analysis and prediction in support of dynamic resource provisioning”, Proceedings of the 4th International Conference on Autonomic Computing, 2007, pp. 12-12.

[8] P. Padala, X. Y. Zhu, M. Uysal, et al, “Adaptive control of virtualized resources in utility computing environments”, EuroSys, 2007, pp. 289-302.

[9] Y. Song, Y. Q. Li, and H. Wang, et al, “A service-oriented priority-based resource scheduling scheme for virtualized utility computing”, Proceedings of the International Conference on High Performance Computing (HiPC), 2008, pp. 220-231.

[10] L. S. Lasdon, “Optimization theory for large systems”, Courier Dover Publications, 2002.

[11] J. Xu, M. Zhao, and J. Fortes, “On the use of fuzzy modeling in virtualized data center management”, Proceedings of the 4th International Conference on Autonomic Computing, Jacksonville, Florida, USA, 2007.

[12] D. A. Menascé, and M. N. Bennani, “Autonomic virtualized environments”, Proceedings of IEEE International Conference on Autonomic and Autonomous Systems, Silicon Valley, California, USA. July 16-21 2006.

[13] J. McKenna, “A Generalization of Little's Law to moments of queue lengths and waiting times in closed, product form queueing networks”, Journal of Applied Probability, 1989, 26, pp. 121-133.

[14] P. A. Meyer, R. T. Smythe, and J. B. Walsh, “Birth and death of Markov processes”, Probability theory, 1972.

[15] R. Burachik, L. M. Grana Drummod, and A. N. Iusem, et al, “Full convergence of the steepest descent method with inexact line searches”, Optimization: A Journal of Mathematical Programming and Operations Research, 1995, vol. 32, pp. 137-146.

[16] F. Glover, and R. Marti, “Tabu Search”, Metaheuristic Procedures for Training Neutral Networks, 2006, vol. 36, pp. 53-69.