

Research Article

Research on Cloud Computing Resources Provisioning Based on Reinforcement Learning

Zhiping Peng,¹ Delong Cui,¹ Jinglong Zuo,¹ and Weiwei Lin²

¹College of Computer and Electronic Information, Guangdong University of Petrochemical Technology, Maoming 525000, China

²School of Computer Science and Engineering, South China University of Technology, Guangzhou, China

Correspondence should be addressed to Delong Cui; delongcui@163.com

Received 27 March 2015; Revised 2 June 2015; Accepted 4 June 2015

Academic Editor: Bao Rong Chang

Copyright © 2015 Zhiping Peng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As one of the core issues for cloud computing, resource management adopts virtualization technology to shield the underlying resource heterogeneity and complexity which makes the massive distributed resources form a unified giant resource pool. It can achieve efficient resource provisioning by using the rational implementing resource management methods and techniques. Therefore, how to manage cloud computing resources effectively becomes a challenging research topic. By analyzing the executing progress of a user job in the cloud computing environment, we proposed a novel resource provisioning scheme based on the reinforcement learning and queuing theory in this study. With the introduction of the concepts of Segmentation Service Level Agreement (SSLA) and Utilization Unit Time Cost (UUTC), we viewed the resource provisioning problem in cloud computing as a sequential decision issue, and then we designed a novel optimization object function and employed reinforcement learning to solve it. Experiment results not only demonstrated the effectiveness of the proposed scheme, but also proved to outperform the common methods of resource utilization rate in terms of SLA collision avoidance and user costs.

1. Introduction

The concept of cloud computing vividly reflects the characteristics of information service in Internet age; meanwhile the pursuit of the vision of cloud computing also brings new challenges to information technology. Acting as a significant application research, the data center is pushing a series of technology innovations to perform the key features of cloud computing, such as on-demand service, elasticity of extension, and massive data storage. The data center widely adopts virtualization technology to achieve the uncoupled mode of physical resource and application. Applications use Virtual Machine (VM) as a package unit to share various physical resources with others. Hence the resource schedule entities are represented by fine-grain VMs instead of coarse grain service machines. Virtualization technology provides convenience for the data center, but the VMs resource provisioning brings more challenges to the efficient management of data center infrastructure.

As one of the core issues for cloud computing, resource management aims to shield the underlying resource heterogeneity and complexity by adopting virtualization technology, which makes the massive distributed resources form a unified giant resource pool. Therefore, it can guarantee the efficient resource provision and use by implementing resources management methods and techniques rationally. Therefore, how to achieve effective management of cloud computing resources is faced with a number of new challenges, which are mainly shown in three types of imbalance.

(i) *First, Imbalance in the Needs of Applications.* Cloud computing application contains various behaviors of workload, from the control-intensive applications (such as search, sort, and analysis) to the data-intensive ones (image processing, simulation, modeling, data mining, etc.). In addition, it also includes the computationally intensive applications (iterative method, numerical method, financial modeling, etc.). The throughput of various applications depends heavily on

the VM resource provisioning, while not even any configuration can make all types of workloads run with optimal efficiency. Furthermore, most applications are featured by multiple types of workload. For example, control-intensive applications require more CPU resources for branch prediction, while data-intensive ones require more memory resources to avoid the frequent operation of reading and writing. The multitenant environment of cloud computing allows heterogeneous applications to share data center resources pool, and the demand for resources for each application is diverse, thus resulting in difficulty measuring the server-loading efficiency. Even for the different resources of the same server, it is prone to cause an imbalance, affecting the resource use efficiency. In cloud computing scenario, while simply the demand for forecasting and the rational purchase cannot solve this problem, a plausible resource provisioning scheme needs to be put forward to further solve the new contradiction between heterogeneous application and the unified resources-sharing pool.

(ii) *Second, Imbalance in the Application Time.* In reality, the server utilization of data center reaches merely 5% to 20%, while the peak workloads of many services are 2–10 times higher than the average. In addition to the different service loads in various periods of a day, most of the services vary in load demands according to the seasonal or other periodic changes (e.g., peak in December before Christmas sales and in photo processing sites after holiday); meanwhile some unexpected events (such as news) lead to changes. Few users deploy less resource than the peak demands, which is prone to waste the resources at the nonpeak time. As a result, the stronger the load is fluctuating, the more resources the users are wasting. In the cloud computing environment, this issue cannot be handled by static configuration mode. At the same time, the VM in the cloud is characterized by isolation performance, but mutual interference resulting from the resource competition between VMs cannot be avoided in the actual operation of the system process, affecting the performance of the whole cloud computing system.

(iii) *Thirdly, Imbalance in the Distribution of Applications.* For load balancing, the node servers of the load equalizers are not fixed physical machines but are VMs of the cloud, which require the load balancer to be equipped with the ability of dynamically adjusting the server cluster to the current user access, so as to avoid the resource waste and the situation that the current resources cannot meet the user's requests [1]. To some extent, numerous academic researches have been conducted on the trend prediction of the load and elastic assignment of resources; however, there are still some disadvantages. Having conducted a comprehensive study of academia and industry current status, we find that the research in this field has the following problems: first, it lacks flexibility. Apart from not considering the dynamic deployment of resources from a service-oriented perspective, some characteristics are not fully reflected, such as the elasticity of cloud computing and the characteristic that resources are adjusted to the user's needs. Second, it does not support the trend prediction, and the resource allocations are

apparently lagging behind, affecting the user's experience and even being unable to meet some of the user requests at times.

In conclusion, the data center must solve the problems such as resource multiplexing, correlation, and dynamic management. High-efficiency dynamic management for virtualization resources is the core issue of the optimal resource scheduler and also the key of how various resource service systems eventually provide appropriate and satisfying resources for users.

Focusing on the accurate scaled cloud computing environment and efficient resources allocation under Service Level Agreement (SLA) and user cost constraints, we introduce two concepts, Segmentation SLA (SSLA) and Utilization Unit Time Cost (UUTC), and then propose an optimization resource provisioning scheme based on reinforcement learning (RL) and queuing theory (QT) in this paper.

This research work has the following theoretical and practical contributions.

- (i) A novel dynamic resources provisioning scheme based on QT and RL is proposed.
- (ii) Two concepts are introduced to evaluate the performance of various resources provisioning scheme.
- (iii) To demonstrate our method, we apply our method to the simulation and real resources provisioning for cloud computing platform.
- (iv) The experiment results demonstrate that our developed method can make accurate provisions at numerous arrival rates of users' job and avoid SLA conflicts.

The remainder of this paper is organized as follows. Section 2 reviews the related work of resources provisioning in cloud computing environment. Section 3 explores the construction of cloud computing platform and detailed analysis on the implementation of user job in proposed system model. Based on the cloud models, we introduce two new concepts: SSLA and UUTC, respectively. In Section 4, we design a resources allocation scheme based on reinforcement learning; then according to the shortage of basic Q learning, we propose an improved Q learning scheme to enhance the algorithm performance. The experiment results are presented in Section 5, and, finally, we reach the conclusions and skeleton of our future work in Section 6.

2. Related Work

Dynamic resource management in cloud computing environment refers to the process of dynamic optimization allocation, organization, coordination, and control of resources. It should not only support task scheduling in interorganizational or management domain, real-time resources monitoring, and job execution, but also maintain the self-management of the local sites, providing the corresponding Quality of Service (QoS) support. It is an advanced form of resource management and also the core component of resource management system in cloud computing environment, so as to shield the heterogeneity and complexity of the underlying resources, to manage the distributed massive resources in the cloud computing, to control the resources effectively, to

improve resource utilization, and to provide the reasonable distribution of resources operation for cloud computing, thus to balance the load.

Since the concept of cloud computing was proposed, resource scheduling, especially dynamic resources provisioning, has been one of the most important research components. Related works of resource provisioning are mainly from different perspectives to construct a cloud computing system model of queueing theory, aiming to attain universal results. However, being affected by factors such as the heterogeneity of the built platforms, the incompatible interface, and the disparity of the underlying physical resources, various research results demonstrate discrepancies, difficult to make a comparable analysis among each other.

In [2], dealing with the combined issues of power and performance management in cloud data centers, the authors proposed a dynamic resource management scheme by leveraging both of the techniques such as dynamic voltage/frequency scaling and server consolidation, thus to achieve energy efficiency and desired application-level performance. The novelty of the proposed scheme was its integration with timing analysis, queuing theory, integer programming, and control theory techniques. In [3], despite the varying event arrival rates, a queuing theory based approach was pursued to achieve specified response time target; by drawing the necessary computing resources from a cloud, a distinct query engine was modeled as an atomic unit to predict response times. Several similar units hosted on a single node were modeled as a multiple class M/G/1 queuing system and the response times were deemed to meet specified targets although being subject to varying event arrival rates over time. Correlation work was also extended to multimedia cloud and large web server clusters [4, 5]. In [4], concentrating on resource allocation problems in multimedia cloud, the authors employed optimization methods and queueing theory; theoretical analysis and computer simulation demonstrate the resource cost minimization problem and the response time minimization problem, respectively. In [6], the authors proposed a dynamic resource allocation scheme to resist distributed denial of service (DDoS) attacks against individual cloud customers. This paper was an early work that discussed mitigating DDoS attacks using resource allocation scheme for individual cloud customers. In [7, 8], the authors proposed an embedded Markov chain analytical model to estimate cloud performance. In [9], the authors focused on SLA-aware service deployment optimization problem, the designed E3-R which is a multiobjective genetic algorithm, to seek individuals and exhibit in cloud computing environments. In order to meet both requirements, E3-R employed two different fitness functions for different kinds of individuals. In [10], the authors focused on the bottleneck of network I/O and the aggregation on the packets delay and proposed a mechanism based on packet aggregation to achieve the best tradeoff between the throughput and packets delay. In [11], the authors investigated the elastic resource provisioning problem under the burstiness of incoming requests and energy consumption, employed the ON-OFF Markov chain and queueing theory to describe burstiness, and proposed a VM consolidation mechanism for each

PM to solve the problem. In [12], the authors focused on cloud backup optimization problem, employing two decision parameters and finite-source queueing theory to maintain the regulated service quality of the cloud platform.

Distinguished from prior works, we establish a model of cloud computing system to strengthen the learning as an optimization tool for dynamic resources provisioning in data center based on the queueing theory in this paper. To the best of the authors' knowledge, not many related papers have appeared in the literature concerning dynamic resources provisioning.

3. System Model

3.1. Cloud Computing Platform Framework. In this section, we will introduce the framework of the cloud computing platform used in this study, as depicted in Figure 1. The details of system organization structure and the functions of various parts are described as follows.

Users Interface (UI). The main function of users interface (UI) is to receive the user's requests, allocating them to the corresponding VM cluster on the basis of provisioning algorithm and then receiving the execution results and returning them to the user.

Users Job Queue (UJQ). Cloud computing platform includes two user-job queues. After they are submitted via the user interface to the cloud computing platform, the jobs enter queue 1 in turn and wait to be scheduled. When they are executed, they enter the user queue 2 in turn, waiting to be sent back to the end user by the transmitter.

Users Job Scheduling (UJS). By employing the job scheduling policy, the user's requests in queue 1 are scheduled to the corresponding VM clusters.

Virtual Machine Cluster (VMC). A plurality of VMs to perform the same operation type forms a VMC. Each VM of a VMC is especially designed for a certain type of operation to enable the highest operating efficiency. Each VMC is equipped with a Virtual Machine Cluster Agent (VMCA), responsible for the VM instance generation, management, and cancellation. For example, when a user job arrival rate increases, the VMs within the current VMC cannot meet QoS or SLA, so VMCA needs to increase the number of the VMs within the VMC to improve its throughput; otherwise, some of the VMs in the VMC need to be cancelled so as to reduce energy consumption.

VM. Responsible for specific job execution, VM removes a user job from the queue for execution, transmits the execution results to the platform interface, and extracts the next user request. Each VM is equipped with a Performance Monitor Agent (PMA) and a Resource Management Agent (RMA). PMA is responsible for the performance indicators to monitor the entire VM, including response time, throughput, and resource utility. RMA is responsible for the VM resource management, mainly including the dynamic scheduling of CPU, memory, bandwidth, and data center resources.

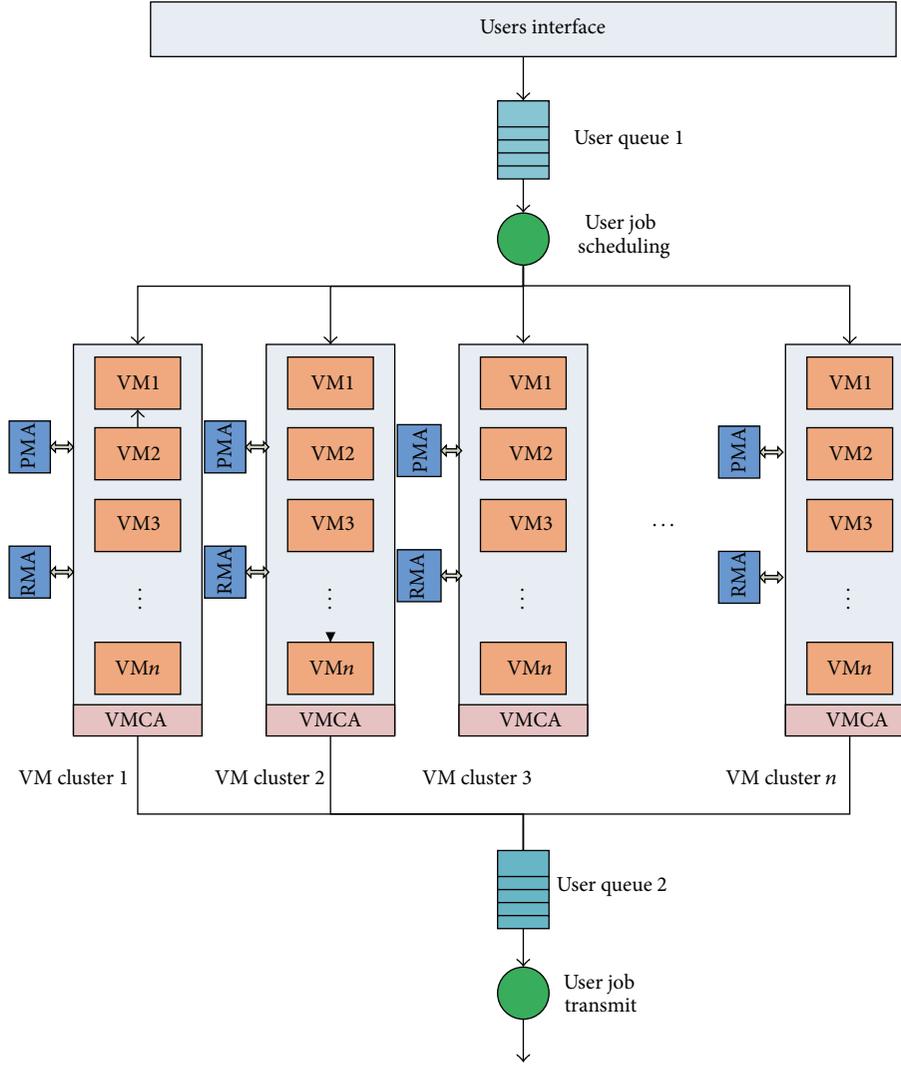


FIGURE 1: Architecture of cloud computing framework.

Users Job Transmit (UJT). The execution results of queue 2 are transmitted to the corresponding user in light of the transmission strategy.

3.2. Job Response Time (JRT). According to the system model shown in Figure 1 and the job execution phase in cloud computing environment, Job Response Time (JRT) hinges on Job Queueing Time (JQT), Job Execution Time (JET), and Job Transfer Time (JTT). In other words, JRT is made up of JQT, JET, and JTT.

In light of the classic queueing theory, given the fact that the job arrival rate of the computing platform is λ , the arrival rate of the j th VM in the cluster i is λ_{ij} , and its service rate is μ_{ij} . Thus, the allocated average queueing time JQT can be described as [13]

$$\text{JQT} = \frac{\rho_{ij}}{\mu_{ij}(1 - \rho_{ij})}, \quad (1)$$

where $\rho_{ij} = \lambda_{ij}/\mu_{ij}$ and the Probability Density Function (PDF) of JQT is

$$f_{\text{JQT}}(t) = (1 - \rho_{ij})\delta(t) + \mu_{ij}\rho_{ij}(1 - \rho_{ij})e^{-\mu_{ij}(1 - \rho_{ij})t} \quad (t \geq 0). \quad (2)$$

Similarly, the response time of JET and JTT [14, 15] is

$$\begin{aligned} \text{JET} &= \frac{1}{\mu_{ij}}, \\ \text{JTT} &= \frac{D_{ij}/B_{ij}}{1 - \lambda_{ij}D_{ij}/B_{ij}}, \end{aligned} \quad (3)$$

where D_{ij} is the result size of user job and B_{ij} is the provisioned bandwidth resources of the user job. And the PDF of JET and JTT is

$$f_{\text{JET}}(t) = \mu_{ij} e^{-\mu_{ij}t} \quad (t \geq 0), \quad (4)$$

$$f_{\text{JTT}}(t) = \frac{B_{ij}}{D_{ij}} e^{-(B_{ij}/D_{ij})t} \quad (t \geq 0), \quad (5)$$

respectively.

Thus the total response time in the VMC can be given as

$$T_{\text{tot}} = \text{JQT} + \text{JET} + \text{JTT}$$

$$= \frac{\rho_{ij}}{\mu_{ij}(1-\rho_{ij})} + \frac{1}{\mu_{ij}} + \frac{D_{ij}/B_{ij}}{1-\lambda_{ij}D_{ij}/B_{ij}}. \quad (6)$$

3.3. Segment SLA. The response time, which is the performance indicator of the cloud computing platform, is constrained by QoS or SLA. In this study, we divide SLA into the varying phases at which the user job is to be executed, so as to provision the resources accurately in the cloud computing environment. When the job is run, the resources can be provisioned according to the different stages by the resource provisioning scheme, thus to enable each phase to be constrained by SLA, as indicated in the following representation:

$$\begin{aligned} \text{JQT} &\leq \text{SLA}_{\text{JQT}}, \\ \text{JET} &\leq \text{SLA}_{\text{JET}}, \\ \text{JTT} &\leq \text{SLA}_{\text{JTT}}. \end{aligned} \quad (7)$$

As long as the job at each phase in the execution meets SLA constraints, the total response time is enabled to satisfy the global SLA constraints. Moreover, the introduction of segment SLA can improve the QoS of the cloud computing platform effectively. For instance, when a JQT violates SLA because of a resource shortage, I/O deadlock, or conflicts, a higher priority is given to the job execution in the upcoming phases, guaranteeing the resources for the job and reducing the corresponding time of JET and JTT, respectively, thus to ensure the overall SLA of the user operation to satisfy the QoS constraints. An example of medical image analysis application in cloud computing environment is shown in Figure 2.

3.4. Utility Unit Time Cost. Currently the commercial cloud computing platforms are mostly paid by being rented per hour. Take the well-known Amazon EC2 cloud computing platform for example, in which the price for standard on-demand instances is illustrated in Table 1.

As for any user job, the Utility Unit Time Cost (UUTC) can be defined as follows:

$$\text{UUTC} = \frac{\text{Total cost}}{T_{\text{tot}}}. \quad (8)$$

Physically, UUTC is the ratio of the operation cost and the actual execution time. It optimizes the constraints in

TABLE 1: Amazon EC2 pricing for standard on-demand instances.

Instance type	Linux (per hour)	Windows (per hour)
Small (default)	\$0.060	\$0.115
Medium	\$0.120	\$0.230
Large	\$0.240	\$0.460
Extra large	\$0.480	\$0.920

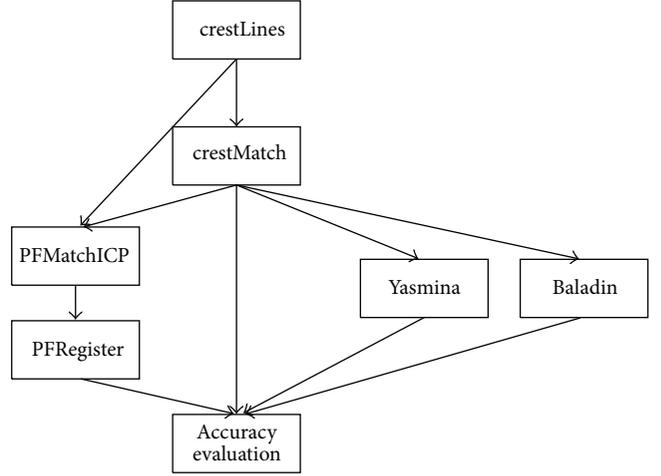


FIGURE 2: Medical image analysis application.

terms of resource utilization rate and improves the optimized function.

As for a user's job, the issue on resource provisioning optimization in cloud computing platform can be denoted as

$$\begin{aligned} &\text{Maximum}_{\{\text{user job}\}} \text{UUTC} \\ &\text{subject to} \quad \text{JQT} \leq \text{SLA}_{\text{JQT}} \\ &\quad \quad \quad \text{JET} \leq \text{SLA}_{\text{JET}} \\ &\quad \quad \quad \text{JTT} \leq \text{SLA}_{\text{JTT}}. \end{aligned} \quad (9)$$

As shown in (9), RAM makes a decision through attaining the performance index at every observant moment on the basis of PAM. Therefore, it is a sequential decision-making problem. Aiming to solve this problem, we propose a scheme in this study by employing reinforcement learning, which is described in detail in Section 4.

4. Resources Provisioning Mechanism

As stated in Section 3, the resource provisioning issue can be viewed as a sequential decision-making problem in the system model; therefore, it can be represented by MDP. Meanwhile, with little difference in the definition for the concepts like state space, action set, and reward function in the various RL-based schemes of resource provisioning in the cloud computing environment, we also define the relevant concepts as follows in this study.

4.1. Concepts Declared

4.1.1. State Space. A physical machine can virtualize a number of VMs, while a VM can only belong to one physical machine. The virtual ones are logically independent within the same physical machine, while they compete with each other in resource provisioning. Resources including VCPU, memory, and bandwidth, in each VM, are regarded as the state space in this study; accordingly, the state space for each VM is expressed in the form of a vector as VCPU, memory, and bandwidth. The value of every element is not beyond the upper bound of physical machine. Supposing a physical machine has four CPUs, 8 G memory, and 100 M bandwidth, an example of state space in VM is (1, 2, 2), which means the VM has 1 VCPU, 2 G memory, and 2 M bandwidth.

4.1.2. Action Space. As for the i th VM resource, a possible action space includes the increase, constancy, or decrease in resource, which can be identified by 1, 0, -1 , respectively, to indicate the corresponding action. Meanwhile, the increased or decreased resource of VCPU, memory, and bandwidth is set as a VCPU, 512 M memory, and 0.5 M bandwidth at each decision-making moment. Then for the i th VM, assuming that its state space is (1, 2, 2), the action at the decision moment can be expressed as (0, 1, -1), which means the number of VCPU remains unchanged, memory increases to 512 M, and the bandwidth reduces to 0.5 M. After the action is implemented, VM's state space is represented as a vector (1, 2.5, 1.5).

4.1.3. Immediate Reward. The immediate reward is used to reflect the correct running state and the efficiency of job scheduling. The three situations are considered by designing a reward function. (1) If the UUTC of current users job is bigger than the mean UUTC and satisfies SAL or QoS constraint, the reward is 1; (2) if the response time of users job violates SLA or QoS constraint, the reward is -1 ; (3) otherwise, the reward is 0.

4.2. Basic Reinforcement Learning Resources Allocation Scheme. As is depicted in the MDP, we employ the Q learning algorithm, a popular reinforcement learning algorithm, to solve the sequential decision problem described in (9). The pseudocode of the basic Q value learning algorithm is illustrated in Algorithm 1.

In order to evaluate the performance of the proposed resources provisioning scheme, we compare it to the utilization ratio provisioning scheme [16] used in the Amazon cloud computing platform. Due to the numerous experimental results, the ones listed here are only the results relevant to the VCPU resource provisioning, as shown Figure 3. In order to make our simulations more convincing, we use the practical parameters and pricing rates of Windows Azure. Windows Azure is a cloud platform developed by Microsoft, which provides on-demand computation and bandwidth resources for services through Microsoft data centers.

We assumed the increasing arrival rate of different jobs and the number of user jobs to be completed at each arrival rate to be the same in the experiments. As seen in Figure 3(a),

```

(1) Initialize Q value table
(2) Initialize state  $s_t$ 
(3) error = 0
(4) repeat
(5) for each state  $s$  do
(6)  $a_t = \text{get\_action}(s_t)$  using  $\epsilon$ -greedy policy
(7) for (step = 1; step < LIMIT; step++) do
(8) take action  $a_t$ , observe  $\mathbf{r}$  and  $S_{t+1}$ 
(9)  $Q_t = Q_t + \alpha * (r + \gamma * Q_{t+1} - Q_t)$ 
(10) error = MAX(error |  $Q_t - Q_{\text{previous-}t}$ )
(11)  $s_t = s_{t+1}$ ,  $a_{t+1} = \text{get\_action}(s_t)$ ,  $a_t = a_{t+1}$ 
(12) end for
(13) end for
(14) Until error <  $\theta$ 

```

ALGORITHM 1: Q value learning algorithm.

based on the different demands for specific resources of different jobs, when the job arrival rate increases, the utilization strategy can make a real-time provision with VCPU resources in virtual machines efficiently and meanwhile avoid the SLA conflict as shown in Figure 3(b). However, under the same experimental conditions, basic Q learning scheme adjusts VCPU resources provisioning frequently, which may result from a wrong performance (e.g., when it is the time to increase VCPU resources, they are not raised but reduced instead) due to the exploration-exploitation mechanism, eventually leading to the frequent resource provisioning and SLA conflicts.

From the comparison results in Figure 3, we can draw some conclusions of basic reinforcement learning used in cloud computing resources provisioning.

- (i) Slow convergence.
- (ii) Ineffective adaptability to the changeable arrival rates; thus the policy needs updating, and sometimes not getting the converged solution.
- (iii) The suboptimal solution is often acquired instead of the optimal solution, even at a fixed job arrival rate.

The above disadvantages of reinforcement learning, especially the slow convergence rate and ineffective adaptive ability, severely limit its practical application in the cloud resource provisioning.

4.3. Improved Reinforcement Learning Resources Allocation Scheme. We design an improved scheme to conquer each weakness of basic reinforcement learning scheme, which is focused on the following aspects.

4.3.1. Offline Learning. With the simulations to the real data sets, the offline training based on the basic Q learning algorithm is employed to acquire the varied job arrival rates, the numbers of the VMs, and the Q value table, the approximate function relation between resource provisioning. During offline learning process, multiple instances can run parallelly in order to learn by dividing the state space, so

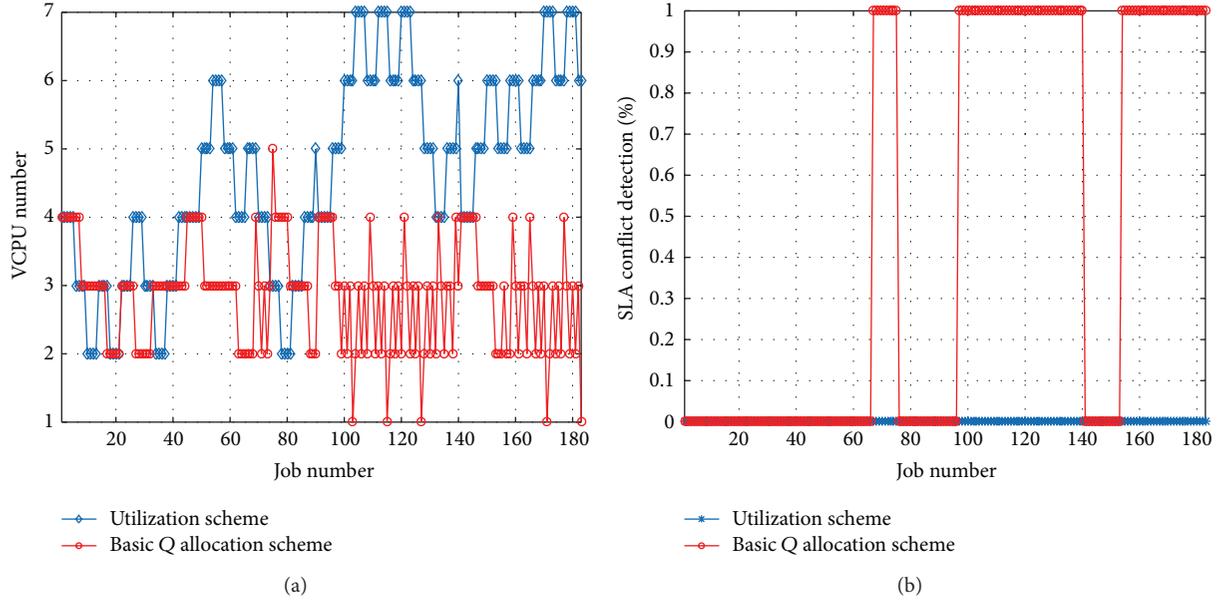


FIGURE 3: Comparison results of VCPU resource provisioning and SLA conflict detection between the basic Q learning scheme and the utilization scheme. (a) Comparison of VCPU number between the basic Q learning scheme and the utilization scheme under various job numbers. (b) SLA conflict detection between the basic Q learning scheme and the utilization scheme under various job numbers.

```

(1) Divide State Space
(2) Repeat
(3)   for each state space partition do
(4)     set upper and low bound of CPU, memory and
         bandwidth
(5)     Obtain running state of cloud computing platform
(6)     Obtain performance index
(7)     for each resources do
(8)       using Algorithm 1
(9)     end for
(10)  end for
(11) Update Q value table
    
```

ALGORITHM 2: Offline reinforcement learning algorithm.

```

(1) Obtain running state of cloud computing platform
(2) Look up Q value table, configure VM resources
(3) Obtain performance index
(4) use belief library
(5) set upper and low bound of VCPU, memory and
         bandwidth
(6)   action space compact
(7)   for each resources do
(8)     using Algorithm 1
(9)   end for
(10) Update Q value table
    
```

ALGORITHM 3: Online reinforcement learning algorithm.

that the acquired relation can be approximately formulated with the regression function. The pseudocode of the Q value offline learning algorithm is illustrated in Algorithm 2.

In spite of the fact that the Q value table resulting from offline learning is rather huge, data index can be used to accelerate the search speed, thus improving search efficiency.

4.3.2. Belief Libraries and Simple Action Space. Set up a VCPU belief library, whose rules are similar to the ways by which the belief library was built in [17]. Based on the established belief library, the action space can be simplified correspondingly. For example, when provisioning resources, if the VCPU utilization approaches the lower bound, the action increase to raise up VCPU resources in action space should be removed; otherwise the decrease to reduce the VCPU resources should be removed. The establishment of

belief library and the simplification of action space avoid the blindness of Q learning action selection effectively and thus improve the convergence speed.

4.3.3. Online Learning. Offline learning environment can only simulate part of the real operating environment. The acquired function is a viable strategy for resource provisioning only when a job arrival rate meets the SLA constraint; even so, it may not be a suboptimal strategy. However, this initial strategy sets up an upper boundary for resource provisioning, and under the guidance of it we can learn online based on this initial strategy to further improve the resource utilization.

The acquired real-time resource utilization rate by using PMA guides RMA learning. The pseudocode of the Q-value online learning algorithm is illustrated in Algorithm 3.

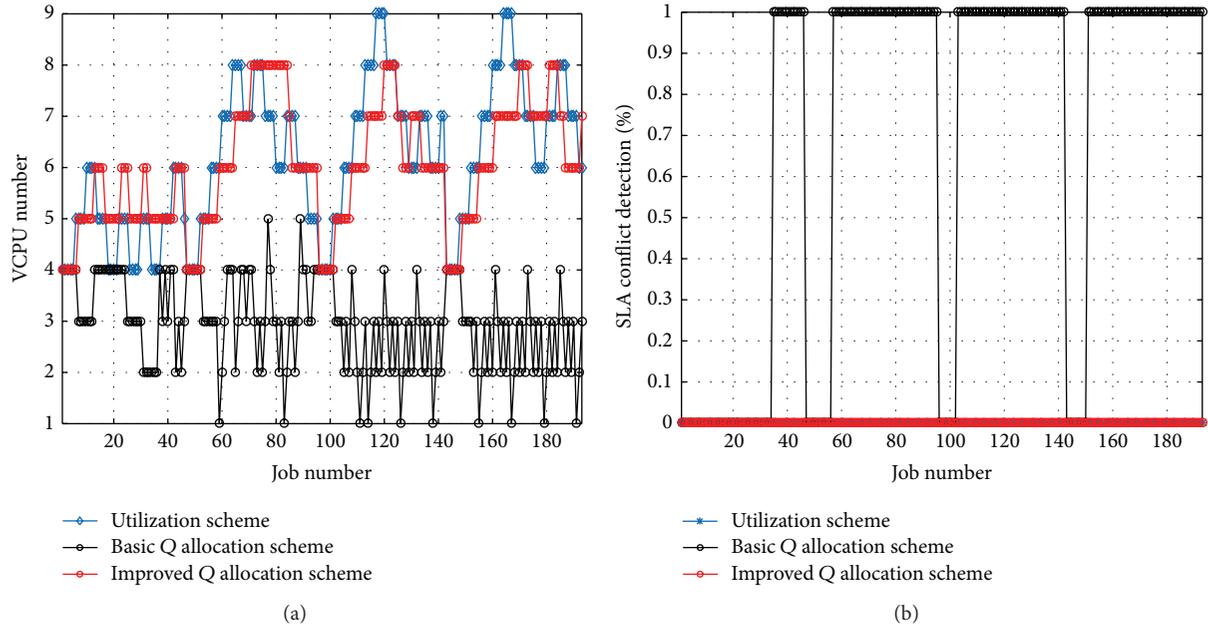


FIGURE 4: Comparison results of VCPU resource provisioning and SLA conflict detection between the improved Q learning scheme, the basic Q learning scheme, and the utilization scheme. (a) Comparison of VCPU number between the improved Q learning scheme, the basic Q learning scheme, and the utilization scheme under various job numbers. (b) SLA conflict detection between the improved Q learning scheme, the basic Q learning scheme, and the utilization scheme under various job numbers.

5. Experience Results

To evaluate the efficiency of our approach, implementations have been performed on the simulation and real cloud computing environment, respectively.

5.1. Simulation Experiment Results. Using MATLAB R2012a by MathWorks, Inc., we have developed a discrete event simulator of the cloud server form to validate the efficiency of resource provisioning solution and have compared the performance information among the alternative schemes in our simulations.

We evaluated the performance of the improved Q learning strategy in Figure 4 and compared it with the utilization rate scheme and the basic Q learning scheme under the same experimental conditions in Section 4.2. From the experimental results in Figure 4, on one hand, we can see that basic Q learning scheme is still likely to make wrong decisions and thus resulted in the SLA conflicts and frequent VCPU resource provisions. On the other hand, the improved Q learning scheme can provision the VCPU resources in real time based on the changes in arrival rates; more importantly, apart from avoiding SLA conflicts, the number of VCPU resources used by it appears less than that used by utilization rate scheme through most of the time. In other words, avoiding the SLA conflict, the improved Q learning method improves the utilization rate of resources.

Next, we compare the improved Q learning scheme with the prevalent resource provisioning schemes: (1) the proposed resources provisioning scheme, denoted by the improved Q learning scheme, in which the cloud computing resources were optimally scheduled to the VMs by improved

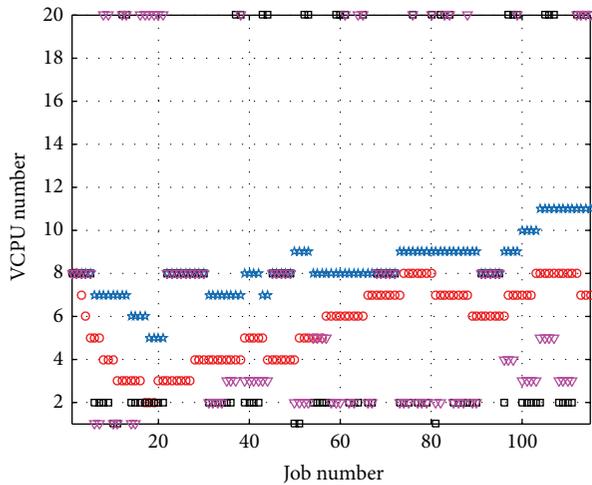
Q learning algorithm, (2) the utilization resources provisioning scheme, denoted by utilization scheme, in which the cloud computing resources were optimally scheduled to the VMs by resources utilization, (3) the genetic algorithm resources provisioning scheme, denoted by GA scheme, in which the cloud computing resources were optimally scheduled to the VMs by genetic algorithm, (4) the nonlinear programming resources provisioning scheme, denoted by nonlinear programming scheme, in which the cloud computing resources were optimally scheduled to the VMs by nonlinear programming.

As seen from the experimental results in Figure 5(a), the number of VCPU resources in the GA scheme and the nonlinear programming scheme may cause frequent changes due to the objective function optimization. Nevertheless, the improved Q learning scheme and the utilization scheme demonstrate the same performances as those in Figure 4(a), respectively. With similar pricing settings to Table 1, Figure 5(b) shows that the total cost of the improved learning Q strategy designed in this paper is lower than that of the compared schemes.

We ran the simulation program 1000 times at various arrival rates, and the average results of various performance indicators are shown in Table 2. As shown in the results, the UUTC at different arrival rates is better than that of the contrast algorithm. In light of the definition of UUTC, the numerator aims to maximize the current operation costs, while the denominator aims to minimize the current execution time. The ratio between the two indicates the utilization rate of cloud computing resources per unit time. Accordingly, the greater the value is, the higher the corresponding utilization rate of the cloud computing resource is.

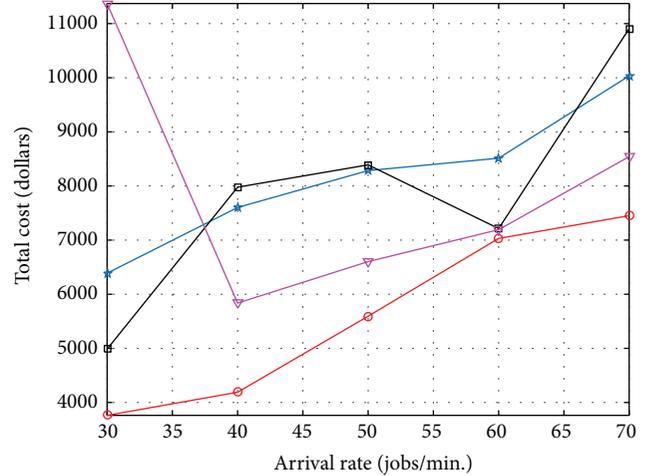
TABLE 2: Comparison results of various resources allocation schemes.

	Arrival rate (jobs per min.)	Total running time (min.)	Total cost (dollars)	Mean VCPU numbers (numbers per arrival rate)	UUTC (dollars per min.)
RPSUT	30	364.61	7270	7.36	2.70
RPSGA		381.68	3930	7.36	1.39
RPSNP		368.36	4353	7.36	1.60
Proposed scheme		368.36	5204	4.76	2.96
RPSUT	40	247.20	7433	8.20	3.66
RPSGA		273.12	5788	8.20	2.58
RPSNP		278.21	5349	8.20	2.34
Proposed scheme		250.30	3727	3.87	3.84
RPSUT	50	216.14	8688	8.18	4.91
RPSGA		244.06	8100	8.18	4.05
RPSNP		232.41	6376	8.18	3.35
Proposed scheme		218.63	5448	4.45	5.59
RPSUT	60	148.39	9540	8.78	7.32
RPSGA		172.75	8718	8.78	5.74
RPSNP		176.30	7569	8.78	4.88
Proposed scheme		149.85	6578	4.43	9.90
RPSUT	70	147.48	9611	9.96	6.54
RPSGA		175.78	10025	9.96	5.72
RPSNP		179.37	9878	9.96	5.529
Proposed scheme		147.87	7820	7.69	6.87



- ★ Utilization scheme
- Improved Q learning scheme
- GA scheme
- ▽ Nonlinear programming scheme

(a)



- ★ Utilization scheme
- Improved Q learning scheme
- GA scheme
- ▽ Nonlinear programming scheme

(b)

FIGURE 5: Comparison results of VCPU resource provisioning and total cost between the improved Q learning scheme, GA provisioning scheme, the utilization scheme, and nonlinear programming provisioning scheme. (a) Comparison of VCPU number between the improved Q learning scheme, GA provisioning scheme, the utilization scheme, and nonlinear programming provisioning scheme under various job numbers. (b) Comparison of total cost between the improved Q learning scheme, GA provisioning scheme, the utilization scheme, and nonlinear programming provisioning scheme under various job arrival rates.

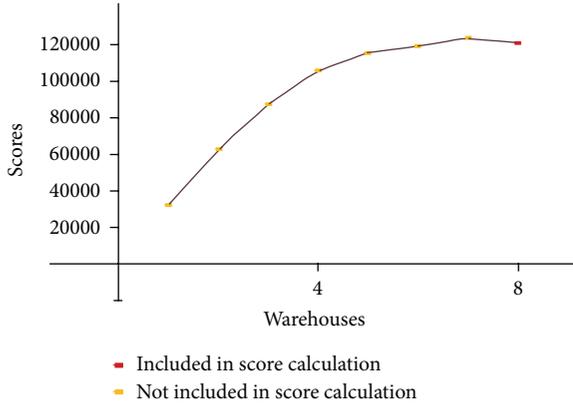


FIGURE 6: SPECjbb2005 throughput in default settings.

The utilization scheme in the figure executes the jobs with the real-time resource provisioning based on the utilization. It is demand-sensitive to the resources and requires frequent application and release of the resources. Based on the job arrival rates, the improved Q provisioning scheme can not only configure the system resources in accordance with Q value table optimization scheme, but also learn the optimal provisioning further on the real-time resource utilization. The improved Q allocation scheme can configure the cloud computing resources based on the arrival rates of the user's job and learn the optimal allocation further based on the optimization scheme. This scheme facilitates the adaptability to the user's job arrival rates, while the basic Q learning scheme needs some exploration before it gets steady, or even worse, it cannot acquire the stable distribution. From the experiment results in Figure 5, a similar conclusion is drawn, which further proves the effectiveness of this provisioning scheme.

5.2. Real Cloud Computing Platform Experiments Results. The machines used in the experiments consist of virtual servers, client, and compute machines. The physical machines for virtual hosting are Lenovo ThinkServer RD630 with 8 CPU and 16 GB memory.

Xen was used as our virtualization environment and the SPECjbb2005 was selected as the workloads running within the VMs. SPECjbb2005, a Java program, is SPEC's benchmark. By simulating a three-tier client/server system with stress on the middle tier, the benchmark measures the performance of server side Java. It implements the Java virtual machine (JVM), just-in-time (JIT) compiler, garbage collection, threads, and part of the operating system. Implemented in a more object-oriented manner, SPECjbb2005 presents new features such as XML processing and BigDecimal computations and furnishes a new enhanced workload to mirror how real application is designed, thus making the benchmark a more realistic reflection of contemporary application [18].

The SPECjbb2005 throughput under various pressures in the default setting can be seen in Figure 6. The default setting in SPECjbb2005 refers to the whole resources in the platform used by the system under any pressure, such as CPU and memory. In the experiments processing, warehouses were

TABLE 3: Comparison results of average throughput under various warehouses (Bops).

Warehouse	Max	Utilization scheme	Proposed scheme
1	32103.43	32847.88	33767.96
2	63252.47	59510.64	59951.13
3	87267.56	82380.99	83919.59
4	105719.64	104081.37	99212.12
5	115648.76	117200.12	116671.41
6	119057.39	121530.59	122071.97
7	123759.19	126187.14	124289.81
8	120844.70	122457.18	122758.30

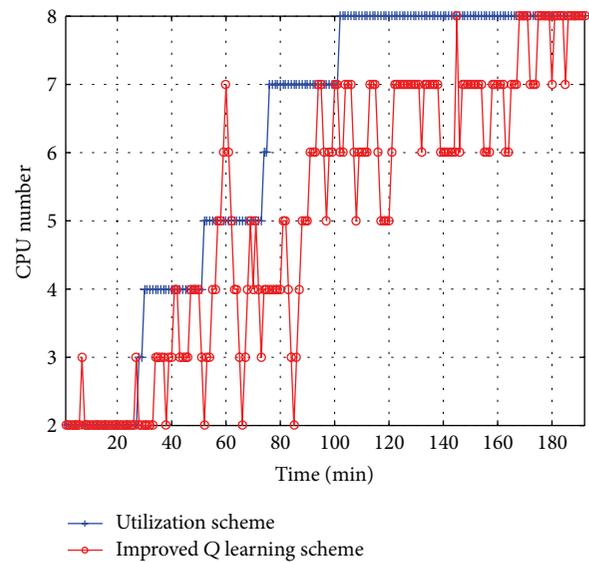


FIGURE 7: Comparison of CPU number between the improved Q learning scheme and the utilization scheme under various warehouses.

dynamically increasing from 1 to 8 every fixed time interval. In our experiments, the time interval was set as 20 minutes. Another fact discovered in the experiments at the same time was that the benchmark was insensitive to memory. Hence, only the results of CPU resources were listed in the next part.

We ran the benchmark 10 times at various warehouses, and the comparison results of average throughput under various warehouses were shown in Table 3. From the results we can see the throughput of the utilization scheme and the proposed scheme was close to the reference Max under various warehouses. It proves these two resource provisioning schemes can provision CPU resources self-adaptively and achieve the maximum throughput according to different warehouses.

The real cloud computing environment experiment results as shown in Figure 7 also demonstrated that the proposed improved Q provisioning scheme outperforms the utilization scheme in that it achieved resources utilization rate under various warehouses and CPU resources constraint.

6. Conclusions and Future Work

In this study, we provide an insightful view about the resource provisioning optimization problem in cloud computing platform; then we propose a novel resources provisioning scheme based on the reinforcement learning and queueing theory. With the introduction of the concepts of SSLA and UUTC, we view the resource provisioning issue in cloud computing as a sequential decision problem, and then we design a novel optimization object function and employ reinforcement learning to solve it. Experiment results not only demonstrate the effectiveness of the proposed scheme, but also prove to outperform the prevalent resource provisioning methods in terms of SLA collision avoidance and user costs. Also, some conclusions can be drawn as below for using Q learning algorithm.

- (i) The Q learning algorithm outperforms the comparative method in three aspects: the number of CPUs in use, SLA conflicts, and costs.
- (ii) Belief library is employed to simplify the action space, thus to enhance the performance of the Q learning algorithm.
- (iii) State space is the number of VCPU/CPU in experiment, with a smaller space, faster convergence, and stronger adaptability.
- (iv) Q learning provisions in two cases, either when the SLA violation occurs for three consecutive times, or when the utilization rate is too high or too low.

In the future, we plan to extend our schemes for the dynamic resource provisioning to get the minimal response time, thus providing highly satisfactory services and avoiding SLA violations. Looking into the cloud entities and considering the details of cloud computing platform, such as VMs failures, VMs migration, costs of communication, and burst arrivals of requests, VMs cluster for different kinds of requests will be other dimensions of extension.

Abbreviations

UI:	Users interface
UJQ:	Users job queue
UJS:	Users Job Scheduling
UJT:	Users Job Transmit
VM:	Virtual Machine
VMC:	Virtual Machine Cluster
VMCA:	Virtual Machine Cluster Agent
PMA:	Performance Monitor Agent
RMA:	Resource Management Agent
JRT:	Job Response Time
JQT:	Job Queueing Time
JET:	Job Execution Time
JTT:	Job Transfer Time
QoS:	Quality of Service
SLA:	Service Level Agreement

SSLA:	Segmentation SLA
UUTC:	Utility Unit Time Cost
MDP:	Markov decision process
PDF:	Probability Density Function
ML:	Machine learning
RL:	Reinforcement learning
λ :	Job arrival rate of cloud computing platform
λ_{ij} :	Job arrival rate of j th VM in i th VMC
μ_{ij} :	Job server rate of j th VM in i th VMC
D_{ij} :	The executed result size of user job
B_{ij} :	The allocation bandwidth resources of the user job.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The work presented in this paper was supported by National Natural Science Foundation of China (nos. 61272382 and 61402183).

References

- [1] H.-S. Wu, C.-J. Wang, and J.-Y. Xie, "TeraScaler ELB-an algorithm of prediction-based elastic load balancing resource management in cloud computing," in *Proceedings of the 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA '13)*, pp. 649–654, IEEE, Barcelona, Spain, March 2013.
- [2] Y. Gao, H. Guan, Z. W. Qi, T. Song, F. Huan, and L. Liu, "Service level agreement based energy-efficient resource management in cloud data centers," *Computers and Electrical Engineering*, vol. 40, pp. 1621–1633, 2013.
- [3] V. Suresh, P. Ezhilchelvan, and P. Watson, "Scalable and responsive event processing in the cloud," *Philosophical Transactions of the Royal Society A*, vol. 371, no. 1983, Article ID 20120095, 2013.
- [4] X. Nan, Y. He, and L. Guan, "Queueing model based resource optimization for multimedia cloud," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 928–942, 2014.
- [5] H. Nguyen, T. N. Minh, and N. Thoai, "Tool-driven strategies for resource provisioning of single-tier web applications in clouds," in *Proceedings of the 5th International Conference on Ubiquitous and Future Networks (ICUFN '13)*, pp. 795–799, July 2013.
- [6] S. Yu, Y. Tian, S. Guo, and D. O. Wu, "Can we beat DDoS attacks in clouds?" *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2245–2254, 2014.
- [7] K. Salah and R. Boutaba, "Estimating service response time for elastic cloud applications," in *Proceedings of the 1st IEEE International Conference on Cloud Networking (Cloud Net '12)*, pp. 12–16, Jussieu, Paris, November 2012.
- [8] H. Khazaei, J. Misic, and V. B. Misic, "Performance analysis of cloud computing centers using M/G/m/m+r queueing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 5, pp. 936–943, 2012.

- [9] H. Wada, J. Suzuki, Y. Yamano, and K. Oba, "Evolutionary deployment optimization for service-oriented clouds," *Software: Practice and Experience*, vol. 41, no. 5, pp. 469–493, 2011.
- [10] M. Bourguiba, K. Haddadou, I. E. Korbi, and G. Pujolle, "Improving network I/O virtualization for cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 673–681, 2014.
- [11] Z. Luo and Z. Qian, "Burstiness-aware server consolidation via queuing theory approach in a computing cloud," in *Proceedings of the 27th IEEE International Symposium on Parallel & Distributed Processing (IPDPS '13)*, pp. 332–341, IEEE, Cambridge, Mass, USA, May 2013.
- [12] F.-C. Jiang, C.-T. Yang, C.-H. Hsu, and Y.-J. Chiang, "Optimization technique on logistic economy for cloud computing using finite-source queuing systems," in *Proceedings of the 4th IEEE International Conference on Cloud Computing Technology and Science (CloudCom '12)*, pp. 827–832, December 2012.
- [13] G. Grimmett and D. Stirzaker, *Probability and Random Processes*, Oxford University Press, 3rd edition, 2010.
- [14] Y. Wu, C. Wu, B. Li, X. Qiu, and F. C. M. Lau, "CloudMedia: when cloud on demand meets video on demand," in *Proceedings of the 31st International Conference on Distributed Computing Systems (ICDCS '11)*, pp. 268–277, IEEE, Minneapolis, Minn, USA, July 2011.
- [15] J. Zheng and E. Regentova, "Qos-based dynamic channel allocation for GSM/GPRS networks," in *Network and Parallel Computing*, vol. 3779 of *Lecture Notes in Computer Science*, pp. 285–294, Springer, Berlin, Germany, 2005.
- [16] A. Wolke and G. Meixner, "Twospot: a cloud platform for scaling out web applications dynamically," in *Proceeding of 3rd European Conference on Service Wave*, pp. 13–24, Ghent, Belgium, 2010.
- [17] X. Bu, J. Rao, and C.-Z. Xu, "A reinforcement learning approach to online web systems auto-configuration," in *Proceedings of the 29th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS '09)*, pp. 2–11, Montreal, Canada, June 2009.
- [18] <http://www.spec.org/jbb2005/>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

