

Research Article

Adaptive Computing Resource Allocation for Mobile Cloud Computing

Hongbin Liang,^{1,2} Tianyi Xing,³ Lin X. Cai,⁴ Dijiang Huang,³ Daiyuan Peng,⁵ and Yan Liu⁶

¹ State Key Laboratory of Information Security, Institute of Information Engineering, The Chinese Academy of Sciences, Beijing 100093, China

² School of Transportation and Logistics, Southwest Jiaotong University, Chengdu, Sichuan 610031, China

³ Arizona State University, 699 S Mill Avenue, Suite 464, Tempe, AZ 85281, USA

⁴ Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada N2L 3G1

⁵ School of Information Science and Technology, Southwest Jiaotong University, Chengdu, Sichuan 610031, China

⁶ School of Software and Microelectronics, Peking University, Beijing 102600, China

Correspondence should be addressed to Yan Liu; ly@ss.pku.edu.cn

Received 6 January 2013; Accepted 24 February 2013

Academic Editor: Jianwei Niu

Copyright © 2013 Hongbin Liang 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.

Mobile cloud computing (MCC) enables mobile devices to outsource their computing, storage and other tasks onto the cloud to achieve more capacities and higher performance. One of the most critical research issues is how the cloud can efficiently handle the possible overwhelming requests from mobile users when the cloud resource is limited. In this paper, a novel MCC adaptive resource allocation model is proposed to achieve the optimal resource allocation in terms of the maximal overall system reward by considering both cloud and mobile devices. To achieve this goal, we model the adaptive resource allocation as a semi-Markov decision process (SMDP) to capture the dynamic arrivals and departures of resource requests. Extensive simulations are conducted to demonstrate that our proposed model can achieve higher system reward and lower service blocking probability compared to traditional approaches based on greedy resource allocation algorithm. Performance comparisons with various MCC resource allocation schemes are also provided.

1. Introduction

Cloud computing is a new computing service model with characteristics such as resource on demand, pay as you go, and utility computing [1]. It provides new computing models for both service providers and individual customers, which can be broadly classified into infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). Furthermore, smart phones are expected to overtake PCs and become the most common web access entities worldwide by 2013 as predicted by Gartner [2]. Since mobile devices (MDs) have more advantages such as mobility, flexibility, and sensing capabilities over fixed terminals, integrating mobile computing and cloud computing techniques is a natural and predictable approach to build new mobile applications, which has attracted a lot of attention in both academia and industry

community. As a result, a new research field, called mobile cloud computing (MCC), is emerging.

In [3], Huang et al. presented a new MCC infrastructure, called MobiCloud, where dedicated virtual machines (VMs) are assigned to mobile users to improve the security and privacy capability. In such an MCC environment, the system computational resources, such as CPU, storage, and memory, are partitioned into several service provisioning domains based on the cluster geographical distribution. Each domain consists of multiple VMs, and each VM handles parts of cloud computing resource (i.e., CPU, storage and memory, etc.). When the MCC service provisioning domain receives a service request from a mobile device, it needs to make a decision on (1) whether to accept the request; and (2) how much Cloud resources should be allocated if the request is accepted. Although the Cloud resource can be considered

as unlimited compared with the computing resource in a single mobile device, in practice, a geographically distributed cloud system usually contains limited resource at a local service provisioning domain. When all the Cloud resources are occupied within the local service provisioning domain, the service request from mobile device will be rejected (or migrated to a nonlocal service provisioning domain) due to the resource unavailability. The rejection of a service request not only degrades the user satisfaction level (i.e., resulting in a long service delay due to the nonlocal service provisioning or service migration to other remote domain), but also reduces the system reward which is usually defined as a metric that includes the system net income and cost.

In general, the Cloud income increases with the number of the accepted services. However, it is definitely not true that cloud service provider (CSP) would like to accept service requests as many as possible, since more accepted services occupy more cloud resources, and more likely a new request will be rejected when the network resource is limited, which degrades the QoS level of users. The rewards of the most existing Cloud resource allocation methods only consider the income on behalf of the CSP. To obtain a comprehensive system reward of MCC, the customer QoS and user satisfaction level should be taken into account in the system reward as well. Therefore, our research goal is to address the following questions: how to obtain the maximal overall system rewards by taking into account from both the service provider side and the customer side while satisfying a certain QoS level.

In this paper, we present an adaptive MCC resource allocation model based on semi-Markov decision process (SMDP) to achieve the objective mentioned above. Our proposed MCC model considers not only the incomes of accepting services, but also the cost resulted from VM occupation in the Cloud. Moreover, other factors including service precessing time of both Cloud and MD battery consumption of mobile device are also taken into account. Thus, the overall economic gain is determined by a comprehensive approach which considers all the factors mentioned above.

The contributions and essence of this proposed model are listed as follows.

- (i) Semi-Markov decision process (SMDP) is applied to derive the optimal resource allocation policy for MCC.
- (ii) The proposed model allows adaptive resource allocations, that is, multiple Cloud resources (i.e., the number of VMs) can be allocated to a service request based on the available Cloud resource in the service domain in order to maximize the resource utilization and enhance the user experience.
- (iii) The maximal system rewards of Cloud can be achieved by using the proposed model and by taking into the considerations the expenses and incomes of both Cloud and mobile devices.

The rest of this paper is organized as follows. We present the related work in Section 2. In Section 3, the basic system model is described. The semi-Markov decision process model

for MCC system is presented in Section 4. Based on our proposed model, we analyze the probabilities for each adaptive allocation scheme and rejection probability in Section 5. We evaluate the performance of the proposed economic model in Section 6 and conclude this paper and discuss the future work in Section 7.

2. Related Work

Recent research work for Cloud computing has shifted its focus from the Cloud for fixed user to Cloud for mobile devices [4], which enables a new model of running applications between resource-constrained devices and Internet-based Cloud. Moreover, resource-constrained mobile devices can outsource computation/communication/storage intensive tasks onto the Cloud. CloneCloud [5] focuses on execution augmentation with less consideration on user preference or device status. Elastic applications for mobile devices via Cloud computing were studied in [6]. In [3], Huang et al. presented an MCC model that allows the mobile device related operations residing either on mobile devices or dedicated VMs in the Cloud. [7] proposes a way using traffic-aware virtual machine (VM) placement to improve the network scalability by optimizing the placement of VMs on host machines.

Although resource management in wireless networks has been extensively studied [8–10], there are few previous works focusing on resource management of Cloud computing and especially mobile cloud computing. In [11], an economic mobile cloud computing model is presented to decide how to manage the computing tasks with a given configuration of the Cloud system. That is, the computing tasks can be migrated between the mobile devices and the Cloud servers. A game theory-based resource allocation model to allocate the Cloud resources according to users' QoS requirements is proposed in [12]. In the past few years, some research work focused on application of specific resource management in Cloud computing using virtual machines or end servers in data center. In [13], authors propose a new operating system which enables resource-aware programming while permitting high-level reusable resource management policies for context-aware applications in Cloud computing. Lorincz et al. [14] address the problem of resource management in semantic event processing applications in Cloud computing. Tesauro et al. [15] propose a reinforcement learning based management system for dynamic allocation of servers trying to maximize the profit of the host data center in Cloud computing. In [16], Boloor et al. propose a generic request allocation and scheduling scheme to achieve desired percentile service level agreements (SLA) goals of consumers and to increase the profits to the cloud provider.

The works discussed above target to achieve a higher Cloud system profit and/or to meet a better service level agreement (SLA). However, they model the problem from service provider's perspective without considering the costs and profits of mobile devices. Therefore, the overall system rewards derived in previous works are sufficient. Generally, a Cloud-based application can be assigned with

multiple resources in terms of VMs (can be in different domains/clusters) to obtain more computation/storage and other capacities. However, to our best knowledge, in the previous literature, none of them addresses the following emerging research problems: (1) how to construct a reward model of MCC system for resources allocation purpose by considering the rewards from both Cloud system and mobile users; (2) how to allocate system resources to service requests to maximize the user satisfaction level of mobile users while obtaining the maximal overall system and user rewards under a given QoS level.

3. System Model

A major benefit of MCC over the traditional client-server mode is that MDs can have more capabilities and better performance (i.e., less processing time, energy saving, etc.) when they outsource their tasks onto the Cloud. The outsourcing procedure can be implemented by using weblets (application components) to link the services between the Cloud and the mobile devices. A weblet can be platform independent such as using Java or .Net bytecode or Python script or platform dependent, using a native code. Some research work [5] focuses on the algorithm to decide whether to offload the weblet from MD to the Cloud (i.e., run on one or more virtual nodes offered by an IaaS provider) or run the weblet on the MD itself. In this way, a mobile device can dynamically expand its capabilities, including computation power, storage capacity, and network bandwidth, by offloading an elastic application service to the Cloud. The choice made by mobile device on whether to offload the task onto the Cloud can refer to the mobile device's status such as CPU processing capability, battery power level, and network connection quality and security. In this paper, the service scenario of the proposed model is the task offloading from MD onto the Cloud. Also, the task offloading procedure can be done in a way that MD sends a service request to the Cloud firstly, then the task is further offloaded to the Cloud once the service request is accepted by the Cloud.

As shown in Figure 1, a VM is responsible for managing the weblet's loading, unloading, and processing in the mobile Cloud. Each VM has the capacity to hold one weblet at a time for handling migrated weblet request, and two types of service requests are defined to be handled by a VM: (i) *paid*: a paid weblet service request is sent to the service provisioning domain from a mobile device; (ii) *free*: a free weblet service request is sent to the service provisioning domain from a mobile device. Figure 1 demonstrates the relationship between the *paid/free* service requests and the VMs of the service provisioning domain.

In this paper, the MCC service architecture is based on the MobiCloud framework presented in [3], in which a VM can handle a portion of Cloud system resources (CPU, memory and storage, etc.) that can satisfy the minimal resource requirement to process an application offloading service in the MCC system. Within the local MobiCloud service provisioning domain, the resource capacity, in terms of the number of VMs, is limited. Thus, if the demands of the arriving service requests exceed the number of available VM

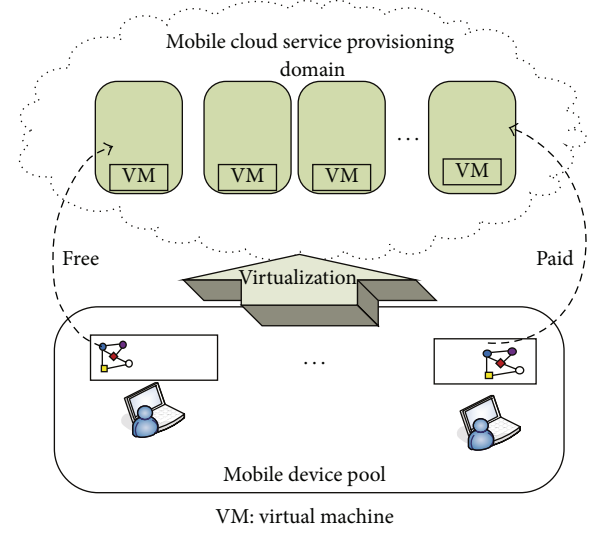


FIGURE 1: Reference model of mobile cloud computing.

resources in a certain service domain, the following service requests will be rejected (or migrated to a remote service provisioning domain). On the other hand, if the demands of the arriving service requests are lower than the number of the available VMs, more VMs can be assigned to one service request to maximally utilize the Cloud resource and achieve a better performance and QoS. Our analytical model is based on a single local service domain. The analysis of local service migrations to remote service domains is regarded as the future study.

3.1. System Description. An MCC system mainly consists of two entities, VM and physical MD. A VM is the minimum set of resources that can be allocated to an MD upon receiving its service request. Since an MD is a wireless node with limited computing capability and energy supply, it can outsource its mobile codes (i.e., weblet) of an application service to the Cloud. Then, the Cloud will decide a number of VMs to be allocated to the arriving service request if the decision for the service request made by the Cloud is accepted.

In this paper, we consider a service provisioning domain with K VMs. The maximum number of VMs that can be allocated to a Cloud service is c VMs (we denote as c allocation scheme), where $c \in \{1, 2, \dots, C\}$, $C \leq K$. Generally, the duration for running a mobile application service in the Cloud depends on the number of VMs allocated to that service. The relationship between the processing time of an application service and the number of allocated VMs in the Cloud can be expressed as a function denoted as $\xi(c)$. Assume that the time to process an application service by using one VM in a service provisioning domain is θ_s , therefore the time to handle the service is $\xi(c)\theta_s$ if c VMs are allocated to that service. The higher computing speed for an application service in a service provisioning domain means the higher user satisfaction level, which is the major part of the whole system reward of the Cloud. Thus, in order to improve the whole system reward of a service provisioning domain by

increasing the user satisfaction level, the traditional greedy algorithm [17] always decides to allocate maximal VMs to the service. But on the other hand, if the Cloud computing resources (denoted by the number of VM) allocated to the current service by the service provisioning domain are too high, then the following several arrival service requests may be rejected by the service provisioning domain because of insufficient available Cloud computing resources, which decrease the user satisfaction level. As a result, the system rewards of that MCC service provisioning domain degrade as well.

It can be more complicated when we consider both the rewards and costs of mobile devices. Cost involved in the MD side should not be neglected, which means that the whole system reward should consider not only the rewards of the mobile Cloud itself, but also the incomes and the costs of MD, such as the saved battery energy if the service is processed in the mobile Cloud and the expense of the battery energy and the processing time of MD if the application service is processed on the MD locally.

To model this complex dynamic MCC resource allocation process, without loss of generality, we assume that the arrival rates of both *paid* and *free* service requests follow Poisson distributions with mean rate of λ_p and λ_f , respectively. The life time of services follows exponential distributions. The mean holding time of a service which is allocated only one VM in the service provisioning domain is $1/\mu$. Thus, the holding time of the service allocated c VMs in the domain is $\xi(c)/\mu$, which implies that the mean departure rate of finished service is $\mu/\xi(c)$.

Since the decision making epoch is randomly generated in the system, we use semi-Markov decision process (SMDP) to model the dynamic MCC resource allocation process based on the system description we presented above. SMDP is a stochastic dynamic programming method, which can be used to model and solve optimal dynamic decision making problems. There are six following elements in the SMDP model: (a) *system states*; (b) *action sets*; (c) *the events that cause the decisions*; (d) *decision epoches*; (e) *transition probabilities*; and (f) *reward*. In the following, we first present the system states, the actions, the events, and the reward model for the MCC system.

3.2. System States. According to the assumption, there are total K VMs in one service provisioning domain, and c VM can be allocated to the service request, which is from 1 to C , where $C \leq K$. However, the arrival of *paid* application service request and *free* application service request and the departure of the finished service are distinct events. Thus, the system states can be described by the number of the running Cloud services which occupy the same number of VMs and the events (including both arrival and departure events) in the service provisioning domain. Here, we use c to indicate the number of VMs allocated to one application service (denoted as c allocation scheme as presented in Section 3.1), $c \in \{1, 2, \dots, C\}$. Therefore, the number of the running Cloud services which occupy c VMs in one service provisioning domain can be denoted as s_c .

In the MCC system model, we can define two types of service events: (1) a *paid* or *free* service request arrives from an MD, denoted by A_p and A_f , respectively; and (2) the departure of a finished application service occupying c VMs in the current service provisioning domain, denoted by F_c . Thus, the event e in the MCC system can be described as $e \in \{A_p, A_f, F_1, F_2, \dots, F_C\}$. Therefore, the system state can be expressed as

$$S = \{s \mid s = \langle s_1, s_2, \dots, s_C, e \rangle\}, \quad (1)$$

where $\sum_{c=1}^C (s_c * c) \leq K$.

3.3. Actions. For a system state of the service provisioning domain with an incoming service request from an MD (i.e., A_p or A_f), the mobile Cloud needs to make a decision on whether to accept the service request and what is the allocation scheme (i.e., how many VMs to allocate to the MD) if the decision is acceptance. If the decision is acceptance, then the c allocation scheme is assigned to the arriving service request; thus, the action to assign the c allocation scheme can be denoted as $a(s) = c$. While if the decision is rejection based on the whole system reward, which means no VM will be assigned, thus the *paid* or *free* service request will be rejected and the application will run on the MD itself. Then, the action to reject the service request can be denoted as $a(s) = 0$.

And for the departure of a finished service in the service provisioning domain (i.e., $e = F_c$), the action for this event can be considered as to calculate the current available Cloud resources and denoted as $a(s) = -1$. Therefore, the action space can be defined as $a(s) \subseteq \text{Act}_s$, where

$$a(s) = \begin{cases} \{0, 1, \dots, C\}, & e \in \{A_p, A_f\}, \\ -1, & e \in \{F_1, F_2, \dots, F_C\}. \end{cases} \quad (2)$$

3.4. Reward Model. Based on the system state and its corresponding action, we can evaluate the whole mobile Cloud system reward (denoted by $r(s, a)$), which is computed based on the income and the cost as follows:

$$r(s, a) = w(s, a) - g(s, a), \quad e \in \{A_p, A_f, F_1, F_2, \dots, F_C\}, \quad (3)$$

where $w(s, a)$ is the net lump sum income for the Cloud and MDs and $g(s, a)$ denotes the system cost.

The net lump sum income should consider the payment from MD to the mobile Cloud, the saved battery energy of MD, and the consumed time of mobile Cloud to process the service if the service is run in the mobile Cloud, the consumed battery energy, and the consumed time of MD if the service is run on MD locally.

Thus, the net lump sum income $w(s, a)$ is computed as

$$w(s, a) = \begin{cases} 0, & a(s) = -1, e \in \{F_1, F_2, \dots, F_C\} \\ -\gamma_d U_d - \theta_d \beta, & a(s) = 0, e \in \{A_p, A_f\} \\ E_d - \delta_d \beta - \xi(c) \theta_s \beta, & a(s) = c, e = A_p \\ -\delta_d \beta - \xi(c) \theta_s \beta, & a(s) = c, e = A_f. \end{cases} \quad (4)$$

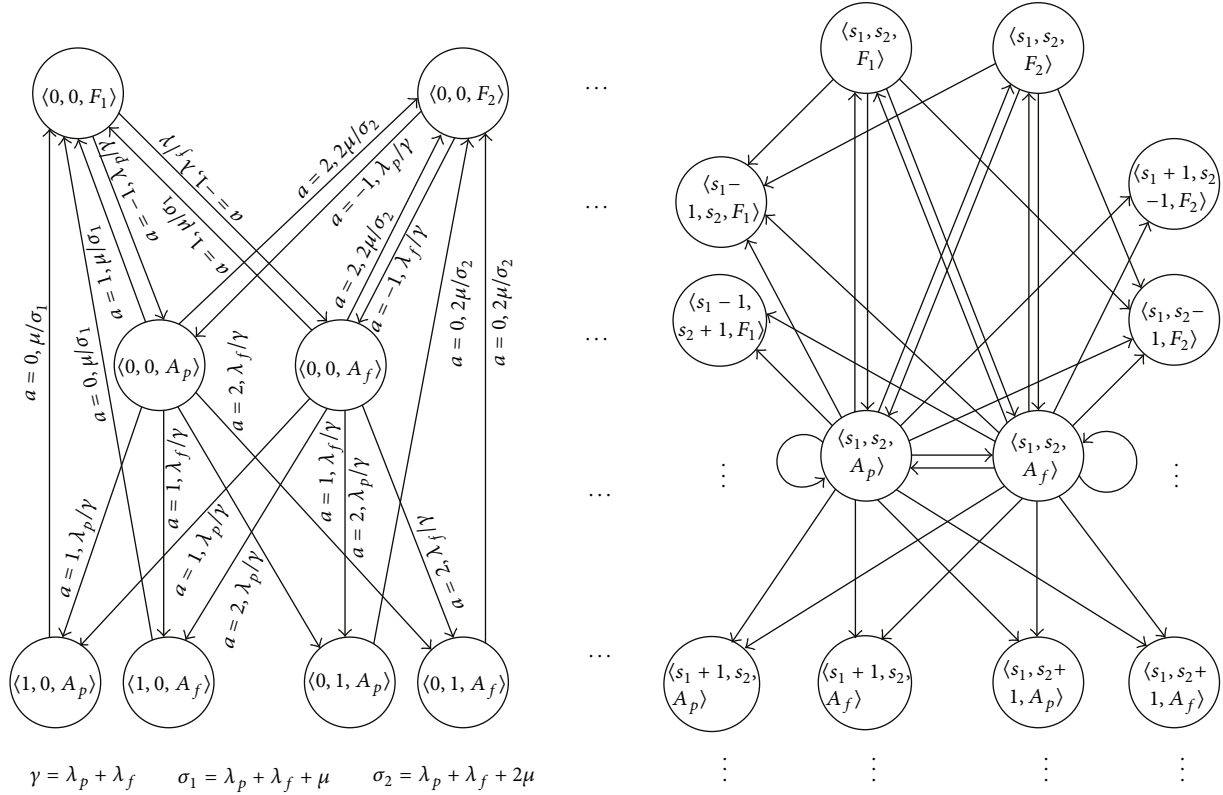


FIGURE 2: An example of state transition probabilities for two allocation schemes. The first item represents the action and the second item represents the state transition probability.

In (4), E_d is the income of the service provisioning domain obtained from the MD when it accepts a *paid* service request from the MD. δ_d denotes the time consumed on transmitting the service request from MD to the service provisioning domain through wireless connection, while β denotes the price per unit time, which has the same measurement unit as the income. Thus, $\delta_d\beta$ denotes the expense measured by the time consumed on transmitting the service request from MD to the service provisioning domain. U_d represents the expense measured by the battery energy consumed by the MD when the service request is rejected by the service provisioning domain and run on the MD locally, which has the same measurement unit as the income. γ_d is the weight factor that satisfies $0 \leq \gamma_d \leq 1$. Let θ_d denote the time to process an application service by using one mobile device, then $\theta_d\beta$ represents the expense measured by the time consumed to process the application using one mobile device. Similarly, $\theta_s\beta$ denotes the expense measured by the time consumed to process the service using one VM in a service provisioning domain. Therefore, $\xi(c)\theta_s\beta$ denotes the expense measured by the time consumed to process the service using c VMs in a service provisioning domain.

In (3), $g(s, a)$ is given by

$$g(s, a) = \tau(s, a) o(s, a), \quad a(s) \in \text{Act}_s. \quad (5)$$

In (5), $\tau(s, a)$ is the average expected service time when the system state transfers from current state s to the next potential state j and the decision a is made; $o(s, a)$ is the cost rate of the service time and it is defined as the number of all occupied VMs; thus, it can be computed as

$$o(s, a) = \sum_{c=1}^C (s_c * c). \quad (6)$$

4. SMDP-Based Mobile Computing Model

Based on the SMDP model, we have already defined the *system states*, *action sets*, the *events*, and *reward* for the MCC system in the last section, then we need to define the *decision epoches* and obtain the *transition probabilities* to calculate the maximum long-term whole system reward.

There are three types of events in the MCC system (i.e., an arrival of a *paid* service request, an arrival of a *free* service request, and a departure of a finished service). The next decision epoch occurs when any of the three types of events takes place. Based on our assumption, the arrival of service request follows Poisson distribution and the departure of finished service follows exponential distribution. Thus, the expected time duration between two decision epoches (i.e., $\tau(s, a)$) follows exponential distribution as well. Then,

TABLE 1: States transition probabilities of system model at $C = 2$. $\gamma = \lambda_p + \lambda_f + s_1\mu + 2s_2\mu$, $\sigma_1 = \lambda_p + \lambda_f + (s_1 + 1)\mu + 2s_2\mu$, $\sigma_2 = \lambda_p + \lambda_f + s_1\mu + 2(s_2 + 1)\mu$.

Current state	Next state	Action (a)	Transition probability
$\langle s_1, s_2, A_p \rangle$	$\langle s_1, s_2, A_p \rangle$	0, -2	λ_p/γ ,
	$\langle s_1, s_2, A_f \rangle$	0, -2	λ_f/γ ,
	$\langle s_1 - 1, s_2, F_1 \rangle$	0, -2	$s_1\mu/\gamma$,
	$\langle s_1 - 1, s_2 + 1, F_1 \rangle$	2	$s_1\mu/\sigma_2$,
	$\langle s_1, s_2, F_1 \rangle$	1	$(s_1 + 1)\mu/\sigma_1$,
	$\langle s_1, s_2, F_2 \rangle$	2	$2(s_2 + 1)\mu/\sigma_2$,
	$\langle s_1 + 1, s_2 - 1, F_2 \rangle$	1	$2s_2\mu/\sigma_1$,
	$\langle s_1, s_2 - 1, F_2 \rangle$	0, -2	$2s_2\mu/\gamma$,
	$\langle s_1 + 1, s_2, A_p \rangle$	1	λ_p/σ_1 ,
	$\langle s_1 + 1, s_2, A_f \rangle$	1	λ_f/σ_1 ,
	$\langle s_1, s_2 + 1, A_p \rangle$	2	λ_p/σ_2 ,
	$\langle s_1, s_2 + 1, A_f \rangle$	2	λ_f/σ_2 ,
$\langle s_1, s_2, A_f \rangle$	$\langle s_1, s_2, A_p \rangle$	0	λ_p/γ ,
	$\langle s_1, s_2, A_f \rangle$	0	λ_f/γ ,
	$\langle s_1 - 1, s_2, F_1 \rangle$	0	$s_1\mu/\gamma$,
	$\langle s_1 - 1, s_2 + 1, F_1 \rangle$	2	$s_1\mu/\sigma_2$,
	$\langle s_1, s_2, F_1 \rangle$	1	$(s_1 + 1)\mu/\sigma_1$,
	$\langle s_1, s_2, F_2 \rangle$	2	$2(s_2 + 1)\mu/\sigma_2$,
	$\langle s_1 + 1, s_2 - 1, F_2 \rangle$	1	$2s_2\mu/\sigma_1$,
	$\langle s_1, s_2 - 1, F_2 \rangle$	0	$2s_2\mu/\gamma$,
	$\langle s_1 + 1, s_2, A_p \rangle$	1	λ_p/σ_1 ,
	$\langle s_1 + 1, s_2, A_f \rangle$	1	λ_f/σ_1 ,
	$\langle s_1, s_2 + 1, A_p \rangle$	2	λ_p/σ_2 ,
	$\langle s_1, s_2 + 1, A_f \rangle$	2	λ_f/σ_2 ,
$\langle s_1, s_2, F_1 \rangle$	$\langle s_1 - 1, s_2, F_1 \rangle$	-1	$s_1\mu/\gamma$,
	$\langle s_1, s_2, A_p \rangle$	-1	λ_p/γ ,
	$\langle s_1, s_2, A_f \rangle$	-1	λ_f/γ ,
$\langle s_1, s_2, F_2 \rangle$	$\langle s_1, s_2, A_f \rangle$	-1	λ_f/γ ,
	$\langle s_1, s_2 - 1, F_2 \rangle$	-1	$2s_2\mu/\gamma$.

the mean rate (denoted as $\gamma(s, a)$) of expected time can be represented as

$$\gamma(s, a) = \tau(s, a)^{-1}$$

$$= \begin{cases} \lambda_p + \lambda_f + \sum_{c=1}^C \frac{s_c\mu}{\xi(c)}, & e \in \{F_1, F_2, \dots, F_C\} \\ \text{or } e \in \{A_p, A_f\}, a = 0, \\ \lambda_p + \lambda_f + \sum_{c=1}^C \frac{s_c\mu}{\xi(c)} + \frac{\mu}{\xi(c)}, & e \in \{A_p, A_f\}, a = c. \end{cases} \quad (7)$$

Thus, the expected discounted reward (denoted as $r(s, a)$) during $\tau(s, a)$ can be obtained based on the discounted

reward model defined in [18, 19],

$$\begin{aligned} r(s, a) &= w(s, a) - o(s, a) E_s^a \left\{ \int_0^\tau e^{-\alpha t} dt \right\} \\ &= w(s, a) - o(s, a) E_s^a \left\{ \frac{[1 - e^{-\alpha \tau}]}{\alpha} \right\} \\ &= w(s, a) - \frac{o(s, a)}{\alpha + \gamma(s, a)}, \end{aligned} \quad (8)$$

where α is a continuous-time discounting factor and $w(s, a)$, $o(s, a)$, and $\gamma(s, a)$ are defined in (4), (6), and (7), respectively.

Then the only element left to be calculated is the transition probabilities. To calculate the transition probabilities, we show an example in Figure 2.

In this example, without loss of generality, we assume that there are only two allocation schemes, which means $C = 2$. Thus, the transition probabilities in this example can be obtained in Table 1.

From the example, the transition probabilities of C allocation schemes can be deduced. Let $q(j | s, a)$ denote the state transition probability from the current state s to the next state j when action a is chosen. Then, the transition probability $q(j | s, a)$ can be expressed as following.

For the state $s = \langle s_1, s_2, \dots, s_c, \dots, s_C, A_p \rangle$, $q(j | s, a)$ can be obtained as

$$\begin{aligned} q(j | s, a) &= \begin{cases} \frac{\lambda_p}{\gamma(s, a)}, & j = \langle s_1, s_2, \dots, s_C, A_p \rangle, a = 0 \\ \frac{\lambda_f}{\gamma(s, a)}, & j = \langle s_1, s_2, \dots, s_C, A_f \rangle, a = 0 \\ \frac{s_c\mu}{\varepsilon(c)\gamma(s, a)}, & j = \langle s_1, s_2, \dots, s_c - 1, \dots, s_C, F_c \rangle, \\ \frac{(s_c + 1)\mu}{\varepsilon(c)\gamma(s, a)}, & j = \langle s_1, s_2, \dots, s_c, \dots, s_C, F_c \rangle, a = c \\ \frac{s_m\mu}{\varepsilon(m)\gamma(s, a)}, & j = \langle s_1, s_2, \dots, s_m - 1, \dots, s_c + 1, \dots, \\ & \quad S_C, F_m \rangle, s_m \geq 1, m \neq c, a = c \\ \frac{\lambda_p}{\gamma(s, a)}, & j = \langle s_1, s_2, \dots, s_c + 1, \dots, s_C, A_p \rangle, \\ & \quad s_c \leq C - 1, a = c \\ \frac{\lambda_f}{\gamma(s, a)}, & j = \langle s_1, s_2, \dots, s_c + 1, \dots, s_C, A_f \rangle, \\ & \quad s_c \leq C - 1, a = c, \end{cases} \end{aligned} \quad (9)$$

where $c \in \{1, 2, \dots, C\}$, $m \in \{1, 2, \dots, C\}$, $m \neq c$.

For the states $s = \langle s_1, s_2, \dots, s_c, \dots, s_C, A_f \rangle$, $q(j | s, a)$ can be obtained

$$q(j | s, a) = \begin{cases} \frac{\lambda_p}{\gamma(s, a)}, & j = \langle s_1, s_2, \dots, s_C, A_p \rangle, a = 0 \\ \frac{\lambda_f}{\gamma(s, a)}, & j = \langle s_1, s_2, \dots, s_C, A_f \rangle, a = 0 \\ \frac{s_c \mu}{\varepsilon(c) \gamma(s, a)}, & j = \langle s_1, s_2, \dots, s_c - 1, \dots, s_C, F_c \rangle, \\ & s_c \geq 1, a = 0 \\ \frac{(s_c + 1) \mu}{\varepsilon(c) \gamma(s, a)}, & j = \langle s_1, s_2, \dots, s_c, \dots, s_C, F_c \rangle, \\ & a = c \\ \frac{s_m \mu}{\varepsilon(m) \gamma(s, a)}, & j = \langle s_1, s_2, \dots, s_m - 1, \dots, \\ & s_c + 1, \dots, s_C, F_m \rangle, \\ & s_m \geq 1, m \neq c, a = c \\ \frac{\lambda_p}{\gamma(s, a)}, & j = \langle s_1, s_2, \dots, s_c + 1, \dots, s_C, A_p \rangle, \\ & s_c \leq C - 1, a = c \\ \frac{\lambda_f}{\gamma(s, a)}, & j = \langle s_1, s_2, \dots, s_c + 1, \dots, s_C, A_f \rangle, \\ & s_c \leq C - 1, a = c, \end{cases} \quad (10)$$

where $c \in \{1, 2, \dots, C\}$, $m \in \{1, 2, \dots, C\}$, and $m \neq c$.

For the states $s = \langle s_1, s_2, \dots, s_c, \dots, s_C, F_c \rangle$, the action for this departure state is always -1 which means $a = -1$, then the transition probability $q(j | s, a)$ can be obtained as

$$q(j | s, a) = \begin{cases} \frac{\lambda_p}{\gamma(s, a)}, & j = \langle s_1, s_2, \dots, s_C, A_p \rangle \\ \frac{\lambda_f}{\gamma(s, a)}, & j = \langle s_1, s_2, \dots, s_C, A_f \rangle \\ \frac{s_c \mu}{\xi(c) \gamma(s, a)}, & j = \langle s_1, s_2, \dots, s_c - 1, \\ & \dots, s_C, F_c \rangle, s_c \geq 1, \end{cases} \quad (11)$$

where $c \in \{1, 2, \dots, C\}$.

Then, the maximal long-term discounted reward is obtained based on the discounted reward model defined in [18, 19] and can be denoted as

$$v(s) = \max_{a \in \text{Act}_s} \left\{ r(s, a) + \lambda \sum_{j \in S} q(j | s, a) v(j) \right\}, \quad (12)$$

where $\lambda = (\gamma(s, a)) / (\alpha + \gamma(s, a))$, and $r(s, a)$ and $q(j | s, a)$ can be obtained in (8), (9), (10), and (11).

In the reward equation (8), the first part is that the revenue is a lump earnings of the reward and the second part is that the cost is a continuous-time payment of the reward. Thus, the reward function needs to be uniformized to obtain the uniformized long-term reward, then the discrete-time discounted Markov decision process can be used in this model. Based on the *assumption 11.5.1* in [19], we need to find a constant ω satisfying $[1 - q(s | s, a)]\gamma(s, a) \leq \omega < \infty$ to obtain the uniformized long-term reward by utilizing (11.5.8) in [19]. Let $\omega = \lambda_f + \lambda_p + K * C * \mu$ and $\bar{q}(j | s, a)$, $\bar{v}(s)$, $\bar{\gamma}(s, a)$ denote the uniformized transition probability, the long-term reward, and the reward function, respectively.

Thus, the transition probability can be uniformized as

$$\bar{q}(j | s, a) = \begin{cases} 1 - \frac{[1 - q(s | s, a)]\gamma(s, a)}{\omega}, & j = s \\ \frac{q(j | s, a)\gamma(s, a)}{\omega}, & j \neq s. \end{cases} \quad (13)$$

For the state $s = \langle s_1, s_2, \dots, s_c, \dots, s_C, A_p \rangle$, the uniformized transition probability $\bar{q}(j | s, a)$ is rewritten as

$$\bar{q}(j | s, a) = \begin{cases} \frac{(\omega + \lambda_p - \gamma(s, a))}{\omega}, & j = \langle s_1, s_2, \dots, s_C, A_p \rangle, a = 0 \\ \frac{\lambda_f}{\omega}, & j = \langle s_1, s_2, \dots, s_C, A_f \rangle, a = 0 \\ \frac{s_c \mu}{\xi(c) \omega}, & j = \langle s_1, s_2, \dots, s_c - 1, \dots, s_C, F_c \rangle, \\ & s_c \geq 1, a = 0 \\ \frac{(s_c + 1) \mu}{\xi(c) \omega}, & j = \langle s_1, s_2, \dots, s_c, \dots, s_C, F_c \rangle, a = c \\ \frac{s_m \mu}{\xi(m) \omega}, & j = \langle s_1, s_2, \dots, s_m - 1, \dots, \\ & s_c + 1, \dots, s_C, F_m \rangle, s_m \geq 1, \\ & m \neq c, a = c \\ \frac{\lambda_p}{\omega}, & j = \langle s_1, s_2, \dots, s_c + 1, \dots, s_C, A_p \rangle, \\ & s_c \leq C - 1, a = c \\ \frac{\lambda_f}{\omega}, & j = \langle s_1, s_2, \dots, s_c + 1, \dots, s_C, A_f \rangle, \\ & s_c \leq C - 1, a = c \\ \frac{(\omega - \gamma(s, a))}{\omega}, & j = s, a = c. \end{cases} \quad (14)$$

Similarly, for the state $s = \langle s_1, s_2, \dots, s_c, \dots, s_C, A_f \rangle$, the uniformized transition probability $\bar{q}(j | s, a)$ can be rewritten

as

$$\bar{q}(j | s, a) = \begin{cases} \frac{(\omega + \lambda_p - \gamma(s, a))}{\omega}, & j = \langle s_1, s_2, \dots, s_C, A_p \rangle, a = 0 \\ \frac{\lambda_f}{\omega}, & j = \langle s_1, s_2, \dots, s_C, A_f \rangle, a = 0 \\ \frac{s_c \mu}{\xi(c) \omega}, & j = \langle s_1, s_2, \dots, s_c - 1, \dots, s_C, F_c \rangle, \\ & s_c \geq 1, a = 0 \\ \frac{(s_c + 1) \mu}{\xi(c) \omega}, & j = \langle s_1, s_2, \dots, s_c, \dots, s_C, F_c \rangle, \\ & a = c \\ \frac{s_m \mu}{\xi(m) \omega}, & j = \langle s_1, s_2, \dots, s_m - 1, \dots, \\ & s_c + 1, \dots, s_C, F_m \rangle, s_m \geq 1, \\ & m \neq c, a = c \\ \frac{\lambda_p}{\omega}, & j = \langle s_1, s_2, \dots, s_c + 1, \dots, s_C, A_p \rangle, \\ & s_c \leq C - 1, a = c \\ \frac{\lambda_f}{\omega}, & j = \langle s_1, s_2, \dots, s_c + 1, \dots, s_C, A_f \rangle, \\ & s_c \leq C - 1, a = c \\ \frac{(\omega - \gamma(s, a))}{\omega}, & j = s, a = c. \end{cases} \quad (15)$$

And for the state $s = \langle s_1, s_2, \dots, s_c, \dots, s_C, F_c \rangle$, the uniformized transition probability $\bar{q}(j | s, a)$ is rewritten as

$$\bar{q}(j | s, a) = \begin{cases} \frac{\lambda_p}{\omega}, & j = \langle s_1, s_2, \dots, s_C, A_p \rangle \\ \frac{\lambda_f}{\omega}, & j = \langle s_1, s_2, \dots, s_C, A_f \rangle \\ \frac{s_c \mu}{\xi(c) \omega}, & j = \langle s_1, s_2, \dots, s_c - 1, \\ & \dots, s_C, F_c \rangle, s_c \geq 1, \\ \frac{(\omega - \gamma(s, a))}{\omega}, & j = s. \end{cases} \quad (16)$$

Using the uniformization equations presented above, then the expected maximal long-term reward in (12) can be uniformized as

$$\bar{r}(s, a) = r(s, a) \frac{\gamma(s, a) + \alpha}{(\alpha + \omega)} \quad (17)$$

and the parameter λ can be uniformized as $\bar{\lambda} = \omega/(\omega + \alpha)$.

Thus, according to the uniformization equations (14), (15), (16), and (17), the uniformized maximal long-term expected reward is obtained as

$$\bar{v}(s) = \max_{a \in \text{Act}_s} \left\{ \bar{r}(s, a) + \bar{\lambda} \sum_{j \in S} \bar{q}(j | s, a) \bar{v}(j) \right\}. \quad (18)$$

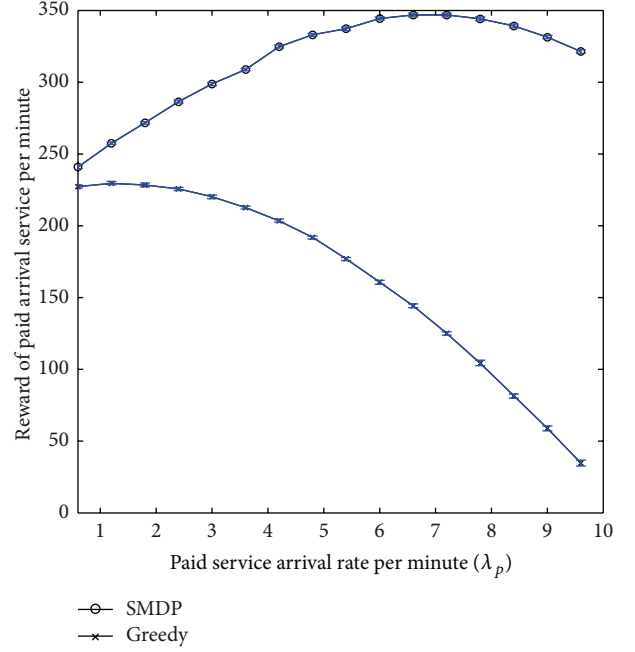


FIGURE 3: System reward of *paid* service compared between SMDP model and greedy method, varying with the arrival rate of *paid* service requests ($\lambda_f = 2.4$, $\mu = 6.6$, $K = 6$).

5. Performance Analysis

The probability of allocation scheme c , which is defined as the probability that c VMs are allocated for a cloud service, is an important performance metric for ensuring the user satisfaction level and the Cloud resource utilization ratio. It is very useful for the operator to manage the system capacity/utilization status based on the system parameters of the service provisioning domain (such as arrival rate, departure rate, and the VM number of Cloud resource). Meanwhile, blocking service request does not only mean the loss of whole system reward, but also means the degradation of users' satisfaction level. Then, the blocking probability, which is the probability that blocking the cloud service requests from mobile device, is another important performance metrics for the service provisioning domain. In this section, we analytically derive the probabilities of each allocation scheme and blocking probability for the proposed economic mobile computing model based on SMDP.

From the reward function (18) and probability equations (14), (15), and (16), the expected total discounted reward $\bar{v}(s)$ at state $s \in S$ is related with the arrival rates of *paid* service request (λ_p) and *free* service request (λ_f), the departure rate ($\mu/\xi(c)$) of each allocation scheme, the occupied Cloud resource expressed by the number of being occupied VMs $\sum_{c=1}^C (s_c * c)$, and the capability of the service provisioning domain (i.e., the total number of VMs- K). For a given service provisioning domain and a certain system state of an arrival of service request (i.e., $\langle s_1, s_2, \dots, s_C, A_p \rangle$ or $\langle s_1, s_2, \dots, s_C, A_f \rangle$), the above parameters λ_p , λ_f , $\mu/\xi(c)$,

$\sum_{c=1}^C (s_c * c)$, and K are fixed. As a result, the steady-state probability of each state can be obtained from the probability equations (14), (15), and (16). Thus, the probabilities of each allocation scheme and blocking probability can also be achieved through the steady-state probability of each state.

Let π_s denote the steady-state probability of the system state s in the service provisioning domain. From the example in Figure 2 and Table 1, the steady-state probability of $\pi_{\langle s_1, s_2, \dots, s_C, e \rangle}$ can be classified as three types: (1) the arrival of a *paid* service request; (2) the arrival of a *free* service request; (3) the departure of a finished service with c allocation scheme. Based on the probability equations (14), (15), and (16), the steady-state probabilities $\pi_{\langle s_1, s_2, \dots, s_C, A_p \rangle}$ and $\pi_{\langle s_1, s_2, \dots, s_C, A_f \rangle}$ can be derived as follows

$$\begin{aligned} \pi_{\langle s_1, s_2, \dots, s_C, A_p \rangle} &= \frac{\lambda_p}{\gamma(s, a)} \rho_{\langle s_1, s_2, \dots, s_C, A_p \rangle} \pi_{\langle s_1, s_2, \dots, s_C, A_p \rangle} \\ &+ \frac{\lambda_p}{\gamma(s, a)} \rho_{\langle s_1, s_2, \dots, s_C, A_f \rangle} \pi_{\langle s_1, s_2, \dots, s_C, A_f \rangle} \\ &+ \frac{\lambda_p}{\gamma(s, a)} \sum_{c=1}^C \rho_{\langle s_1, s_2, \dots, s_{c-1}, \dots, s_C, A_p \rangle} \pi_{\langle s_1, s_2, \dots, s_{c-1}, \dots, s_C, A_p \rangle} \\ &+ \frac{\lambda_p}{\gamma(s, a)} \sum_{c=1}^C \rho_{\langle s_1, s_2, \dots, s_{c-1}, \dots, s_C, A_f \rangle} \pi_{\langle s_1, s_2, \dots, s_{c-1}, \dots, s_C, A_f \rangle} \\ &+ \frac{\lambda_p}{\gamma(s, a)} \sum_{c=1}^C \pi_{\langle s_1, s_2, \dots, s_C, F_c \rangle} \end{aligned} \quad (19)$$

$$\begin{aligned} \pi_{\langle s_1, s_2, \dots, s_C, A_f \rangle} &= \frac{\lambda_f}{\gamma(s, a)} \rho_{\langle s_1, s_2, \dots, s_C, A_p \rangle} \pi_{\langle s_1, s_2, \dots, s_C, A_p \rangle} \\ &+ \frac{\lambda_f}{\gamma(s, a)} \rho_{\langle s_1, s_2, \dots, s_C, A_f \rangle} \pi_{\langle s_1, s_2, \dots, s_C, A_f \rangle} \\ &+ \frac{\lambda_f}{\gamma(s, a)} \sum_{c=1}^C \rho_{\langle s_1, s_2, \dots, s_{c-1}, \dots, s_C, A_p \rangle} \pi_{\langle s_1, s_2, \dots, s_{c-1}, \dots, s_C, A_p \rangle} \\ &+ \frac{\lambda_f}{\gamma(s, a)} \sum_{c=1}^C \rho_{\langle s_1, s_2, \dots, s_{c-1}, \dots, s_C, A_f \rangle} \pi_{\langle s_1, s_2, \dots, s_{c-1}, \dots, s_C, A_f \rangle} \\ &+ \frac{\lambda_f}{\gamma(s, a)} \sum_{c=1}^C \pi_{\langle s_1, s_2, \dots, s_C, F_c \rangle}, \end{aligned} \quad (20)$$

where $\rho_{\langle s_1, s_2, \dots, s_C, A_p \rangle}$, $\rho_{\langle s_1, s_2, \dots, s_C, A_f \rangle}$, $\rho_{\langle s_1, s_2, \dots, s_{c-1}, \dots, s_C, A_p \rangle}$ and $\rho_{\langle s_1, s_2, \dots, s_{c-1}, \dots, s_C, A_f \rangle}$, are the parameters decided by the

correlative actions respectively as follows:

$$\begin{aligned} \rho_{\langle s_1, s_2, \dots, s_C, A_p \rangle} &= \begin{cases} 1, & a_{\langle s_1, s_2, \dots, s_C, A_p \rangle} = 0, \\ 0, & \text{otherwise,} \end{cases} \\ \rho_{\langle s_1, s_2, \dots, s_C, A_f \rangle} &= \begin{cases} 1, & a_{\langle s_1, s_2, \dots, s_C, A_f \rangle} = 0, \\ 0, & \text{otherwise.} \end{cases} \\ \rho_{\langle s_1, s_2, \dots, s_{c-1}, \dots, s_C, A_p \rangle} &= \begin{cases} 1, & a_{\langle s_1, s_2, \dots, s_{c-1}, \dots, s_C, A_p \rangle} = c, \quad c \in \{1, 2, \dots, C\}, \\ 0, & \text{otherwise,} \end{cases} \\ \rho_{\langle s_1, s_2, \dots, s_{c-1}, \dots, s_C, A_f \rangle} &= \begin{cases} 1, & a_{\langle s_1, s_2, \dots, s_{c-1}, \dots, s_C, A_f \rangle} = c, \quad c \in \{1, 2, \dots, C\}, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (21)$$

Similarly, the steady-state probability $\pi_{\langle s_1, s_2, \dots, s_C, F_c \rangle}$ can be attained as

$$\begin{aligned} \pi_{\langle s_1, s_2, \dots, s_C, F_c \rangle} &= \frac{(s_c + 1) \mu}{\xi(c) \gamma(s, a)} \rho_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_C, A_p \rangle} \pi_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_C, A_p \rangle} \\ &+ \frac{(s_c + 1) \mu}{\xi(c) \gamma(s, a)} \rho_{\langle s_1, s_2, \dots, s_C, A_p \rangle} \pi_{\langle s_1, s_2, \dots, s_C, A_p \rangle} \\ &+ \frac{(s_c + 1) \mu}{\xi(c) \gamma(s, a)} \sum_{m=1, m \neq c}^C \rho_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_{m-1}, \dots, s_C, A_p \rangle} \\ &\quad \times \pi_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_{m-1}, \dots, s_C, A_p \rangle} \\ &+ \frac{(s_c + 1) \mu}{\xi(c) \gamma(s, a)} \rho_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_C, A_f \rangle} \pi_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_C, A_f \rangle} \\ &+ \frac{(s_c + 1) \mu}{\xi(c) \gamma(s, a)} \rho_{\langle s_1, s_2, \dots, s_C, A_f \rangle} \pi_{\langle s_1, s_2, \dots, s_C, A_f \rangle} \\ &+ \frac{(s_c + 1) \mu}{\xi(c) \gamma(s, a)} \sum_{m=1, m \neq c}^C \rho_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_{m-1}, \dots, s_C, A_f \rangle} \\ &\quad \times \pi_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_{m-1}, \dots, s_C, A_f \rangle} \\ &+ \frac{(s_c + 1) \mu}{\xi(c) \gamma(s, a)} \sum_{m=1}^C \pi_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_C, F_m \rangle}, \end{aligned} \quad (22)$$

where

$\rho_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_C, A_p \rangle}$, $\rho_{\langle s_1, s_2, \dots, s_C, A_p \rangle}$, $\rho_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_{m-1}, \dots, s_C, A_p \rangle}$, $\rho_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_C, A_f \rangle}$, $\rho_{\langle s_1, s_2, \dots, s_C, A_f \rangle}$ and $\rho_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_{m-1}, \dots, s_C, A_f \rangle}$ are defined by the related actions

TABLE 2: Simulation parameters.

Parameter	Value
E_d	50
δ_d	30
β	1
γ_d	1
U_d	10
θ_d	60
θ_s	12

respectively as

$$\rho_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_C, A_p \rangle} = \begin{cases} 1, & a_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_C, A_p \rangle} = 0, \\ 0, & \text{otherwise,} \end{cases}$$

$$\rho_{\langle s_1, s_2, \dots, s_C, A_p \rangle} = \begin{cases} 1, & a_{\langle s_1, s_2, \dots, s_C, A_p \rangle} = c, \quad c \in \{1, 2, \dots, C\}, \\ 0, & \text{otherwise,} \end{cases}$$

$$\rho_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_{m-1}, \dots, s_C, A_p \rangle}$$

$$= \begin{cases} 1, & a_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_{m-1}, \dots, s_C, A_p \rangle} = m, \quad c \in \{1, 2, \dots, C\}, \\ & m \in \{1, 2, \dots, C\}, \quad m \neq c, \\ 0, & \text{otherwise,} \end{cases}$$

$$\rho_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_C, A_f \rangle} = \begin{cases} 1, & a_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_C, A_f \rangle} = 0, \\ 0, & \text{otherwise,} \end{cases}$$

$$\rho_{\langle s_1, s_2, \dots, s_C, A_f \rangle} = \begin{cases} 1, & a_{\langle s_1, s_2, \dots, s_C, A_f \rangle} = c, \quad c \in \{1, 2, \dots, C\}, \\ 0, & \text{otherwise,} \end{cases}$$

$$\rho_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_{m-1}, \dots, s_C, A_f \rangle}$$

$$= \begin{cases} 1, & a_{\langle s_1, s_2, \dots, s_{c+1}, \dots, s_{m-1}, \dots, s_C, A_f \rangle} = m, \quad c \in \{1, 2, \dots, C\}, \\ & m \in \{1, 2, \dots, C\}, \quad m \neq c, \\ 0, & \text{otherwise.} \end{cases} \quad (23)$$

Since the sum of the steady-state probabilities for all states equals to 1, we have

$$\sum_S \left(\pi_{\langle s_1, s_2, \dots, s_C, A_p \rangle} + \pi_{\langle s_1, s_2, \dots, s_C, A_f \rangle} + \pi_{\langle s_1, s_2, \dots, s_C, F_c \rangle} \right) = 1. \quad (24)$$

Therefore, the steady-state probability of each state in an MCC service provisioning domain can be obtained by solving (19), (20), (22), and (24). Thus, as a result, for the service request arrival states (i.e., $\langle s_1, s_2, \dots, s_C, A_p \rangle$ and $\langle s_1, s_2, \dots, s_C, A_f \rangle$) in one service provisioning domain, the probability of each action can be achieved, which is the ratio of the sum of all steady-state probabilities with the same action to the sum of the steady-state probabilities of all service request arrival states (i.e., $\langle s_1, s_2, \dots, s_C, A_p \rangle$ or $\langle s_1, s_2, \dots, s_C, A_f \rangle$) in one domain. Let Pp_a and Pf_a denote the probability of each action for *paid* service request and

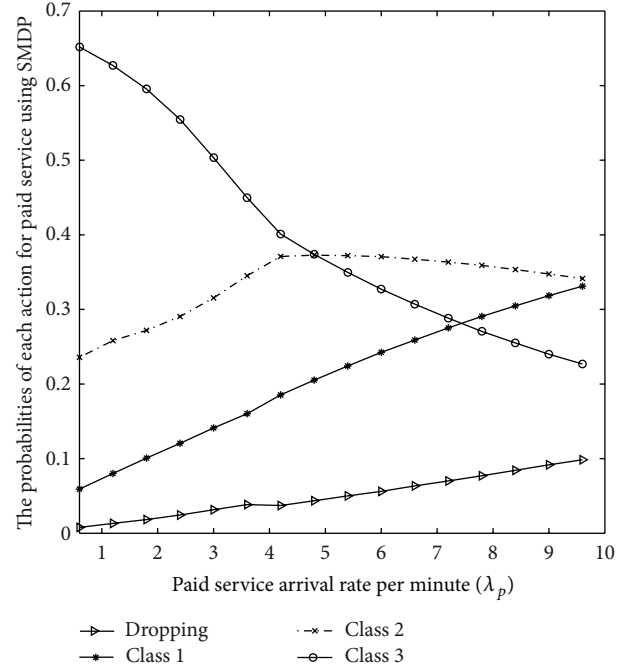


FIGURE 4: Probabilities for each action of *paid* service using SMDP model, varying with the arrival rate of *paid* service requests ($\lambda_f = 2.4$, $\mu = 6.6$, $K = 6$).

free service request, respectively, then, Pp_a and Pf_a can be expressed as

$$Pp_a = \frac{\sum_{a_{\langle s_1, s_2, \dots, s_C, A_p \rangle} = a} \pi_{\langle s_1, s_2, \dots, s_C, A_p \rangle}}{\sum_{m=0}^C \left(\sum_{a_{\langle s_1, s_2, \dots, s_C, A_p \rangle} = m} \pi_{\langle s_1, s_2, \dots, s_C, A_p \rangle} \right)}, \quad (25)$$

$$a = \{0, 1, 2, \dots, C\},$$

$$Pf_a = \frac{\sum_{a_{\langle s_1, s_2, \dots, s_C, A_f \rangle} = a} \pi_{\langle s_1, s_2, \dots, s_C, A_f \rangle}}{\sum_{m=0}^C \left(\sum_{a_{\langle s_1, s_2, \dots, s_C, A_f \rangle} = m} \pi_{\langle s_1, s_2, \dots, s_C, A_f \rangle} \right)}, \quad (26)$$

$$a \in \{0, 1, 2, \dots, C\}.$$

Based on (26) and (25), the blocking probability for the service request arrival states (i.e., $\langle s_1, s_2, \dots, s_C, A_p \rangle$ and $\langle s_1, s_2, \dots, s_C, A_f \rangle$) in one service provisioning domain can be obtained and denoted as Pp_0 and Pf_0 , respectively.

The high values of Pp_0 and Pf_0 do not only mean the loss of the whole system reward but also the decrease of the QoS of the service provisioning domain. Thus, the blocking probabilities Pp_0 and Pf_0 are very important metrics to measure the capability and QoS of a service provisioning domain. In the next section, we will illustrate the relationships between the blocking probability (i.e., Pp_0 and Pf_0) and the parameters (such as λ_p , λ_f , μ , and K) based on the simulation results.

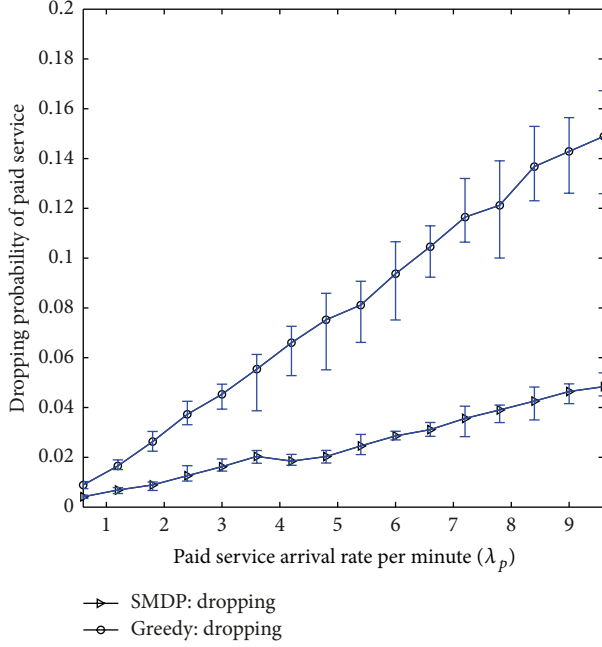


FIGURE 5: Dropping probability of *paid* service compared between SMDP model and greedy method, varying with the arrival rate of *paid* service requests ($\lambda_f = 2.4$, $\mu = 6.6$, $K = 6$).

TABLE 3: Resource allocation decision table for each state of paid service ($\lambda_p = 7.2$, $\lambda_f = 2.4$, $\mu = 6.6$, $K = 10$, $s_3 = 0$).

$s_1 \setminus s_2$	0	1	2	3	4	5
0	3	3	3	3	1	0
1	3	3	3	2	1	—
2	3	3	3	1	0	—
3	3	3	2	1	—	—
4	3	2	1	0	—	—
5	3	2	1	—	—	—
6	2	1	0	—	—	—
7	2	1	—	—	—	—
8	1	0	—	—	—	—
9	1	—	—	—	—	—
10	0	—	—	—	—	—

6. Performance Evaluation

In this section, we evaluate the performance of the proposed economic MCC model based on SMDP by using an event driven simulator compiled by Matlab [20] and compare our proposed model with the traditional greedy algorithm. Since the *paid* service demands a higher QoS level compared with other *free* services, thus our simulation mainly focuses on the performance of *paid* service.

In our simulation, the maximal number of VMs is $C = 3$, and the scheme that allocates $c_1 = 1$, $c_2 = 2$, and $c_3 = 3$ VMs to a service is denoted as allocation scheme c_i . The time to process an application service by the Cloud is assumed as a linear function of the number of VMs allocated to the service, which can be denoted as $\xi(c) = 1/c$. Thus, the

TABLE 4: Resource allocation decision table for each state of paid service ($\lambda_p = 60$, $\lambda_f = 2.4$, $\mu = 6.6$, $K = 10$, $s_3 = 0$).

$s_1 \setminus s_2$	0	1	2	3	4	5
0	3	3	2	1	1	0
1	3	3	2	1	1	—
2	2	2	1	1	0	—
3	2	2	1	1	—	—
4	2	1	1	0	—	—
5	1	1	1	—	—	—
6	1	1	0	—	—	—
7	1	1	—	—	—	—
8	1	0	—	—	—	—
9	1	—	—	—	—	—
10	0	—	—	—	—	—

value of $\xi(c_1)$, $\xi(c_2)$ and $\xi(c_3)$ can be obtained as $\xi(c_1) = 1$, $\xi(c_2) = 1/2$, and $\xi(c_3) = 1/3$. The total resource capability of the service provisioning domain is up to $K = 10$ VMs. Unless otherwise specified, the arrival rates of the *paid* and *free* service request are $\lambda_p = 7.2$ and $\lambda_f = 2.4$, respectively, and the departure rate of finished service occupying one VM is $\mu = 6.6$. Since the time to process the application service occupying one VM is $1/\mu$, then the departure rate of finished service occupying multiple VMs is $\mu/\xi(c)$ which is described in Section 3. Thus, the departure rates of finished service occupying one, two, and three VMs are $\mu_{c_1} = 6.6$, $\mu_{c_2} = 13.2$, and $\mu_{c_3} = 19.8$, respectively. To assure reward computation convergence, the continuous-time discounting factor α is set to be 0.1. The simulation results are collected with each experiment running 18000 s, and each experiment runs 1000 rounds. The other parameters used in this simulation are listed in Table 2.

6.1. Optimal Actions. Tables 3 and 4 illustrate the actions of optimal resource allocation at each system state with different arrival rates of the *paid* service λ_p . The numbers in the tables represent the optimal decisions made on state $\langle s_1, s_2, s_3, e \rangle$. The symbol “—” in the tables denotes that the state does not exist. When no user is in the service provisioning domain, 3 VMs (which implies that the action $a = 3$ is made) are allocated to the *paid* service in both two scenarios, when a *paid* service request arrives. If there are $s_2 = 3$ services in the service provisioning domain, which means that the number of the occupied VMs is 6, thus, there are 4 unoccupied VMs available in the service provisioning domain. Our proposed model allocates 3 VMs ($a = 3$) to the *paid* service request when the arrival rate of *paid* service requests is low ($\lambda_p = 7.2$) and allocates 2 VMs ($a = 2$) to the *paid* service request when the arrival rate of *paid* service requests is high ($\lambda_p = 60$), which implies that when the arrival rate of *paid* service requests increases, our model becomes more conservative to allocate resources to the *paid* service requests. The reason is, for example, for the state $\langle 0, 3, 0, A_p \rangle$, the corresponding lump incomes $w(s, a)$ for c_1 , c_2 , and c_3 are 8, 14, and 16, respectively. Due to the small variance between the lump incomes obtained by allocating c_2 and c_3 VMs to the *paid*

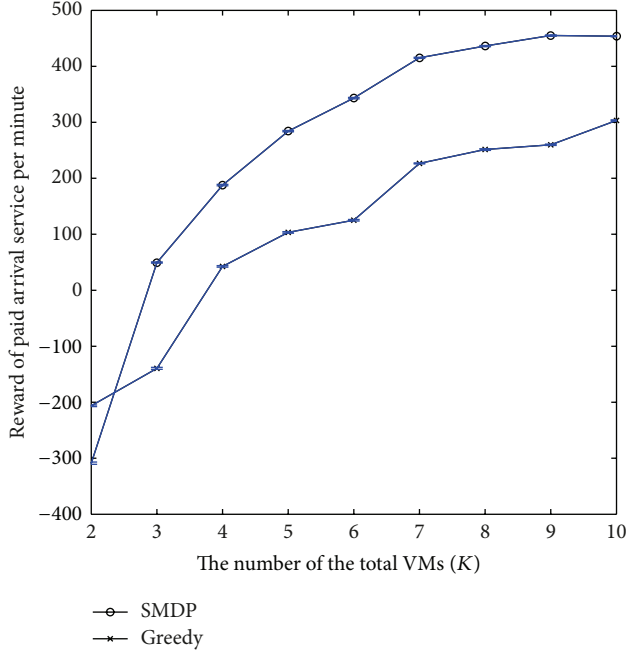


FIGURE 6: System reward of *paid* service compared between SMDP model and greedy method, varying with the number of VMs (K) ($\lambda_p = 7.2$, $\lambda_f = 2.4$, $\mu = 6.6$).

service request, when the arrival rate of *paid* service requests increases (i.e., $\lambda_p = 60$), our model prefers action $a = 2$ other than action $a = 3$, since action $a = 2$ can accommodate more *paid* services to gain higher rewards of the MCC system than action $a = 3$, which consumes more Cloud resources of the service provisioning domain.

6.2. System Rewards and Blocking Probability. To evaluate the performance of the proposed dynamic resource allocation model, we compare the long-term reward and blocking probability of the *paid* service between our model and greedy method in Figures 3, 4, and 5. In Figure 3, the reward of *paid* service of our model increases at the beginning, then falls down with the increase of the arrival rate of *paid* service requests (λ_p), while the reward of *paid* service using the greedy method declines always. It can be seen in this figure that the reward of the *paid* service of our proposed model performs much better than that of greedy method. In Figure 4, with the increase of the arrival rate of the *paid* service requests, our model would rather to allocate more c_1 and c_2 VMs to the *paid* service request other c_3 VM; thus, the dropping probability of our model is lower than that of the greedy method which can be seen in Figure 5 as well. As the rejection has more impact on the system lump income compared with acceptance (in our simulation, the lump income $w(s, a)$ or fine of rejection is -70 , while the corresponding lump incomes $w(s, a)$ for c_1 , c_2 , and c_3 are 8, 14, and 16, resp.), thus the lower dropping probability of our model gains more rewards of *paid* service than the greedy method. We can also see in Figure 4 that when the arrival rate of the *paid* service requests is over 7, the probabilities to

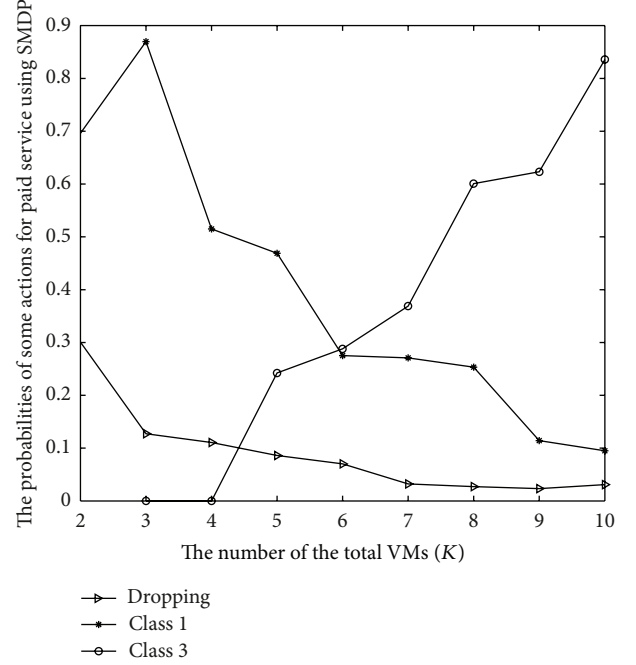


FIGURE 7: Probabilities for each action of *paid* service using SMDP model, varying with the number of VMs (K) ($\lambda_p = 7.2$, $\lambda_f = 2.4$, $\mu = 6.6$).

allocate c_1 and c_2 VMs (especially the probability of c_1 VM) exceed the probability to allocate c_3 VM, which explains the reason why the reward of *paid* service of our proposed model falls down when the arrival rate of *paid* service requests exceeds 7 as shown in Figure 3. In a word, our model can achieve higher reward of *paid* service while keeping lower dropping probability of *paid* service requests at the same time comparing with the greedy method, which are shown in Figures 3 and 5, respectively. Thus, our model outperforms the greedy method with the increase of arrival rate of *paid* service requests.

To further illustrate the performance of our model, we compare the reward of *paid* service and the blocking probability with the greedy method under the scenario of different number of VMs (K). In Figure 6, the rewards of both our model and greedy method increase with the increase of the number of total VMs in the service provisioning domain.

When the number of VMs (K) is less than 2, the rewards of both our model and greedy method are negative. This is because the absolute value of rejection cost (-70) is much higher than the net lump rewards of acceptance (8, 14, and 16 for c_1 , c_2 , and c_3 , resp.) in our simulation.

When the number of total VMs in the service provisioning domain is low (1 and 2), the rejection probability of *paid* service requests is as high as 30% as shown in Figures 7 and 8, which results in the negative rewards for both our model and greedy algorithm. We also observed that when K is less than 3, the reward of *paid* service of our model is lower than that of the greedy method.

The reason is that our model does not only consider the instant and future long-term income but also the cost of

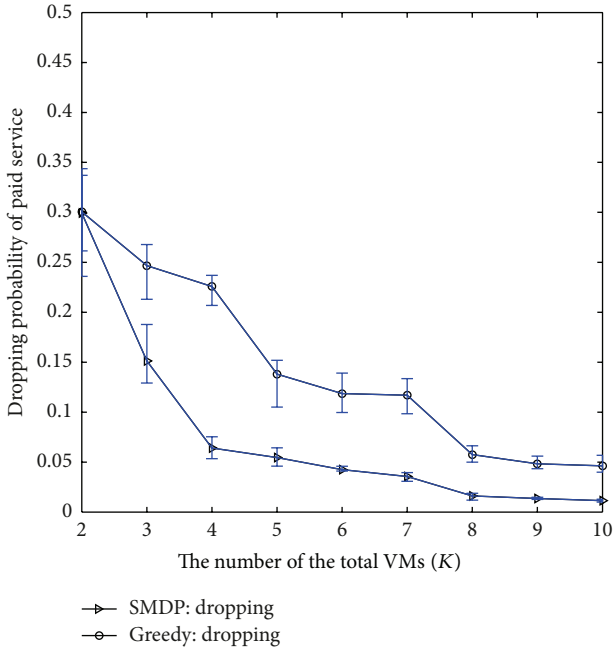


FIGURE 8: Dropping probability of *paid* service compared between SMDP model and greedy method, varying with the number of VMs (K) ($\lambda_p = 7.2$, $\lambda_f = 2.4$, $\mu = 6.6$).

resource occupation of all running services in the service provisioning domain when deciding to allocate the Cloud resources to the *paid* service request, while the greedy method only considers the current income of *paid* service of the service provisioning domain. Then, when the Cloud resource of the service provisioning domain is less than 3 VMs, our model is more conservative than the greedy method to allocate Cloud resources to the *paid* service request.

In Figure 6, we can also see that when the number of VMs (K) is less than 7, the reward of *paid* service of our model increases rapidly with the increase of K , while when K is greater than 7, the reward of *paid* service of our model increases slowly with the increase of K , which implies that when the Cloud resource of the service provisioning domain exceeds the threshold, for the given arrival rate and departure rate, it has limited impact to increase the reward of *paid* service through increasing the Cloud resource of the service provisioning domain. Comparing the rewards of *paid* service between our model and the greedy method in Figure 6, it can be seen that our model outperforms over 50% averagely than the greedy method. Meanwhile, as shown in Figure 8, the dropping probability of *paid* service requests of our model is lower than that of the greedy method over 50% averagely as well, which proves that our model performs better than the greedy method with the increase of the total number of VMs (or Cloud resources) of the service provisioning domain as well.

Figure 9 shows the total rewards (rewards of *paid* service plus *free* service) of different arrival rates of *free* service requests of our proposed model, varying with the increase of arrival rate of *paid* service requests in the service provisioning

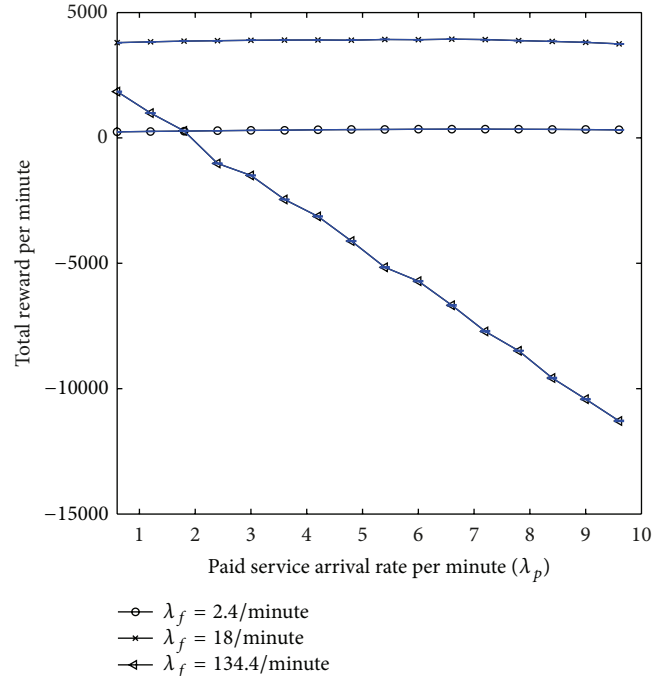


FIGURE 9: Total system reward with different arrival rate of *free* service requests using SMDP model, varying with the arrival rate of *paid* service requests ($\mu = 6.6$, $K = 6$).

domain. It can be seen that when the values of the arrival rates between *paid* service request and *free* service request are comparable, the total reward of our model increases with the increase of arrival rate of *free* service requests. On the other hand, when the arrival rate of *free* service requests is much larger than that of *paid* service requests, the total reward decreases rapidly, which results from the large increase of the arrival rate of *free* service requests which may cause more rejections for the following service requests.

7. Conclusion

In this paper, we propose an SMDP-based model to adaptively allocate Cloud resources in terms of VMs based on requests from mobile users. By considering the benefits and expenses of both Cloud and mobile devices, the proposed model is able to dynamically allocate different numbers of VMs to mobile applications based on the Cloud resource status and system performance, thus to obtain the maximal system rewards and to achieve various QoS levels for mobile users. We further derive the Cloud service blocking probability and the probabilities of different Cloud resource allocation schemes in our proposed model. Simulation results show that the proposed model can achieve a higher system reward and a lower service blocking probability compared with the traditional greedy resource allocation algorithm. In the future, we will study a more complex decision making model with different types of mobile application services, for example, the mobile application services which require different serving priorities. We will also investigate the optimal Cloud resource planning by

determining the minimal Cloud network resources to achieve the maximal system rewards under given QoS constraints.

Acknowledgments

This work was supported in part by the State Key Development Program for Basic Research of China (Grant no. 2011CB302902), the “Strategic Priority Research Program” of the Chinese Academy of Sciences (Grant no. XDA06040100), the National Key Technology R&D Program (Grant no. 2012BAH20B03), US NSF Grants CNS-1029546, and the Office of Naval Research’s (ONR) Young Investigator Program (YIP).

References

- [1] M. Armbrust, A. Fox, R. Griffith et al., “Above the clouds: a berkeley view of cloud computing,” Tech. Rep. UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Calif, USA, 2009.
- [2] M. Walshy, “Gartner: Mobile to outpace desktop web by 2013,” Online Media Daily.
- [3] D. Huang, X. Zhang, M. Kang, and J. Luo, “Mobicloud: a secure mobile cloud frame-work for pervasive mobile computing and communication,” in *Proceedings of 5th IEEE International Symposium on Service-Oriented System Engineering*, 2010.
- [4] X. H. Li, H. Zhang, and Y. F. Zhang, “Deploying mobile computation in cloud service,” in *Proceedings of the 1st International Conference for Cloud Computing (CloudCom '09)*, p. 301, 2009.
- [5] B. Chun and P. Maniatis, “Augmented smartphone applications through clone cloud execution,” in *Proceedings of the 12th USENIX HotoS*, 2009.
- [6] X. Zhang, J. Schiffman, S. Gibbs, A. Kunjithapatham, and S. Jeong, “Securing elastic applications on mobile devices for cloud computing,” in *Proceedings of the ACM workshop on Cloud Computing Security*, pp. 127–134, 2009.
- [7] X. Meng, V. Pappas, and L. Zhang, “Improving the scalability of data center networks with traffic-aware virtual machine placement,” in *Proceedings of the IEEE INFOCOM*, San Diego, Calif, USA, March 2010.
- [8] L. X. Cai, L. Cai, X. Shen, and J. W. Mark, “Resource management and QoS provisioning for IPTV over mmWave-based WPANs with directional antenna,” *ACM Mobile Networks and Applications*, vol. 14, no. 2, pp. 210–219, 2009.
- [9] H. T. Cheng and W. Zhuang, “Novel packet-level resource allocation with effective QoS provisioning for wireless mesh networks,” *IEEE Transactions on Wireless Communications*, vol. 8, no. 2, pp. 694–700, 2009.
- [10] L. X. Cai, X. Shen, and J. W. Mark, “Efficient MAC protocol for ultra-wideband networks,” *IEEE Communications Magazine*, vol. 47, no. 6, pp. 179–185, 2009.
- [11] H. Liang, D. Huang, and D. Peng, “On economic mobile cloud computing model,” in *Proceedings of the International Workshop on Mobile Computing and Clouds (MobiCloud '10)*, 2010.
- [12] G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong, “A game-theoretic method of fair resource allocation for cloud computing services,” *The Journal of Supercomputing*, vol. 54, no. 2, pp. 252–269, 2009.
- [13] K. Lorincz, B. R. Chen, J. Waterman, G. Werner-Allen, and M. Welsh, “Resource aware programming in the pixie os,” in *Proceedings of the SenSys*, Raleigh, NC, USA, November 2008.
- [14] K. Lorincz, B. Chen, J. Waterman, G. Werner-Allen, and M. Welsh, “A stratified approach for supporting high throughput event processing applications,” in *Proceedings of the DEBS*, Nashville, Tenn, USA, July 2009.
- [15] G. Tesauro, N. K. Jong, R. Das, and M. N. Bennani, “A hybrid reinforcement learning approach to autonomic resource allocation,” in *Proceedings of the ICAC*, Dublin, Ireland, June 2006.
- [16] K. Bloor, R. Chirkova, Y. Viniotis, and T. Salo, “Dynamic request allocation and scheduling for context aware applications subject to a percentile response time sla in a distributed cloud,” in *Proceedings of the 2nd IEEE International Conference on Cloud Computing Technology and Science*, Indianapolis, Ind, USA, November 2010.
- [17] R. Ramjee, D. Towsley, and R. Nagarajan, “On optimal call admission control in cellular networks,” *Wireless Networks*, vol. 3, no. 1, pp. 29–41, 1997.
- [18] S. O. H. Mine and M. L. Puterman, *Markovian Decision Process*, Elsevier, Amsterdam, The Netherlands, 1970.
- [19] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley & Sons, New York, NY, USA, 2005.
- [20] MathWorks, “Matlab,” <http://www.mathworks.com/>.

