On Economic Mobile Cloud Computing Model

Hongbin Liang[1,3], Dijiang Huang[2], and Daiyuan Peng[1]

[1] School of Information Science and Technology, Southwest Jiaotong University
[2] School of Computing Informatics and Decision Systems Engineering,
Arizona State University
[3] Department of Electrical and Computer Engineering, University of Waterloo

Abstract. Cloud has become a promising service model for mobile devices. Using cloud services, mobile devices can outsource its computationally intensive operations to the cloud, such as searching, data mining, and multimedia processing. In this service computing model, how to build an economic service provisioning scheme is critical for mobile cloud service providers. Particularly when the mobile cloud resource is restricted. In this paper, we present an economic mobile cloud computing model using Semi-Markov Decision Process for mobile cloud resource allocation. Our model takes the considerations the cloud computing capacity, the overall cloud system gain, and expenses of mobile users using cloud services. Based on the best of our knowledge, our presented model is the first to address the economic service provisioning for mobile cloud services. In the performance evaluation, we showed that the presented economic mobile cloud computing model can produce the optimal system gain with a given cloud service inter-domain transfer probability.

Keywords: Mobile Cloud Computing, Semi-Markov Decision Process.

1 Introduction

With the development of wireless access technologies such as 3/4G, LTE, and WiMax, mobile devices can gain access to the network core over longer distances and larger bandwidths. This allows for very effective communication between mobile devices and the cloud infrastructure. A new service architecture is necessary to address the requirements of users in their unique operational environment and create new mobile applications. Cloud computing is a new business model focusing on resource-on-demand, pay-as-you-go, and utility-computing [1]. Cloud computing can be broadly classified as infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), and software-as-a-service (SaaS). Critical research issues for cloud computing such as computation offloading, remote execution, and dynamic composition have been extensively discussed in previous literature.

Recent research have been focused on cloud computing for mobile devices [4,6, 9]. Cloud computing for mobile devices has a major benefit in that it enables running applications between resource-constrained devices and Internet-based Clouds. Moreover, resource-constrained devices can outsource computation/communication/resource intensive operations to the cloud. CloneCloud [2]

M. Griss and G. Yang (Eds.): MOBICASE 2010, LNICST 76, pp. 329–341, 2012.

focuses on execution augmentation with less consideration on user preference or device status. Samsung has proposed the concept of elastic applications, which can offload components of applications from mobile devices to cloud [10]. We generalize mobile cloud services based on the MobiCloud computing model presented in [3], which is shown in Figure 1. Mobile cloud uses weblets (application components) to link the cloud services and 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. However, its execution location can be run on a mobile device or migrated to the cloud, i.e., run on one or more virtual nodes offered by an IaaS provider. In this way, an elastic application can dynamically augment the capabilities of a mobile device, including computation power, storage, and network bandwidth, based on the devices' status with respect to CPU load, battery power level, network connection quality, security, etc. One or more weblets are running in the Weblet Container (WC).

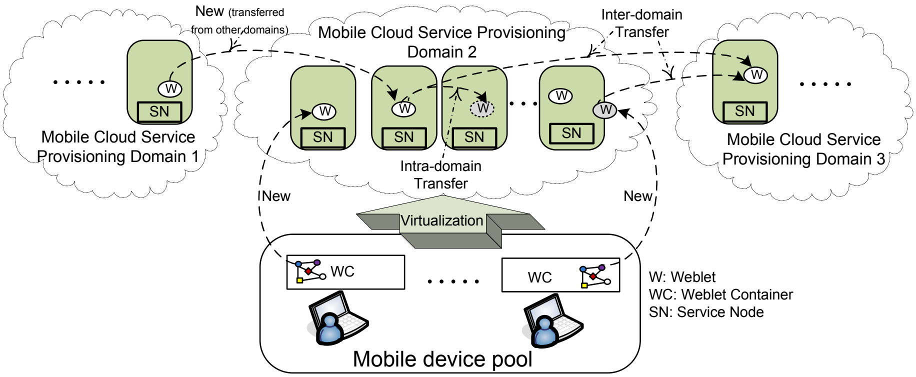
New (transferred from other domains)
Mobile Cloud Service Provisioning Domain 2
Inter-domain Transfer
.....
W
SN
Mobile Cloud Service Provisioning Domain 1
Intra-domain Transfer
Mobile Cloud Service Provisioning Domain 3
New
New
Virtualization
WC
WC
Mobile device pool
W: Weblet
WC: Weblet Container
SN: Service Node

Fig. 1. Reference Model of Mobile Cloud Computing

In the cloud, a service node (SN) is responsible for managing the weblet's loading and unloading in the virtual image. An SN can only handle one weblet from either a new migrated weblet request or a transferred weblet request. Each virtual image has a capacity to hold one weblet at a time. The SN can handle three types of service requests: (i) *New*: a new weblet migration request received from a mobile device or transferred from other mobile cloud service provisioning domains, (ii) *Intra-domain transfer*: an existing weblet transferred from one SN to another within the same mobile cloud service provisioning domain, and (iii) *Inter-domain transfer*: a weblet transferred from current mobile cloud service provisioning domain to another one.

In our presented mobile cloud service model, the cloud can provide a large numbers of virtual images (one virtual image is associated with one CPU, and the CPU can only handle one weblet at a time), however, in reality, the number of virtual images is restricted by the capacity of the cloud hardware configuration. The inter-domain weblet transfer (the third type of service request) means the lose of revenue for the current mobile service provider. As a result, an economic mobile computing model is desired to maximally utilize the cloud resource and achieve the maximum benefit (economic gain) at the same time. The presented economic mobile computing model consists three types of weblet migrations. We differentiate these migrations based on their economic gains, in which a intra-domain weblet transfer migration from another SN usually generates higher economic gain than a new weblet migration from the mobile device or another mobile cloud service provisioning domain, and the inter-domain transfer migration means the lose of revenue. Besides the economic gain, the presented economic mobile computing model also needs to consider the cost due to CPU (or virtual image) occupation. Moreover, the model also needs to consider the trade-offs of the battery consumptions of mobile devices vs. the expenses of using cloud services. Thus, the total economic gain is determined by a comprehensive approach taking all the above mentioned considerations.

In this paper, we present an economic mobile computing model based on Semi-Markov Decision Process (SMDP) model. The contributions of our solutions are in three-fold:

- We firstly apply the Semi-Markov Decision Process to derive the optimal resource allocation policy for mobile cloud computing.
- Our model can take into the considerations both maximizing the system gain of the cloud and reducing the expenses for mobile users.
- Finally, our model can be used to achieve the maximum system gain with a given inter-domain transfer probability constraint.

The rest of this paper is arranged as follows: In Section 2, we present basic system models. In Section 3, the Semi-Markov Decision Process model for mobile cloud computing is presented. We present the inter-domain transfer probability in Section 4. The performance evaluation is presented in Section 5. Finally, we conclude our work in Section 6.

2 System Description

We consider that a mobile cloud consists of two types of nodes, virtual SNs and physical mobile devices (MDs). An MD is a wireless node with limited computing capability and energy supply. An MD can migrate its mobile codes (i.e., weblet) to the cloud. When the cloud receives a migration request, it will decide: wether the SN should accept the request or perform inter-domain transfer based on the consideration if the overall system gain of acceptance.

In the following, we present the system assumptions and states, and the reward model for the mobile cloud computing system.

2.1 System Assumptions

We assume that a service running at an MD or an SN in the cloud costs differently. For simplicity, we also assume the CPU in the cloud is single thread, thus, each weblet in processing occupies one CPU. There are K CPUs in the cloud system. We reserve $K-L$ ($L<K$) CPUs for the intra-domain transfer to ensure that the weblet transfers mostly occur within a mobile cloud service provisioning domain. The distribution for a new weblet migration and an intra-domain transfer weblet migration follows the Poisson distribution with mean rate λ_n and λ_t, respectively. The CPU occupation time of a new weblet and that of an intra-domain transfer weblet in an SN follow exponential distribution with mean rate μ_n and μ_t, respectively.

2.2 System States

The system states can be described based on the service events (including both arrival and leave events) and the service load. In mobile cloud computing system model, we can define three service events: 1) a new weblet request arrives from an MD or another mobile cloud service provisioning domain, denoted by A_n; 2) an intra-domain transfer weblet request arrives from one SN to another within the same mobile cloud service provisioning domain, denoted by A_t; and 3) a weblet leaves current mobile cloud domain, denoted by F. The service load can be represented as the numbers of new weblets and intra-domain transfer weblets in the mobile cloud, which are denoted as s_n and s_t, respectively. Therefore, the system state can be expressed as

$$S=\{\hat{s}|\hat{s}=(s_n,s_t,e)\},$$

where $0\leq s_n+s_t\leq K$, $0\leq s_n\leq L$, $L<K$, K is the number of CPUs, $K-L$ is the number of reserved CPUs for the intra-domain transfer weblet migration. Here, L is the maximal number of CPUs for the new weblet migration and $e\in\{A_n,A_t,F\}$.

2.3 Reward Model

For a system state with an incoming weblet migration service request (i.e., A_n or A_t), two actions can be adopted by the mobile cloud: *accept* or *transfer* (without speciall notice, the "transfer" means inter-domain transfer in the rest of this paper). We denote the action to accept the request as $a_{<s,e>}=1$ and the action to transfer the request as $a_{<s,e>}=0$, where $s=(s_n,s_t)$ and $e\in\{A_n,A_t\}$. On the other hand, for a system state with a weblet leave, there is no action to be performed and we define the action as $a_{<s,F>}=0$. Then, the action space is defined as $Act_{\hat{s}}$, where

$$Act_{\hat{s}}=\begin{cases}0 \text{ (no action)}, & e=F\\ 0 \text{ (transfer)}, & e\in\{A_n,A_t\}\\ 1 \text{ (accept)}, & e\in\{A_n,A_t\}.\end{cases} \tag{1}$$

We also simplify the action as a, where $a\in Act_{\hat{s}}$.

Based on the system state and its corresponding action, one can evaluate the reward to the cloud, which is computed based on the income and the cost as follows:

$$r_{<s,e>} = w_{<s,e>} + g_{<s,e>},\ e \subseteq \{A_n, A_t, F\}, \tag{2}$$

where $w_{<s,e>}$ is the net lump sum income for the cloud and a mobile device, and it is computed as:

$$w_{<s,e>} = \begin{cases} 0, & a_{<s,e>} = 0,\ e \in \{A_n, A_t, F\} \\ (\alpha_s - \alpha_d)\, E_n + \gamma_d U_d, & a_{<s,A_n>} = 1 \\ (\alpha_s - \alpha_d)\, E_t + \gamma_d U_d, & a_{<s,A_t>} = 1. \end{cases} \tag{3}$$

Here, α_s and α_d are weight factors for cloud and mobile device, respectively. They satisfy $0 \leq \alpha_s, \alpha_d \leq 1$ and $\alpha_s + \alpha_d = 1$. E_n and E_t are the incomes of the cloud when it accepts a new weblet migration request from an MD, or an intra-domain transfer weblet migration request from a different SN. Here, U_d represents the income measured by the saved battery energy for the MD when the cloud accepts the weblet migration. γ_d is the weight factor that satisfies $0 \leq \gamma_d \leq 1$.

In (2), $g_{<s,e>}$ denotes the system cost and it is given by:

$$g_{<s,e>} = \tau_{<s,e>} o_{<s,e>},\ a_{<s,e>} \in Act_{\hat{s}}. \tag{4}$$

In (4), $\tau_{<s,e>}$ is the average service time when the system state transfers from $< s, e >$ to the next potential state; $o_{<s,e>}$ is the cost rate of the service time, and it is defined as

$$o_{<s,e>} = \begin{cases} -f(s_n, s_t), & a_{<s,e>} = 0,\ e \in \{A_n, A_t, F\} \\ -f(s_n + 1, s_t), & a_{<s,A_n>} = 1 \\ -f(s_n, s_t + 1), & a_{<s,A_t>} = 1, \end{cases} \tag{5}$$

where $f(\cdot)$ is a linear function of s_n and s_t.

3 SMDP Based Mobile Computing Model

SMDP known as stochastic dynamic programming can be used to model and solve dynamic decision making problems. The SMDP model has the following elements: *system states, action sets, the events cause the decision, decision epochs, transition probabilities,* and *rewards.* We use standard notations and definitions as defined in [7] for our SMDP-based problem formulation.

Based on the SMDP model, to obtain the maximum long term reward, we need to calculate the transition probabilities between each system state. There are only three events in the cloud (i.e., a new weblet migration request arrival, an intra-domain transfer weblet migration request arrival, and a weblet leave). The next decision epoch occurs when any of the events takes place. T_{A_n} and T_{A_t} denote the time intervals from current state to the next weblet migration event, and T_F denotes the time interval from current state to the next weblet leave event. Then, the next decision epoch T satisfies $T = \min(T_{A_n}, T_{A_t}, T_F)$. T_{A_n}, T_{A_t}, and T_F follow exponential distributions with rate λ_n, λ_t, and $(s_n\mu_n + s_t\mu_t)$, respectively. Thus, T follows exponential distribution with rate $\lambda_n + \lambda_t + s_n\mu_n + s_t\mu_t$. Then, the expected time between current state and a new state can be expressed as:

$$\tau(\hat{s}, a) = \begin{cases} [s_n\mu_n + s_t\mu_t + \lambda_n + \lambda_t + a_{<s,A_n>}\mu_n]^{-1}, & e = A_n \\ [s_n\mu_n + s_t\mu_t + \lambda_n + \lambda_t + a_{<s,A_t>}\mu_t]^{-1}, & e = A_t \\ [s_n\mu_n + s_t\mu_t + \lambda_n + \lambda_t]^{-1}, & e = F. \end{cases} \tag{6}$$

$q(j|\hat{s}, a)$ denotes the state transition probability from the current state $\hat{s}$ to the next state j when action a is chosen. For a states $\hat{s} =< s, e >$ $(e \in \{A_n, A_t, F\})$ with action $a = 0$, $q(j|\hat{s}, a)$ can be obtained as follow:

$$q(j|\hat{s}, a) = \begin{cases} \lambda_n\tau(\hat{s}, a), & j =< s_n, s_t, A_n >, s_n \geq 0, s_t \geq 0 \\ \lambda_t\tau(\hat{s}, a), & j =< s_n, s_t, A_t >, s_n \geq 0, s_t \geq 0 \\ s_n\mu_n\tau(\hat{s}, a), & j =< s_n - 1, s_t, F >, s_n \geq 1, s_t \geq 0 \\ s_t\mu_t\tau(\hat{s}, a), & j =< s_n, s_t - 1, F >, s_n \geq 0, s_t \geq 1. \end{cases} \tag{7}$$

where $0 \leq s_n + s_t \leq K$, $0 \leq s_n \leq L$.

For a states $\hat{s} =< s_n, s_t, e >$ $(e \in \{A_n, A_t\})$ with action $a = 1$, $q(j|\hat{s}, a)$ can be obtained as follow:

$$q(j|\hat{s}, a) = \begin{cases} \lambda_n\tau(\hat{s}, a), & j =< s_n + 1, s_t, A_n >, s_n \geq 0, s_t \geq 0 \\ \lambda_t\tau(\hat{s}, a), & j =< s_n, s_t + 1, A_t >, s_n \geq 0, s_t \geq 0 \\ (s_n + 1)\mu_n\tau(\hat{s}, a), & j =< s_n - 1, s_t, F >, s_n \geq 1, s_t \geq 0 \\ (s_t + 1)\mu_t\tau(\hat{s}, a), & j =< s_n, s_t - 1, F >, s_n \geq 0, s_t \geq 1. \end{cases} \tag{8}$$

where $0 \leq s_n + s_t \leq K$, $0 \leq s_n \leq L$.

Figure 2 shows the state transition probabilities when there exists only one type of weblet migrations in the mobile cloud.

Since the time between two decision epochs can be regarded as exponentially distributed and the expected time between two decision epochs is $\tau(\hat{s}, a)$. Then the distribution of the time between two decision epochs is given as:

$$F(\bar{t}|\hat{s}, a) = 1 - e^{-\tau(\hat{s},a)^{-1}\bar{t}}, \bar{t} \geq 0. \tag{9}$$

Then we have

$$Q(\bar{t}, j|\hat{s}, a) = q(j|\hat{s}, a)F(\bar{t}|\hat{s}, a), \tag{10}$$

where (10) denotes if at a decision epoch the system occupies state $\hat{s} \in S$, after the cloud chooses an action a from the set of $Act_{\hat{s}}$ at state $\hat{s}$. The next decision epoch occurs at or before time $\bar{t}$, and the system state at that decision epoch equals j with probability $Q(\bar{t}, j|\hat{s}, a)$. We use $Q(d\bar{t}, j|\hat{s}, a)$ and $F(d\bar{t}|\hat{s}, a)$to represent the time-differential.

Then we can get the reward of the system when an event (arrival or leave) occurs. To incorporate the action into the notations, we let $r(\hat{s}, a)$ denote $r_{<s,e>}$, $h(\hat{s}, a)$ denote $h_{<s,e>}$, and $o(\hat{s}, a)$ denote $o_{<s,e>}$. As the system state does not

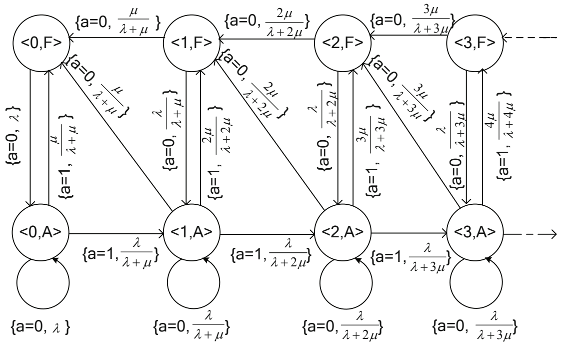
<0,F>
<1,F>
<2,F>
<3,F>
<0,A>
<1,A>
<2,A>
<3,A>
{a=0, $\frac{\mu}{\lambda+\mu}$ }
{a=0, $\frac{2\mu}{\lambda+2\mu}$}
{a=0, $\frac{3\mu}{\lambda+3\mu}$}
{a=0, λ}
{a=1, $\frac{\mu}{\lambda+\mu}$}
{a=0, $\frac{\mu}{\lambda+\mu}$}
{a=0, $\frac{\lambda}{\lambda+\mu}$}
{a=1, $\frac{2\mu}{\lambda+2\mu}$}
{a=0, $\frac{2\mu}{\lambda+2\mu}$}
{a=0, $\frac{\lambda}{\lambda+2\mu}$}
{a=1, $\frac{3\mu}{\lambda+3\mu}$}
{a=0, $\frac{3\mu}{\lambda+3\mu}$}
{a=0, $\frac{\lambda}{\lambda+3\mu}$}
{a=1, $\frac{4\mu}{\lambda+4\mu}$}
{a=1, $\frac{\lambda}{\lambda+\mu}$}
{a=1, $\frac{\lambda}{\lambda+2\mu}$}
{a=1, $\frac{\lambda}{\lambda+3\mu}$}
{a=0, λ }
{a=0, $\frac{\lambda}{\lambda+\mu}$}
{a=0, $\frac{\lambda}{\lambda+2\mu}$}
{a=0, $\frac{\lambda}{\lambda+3\mu}$}

Fig. 2. An example of state transition probabilities for only one type of weblet migrations. The first item in the brackets is the action and the second item in the brackets is the state transition probability.

change between two decision epochs, the expected discounted reward during $\tau(\hat{s}, a)$ satisfies:

$$r(\hat{s}, a) = h(\hat{s}, a) + o(\hat{s}, a)E^a_{\hat{s}}\left\{\int_0^{\tau_1} e^{-\alpha\bar{t}}d\bar{t}\right\} = h(\hat{s}, a) + o(\hat{s}, a)E^a_{\hat{s}}\left\{\frac{[1-e^{-\alpha\tau_1}]}{\alpha}\right\}$$
$$= h(\hat{s}, a) + \frac{o(\hat{s}, a)\tau(\hat{s}, a)}{1+\alpha\tau(\hat{s}, a)}, \tag{11}$$

where α is the discounted rate.

Let D denote the class of deterministic Markovian decision rules and d denote each deterministic Markovian decision rule. There exists a stationary deterministic optimal policy denoted by d^∞. At the current decision epoch, state $\hat{s}$ is occupied and the cloud makes the decision to choose action $d(\hat{s}) \in Act_{\hat{s}}$ under the deterministic Markovian decision rule d. Then, when the system occupies state j at the next decision epoch, for each d, $q(j|\hat{s}, d(\hat{s}))$, $r(j|\hat{s}, d(\hat{s}))$, and $\tau(\hat{s}, d(\hat{s}))$ denote the state transition probability, reward, and expected occupation time between two states, respectively, in which they are defined as follows:

$$q_d(j|\hat{s}) = q(j|\hat{s}, d(\hat{s})); \;\; r_d(j|\hat{s}) = r(j|\hat{s}, d(\hat{s})); \;\; \tau_d(\hat{s}) = \tau(\hat{s}, d(\hat{s})).$$

Thus, under the decision rule d, we define the distribution of the time between two decision epochs as $F_d(\bar{t}|\hat{s}) = F(\bar{t}|\hat{s}, d(\hat{s}))$, and then we can rewrite (10) as:

$$Q_d(\bar{t}, j|\hat{s}) = Q(\bar{t}, j|\hat{s}, d(\hat{s})).$$

The expected infinite-horizon discrete-time discounted reward is

$$\nu_\alpha^{d^\infty}(\hat{s}) = r_d(\hat{s}) + \sum_{j\ \in S}\int_0^\infty e^{\,-\,\alpha\bar{t}}Q_d(d\bar{t}, j|\hat{s})\ \nu_\alpha^{d^\infty}(j), \tag{12}$$

where $Q_d(d\bar{t}, j|\hat{s}) = q_d(j|\hat{s})F_d(d\bar{t}|\hat{s})$ is derived from (10).

According to (9), the long-term reward (12) can be simplified as:

$$\begin{aligned}\nu_\alpha^{d^\infty}(\hat{s}) &= r_d(\hat{s}) + \sum_{j\in S}\left[\int_0^\infty \tau_d(\hat{s})^{-1} e^{-[\alpha+\tau_d(\hat{s})^{-1}]\bar{t}} d\bar{t}\right] q_d(j|\hat{s})\nu_\alpha^{d^\infty}(j) \\ &= r_d(\hat{s}) + \tfrac{1}{1+\tau_d(\hat{s})\alpha}\sum_{j\in S} q_d(j|\hat{s})\nu_\alpha^{d^\infty}(j).\end{aligned} \tag{13}$$

To simplify the calculation, we assume that $\tau_d(\hat{s})^{-1}$ is a constant, and $\tau_d(\hat{s})^{-1} = k$ for all $\hat{s} \in S$. Then the equation (13) can be rewritten as:

$$\nu_\alpha^{d^\infty}(\hat{s}) = r_d(\hat{s}) + \lambda \sum_{j\ \in S} q_d(j|\hat{s})\nu_\alpha^{d^\infty}(j), \tag{14}$$

where $\lambda = \frac{k}{k+\alpha}$. Thus the optimal reward has the discrete-time discounted evaluation equation as:

$$\nu(\hat{s}) = \max_{a\in Act_{\hat{s}}}\left\{ r(\hat{s}, a) + \lambda \sum_{j\in S} q(j|\hat{s}, a)\nu(j) \right\}. \tag{15}$$

Since the system cost $g_{<s,e>}$ is a continuous-time Markov decision process with constant transition rate k, it can be uniformized so that the results and algorithms for discrete-time discounted models can be used directly. We define an uniformization of the continuous-time Markov decision process with components denoted by "~". Let $\tilde{S} = S$, $\tilde{A}ct_{\hat{s}} = Act_{\hat{s}}$, $\tilde{Q}_d$ denote the matrix with components $q_d(j|\hat{s})$ for all $\hat{s} \in \tilde{S}$. We use the same assumption given by [7], where

$$[1 - q(\hat{s}|\hat{s}, a)]\tau(\hat{s}, a)^{-1} \le \tilde{k}. \tag{16}$$

Based on this assumption, we define a constant $\tilde{k} = \lambda_n + \lambda_t + K * \max(\mu_n, \mu_t) < \infty$ satisfying any $\hat{s} \in S$. The uniformization maximum $v(\hat{s})$ of optimal rule d can be obtained as:

$$\nu(\hat{s}) = \max_{a\in \tilde{A}ct_{\hat{s}}}\left\{ \tilde{r}(\hat{s}, a) + \lambda \sum_{j\in S} \tilde{q}(j|\hat{s}, a)\nu(j) \right\}. \tag{17}$$

where $\lambda = \frac{\tilde{k}}{\tilde{k}+\alpha}$, $\tilde{r}(\hat{s}, a) \equiv r(\hat{s}, a)\frac{1+\alpha\tau(\hat{s},a)}{(\alpha+\tilde{k})\tau(\hat{s},a)}$, and

$$\tilde{q}(j|\hat{s}, a) = \begin{cases} 1 - \frac{[1-q(\hat{s}|\hat{s},a)]}{\tau(\hat{s},a)\tilde{k}}, j = \hat{s} \\ \frac{q(j|\hat{s},a)}{\tau(\hat{s},a)\tilde{k}}, \qquad j \neq \hat{s}. \end{cases} \tag{18}$$

Since the state space and action space is limited, then the maximum of equation (17) exists for all $\nu \in V$. In [7], the author proved that if the maximum of (17) is obtained for each $\nu \in V$, then there exists a stationary deterministic optimal policy d^*. Thus, we have

$$d^* \in \arg\max_{d\in D}\left\{ \tilde{r}_d + \lambda \tilde{Q}_d \nu(\hat{s}) \right\}, \tag{19}$$

which means that $(d^*)^\infty$ is optimal. To obtain the maximum $\nu(\hat{s})$ and optimal d^*, we can use Value Iteration Algorithm that is described in [7].

4 Inter-domain Transfer Probability

One of an important QoS metrics of the cloud system is the inter-domain transfer probability for end users. This is because the inter-domain service transfer may cause service disruptions or incur longer service delay. In this section, we discuss and attain the inter-domain probability based on the presented SMDP-based mobile cloud computing model.

From (17), the expected total discounted reward $\nu(\hat{s})$ at state $\hat{s} \in S$ is only related with $\lambda_n, \lambda_t, \mu_n, \mu_t$ and K. For a state of weblet leave, there is no action (i.e., $a = 0$). Therefore, we only need to consider the state with weblet migration arrivals. If $\lambda_n, \lambda_t, \mu_n, \mu_t$ and K are fixed, then $\nu(\hat{s})$ is also fixed at state $< s_n, s_t, A >, A \in \{A_n, A_t\}$. Moreover, the action $a \in \{0, 1\}$ at state $< s_n, s_t, A >, A \in \{A_n, A_t\}$ is fixed, i.e., accept or transfer (i.e., inter-domain service transfer). From the system point of view, the purpose to accept or transfer a weblet migration request is to achieve higher long-term rewards at state $< s_n, s_t, A >, A \in \{A_n, A_t\}$. Let $\pi_{<s_n,s_t,e>}, e \in \{A_n, A_t, F\}$ denote the steady-state probability of state $< s_n, s_t, e >, e \in \{A_n, A_t, F\}$, $\pi_{<s_n,s_t,A>}, A \in \{A_n, A_t\}$ denote arrival steady-state probability of state $< s_n, s_t, A >, A \in \{A_n, A_t\}$. From [8], we can simply use $P_{inter-transfer} = P^n_{inter-transfer} + P^t_{inter-transfer}$ as the inter-domain transfer probability for the entire system, where $P^n_{inter-transfer}$ and $P^t_{inter-transfer}$ are inter-domain transfer probabilities for new weblet migration requests and intra-domain transfer requests, respectively. The entire system inter-domain transfer probability $P_{inter-transfer}$ is a ratio of all inter-domain transferred weblets migration requests to all arrived weblets migration requests, which is defined as:

$$P_{inter-transfer} = \frac{\sum_{s_n = 0}^{N} \sum_{s_t = 0}^{H} \left((1-a_{<s_n,s_t,A_n>})\pi_{<s_n,s_t,A_n>} + (1-a_{<s_n,s_t,A_t>})\pi_{<s_n,s_t,A_t>}\right)}{\sum_{s_n = 0}^{N} \sum_{s_t = 0}^{H} \left(\pi_{<s_n,s_t,A_n>} + \pi_{<s_n,s_t,A_t>}\right)}, \quad 0 \le N+H \le K, 0 \le N \le L \tag{20}$$

where $a_{<s_n,s_t,A_n>} \in \tilde{Act}_{\hat{s}}$ is the action adopted at state $< s_n, s_t, A_n >$ and $a_{<s_n,s_t,A_t>} \in \tilde{Act}_{\hat{s}}$ is the action adopted at state $< s_n, s_t, A_t >$.

According to the result of [5], we can derive $\pi_{<s_n,s_t,e>}, e \in \{A_n, A_t, F\}$ as:

$$\begin{aligned} \pi_{<s_n,s_t,A_n>} = {} & \\ & (1 - a_{<s_n,s_t,A_n>})\pi_{<s_n,s_t,A_n>} \frac{\tilde{k}+\lambda_n-\beta}{\tilde{k}} + (1 - a_{<s_n,s_t,A_t>})\pi_{<s_n,s_t,A_t>} \frac{\lambda_n}{\tilde{k}} \\ & + a_{<s_n,s_t,A_n>}\pi_{<s_n,s_t,A_n>} \frac{\tilde{k}-\beta-\mu_n}{\tilde{k}} + \pi_{<s_n,s_t,F>} \frac{\lambda_n}{\tilde{k}} \\ & + a_{<s_n^{\max},s_t,A_n>}\pi_{<s_n^{\max},s_t,A_n>} \frac{\lambda_n}{\tilde{k}} + a_{<s_n,s_t^{\max},A_t>}\pi_{<s_n,s_t^{\max},A_t>} \frac{\lambda_n}{\tilde{k}}, \end{aligned}$$

$$\begin{aligned} \pi_{<s_n,s_t,A_t>} = {} & \\ & (1 - a_{<s_n,s_t,A_n>})\pi_{<s_n,s_t,A_n>} \frac{\lambda_t}{\tilde{k}} + (1 - a_{<s_n,s_t,A_t>})\pi_{<s_n,s_t,A_t>} \frac{\tilde{k}+\lambda_t-\beta}{\tilde{k}} \\ & + a_{<s_n,s_t,A_t>}\pi_{<s_n,s_t,A_t>} \frac{\tilde{k}-\beta-\mu_t}{\tilde{k}} + \pi_{<s_n,s_t,F>} \frac{\lambda_t}{\tilde{k}} \\ & + a_{<s_n^{\max},s_t,A_n>}\pi_{<s_n^{\max},s_t,A_n>} \frac{\lambda_t}{\tilde{k}} + a_{<s_n,s_t^{\max},A_t>}\pi_{<s_n,s_t^{\max},A_t>} \frac{\lambda_t}{\tilde{k}}, \end{aligned}$$

$$\begin{aligned}
\pi_{<s_n,s_t,F>} &= \pi_{<s_n^{\min},s_t,F>}\frac{s_n^{\min}\mu_n}{\tilde{k}} + \pi_{<s_n,s_t^{\min},F>}\frac{s_t^{\min}\mu_t}{\tilde{k}} + \pi_{<s_n,s_t,F>}\frac{\tilde{k}-\beta}{\tilde{k}} \\
&+ (1-a_{<s_n^{\min},s_t,A_n>})\pi_{<s_n^{\min},s_t,A_n>}\frac{s_n^{\min}\mu_n}{\tilde{k}} \\
&+ (1-a_{<s_n,s_t^{\min},A_n>})\pi_{<s_n,s_t^{\min},A_n>}\frac{s_t^{\min}\mu_t}{\tilde{k}} \\
&+ (1-a_{<s_n^{\min},s_t,A_t>})\pi_{<s_n^{\min},s_t,A_t>}\frac{s_n^{\min}\mu_n}{\tilde{k}} \\
&+ (1-a_{<s_n,s_t^{\min},A_t>})\pi_{<s_n,s_t^{\min},A_t>}\frac{s_t^{\min}\mu_t}{\tilde{k}} \\
&+ a_{<s_n,s_t,A_n>}\pi_{<s_n,s_t,A_n>}\frac{s_n^{\min}\mu_n}{\tilde{k}} + a_{<s_n^{\max},s_t^{\min},A_n>}\pi_{<s_n^{\max},s_t^{\min},A_n>}\frac{s_t^{\min}\mu_t}{\tilde{k}} \\
&+ a_{<s_n,s_t,A_t>}\pi_{<s_n,s_t,A_t>}\frac{s_t^{\min}\mu_t}{\tilde{k}} + a_{<s_n^{\min},s_t^{\max},A_t>}\pi_{<s_n^{\min},s_t^{\max},A_t>}\frac{s_n^{\min}\mu_n}{\tilde{k}},
\end{aligned} \tag{21}$$

where $\beta = s_n\mu_n + s_t\mu_t + \lambda_n + \lambda_t$, $0 \le s_n + s_t \le K$ and $0 \le s_n \le L$. To cover the boundary conditions, we define $s_n^{\min} = \min(s_n+1, K-s_t, L)$, $s_t^{\min} = \min(s_t+1, K-s_n)$, $s_n^{\max} = \max(s_n-1, 0)$ and $s_t^{\max} = \max(s_t-1, 0)$.

The summation of the steady-state probability for all states is equal to 1, and thus we have:

$$\sum_{s_n\ =\ 0}^{N}\sum_{s_t\ =\ 0}^{H}\pi_{<s_n,s_t,e>} = 1, e \in \{A_n, A_t, F\},\ \ 0 \le N+H \le K, 0 \le N \le L. \tag{22}$$

Based on Equations (21) and (22), the steady-state occurring probability $\pi_{<s_n,s_t,e>}, e \in \{A_n, A_t, F\}$ can be obtained. Thus, the entire system inter-domain transfer probability $P_{inter-transfer}$ can be attained.

5 Performance Evaluation

The inter-domain transfer probabilities of our presented SMDP-based mobile cloud computing model are compared with that computed by using *Guard occupation model* [8]. We conduct a simulation-based study, in which the comparative results are presented in Figure 3. In this simulation, we set the new weblet migration request arrival rate λ_n as 5, the intra-domain transfer weblet migration request arrival rate λ_t as 2, and the leave rates of both new and intra-domain transfer migration requests (μ_n and μ_t) as 4. We set the maximal number of CPUs for the new weblet migration requests $L = \lfloor 0.8K \rfloor$. Thus, the number of reserved CPUs for the weblet intra-domain transfer migration requests is $K - L$. For each value of K, we run the simulation for 5 times.

In Figure 3, we observe that the inter-domain transfer probabilities of both SMDP-based occupation model and Guard occupation model decrease with the increase of the number of CPUs. Additionally, we can see that the inter-domain transfer probability is only related to the total number of CPUs in the cloud when the arrival rate and leave rate of weblets are fixed. This also confirms our discussion about the inter-domain tranfer probability presented in Section 4.

We also observe that if the number of CPUs is smaller than 10 or larger than 22, the differences of the inter-domain transfer probabilities between the SMDP-based occupation model and Guard occupation model is very small. In addition,

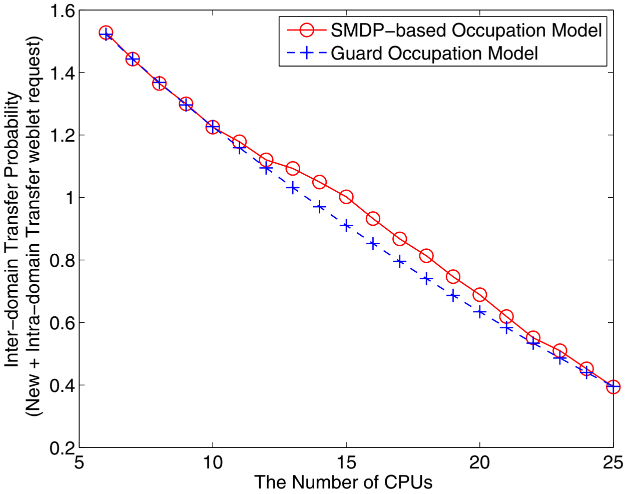

Fig. 3. An example to compare the inter-domain transfer probabilities of using SMDP occupation model and Guard occupation model

if the number of CPUs is between 10 and 22, then, the differences of the inter-domain transfer probabilities are increased. This phenomena can be explained as follows:

– If the number of CPUs is small (i.e., less than 10 in our simulation), then, both SMDP-based occupation model and Guard occupation model cannot accommodate the incoming weblet migration requests for the given simulation setting. As a result, both inter-domain transfer probabilities are high.
– If the number of CPUs is large (i.e., larger than 22 in our simulation), then, SMDP-based occupation model and Guard occupation model have sufficient CPUs to accommodate the coming weblet migration requests for the given simulation setting. Thus, both inter-domain transfer probabilities are low.
– If the number of CPUs is moderate (i.e., between 10 and 22 in our simulation), the inter-domain transfer probability of the SMDP-based occupation model is higher than that computed by using the Guard occupation model. This is because the SMDP-based occupation model focuses more on the maximal system reward that involves the system income and cost, service expenses of MDs, and conservation of energy consumption of MDs. However, the Guard occupation model purely focuses on the reduction of inter-domain transfer rate, which may not be the optimal in terms of system reward.

In general, the increase of the inter-domain transfer probability not only means the decrease of the revenue of a mobile cloud service provider, but also means the disruption of a service. Thus, the system reward should be obtained under a given inter-domain transfer probability to satisfy the desired QoS, i.e., the optimal policy d^* should also consider the restriction enforced by the given inter-domain transfer probability.

If the inter-domain transfer probability is given by P_B, then the system reward (17) can be rewritten as:

$$\nu(\hat{s}) = \max_{P_{inter-transfer} \leq P_B, a \in Act_{\hat{s}}} \left\{ \tilde{r}(\hat{s}, a) + \lambda \sum_{j \in S} \tilde{q}(j|\hat{s}, a)\nu(j) \right\}. \tag{23}$$

The optimal policy (17) can be rewritten as:

$$d^* \in \arg \max_{P_{inter-transfer} \leq P_B, d \in D} \left\{ \tilde{r}_d + \lambda \tilde{Q}_d \nu(\hat{s}) \right\}. \tag{24}$$

6 Conclusion

In this paper, we present an economic mobile cloud computing model based on Semi-Markov Decision Process. In our approach, both the maximal system reward and expenses of mobile devices are considered. We present the inter-domain transfer probability of the SMDP-based mobile cloud computing model using both theoretical approach and simulation comparative studies. Particularly, we derive both the constraint maximal system reward and the optimal decision policy under a given inter-domain transfer probability. In the future, we will incorporate more system metrics into the constructions of the reward function such as different application tasks or security levels based on multi-threads CPUs. Moreover, we will investigate the optimal CPUs allocation issues using the SMDP-based occupation model.

Acknowledgement. Hongbin Liang's research is supported by the China Scholarship Council and Dijiang Huang's research is supported by the Office of Naval Research's Young Investigator Program.

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., et al.: Above the clouds: A berkeley view of cloud computing. EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28 (2009)
2. Chun, B., Maniatis, P.: Augmented Smartphone Applications Through Clone Cloud Execution. In: Proceedings of USENIX HotOS XII (2009)
3. Huang, D., Zhang, X., Kang, M., Luo, J.: Mobicloud: A secure mobile cloud framework for pervasive mobile computing and communication. In: Proceedings of 5th IEEE International Symposium on Service-Oriented System Engineering (2010)
4. Lyons, K., Pering, T., Rosario, B., Sud, S., Want, R.: Multi-display Composition: Supporting Display Sharing for Collocated Mobile Devices. In: Gross, T., Gulliksen, J., Kotzé, P., Oestreicher, L., Palanque, P., Prates, R.O., Winckler, M. (eds.) INTERACT 2009. LNCS, vol. 5726, pp. 758–771. Springer, Heidelberg (2009)

5. Ni, W., Li, W., Alam, M.: Determination of optimal call admission control policy in wireless networks. IEEE Transactions on Wireless Communications 8(2), 1038–1044 (2009)
6. Pering, T., Want, R., Rosario, B., Sud, S., Lyons, K.: Enabling pervasive collaboration with platform composition. In: Proceedings of Perviasive (2009)
7. Puterman, M.: Markov decision processes: Discrete stochastic dynamic programming. John Wiley & Sons, Inc., New York (2005)
8. Ramjee, R., Towsley, D., Nagarajan, R.: On optimal call admission control in cellular networks. Wireless Networks 3(1), 29–41 (1997)
9. Li, X., Zhang, H., Zhang, Y.: Deploying Mobile Computation in Cloud Service. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) Cloud Computing. LNCS, vol. 5931, pp. 301–311. Springer, Heidelberg (2009)
10. Zhang, X., Schiffman, J., Gibbs, S., Kunjithapatham, A., Jeong, S.: Securing elastic applications on mobile devices for cloud computing. In: Proceedings of the 2009 ACM Workshop on Cloud Computing Security, pp. 127–134 (2009)