INFORMATICA, 2015, Vol. 26, No. 1, 89–110

DOI: http://dx.doi.org/10.15388/Informatica.2015.40

An Optimization Approach for Utilizing Cloud Services for Mobile Devices in Cloud Environment

Chunlin LI[1,2]*, Layuan LI[2]

[1]*Department of Computer Science, Wuhan University of Technology, Wuhan 430063, PR China*
[2]*The Key Laboratory of Embedded System and Service Computing, Ministry of Education Tongji University, Shanghai 200092, PR China*
e-mail: chunlin74@aliyun.com, jwtu@public.wh.hb.cn

Received: January 2014; accepted: September 2014

Abstract. Mobile cloud computing has emerged aiming at assisting mobile devices in processing computationally or data intensive tasks using cloud resources. This paper presents an optimization approach for utilizing cloud services for mobile client in mobile cloud, which considers the benefit of both mobile device users and cloud datacenters. The mobile cloud service provisioning optimization is conducted in parallel under the deadline, budget and energy expenditure constraint. Mobile cloud provider runs multiple VMs to execute the jobs for mobile device users, the cloud providers want to maximize the revenue and minimize the electrical cost. The mobile device user gives the suitable payment to the cloud datacenter provider for available cloud resources for optimize the benefit. The paper proposes a distributed optimization algorithm for utilizing cloud services for mobile devices. The experiment is to test convergence of the proposed algorithm and also compare it with other related work. The experiments study the impacts of job arrival rate, deadline and mobility speeds on energy consumption ratio, execution success ratio, resource allocation efficiency and cost. The experiment shows that the proposed algorithm outperforms other related work in terms of some performance metrics such as allocation efficiency.

Key words: mobile cloud, cloud standards, mobile device, energy efficiency.

1. Introduction

Mobile devices are increasingly becoming the most effective and convenient communication tools in our daily life. However, the mobile devices are facing many challenges in their limited resources (e.g., battery life, storage, and bandwidth) and communications (e.g., mobility and security), which significantly impede the improvement of service qualities. The application of cloud computing is not limited only to PC, but also to mobile devices and mobile terminals. With the rapid development of mobile Internet, cloud computing services providing for mobile phones and other mobile terminals have emerged. Currently, mobile devices are becoming the popular instrument for accessing the cloud environment. While roaming around with a mobile handheld device, a user can enjoy

*Corresponding author.

the vast computation power and diverse capabilities with the support of the cloud environment. In mobile cloud environment, users access cloud services with mobile devices. Mobile cloud computing can enhance user experience because mobile devices have limited computing power, storage space, and battery life. Storage and computing capabilities of mobile devices can be extended by utilizing cloud resources (Dinh *et al.*, 2007).

Mobile cloud computing introduces a variety of benefits over traditional mobile services. With cloud computing, mobile applications can transfer some computation task to be executed on the cloud center. Many mobile applications are developed under mobile cloud computing concept including e-commerce, healthcare, and computer games. Cloud computing with mobile devices enables many new applications that were unavailable in the past. For example, a mobile cloud could be made of smart phones to process real-time data (e.g., video and audio feeds, GPS coordinates). Other mobile cloud applications might involve real time control/actuation in wireless powered sensor networks.

Mobile cloud computing has emerged aiming at assisting mobile devices in processing computationally or data intensive tasks using cloud resources. This paper presents an optimization approach for utilizing cloud services for mobile client in mobile cloud, which considers the benefit of both mobile device users and cloud datacenters. The paper proposes a distributed optimization algorithm for utilizing cloud services for mobile devices. The experiment is to test convergence of the proposed algorithm and also compare it with other related work. The experiments study the impacts of job arrival rate, deadline and mobility speeds on energy consumption ratio, execution success ratio, resource allocation efficiency and cost. The experiment shows that the proposed algorithm outperforms other related work in terms of some performance metrics such as resource allocation efficiency.

The rest of the paper is structured as followings. Section 2 discusses the related works. Section 3 presents an optimization approach for utilizing cloud services for mobile client in mobile cloud. Section 4 presents distributed optimization algorithm for utilizing cloud services for mobile devices. In Section 5 the experiments are conducted and discussed. Section 6 gives the conclusions to the paper.

2. Related Works and Motivations

Currently, many researches focused on cloud computing with mobile devices. Mobile cloud computing combines wireless access service and cloud computing to improve the performance of mobile applications. Ferber and Rauber (2012) propose a pipelining task concept on a single encrypted channel between a mobile device and a cloud resource to better use the 3G network connections. Barbera *et al.* (2012) study the mobile software/data backup of both mobile computation offloading and mobile software/data backups in real-life scenarios. Zhang *et al.* (2012) investigate the scheduling policy for collaborative execution in mobile cloud computing. The design objective is to minimize the energy consumed by the mobile device, while meeting a time deadline. They formulate this minimum-energy task scheduling problem as a constrained shortest path problem on a directed acyclic graph, and adapt the canonical "LARAC" algorithm to solving this problem approximately. Rahimi *et al.* (2013) study mobility-aware optimal service allocation in mobile cloud computing. They propose a framework to model mobile applications as location-time workflows (LTW) of tasks, here user mobility patterns are translated to a mobile service usage patterns. Abolfazli *et al.* (2013) propose a market-oriented architecture based on SOA to stimulate publishing, discovering, and hosting services on nearby mobiles, which reduces long WAN latency and creates a business opportunity that encourages mobile owners to embrace service hosting. Group of mobile phones simulate a nearby cloud computing platform. Barbarossa *et al.* (2013) proposes joint allocation of computation and communication resources in multiuser mobile cloud computing. They propose a method to jointly optimize the transmit power, the number of bits per symbol and the CPU cycles assigned to each application in order to minimize the power consumption at the mobile side.

Lee *et al.* (2013) propose a collaborative framework that lets the mobile device participate in the computation of Cloud Computing system by dynamically partitioning the workload across the device and the system. The objective is to mitigate the deterioration of network performance and improve the overall system performance. Park *et al.* (2012) study mobile devices as resources in mobile cloud environments. They propose a resource allocation technique which offers reliable resource allocation considering the availability of mobile resources and movement reliability of mobile resources. Shiraz *et al.* (2014) study runtime partitioning of elastic mobile applications for mobile cloud computing. They investigate the overhead of runtime application partitioning on SMD by analyzing additional resources utilization on SMD in the mechanism of runtime application profiling and partitioning. Ge *et al.* (2012) propose a game-theoretic approach to optimize the overall energy in a mobile cloud computing system. They formulate the energy minimization problem as a congestion game, where each mobile device is a player and the strategy is to select one of the servers to offload the computation while minimizing the overall energy consumption. Wang *et al.* (2013) consider a mobile cloud computing (MCC) interaction system consisting of multiple mobile devices and the cloud computing facilities. They provide a nested two stage game formulation for the MCC interaction system. In the first stage, each mobile device determines the portion of its service requests for remote processing in the cloud. In the second stage, the cloud computing facilities allocate a portion of its total resources for service request processing depending on the request arrival rate from all the mobile devices.

Zhang *et al.* (2013) model the resource allocation process of a mobile cloud computing system as an auction mechanism with premium and discount factors. The premium and discount factors indicate complementary and substitutable relations among cloud resources provided by the service provider. Makris *et al.* (2012) propose a deployment scheme to offload expensive computational tasks from thin, mobile devices to the cloud datacenter, so that the proposed method prolongs battery life for mobile devices; meanwhile provide rich user experiences for such mobile applications. Shiraz *et al.* (2013a) analyzes the impact of VM deployment and management on the execution time of application in mobile cloud computing. They investigate VM deployment and management for application processing in simulation environment by using CloudSim. Hussain and Hussain (2011) discuss and formalize the issue of cloud service selection in general and propose a multi-criteria cloud service selection methodology, they consider cost, pricing policy, performance etc as multiple criteria.

Shiraz *et al.* (2013b) propose distributed application processing frameworks in smart mobile devices for mobile cloud computing. The author reviews existing Distributed Application Processing Frameworks (DAPFs) for smart mobile devices in mobile cloud computing domain. The objective is to highlight issues and challenges to existing DAPFs in developing, implementing, and executing computational intensive mobile applications within mobile cloud. Gkatzikis and Koutsopoulos (2013) advocate the need for novel cloud architectures and migration mechanisms that effectively bring the computing power of the cloud closer to the mobile user. They consider a cloud computing architecture that consists of a back-end cloud and a local cloud, which is attached to wireless access infrastructure. Lu *et al.* (2013) introduce the concept of an Internet-based Virtual Computing Environment (iVCE), which aims to provide Cloud services by a dynamic combination of data centers and other multi-scale computing resources on the Internet. Li *et al.* (2009) propose Armada, an efficient range query processing scheme to support delay-bounded single-attribute and multiple-attribute range queries. Bessis *et al.* (2013) propose a model for supporting the decision making process of the cloud policy for the deployment of virtual machines in cloud environments. They explore two configurations, the static case and the dynamic case to study deployment of virtual machines. Choi *et al.* (2013) study proposes a method of designing a scheme for applying MapReduce of the FP-Growth algorithm.

Our previous work (Li and Li, 2014) focuses on batch processing applications for mobile cloud computing environment. The mobile device's user requirements arrive in batches into the mobile cloud systems. For example, mobile device's users submit batch jobs (e.g., financial analytics, scientific simulations) to mobile cloud system for fast processing. The paper of Li and Li (2014) proposes a phased scheduling for batch processing applications in mobile cloud, which is divided into mobile device's batch application optimization and mobile device's job level optimization. The formulation model and method of this paper is different from Li and Li (2014).

Li and Li (2012) is oriented to SaaS cloud, it combines three perspectives: SaaS user, SaaS provider and cloud resource providers, Li and Li (2012) presents an efficient cloud resource provisioning algorithm, which is beneficial for SaaS users, SaaS provider and cloud resource provider. This paper presents an optimization approach for utilizing cloud services for mobile client in mobile cloud. It focus on mobile cloud service provisioning. The research object and formulation model of this paper is different from Li and Li (2012).

3. An Optimization Approach for Utilizing Cloud Services for Mobile Devices

3.1. *Problem Description*

The proposed mobile cloud system includes mobile devices, cloud data center and mobile cloud proxy. The mobile cloud users run the interactive application on their devices.

Table 1
The description of notations.

Notations	Meanings
C_j^{cpu}	The maximum capacity of CPU of cloud provider j
C_j^{ram}	The maximum capacity of memory of cloud provider j
ed_i^j	Energy dissipation of cloud resource j for provisioning VM for mobile device user i
ec_j	Energy consumption rate of cloud resource j for provisioning VM
v_{ij}^{cpu}	CPU of cloud resource j required by a VM for mobile device user i
v_{ij}^{ram}	Memory of cloud resource j required by a VM for mobile device user i
u_{ij}^{cpu}	Payment of mobile device user i to the cloud resource j for CPU required by a VM
u_{ij}^{ram}	Payment of mobile device user i to the cloud resource j for memory required by a VM
p_j^{cpu}	The price of CPU of cloud resource j
p_j^{ram}	The price of memory of the cloud resource j
B_i	The expense budget of mobile device user i
EB_j	Upper limit of energy expenditure of cloud provider j
pe_j	Electricity price
T_i	Time limits given by mobile device user i to complete all jobs
$ecost_j$	The payment for the electricity usage of cloud provider j
e_i^n	Energy dissipation caused by mobile device user's nth job
er_i^n	Energy consumption rate for the mobile device user i to complete nth job
E_i	Energy limit of mobile device user i

Wireless access points provide radio resource (i.e., bandwidth), while data centers provide computing resources (e.g., CPU, memory, and storage) to support different mobile applications. Mobile device can access the Internet through an access point (AP) that is either Wireless Local Area Network (WLAN) or cellular 3G network. The mobile cloud proxy is responsible for managing all communications between the mobile device and the cloud datacenter. The notations used in sections are listed in Table 1.

Let v_{ij}^{cpu}, v_{ij}^{ram} denote the amount of CPU and memory network required by a VM for mobile device i from cloud provider j. VM can be expressed as $vm = \{v_{ij}^{cpu}, v_{ij}^{ram}\}$. Let $S = \{s_1, s_2, \dots, s_j\}$ denote the set of cloud providers. Each cloud provider supplies a pool of resources to host VM for the mobile device application. The cloud provider supply CPU capacity expressed in MHz, the memory capacity expressed in megabytes. The processing power of a cloud resource provider s_j is measured by the average CPU speed. C_j^{cpu}, C_j^{ram} denote the maximum capacity of CPU and memory, which cloud provider s_j can support VM for the mobile device. $B = (b_1, b_2, \dots, b_i)$ denote the set of mobile devices. The cloud provider provides the VMs for executing mobile device user's applications. A mobile device estimates its energy consumption rate er_l for executing the application set $A = \{A_1, A_2, \dots, A_i\}$, and the energy consumption constraint is C_l.

Energy consumption rate of each mobile device in the mobile cloud is measured by Joule per unit time. Let e_i^n be energy exhausted by mobile device user i's nth job, t_i^n is the execution time of mobile device application i' n-th job on the mobile cloud datacenter.

The energy consumption rate of the cloud datacenter is denoted by er_i^n, which can be written as $e_i^n = er.t_i^n$. Service provisioning in mobile cloud computing is to maximize the mobile cloud utility is subject to the resource constraints of cloud datacenter and QoS constraints of mobile device users. The service provisioning for mobile client in mobile cloud is formulated as the follows:

$$
\begin{aligned}
&MaxU_{MCC} \\
&\text{s.t. } B_i \geqslant \sum_j (u_{ij}^{cpu} + u_{ij}^{ram}), \\
&T_i \geqslant \sum_n t_i^n, \\
&C_j^{cpu} \geqslant \sum_i v_{ij}^{cpu}, \\
&ecost_j \leqslant EB_j, \\
&C_j^{ram} \geqslant \sum_i v_{ij}^{ram}.
\end{aligned} \tag{3.1}
$$

The service provisioning in mobile cloud computing aims to maximize U_{mcc} subject to the constraints. v_{ij}^{cpu} is CPU required by a VM for mobile device user i from the mobile cloud provider j, v_{ij}^{ram} is the memory required by a VM for mobile device user i from the mobile cloud provider j. The constraint implies that the provisioned CPU does not exceed the total capacity C_j^{cpu} of mobile cloud provider j, the provisioned memory units do not exceed the total resource C_j^{ram} of mobile cloud provider j. Also there are some constraints related with mobile device user. Mobile device user should complete all jobs under the deadline and the budget. Mobile device user needs to complete jobs in the deadline, T_{i}, while the cost cannot exceed the budgetB_i. u_{ij}^{cpu}, u_{ij}^{ram} are the payments of the mobile device user i to the cloud datacenter provider j for CPU and memory required by a VM respectively. B_i is the budget of mobile device user i. For the mobile cloud datacenter providers, cloud datacenter providers can't pay for the electricity cost more than EB_j, which is the electricity expenditure of mobile cloud datacenter providers.

$$
\begin{aligned}
MaxU_{MCC} = {} & \left(B_i - \sum_j \left(u_{ij}^{cpu} + u_{ij}^{ram}\right)\right) + \left(E_i - \sum_{n=1}^{N} e_i^n\right) \\
& + \sum \left(u_{ij}^{cpu} \log v_{ij}^{cpu} + u_{ij}^{ram} \log v_{ij}^{ram}\right) \\
& + \left(T_i - \sum_n t_i^n\right) + (EB_j - ecost_j) \\
\text{s.t. } & B_i \geqslant \sum_j \left(u_{ij}^{cpu} + u_{ij}^{ram}\right), \quad C_j^{cpu} \geqslant \sum_i v_{ij}^{cpu}, \quad C_j^{ram} \geqslant \sum_i v_{ij}^{ram}, \\
& T_i \geqslant \sum_n t_i^n, \quad ecost_j \leqslant EB_j
\end{aligned} \tag{3.2}
$$

Mobile cloud utility functions are maximized with specific parameter constraints of mobile cloud provider and mobile device user. $u_{ij}^{cpu} \log v_{ij}^{cpu} + u_{ij}^{ram} \log v_{ij}^{ram}$ is the revenue of cloud provider j obtained from mobile device user i. $(E_i - \sum_{n=1}^{N} e_i^n)$ represents remaining energy of mobile device users. We can apply the Lagrangian method to above problem. Let us consider the Lagrangian form of service provisioning optimization in mobile cloud:

$$\begin{aligned} L \;=\; &(EB_j - ecost_j) + \left(T_i - \sum_n t_i^n\right) + \sum \left(u_{ij}^{cpu} \log v_{ij}^{cpu} + u_{ij}^{ram} \log v_{ij}^{ram}\right) \\ &+ \left(B_i - \sum_j \left(u_{ij}^{cpu} + u_{ij}^{ram}\right)\right) + \left(E_i - \sum_{n=1}^{N} e_i^n\right) + \lambda\left(C_j^{cpu} - \sum_i v_{ij}^{cpu}\right) \\ &+ \beta\left(C_j^{ram} - \sum_i v_{ij}^{ram}\right) + \omega\left(B_i - \sum_j \left(u_{ij}^{cpu} + u_{ij}^{ram}\right)\right) \\ &+ \rho\left(T_i - \sum_n t_i^n\right) + \theta(EB_j - ecost_j). \end{aligned} \tag{3.3}$$

Since the Lagrangian is separable, $MaxU_{MCC}$ can be decomposed into the subproblems P_{md} and P_{cp} which are separately conducted by mobile device user and cloud datacenter provider as follows:

$$\begin{aligned} &MaxP_{cp} = Max \sum (EB_j - ecost_j) + \left(u_{ij}^{cpu} \log v_{ij}^{cpu} + u_{ij}^{ram} \log v_{ij}^{ram}\right) \\ &\text{s.t. } ecost_j \leqslant EB_j, \qquad C_j^{cpu} \geqslant \sum_i v_{ij}^{cpu}, \qquad C_j^{ram} \geqslant \sum_i v_{ij}^{ram}, \end{aligned} \tag{3.4}$$

$$\begin{aligned} &MaxP_{md} = Max \left\{ (B_i - \sum_j \left(u_{ij}^{cpu} + u_{ij}^{ram}\right)\right) + \left(E_i - \sum_{n=1}^{N} e_i^n\right) + \left(T_i - \sum_n t_i^n\right) \right\}, \\ &\text{s.t. } T_i \geqslant \sum_n t_i^v, \qquad B_i \geqslant \sum_j \left(u_{ij}^{cpu} + u_{ij}^{ram}\right). \end{aligned} \tag{3.5}$$

Problem P_{cp} is conducted by the mobile cloud provider, different mobile cloud providers provision optimal resource allocation to run VMs for mobile device users. The objective of mobile cloud providers is to maximize the revenue and minimize electrical cost $ecost_j$ under the constraints of electrical expenditure and physical resource capacity. $EB_j - ecost_j$ represents the surplus of cloud provider which is obtained by electrical expenditure subtracting actual cost for running VMs. The mobile device user pays the cloud datacenter provider for available cloud resources to run VMs. Mobile device user i submits the payment u_{ij}^{cpu}, u_{ij}^{ram} to cloud datacenter provider j for CPU and memory required by VMs. $B_i - \sum_j (u_{ij}^{cpu} + u_{ij}^{ram})$ represents the surpluses of mobile device user's budget. Problem P_{md} is conducted by mobile device users; the mobile device user gives the unique optimal payment to cloud datacenter provider under the deadline constraint to maximize the mobile device user's QoS satisfaction. $T_i - \sum_n t_i^n$ represents the deadline subtracting actual processing time. $(E_i - \sum_{n=1}^{N} e_i^n)$ is remaining energy of mobile device users. So, the objective of problem P_{md} is to save the energy and the cost, and complete task for mobile device user as soon as possible.

3.2. *Solution for Mobile Cloud Optimization Problem*

In mobile device user optimization problem, the mobile device user gives the unique optimal payment to the cloud datacenter provider under the deadline, budget and energy constraint to maximize the mobile device user's satisfaction. Mobile device user optimization problem is written as follows.

$$MaxP_{md} = Max\left\{\left(B_i - \sum_j \left(u_{ij}^{cpu} + u_{ij}^{ram}\right)\right) + \left(E_i - \sum_{n=1}^{N} e_i^n\right) + \left(T_i - \sum_n t_i^n\right)\right\},$$

$$\text{s.t. } T_i \geqslant \sum_n t_i^n, \qquad B_i \geqslant \sum_j \left(u_{ij}^{cpu} + u_{ij}^{ram}\right).$$

$u^{cpu} = [u_1^{cpu}, \ldots, u_j^{cpu}]$ represents all payments of mobile device users for cloud datacenter provider j's CPU, $u^{ram} = [u_1^{ram}, \ldots, u_j^{ram}]$ represents all payments of mobile device users for cloud datacenter provider j's memory. Let $f_i = \sum_j (u_{ij}^{cpu} + u_{ij}^{ram})$, f_i is the total payment of the ith mobile device user. N mobile device users compete for cloud datacenter provider's resources with finite capacity. The cloud datacenter resource is provisioned using a market mechanism, where the cloud datacenter resource allocation depends on the relative payments of the mobile device users. sq_i^n is the storage requirement of ith mobile device user's nth job, and cq_i^n is the processor requirement of ith mobile device user's nth job. $q_i^n = sq_i^n + cq_i^n$, q_i^n is total service requirement of ith mobile device user's nth job. Let p_j^{cpu} denote the price of CPU of cloud datacenter resource j, p_j^{ram} denote the price of memory of the cloud datacenter resource j. $p^{cpu} = (p_1^{cpu}, p_2^{cpu}, \ldots, p_j^{cpu})$ denote the set of CPU prices of cloud datacenter providers, $p^{ram} = (p_1^{ram}, p_2^{ram}, \ldots, p_j^{ram})$ denote the set of memory prices of cloud datacenter providers. The ith mobile device user receives the cloud resources proportional to the payment. Let v_{ij}^{cpu}, v_{ij}^{ram} be the fraction of CPU and memory provisioned to mobile device user i by cloud datacenter provider j.

$$v_{ij}^{ram} = C_j^{ram} \frac{q_i^n u_{ij}^{ram}}{p_j^{ram}}, \qquad v_{ij}^{cpu} = C_j^{cpu} \frac{q_i^n u_{ij}^{cpu}}{p_j^{cpu}}.$$

The problem of mobile device user can be reformulated as

$$\begin{aligned} MaxP_{md} = Max\Bigg\{&\left(B_i - \sum_j \left(u_{ij}^{cpu} + u_{ij}^{ram}\right)\right) + \left(T_i - \sum_n \left(\frac{q_i^n p_j^{ram}}{C_j^{ram} u_{ij}^{ram}} + \frac{q_i^n p_j^{cpu}}{C_j^{cpu} u_{ij}^{cpu}}\right)\right) \\ &+ \left(E_i - \sum_{n=1}^{N} er\left(\frac{q_i^n p_j^{ram}}{C_j^{ram} u_{ij}^{ram}} + \frac{q_i^n p_j^{cpu}}{C_j^{cpu} u_{ij}^{cpu}}\right)\right)\Bigg\}, \end{aligned}$$

$$\text{s.t. } B_i \geqslant \sum_j \left(u_{ij}^{cpu} + u_{ij}^{ram}\right), \qquad T_i \geqslant \sum_n t_i^n. \tag{3.6}$$

We take derivative and second derivative of P_{md} with respect to u_{ij}^{ram}:

$$P''_{md}\left(u_{ij}^{ram}\right) = -\sum \frac{q_i^n p_j^{ram}}{C_j^{ram}(u_{ij}^{ram})^3} - \sum er \frac{q_i^n p_j^{ram}}{C_j^{ram}(u_{ij}^{ram})^3}$$

$P''_{md}(u_{ij}^{ram}) < 0$ is negative. The extreme point is the unique value minimizing the mobile device user's payment under the deadline and the budget. The Lagrangian for the mobile device user's utility is $L_{md}(u_{ij}^{ram}, u_{ij}^{cpu})$.

$$\begin{aligned} L_{md} = {} & \left(E_i - \sum_{n=1}^{N} er\left(\frac{q_i^n p_j^{ram}}{C_j^{ram} u_{ij}^{ram}} + \frac{q_i^n p_j^{cpu}}{C_j^{cpu} u_{ij}^{cpu}}\right)\right) \\ & + \left(T_i - \sum_n \left(\frac{q_i^n p_j^{ram}}{C_j^{ram} u_{ij}^{ram}} + \frac{q_i^n p_j^{cpu}}{C_j^{cpu} u_{ij}^{cpu}}\right)\right) \\ & + \left(B_i - \sum_j \left(u_{ij}^{cpu} + u_{ij}^{ram}\right)\right) + \eta\left(B_i - \sum_j \left(u_{ij}^{cpu} + u_{ij}^{ram}\right)\right) \\ & + \psi\left(T_i - \sum_n \left(\frac{q_i^n p_j^{ram}}{C_j^{ram} u_{ij}^{ram}} + \frac{q_i^n p_j^{cpu}}{C_j^{cpu} u_{ij}^{cpu}}\right)\right), \end{aligned} \tag{3.7}$$

where η, ψ is the Lagrangian constant. From Karush–Kuhn–Tucker theorem we know that the optimal solution is given $\partial L_{md}(u_{ij}^{ram}, u_{ij}^{cpu})/\partial u_{ij}^{cpu} = 0$ for $\eta, \psi > 0$.

$$\begin{aligned} \frac{\partial L_{md}}{u_{ij}^{cpu}} = {} & \sum_n \left(\frac{q_i^n p_j^{cpu}}{C_j^{cpu}(u_{ij}^{cpu})^2}\right) - 1 + \sum_n er\left(\frac{q_i^n p_j^{cpu}}{C_j^{cpu}(u_{ij}^{cpu})^2}\right) \\ & - \eta + \psi \sum_n \left(\frac{q_i^n p_j^{cpu}}{C_j^{cpu}(u_{ij}^{cpu})^2}\right). \end{aligned}$$

Let $\partial L_{md}(u_{ij}^{ram}, u_{ij}^{cpu})/\partial u_{ij}^{cpu} = 0$ to obtain

$$u_{ij}^{cpu} = \left(\frac{(1 + er + \psi) q_i^n p_j^{cpu}}{(1 + \eta) C_j^{cpu}}\right)^{1/2}. \tag{3.8}$$

Using this result in the constraint equation, we can determine $\theta = \frac{(1+er+\psi)}{(1+\eta)}$ as

$$(\theta)^{-1/2} = \frac{T_i}{\sum_{n=1}^{N} \left(\frac{q_i^n p_j^{cpu}}{C_j^{cpu}}\right)^{1/2}}.$$

We obtain u_{ij}^{cpu*}

$$u_{ij}^{cpu*} = \left(\frac{q_i^n p_j^{cpu}}{C_j^{cpu}}\right)^{1/2} \frac{\sum_{n=1}^{N} \left(\frac{q_i^n p_j^{cpu}}{C_j^{cpu}}\right)^{1/2}}{T_i}. \tag{3.9}$$

It means that mobile device user i wants to pay $u_{ij}^{cpu^*}$ to cloud datacenter resource j for CPU to run VMs under the deadline and the budget.

Using the similar method, we can solve optimal payment $u_{ij}^{ram^*}$ of cloud datacenter provider j.

Let $\partial L_{md}(u_{ij}^{ram}, u_{ij}^{cpu})/\partial u_{ij}^{ram} = 0$ to obtain

$$u_{ij}^{ram^*} = \left(\frac{q_i^n p_j^{ram}}{C_j^{ram}}\right)^{1/2} \frac{\sum_{n=1}^{N} \left(\frac{q_i^n p_j^{ram}}{C_j^{ram}}\right)^{1/2}}{T_i}. \tag{3.10}$$

It means that mobile device user i pay $u_{ij}^{ram^*}$ to cloud datacenter resource j for memory to run VMs under the deadline and the budget.

A mobile cloud provider run multiple VMs to execute the jobs for mobile device users, the cloud datacenter providers pay for the electrical cost depending on the electricity price pe_j. The cloud datacenter provider optimization aims to maximize the revenue for provisioning VMs to mobile device users and minimize the electrical cost. The revenue optimization of cloud datacenter provider is to maximize the benefit function P_{cp} without exceeding the electrical expenditure of cloud provider and physical resource capacity

$$MaxP_{cp} = Max \sum (EB_j - ecost_j) + \left(u_{ij}^{cpu} \log v_{ij}^{cpu} + u_{ij}^{ram} \log v_{ij}^{ram}\right)$$
$$\text{s.t. } ecost_j \leqslant EB_j, \qquad C_j^{cpu} \geqslant \sum_i v_{ij}^{cpu}, \qquad C_j^{ram} \geqslant \sum_i v_{ij}^{ram}.$$

For the mobile cloud provider's optimization problem, the mobile cloud provider aims to maximize their revenue and minimize the electrical cost for provisioning VMs to mobile device user. The revenue of cloud datacenter provider is affected by the payments of mobile device users and the cost of the electricity. The cloud datacenter provider' revenue increases when provisioned resources for running VMs increase, payments increase, and the electricity cost decreases. $\sum(u_{ij}^{cpu} \log v_{ij}^{cpu} + u_{ij}^{ram} \log v_{ij}^{ram})$ presents the revenue obtained by cloud provider j from mobile device users. The objective of mobile cloud datacenter providers is to maximize the revenue and minimize electrical cost $ecost_j$. $EB_j - ecost_j$ represents the electrical expenditure subtracting actual cost for running VMs. Mobile device user adaptively adjusts cloud resource demand based on the cloud datacenter's resource supply conditions, while the mobile cloud provider adaptively provisions cloud resource for the mobile device user. The interaction between mobile device user demand and the supply of cloud datacenter is controlled through price sets p_j^{cpu}, p_j^{ram}, which are charged from mobile device users by cloud datacenter. For the cloud datacenter provider's optimization problem, different mobile cloud providers compute optimal resource allocation for maximizing the benefit under the limit of energy expenditure

$$ecost_j = pe_j \sum_{i=1}^{N} ed_i^j, \tag{3.11}$$

pe_j denote electricity price. The energy consumption rate of cloud provider j for hosting VMs to mobile device users is denoted as ec_j. The energy cost of mobile cloud provider for hosting VMs can't exceed EB_j. The electrical usage of mobile cloud provider j for providing VM to mobile device i can be presented as $ed_i^j = ec_j(v_{ij}^{ram} + v_{ij}^{cpu})$. Mobile cloud provider's optimization can be presented as

$$\begin{aligned} MaxP_{cp} &= Max \sum (EB_j - pe_j ec_j \sum_{i=1}^{N} \left(v_{ij}^{cpu} + v_{ij}^{ram}\right) \\ &\quad + \left(u_{ij}^{cpu} \log v_{ij}^{cpu} + u_{ij}^{ram} \log v_{ij}^{ram}\right). \end{aligned} \tag{3.12}$$

We take derivative and second derivative with respect to v_{ij}^{cpu}:

$$P'_{cp}\left(v_{ij}^{cpu}\right) = \frac{u_{ij}^{cpu}}{v_{ij}^{cpu}} - pe_j ec_j, \qquad P'_{cp}\left(v_{ij}^{cpu}\right) = -\frac{u_{ij}^{cpu}}{(v_{ij}^{cpu})^2}$$

$P''_{cp}(v_{ij}^{cpu}) < 0$ is negative due to $0 < v_{ij}^{cpu}$. The extreme point is the unique value for mobile cloud provider' optimization. The Lagrangian for P_{cp} is

$$\begin{aligned} L_{cp} &= Max\left(EB_j - pe_j ec_j \sum_{i=1}^{N} \left(v_{ij}^{cpu} + v_{ij}^{ram}\right)\right) + \sum \left(u_{ij}^{cpu} \log v_{ij}^{cpu} + u_{ij}^{ram} \log v_{ij}^{ram}\right) \\ &\quad + \xi \left(C_j^{ram} - \sum_i v_{ij}^{ram}\right) + \kappa \left(C_j^{cpu} - \sum_i v_{ij}^{cpu}\right) \\ &\quad + \alpha \left(EB_j - pe_j ec_j \sum_{i=1}^{N} \left(v_{ij}^{cpu} + v_{ij}^{ram}\right)\right), \end{aligned} \tag{3.13}$$

where α, ξ, κ is the Lagrangian constant. From Karush–Kuhn–Tucker theorem we know that the optimal solution is given $\partial L(v_{ij}^{cpu})/\partial v_{ij}^{cpu} = 0$

$$v_{ij}^{cpu} = \frac{u_{ij}^{cpu}}{pe_j ec_j (1+\alpha)}.$$

Using this result in the constraint equation, we can determine $\gamma = 1 + \alpha$ as $\gamma = \frac{\sum u_{ij}^{cpu}}{EB_j}$.

We obtain

$$v_{ij}^{cpu*} = \frac{u_{ij}^{cpu} EB_j}{pe_j ec_j \sum u_{ij}^{cpu}}. \tag{3.14}$$

It means that cloud datacenter providers allocate optimal CPU v_{ij}^{cpu*} for hosting VM for mobile device i while maximizing its own profit.

Using the similar method, we can solve memory allocation optimization v_{ij}^{ram*} of cloud datacenter provider j.

$$v_{ij}^{ram*} = \frac{u_{ij}^{ram} EB_j}{pe_j ec_j \sum u_{ij}^{ram}} \tag{3.15}$$

It means that cloud datacenter providers allocate optimal memory v_{ij}^{ram*} for hosting VM for mobile device user i.

4. Optimization Algorithm for Utilizing Cloud Services for Mobile Devices

The distributed optimization algorithm for utilizing cloud services for mobile devices (Algorithm 1) is executed by mobile device users and the cloud datacenter providers. In mobile device optimization problem, the mobile device user gives the unique optimal payment to the cloud datacenter provider under the deadline, the budget and energy constraint to maximize the mobile device user's satisfaction. In mobile device user optimization problem, mobile cloud provider runs multiple VMs to execute the jobs for mobile device users, the cloud datacenter providers pay for the electrical cost.

5. Experiments

5.1. *Environment Description*

We simulate a mobile cloud environment with a 2 dimension area of 500 m × 500 m to study mobile device's behavior. Each mobile device in the simulated environment has a maximal radio range of 100 m, and moves following a random walking mobility model. The average speed of each mobile device is 5 meters per second. Each mobile device's battery capacity is initialized with a random value in the range of [700, 800], and reduced automatically by a random value in the range of [0, 5] in each iteration. The initial price of electrical energy for cloud datacenter is set from 1 to 100 dollars. The initial price of VM is set from 10 to 500 dollars.

There are 40 mobile devices and 12 cloud resource provider, all of which contribute resources to the mobile cloud environment. Wi-Fi interfaces operate at a rate of 11 Mb/s. All Ethernet interfaces operate at a rate of 10 Gb/s. Energy consumption is represented

Algorithm 1 Optimization Algorithm for Utilizing Cloud Services for Mobile Devices (OAUCS)

cloud datacenter provider

Input: new price $p_j^{cpu^{(n)}}$, $p_j^{ram^{(n)}}$ from cloud datacenter provider

Step 1: Calculates $u_{ij}^{cpu^{*(n+1)}}$ to cloud datacenter resource j for needed CPU to run VMs for mobile device user, and maximize the mobile device user's benefit function P_{md}

$u_{ij}^{cpu^{*(n+1)}} = MaxP_{md}(u_{ij}^{cpu}, u_{ij}^{ram})$;

Step 2: If $B_i \geqslant \sum_j (u_{ij}^{cpu} + u_{ij}^{ram})$

Then Return payment $u_{ij}^{cpu^{*(n+1)}}$ to cloud datacenter resource j;

Else Return Null;

Step 3: Calculates optimal payment $u_{ij}^{ram^{*(n+1)}}$ to cloud datacenter resource j for needed memory

$u_{ij}^{ram^{*(n+1)}} = MaxP_{md}(u_{ij}^{cpu}, u_{ij}^{ram})$;

Step 4: If $B_i \geqslant \sum_j (u_{ij}^{cpu} + u_{ij}^{ram})$

Then Return $u_{ij}^{ram^{*(n+1)}}$ to cloud datacenter resource j;

Else Return Null

Output: $u_{ij}^{cpu^{*(n+1)}}$, $u_{ij}^{ram^{*(n+1)}}$ to cloud datacenter resource j.

mobile device user

Input: optimal payments $u_{ij}^{cpu^{(n)}}$, $u_{ij}^{ram^{(n)}}$ from mobile device user i

Step 1: Calculates $v_{ij}^{cpu(n+1)^*}$ to host VM for mobile device user i, and maximize the cloud datacenter provider's benefit function P_{cp}

$v_{ij}^{cpu(n+1)^*} = MaxP_{cp}(v_{ij}^{cpu}, v_{ij}^{ram}$;

Step 2 Computes new price of CPU

$p_j^{cpu(n+1)} = \max\{\varepsilon, p_j^{cpu(n)} + \eta(\sum_i v_{ij}^{cpu} p_j^{cpu(n)} - C_j^{cpu})\}$;

// $\eta > 0$ is a small step size parameter, n is iteration number

Step 3: Return CPU price $p_j^{cpu(n+1)}$ to all mobile device users;

Step 4: Calculates $v_{ij}^{ram(n+1)^*}$ to host VM for mobile device user i

$v_{ij}^{ram(n+1)^*} = MaxP_{cp}(v_{ij}^{ram}, v_{ij}^{cpu})$;

Step 5: Computes new price of the memory

$p_j^{ram(n+1)} = \max\{\varepsilon, p_j^{ram(n)} + \eta(\sum_i v_{ij}^{ram} p_j^{ram(n)} - C_j^{ram})\}$;

// $\eta > 0$ is a small step size parameter, n is iteration number

Step 6: Return memory price $p_j^{ram(n+1)}$ to all mobile device users;

Output: price $p_j^{cpu(n+1)}$, $p_j^{ram(n+1)}$ to mobile device users

Table 2
Simulation parameters.

Simulation parameter	Value
Total number of mobile device users	40
Total number of cloud providers	12
Mobility model	Random-walking mobility
Average speed of mobile device	5 m/s
Transmission range	[10 m, 250 m]
Total number of nodes	30
Traffic type	CBR
Initial price of VM	[10, 500]
Deadline	[100 ms, 400 ms]
Expense budget	[100, 1500]
Electrical energy	[0.1, 1.0]
Bandwidth	[100 kbps, 1000 kbps]
Computing power	[100, 1000]
RAM	[100 MB, 2000 MB]
Energy price	[1, 100]
Job arrival rate	[0.1, 0.6]

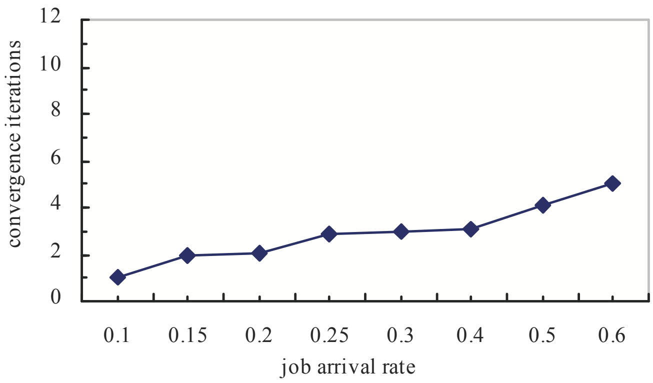

Fig. 1. Convergence iterations under various job arrival rate.

as a percentage of the total energy required to meet all job deadlines. Jobs arrive at each cloud node s_i, $i = 1, 2, \dots, n$ according to a Poisson process with rate α. The energy cost can be expressed in the dollar that can be defined as unit energy processing cost. The initial price of energy is set from 10 to 500 cloud dollars. The deadlines of mobile device user are chosen from 100 ms to 400 ms. The budgets of mobile device users are set from 100 to 1500 dollars. Simulation parameters are listed in Table 2.

5.2. *Convergence of the Algorithm*

The experiment is to test convergence of optimization algorithm for utilizing cloud services for mobile devices (*OAUCS*). Convergence iterations are tested under varying job arrival rate (a). In Fig. 1, when job arrival rate is low ($a = 0.1$), the convergence iteration of *OAUCS* is one. When job arrival rate increases to 0.25, the convergence iterations

Algorithm 2 MuSIC algorithm

Step 1: computing the center of mobility l_{cm}^{uk} of each user u_k
Step 2: uses the service selection function $Find_{service}$
$Condidote_{service} = Find_{service}(W(u_k), l_{cm}^{uk})$
Step 3: $Util_0 = Compute_{Fmobile}(Condidote_{service})$
Step 4: For $i = 1$ to maxiter do
Step 5: $Condidote_{service} = Find_{service}(W(u_k), l_{cm}^{uk})$
Step 6: $Util_1 = Compute_{Fmobile}(Condidote_{service})$
Step 7: $\Delta = Util_0 - Util_1$
Step 8: If $\Delta > 0$
Step 9: $Util_0 = Util_1$
Step 10: End IF
Step 11: End For
Return $Condidote_{service}$, $Util_0$

of *OAUCS* is three. When job arrival rate increases to 0.5, the convergence iterations of *OAUCS* is four. When job arrival rate increases to 0.6, the convergence iterations of *OAUCS* increase to five. As the demand of cloud datacenter resource is more than the supply, the price of the cloud resources increases. When job arrival rate increases quickly, the cloud resource supply is not enough to be allocated to mobile device user, the price of the cloud datacenter resource will increase, and some mobile device users will not afford the cloud resources.

5.3. *Comparison Experiment*

The next experiments aimed at comparing our distributed optimization algorithm for utilizing cloud services for mobile devices (*OAUCS*) with mobility-aware optimal service allocation in mobile cloud computing (*MuSIC*) proposed by Rahimi *et al.* (2013). Rahimi *et al.* (2013) develop *MuSIC* (Mobility-Aware Service Allocation on Cloud), an efficient heuristic for tiered-cloud resource allocation which takes into consideration user mobility information. The *MuSIC* algorithm is a greedy heuristic that generates a near-optimal solution to the tiered cloud resource allocation problem using a simulated annealing-based approach. *MuSIC* uses simulated annealing as the core approach in selecting and refining service selection; custom policies have been designed to make it efficient for the 2-tiered cloud architecture with mobile applications. Given a cloud service set, *MuSIC* starts by computing the center of mobility of each user by LTWs (location-time workflows). The general goal is to select services near that location. *MuSIC* then uses the service selection function that returns the list of services near the user center of mobility, which can realize the LTW and satisfy the required constraints. The pseudo code for the *MuSIC* algorithm is shown in Algorithm 2.

To evaluate the performance of our optimization algorithm for utilizing cloud services for mobile devices (*OAUCS*) against mobility-aware optimal service allocation in mobile

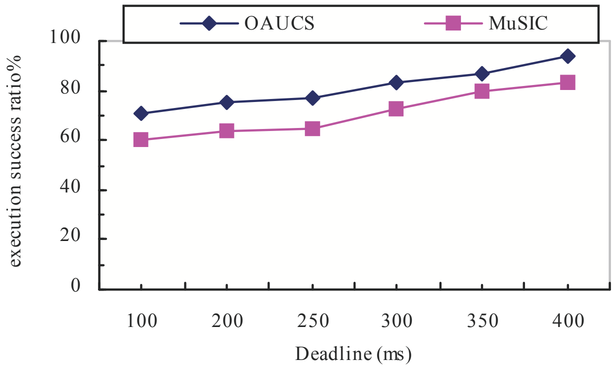

Fig. 2. Execution success ratio under various deadline.

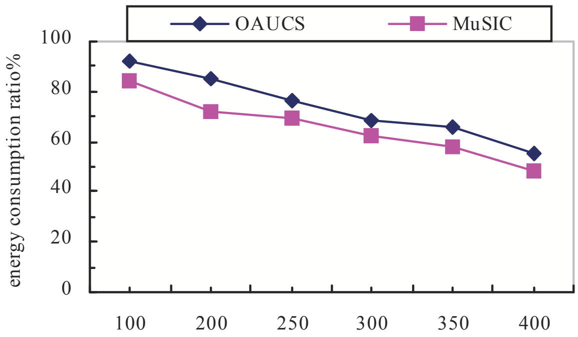

Fig. 3. Energy consumption ratio under various deadline.

cloud computing (*MuSIC*) (Rahimi *et al.*, 2013), we adopt the metrics: energy consumption ratio, execution success ratio, resource allocation efficiency and cost. The experiments study the impacts of deadline, mobility speeds, and the node number on performance metrics.

How the deadline effects on energy consumption ratio, execution success ratio, resource allocation efficiency and cost were illustrated in Figs. 2–5 respectively. Figure 2 is to show the effect of the deadline on execution success ratio. When the deadline is low, the execution success ratios of *MuSIC* and *OAUCS* are low. When increasing deadline, execution success ratios of two algorithms become higher. Because under low deadline, some jobs of mobile device user can't be completed on time. When deadline is 200 ($T = 200$), execution success ratio of *MuSIC* falls to 60%, execution success ratio of *OAUCS* falls to 73%. Figure 3 shows the energy consumption ratio varies under different deadlines. From the results in Fig. 9, when the deadline is low, there is intensive demand for the cloud resources in short time, some mobile device user need choose more energy exhausting resource to process the jobs, energy consumption ratio is high. However, when the deadline changes to higher, it is likely that mobile device user's jobs can be completed before the deadline, so mobile device user considers using the energy saving resources to complete

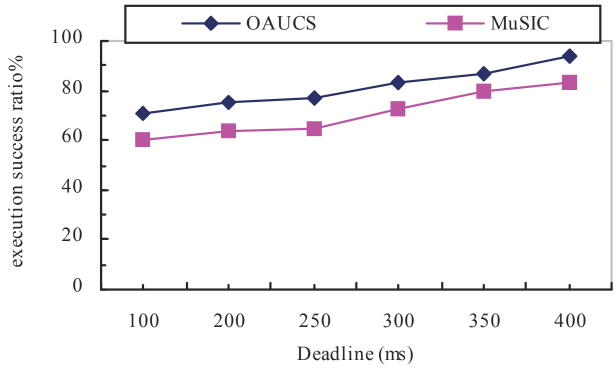

Fig. 4. Allocation efficiency under various deadline.

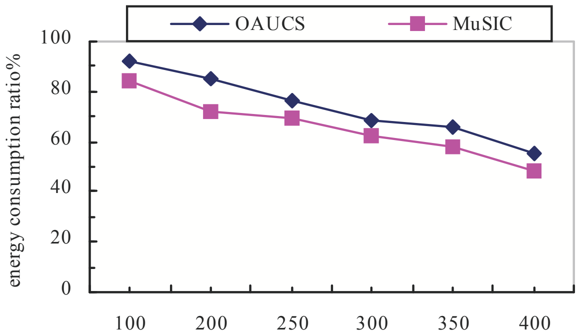

Fig. 5. Cost Vs deadline.

jobs to maximize the benefit, energy consumption ratio is low. When the deadline is 250, energy consumption ratio of *MuSIC* is 67%; energy consumption ratio of *OAUCS* is 73%. Compared with *MuSIC*, *OAUCS* consume more energy. For the resource allocation efficiency under different deadline constraints, from the results in Fig. 4, when increasing the deadlines, the impact on the resource allocation efficiency is obvious. A larger deadline values brings out higher resource allocation efficiency. When the deadline is 400, the allocation efficiency of *OAUCS* is 20% higher than *MuSIC*. Under the same deadline, *OAUCS* has higher resource allocation efficiency than *MuSIC*. Figure 5 represents the impact of different deadline constraint on the renting cost. When the deadlines are small, *OAUCS* spends more payment for obtaining cloud resources to complete jobs, because the mobile cloud user need buy expensive cloud service from cloud datacenter provider. When the deadline is 400, the cost of *OAUCS* is 27% less than the deadline is 200. When increasing deadline by $T = 400$, the cost of *MuSIC* is as much as 39% more than *OAUCS*. When $T = 400$, the cost of *MuSIC* decreases to nearly 34% compared with $T = 100$.

In Figs. 6–7, we study the impact of the different average speeds of mobile cloud users on allocation efficiency and execution success ratio. Figures 8 and 9 are to measure the effect of different cloud resource nodes on execution success ratio and allocation efficiency

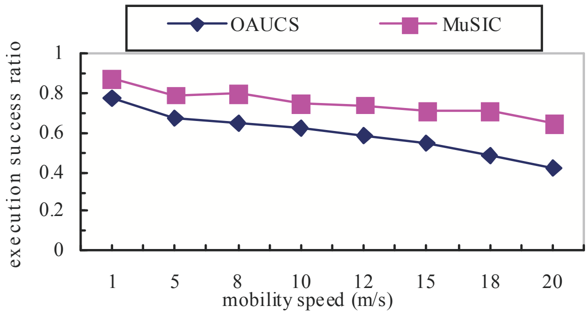

Fig. 6. Execution success ratio Vs. the mobility speeds.

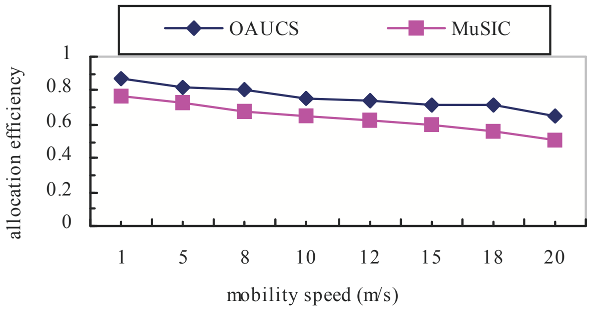

Fig. 7. Allocation efficiency Vs. the mobility speeds.

respectively. Figure 6 shows how the execution success ratio is affected by mobility speed. The execution success ratio of *OAUCS* and *MuSIC* decreases when the mobility speed increases. When $s = 20$, execution success ratio of *OAUCS* is as much as 35% lower than that by $s = 1$. Under the same mobility speed $s = 20$, *OAUCS* has 17% less execution success ratio than *MuSIC*. This is because *MuSIC* is the tiered-cloud resource allocation which takes into consideration user mobility information. *OAUCS* do not take into account the movement characteristics of the mobile device. Figure 7 shows the effect of mobility speed on the allocation efficiency. When the mobility speed increases to 15, the allocation efficiency of *OAUCS* is as much as 24% less than that with $s = 1$. The allocation efficiency is larger when the mobility speed is smaller. When the mobility speed increases, some mobile cloud users' requirements can't be processed on time. The allocation efficiency of *OAUCS* decreases slowly than *MuSIC* when the mobility speed increases. When the mobility speed is 20, the allocation efficiency of *MuSIC* decreases to 51%, the allocation efficiency of *OAUCS* decreases to 64%. Figure 8 shows when the number of cloud resource node increases to 10 ($N = 10$), execution success ratio of *OAUCS* is as much as 21% less than that with $N = 2$. The execution success ratio is larger when the number of cloud resource node is smaller. The execution success ratio of *OAUCS* decreases slowly than *MuSIC* when the number of cloud resource node increases. When the number of cloud resource node is 12, the execution success ratio of *MuSIC* decreases to 65%, the execution success ratio of *OAUCS* decreases to 73%. Considering allocation efficiency, as shown

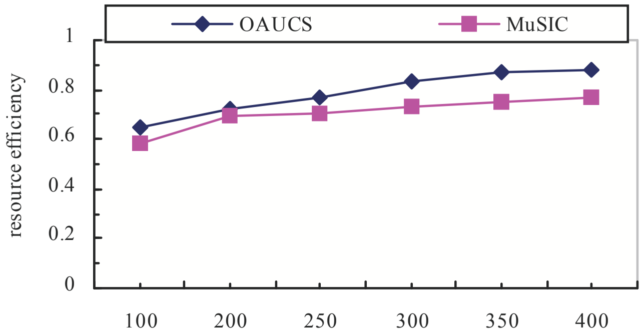

Fig. 8. Execution success ratio vs. number of cloud resource nodes.

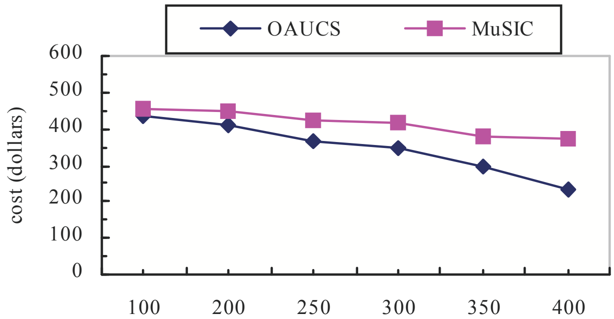

Fig. 9. Allocation efficiency vs. number of cloud resource nodes.

in Fig. 9, when cloud resource node increases, allocation efficiency deteriorates. When the number of cloud resource nodes is 10, the allocation efficiency of *OAUCS* is as 31% less than $N = 2$. When the number of cloud resource node is 12, allocation efficiency of *OAUCS* decreases to 69%, resource allocation efficiency of *MuSIC* decreases to 51%.

6. Conclusions

This paper presents an optimization approach for utilizing cloud services for mobile client in mobile cloud. Mobile cloud service provisioning optimization considers the benefit of both mobile device users and cloud datacenters. The paper proposes a distributed optimization algorithm for utilizing cloud services for mobile devices. Firstly, the experiment is to test convergence of the proposed algorithm. Secondly, the experiments aim to compare proposed optimization algorithm for utilizing cloud services for mobile devices with mobility-aware optimal service allocation in mobile cloud (*MuSIC*). The experiments study the impacts of job arrival rate, deadline, mobility speeds, and the node number on energy consumption ratio, execution success ratio, resource allocation efficiency and cost. The experiment shows that our algorithm outperforms *MuSIC* in terms of some performance metrics such as allocation efficiency.

Acknowledgements. The authors thank the editors and the anonymous reviewers for their helpful comments and suggestions. The work was supported by the National Natural Science Foundation (NSF) under grants (Nos. 61472294, 61171075), Key Natural Science Foundation of Hubei Province (No. 2014CFA050), Applied Basic Research Project of WuHan, National Key Basic Research Program of China (973 Program) under Grant No.2011CB302601, Program for the High-end Talents of Hubei Province, and Open Fund of the State Key Laboratory of Software Development Environment (SKLSDE). Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the above agencies.

References

Abolfazli, S., Sanaei, Z., Shiraz, M., Gani, A. (2012). MOMCC: market-oriented architecture for mobile cloud computing based on service oriented architecture. In: *1st IEEE International Conference on Communications in China Workshops (ICCC)*. IEEE Press, New York, pp. 8–13.

Barbarossa, S., Sardellitti, S., Di Lorenzo, P. (2013). Joint allocation of computation and communication resources in multiuser mobile cloud computing. In: *Signal Processing Advances in Wireless Communications (SPAWC), 2013 IEEE 14th Workshop on*. IEEE Press, New York, pp 26–30.

Barbera, M.V., Kosta. S., Mei, A., Stefa, J. (2012). To offload or not to offload? The bandwidth and energy costs of mobile cloud computing. In: *INFOCOM, 2013 Proceedings IEEE*. IEEE Press, New York, pp. 1285–1293.

Bessis, N., Sotiriadis, S., Xhafa, F., Asimakopoulou, E. (2013). Cloud scheduling optimization: a reactive model to enable dynamic deployment of virtual machines instantiations. *Informatica*, 24(3), 357–380.

Choi, J., Choi, C., Yim, K., Kim, J., Kim, P. (2013). Intelligent reconfigurable method of cloud computing resources for multimedia data delivery. *Informatica*, 24(3), 381–394.

Dinh, H.T., Lee, C., Niyato, D., Wang, P. (2013). A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless Communications and Mobile Computing*, 13(18), 1587–1611.

Ferber, M., Rauber, T. (2012). Mobile cloud computing in 3G cellular networks using pipelined tasks. In: *ESOCC'12 Proceedings of the First European Conference on Service-Oriented and Cloud Computing*, pp. 192–199.

Ge, Y., Zhang, Y., Qiu, Q., Lu, Y.-H. (2012). A game theoretic resource allocation for overall energy minimization in mobile cloud computing system. In: *Proceedings of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design*. ACM, New York, pp. 279–284.

Gkatzikis, L., Koutsopoulos, I. (2013). Migrate or not? Exploiting dynamic task migration in mobile cloud computing systems. *IEEE Wireless Communications Magazine*, 20(3), 24–32.

Hussain, F.K., Hussain, O.K. (2011). Towards multi-criteria cloud service selection. In: *2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*. IEEE Press, New York, pp. 44–48.

Lee, W., Lee, S.K., Yoo, S., Kim, H. (2013). A collaborative framework of enabling device participation in mobile cloud computing. In: *Mobile and Ubiquitous Systems: Computing, Networking, and Services*. Springer, Berlin, pp. 37–49.

Li, C., Li, L. (2012). Optimal resource provisioning for cloud computing environment. *Journal of Supercomputing*, 62(2), 989–1022.

Li, C., Li, L. (2014). Phased scheduling for resource-constrained mobile devices in mobile cloud computing. *Wireless Personal Communications*, 77(4), 2817–2837.

Li, D., Cao, J., Lu, X., Chen, K.C.C. (2009). Efficient range query processing in peer-to-peer systems. *IEEE Transactions on Knowledge and Data Engineering*, 21(1), 78–91.

Lu, X., Wang, H., Wang, J., Xu, J., Li, D. (2013). Internet-based virtual computing environment: beyond the datacenter as a computer. *Future Generation Computer Systems*, 29, 309–322.

Makris, P., Skoutas, D.N., Skianis, C. (2012). On networking and computing environments' integration: a novel mobile cloud resources provisioning approach. In: *2012 International Conference on Telecommunications and Multimedia (TEMU)*. IEEE Press, New York, pp. 71–76.

Park, J.S., Yu, H.C., Lee, E.Y. (2012). Resource allocation techniques based on availability and movement reliability for mobile cloud computing. In: *Distributed Computing and Internet Technology*. Springer, Heidelberg, pp. 263–264.

Rahimi, M.R., Venkatasubramanian, N., Vasilakos, A.V. (2013). MuSIC: mobility-aware optimal service allocation in mobile cloud computing. In: *2013 IEEE Sixth International Conference on Cloud Computing (CLOUD)*. IEEE Press, New York, pp. 75–82.

Shiraz, M., Abolfazli, S., Sanaei, Z., Gani, A. (2013). A study on virtual machine deployment for application outsourcing in mobile cloud computing. *The Journal of Supercomputing*, 63(3), 946–964.

Shiraz, M., Gani, A., Khokhar, R.H., Buyya, R. (2013). A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing. *IEEE Communications Surveys & Tutorials*, 15(3), 1294–1313.

Shiraz, M., Ahmed, E., Gani, A., Han, Q. (2014) Investigation on runtime partitioning of elastic mobile applications for mobile cloud computing. *The Journal of Supercomputing*, 67(1), 84–103.

Wang, Y., Lin, X., Pedram, M. (2013). A nested two stage game-based optimization framework in mobile cloud computing system. In: *2013 IEEE 7th International Symposium on Service Oriented System Engineering (SOSE)*, pp. 494–502.

Zhang, W., Wen, Y., Wu, D.O. (2013). Energy-efficient scheduling policy for collaborative execution in mobile cloud computing. In: *INFOCOM, 2013 Proceedings IEEE*. IEEE Press, New York, pp. 190–194.

Zhang, Y., Niyato, D., Wang, P. (2013). An auction mechanism for resource allocation in mobile cloud computing systems. In: *Wireless Algorithms, Systems, and Applications*. Springer, Berlin, pp. 76–87.

C. Li received the ME in computer science from Wuhan Transportation University in 2000, and PhD degree in Computer Software and Theory from Huazhong University of Science and Technology in 2003. She now is a professor of Computer Science in Wuhan University of Technology. Her research interests include cloud computing, grid computing, and distributed computing. She has published over 15 papers in international journals.

L. Li received the BE degree in Communication Engineering from Harbin Institute of Military Engineering, China in 1970 and the ME degree in Communication and Electrical Systems from Huazhong University of Science and Technology, China in 1982. Since 1982, he has been with the Wuhan University of Technology, China, where he is currently a Professor and PhD tutor of Computer Science, and Editor in Chief of the Journal of WUT. He is Director of International Society of High-Technology and Paper Reviewer of IEEE INFOCOM, ICCC and ISRSDC. His research interests include high speed computer networks, protocol engineering and image processing. Professor Li has published over 150 technical papers and is the author of six books. He also was awarded the National Special Prize by the Chinese Government in 1993.

Debesų paslaugų mobiliesiems įrenginiams optimizavimas

Chunlin LI, Layuan LI

Mobilioji debesų kompiuterija atsirado siekiant padėti mobiliesiems įrenginiams atlikti daug skaičiavimų arba duomenų reikalaujančius uždavinius naudojant debesų išteklius. Šiame darbe pristatome debesų paslaugų mobiliajam klientui mobiliajame debesyje optimizavimo būdą, kuris įvertiną naudą mobiliųjų įrenginių naudotojams ir debesų duomenų centrams. Mobiliųjų debesų paslaugų optimizavimas atliekamas įvertinant termino, biudžeto ir energijos išlaidų apribojimus. Šiame straipsnyje siūlome paskirstytą debesų paslaugų mobiliesiems įrenginiams optimizavimo algoritmą. Eksperimentu ištyrėme pasiūlyto algoritmo konvergavimą ir atlikome palyginimą su susijusiais darbais. Eksperimento rezultatai rodo, kad pasiūlytas algoritmas pranoksta kitus kai kuriais rodikliais, pavyzdžiui, paskirstymo efektyvumą.