Fundamenta Informaticae 138 (2015) 435–455
DOI 10.3233/FI-2015-1220
IOS Press

Hybrid Cloud Scheduling Method for Cloud Bursting

Li Chunlin*
Department of Computer Science, Wuhan University of Technology
Wuhan 430063, P.R.China and
State Key Laboratory of Software Engineering, Wuhan University
chunlin74@aliyun.com

Li LaYuan
Department of Computer Science, Wuhan University of Technology
Wuhan 430063, P.R.China
jwtu@public.wh.hb.cn

Abstract. In the paper, we consider the hybrid cloud model used for cloud bursting, when the computational capacity of the private cloud provider is insufficient to deal with the peak number of customers' applications, the private cloud will rely on the resources leased from public cloud providers for the execution of private cloud applications. The paper proposes the model and algorithm of hybrid cloud scheduling optimization for cloud bursting. Public cloud providers and private cloud users communicate by the hybrid cloud marketplace. The hybrid cloud scheduling optimization is conducted at different levels. According to the model formulation and mathematic solutions of hybrid cloud scheduling, hybrid cloud scheduling algorithms for cloud bursting are proposed, which includes the routines of public cloud optimization, private cloud application optimization and private cloud job optimization. In the simulations, compared with other related algorithm, our proposed hybrid cloud scheduling algorithms achieve the better performance.

Keywords: hybrid cloud; cloud bursting; multicriteria optimization

1. Introduction

Private cloud management platforms have been emerging in the last several years providing new opportunities for efficient management of internal infrastructures leading to high utilization. The resources in a private cloud may become insufficient to satisfy the demands for higher computational power or storage

*Address for correspondence: Department of Computer Science, Wuhan University of Technology, Wuhan 430063, P.R.China.

Received October 2013; revised November 2014.

capacities. In cases where the local provisioning is not sufficient, the private resources can be extended with resources from other public remote infrastructures. In other words, a private cloud can be extended to a hybrid cloud by adding resources to their capacity from public cloud providers where required.

In a hybrid cloud, a company maintains its own private cloud, i.e. a virtualized data center, and can scale out to a public cloud if needed. The hybrid cloud approach extends the private cloud model by using both local and remote resources. It is usually used to handle flash crowds by scaling out when the local capacity is exhausted. Private clouds can exploit a hybrid cloud model by supplementing local infrastructure with computing capacity from an external public cloud. The aim of the private cloud deployments is not to sell capacity over the Internet, but to provide local users with a flexible private infrastructure to run service workloads within their administrative domain. Multiple public cloud providers such as Amazon, Google, Microsoft, etc., coexisting in a cloud computing market provide similar services. For example Google Docs and Microsoft Office Live provide its clients similar software service. The decision of whether a public cloud provider hosts a service depends jointly on the price it sets and the QoS guarantees it provides to its customers. Resource allocation and scheduling of multiple services is an important challenge in hybrid cloud, where there may be some free resources available from private clouds but some fee-paying resources from public clouds.

Our contributions are as follows: 1) The paper proposes a scheduling optimization scheme in hybrid cloud computing for cloud bursting. 2) The scheduling optimization is conducted at different levels: hybrid cloud system level and private cloud local level. 3) In the simulations, compared with other related algorithm, our proposed hybrid cloud scheduling algorithms achieve the better performance.

The rest of the paper is structured as followings. Section 2 discusses the related works. Section 3 presents hybrid cloud scheduling model for cloud bursting. Section 4 describes mathematic formulations of hybrid cloud scheduling for cloud bursting. Section 5 proposes hybrid cloud scheduling algorithm for cloud bursting. Section 6 describes the experiments. Section 7 gives the conclusions to the paper.

2. Related Works

Hybrid cloud is a combination of both public and private clouds, which provide services to its customers. Private clouds mean a cloud computing capability dedicated to one organization having a limited capacity. We refer as public cloud the utility computing made available in a pay-as-you-go manner to the general public. Dynamic provisioning of resources is one of most challenging problems in hybrid cloud computing. Resource provisioning and scheduling in hybrid cloud computing has attracted attention of the research community in the last years. However, little has been addressed from the public cloud provider and private provider's perspective to consider hybrid cloud hierarchical scheduling, also on how their interactions are modeled to maximize their benefit.

In [2, 3] Oleksiy Mazhelis et al. study economic aspects of hybrid cloud Infrastructure and address the issue of efficient division of the load between the private and the public portion of a hybrid cloud. They propose an analytical model of hybrid cloud costs, including the costs of computing and data communication. In [4] Prodromos Makris et al. propose a user-oriented, highly customizable infrastructure sharing approach, namely IaaS Request Admission Control (IRAC) for Hybrid Cloud Computing Environments. Several conflicting parameters such as the type of the hybrid cloud infrastructure being deployed, multiple user priority groups, security, energy efficiency and financial costs are taken into account. In [5] Vecchiola C. et al. present Aneka's deadline-driven provisioning mechanism in hybrid clouds composed of resources obtained from a variety of sources, which is responsible for supporting quality of service (QoS)-aware execution of scientific applications. In [6] Van den Bossche R. et al. proposing a set of algorithms to cost-efficiently schedule the deadline-constrained bag-of-tasks applications on both public cloud providers and private infrastructure. The proposed algorithms take into account both computational and data transfer costs as well as network bandwidth constraints. In [7] Jiayuan Yue propose a method for high throughput task processing to self-adapt peak workload on the hybrid infrastructure combining fixed-size system with public cloud. In [8] Xuanjia Qiu et al. present an optimization framework for dynamic, cost-minimizing migration of content distribution services into a hybrid cloud infrastructure that spans geographically distributed data centers. In [9] Faouzi Ben Charrada et al. explores an efficient and secure mechanism to partition computations across public and private machines in a hybrid cloud setting. In [10] Mustafizur Rahman et al. propose an adaptive heuristic for user constrained data-analytics workflow scheduling in hybrid Cloud environment by integrating the dynamic nature of heuristic based approaches as well as workflow-level optimization capability of meta-heuristic based approaches. In [11] Brock M. et al. study the execution of compute intensive applications on hybrid clouds. They propose a new form of cloud called HPC Hybrid Deakin (H2D) Cloud – an experimental hybrid cloud capable of utilizing both local and remote computational services for single large parallel applications. In [12] Den Bossche R.V. et al. study cost-optimal scheduling in a multi-provider hybrid cloud setting with deadline-constrained and preemptible but non-provider-migratable workloads that are characterized by memory, CPU and data transmission requirements. In [13] Linton A. et al. expands on previous work with the Virtual Organization Cluster Model by demonstrating its scalability across multiple grid sites with the use of a structured peer-to-peer overlay networking system.

In [14] Tekin Bicer et al. describe a modeling-driven resource allocation framework to support both time and cost sensitive execution for data-intensive applications executed in a hybrid cloud setting. In [15] Bjorkqvist M. et al. propose a novel algorithm to dynamically optimize the allocation of private and public nodes across services, with special focus on the performance-cost tradeoff between private and public nodes. In [16] Quarati A. et al. presents a cloud brokering algorithm delivering services with different level of non-functional requirements, to the private or public resources, on the basis of different scheduling criteria. In [17] D'Agostino D. et al. focus on the design of a brokering tool for hybrid clouds capable to adequately respond to specific Quality of Service (QoS) constraints. The proposed method is aimed at satisfying the highest number of user requests while trying maximizing the profit of the private provider. In [18] Zinnen A. presents experimental results on different optimization strategies for cost-optimal dynamic scheduling in hybrid cloud environments. In [19] Mohammad Mahdi Kashef et al. suggest a cost model for hybrid Clouds (i.e., the combinations of a private data center (private Cloud) and the public Cloud). In [20] Charrada F.B. et al. study approximate placement of service-Based applications in hybrid clouds. They propose a new algorithm that approximates the optimal placement based on communication and hosting costs induced by the deployment of components in the public cloud. In [21] Yongquan Fua et al. present general scalable and accurate decentralized level monitoring method for large-scale dynamic service provision in hybrid clouds. They propose a novel distributed level monitoring method HPM (Hierarchical Performance Measurement) satisfying these requirements. In [22] Bittencourt et al. introduces the scheduling problem in hybrid clouds presenting the main characteristics to be considered when scheduling workflows, as well as a brief survey of some of the scheduling algorithms. In [23] Time and Cost Optimization for Hybrid Clouds (TCHC) algorithm is proposed to reduce the execution time and cost of multiple workflows scheduling. In [24] Van Hoecke S. et al. present a virtual infrastructure management tool that allows to set-up and manage hybrid clouds efficiently in a user-friendly way. The tool provides automatic load balancing between the private and public clouds at the virtual machine level. In [25] Wei-JenWang et al. propose the Adaptive Scheduling with QoS Satisfaction algorithm, namely *AsQ*, for the hybrid cloud environment to raise the resource utilization rate of the private cloud and to diminish task response time as much as possible. In [26, 27] Bittencourt L.F. et al. deal with this problem, presenting HCOC: The Hybrid Cloud Optimized Cost scheduling algorithm. HCOC decides which resources should be leased from the public cloud and aggregated to the private cloud to provide sufficient processing power to execute a workflow within a given execution time. The paper [28] combines perspectives of SaaS user and SaaS provider, and presents an efficient SaaS cloud resource provisioning approach, which is beneficial for SaaS users and SaaS provider.

3. Hybrid Cloud Scheduling Model for Cloud Bursting

The hybrid cloud extends the private cloud model by using the resources of a public cloud to maintain service levels in the face of rapid workload fluctuations. Public cloud provider offer services in a pay-as-you-go manner to the general public. The number of cloud users requesting services from the private cloud built by the enterprises can be variable over the time. In this paper, we consider a private cloud which executes a set of applications for its customers. The customers submit a set of applications to be executed by the private cloud along with a deadline to be obeyed and cost budget. The private cloud provider aims to execute the customer's jobs within the deadline using computational resources leased from multiple public cloud providers. In hybrid cloud, public cloud providers and private cloud users communicate with one another by the means of hybrid cloud marketplace.

In order to achieve hybrid cloud hierarchical scheduling, the scheduling optimization is deployed at two levels: hybrid cloud system level and private cloud local level. At the high level, the hybrid cloud system scheduler implements the allocation of public cloud resources to the private cloud application groups. On the other hand, the private cloud application group coordinates the deployments of all private cloud applications that consume the allocation of public cloud resources. At the low level, the schedulers of the private cloud local level adjust the cloud resource usages to optimize the utility of single private cloud application. The private cloud local level schedulers are responsive; the hybrid cloud system level schedulers take more time to decide how to maximize private cloud application group's utility. The private cloud local level schedulers attempt to greedily optimize their own benefit. Hybrid cloud scheduling model is shown in Fig.1.

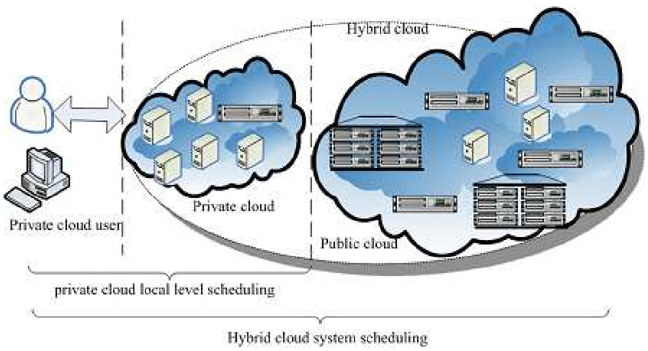

Figure 1. Hybrid cloud scheduling model

4. Mathematic Formulations and Solution for Hybrid Cloud Scheduling

4.1. Problem Formulation

To be able to provide a good quality of service (QoS), the private cloud must be prepared to attend a peak of demand. To contour this situation, the private cloud may rely on resources leased from public cloud providers. In the paper, we consider the hybrid cloud model used for cloud bursting, when the computational capacity of the private cloud provider is insufficient to deal with the peak number of customers' applications, the private cloud will rely on resources leased from public cloud providers for the execution of private cloud applications.

The notations used in the following sections are listed on Table 1.

Table 1. The description of notations

Notations	Meanings
$x_i^{j(cpu)}$	CPU required by a VM for private cloud application i from the public cloud provider j
$x_i^{j(ram)}$	RAM required by a VM for private cloud application i from the public cloud provider j
T_s	The deadline given by the private cloud applications group s to complete its all applications
$C_i^{(cpu)}$	the maximum capacity of CPU of public cloud provider j
$C_i^{(ram)}$	the maximum capacity of memory of public cloud provider j
t_i^n	the time taken by the private cloud application i to complete nth job
t_s^i	the time taken by i th application in the private cloud application group s
$u_i^{j(cpu)}$	the payments of the private cloud application i to public cloud provider j for CPU required by VM
$u_i^{J(cpu)}$	the payments of the private cloud application i to public cloud provider j for RAM required by VM
E_s	the budget of private cloud application group s
T_i	The deadline given by the private cloud application i
$energy_i$	The energy dissipation used by jth public cloud provider to support VMs for private cloud application
EC_i	limited energy budget of public cloud provider j
$E_i^{prirateciouа}$	the budget of private cloud application i
y_i^n	the payment of the nth job of private cloud application i
z_i^n	computation task of ith private cloud application's nth job

It is assumed that the hybrid cloud system consists of multiple public cloud providers, private cloud provider and private cloud applications. A public cloud system consists of a collection of datacenter connected by a communication network or internet. It is assumed that the public cloud datacenter consists of a set of physical machines which can host multiple virtual machines. It is hosted by a cloud service provider who sells its capacity using a pay-per-use payment model. Private cloud infrastructure is hosted within the data center of an organization and used by local users only, and it has limited capacity. The private cloud local users use the private cloud resource freely. Each public cloud provider may have different resource such as network, storage and compute power. Let $X = \{x_1, x_2 \dots x_i \dots x_n\}$ denote n classes of VM. The number of VMs in each class can be different, depending on the demand from the private cloud applications. Let $x_i^{j(cpu)}, x_i^{j(ram)}$, denote the amount of CPU and memory required by a VM for private cloud application i from public cloud provider j. VM can be expressed as $x = x_i^{j(cpu)}, x_i^{j(ram)}$. Let $S = \{s_1, s_2, \dots s_i\}$ denote the set of public cloud providers. The public cloud provider supply CPU capacity expressed in MHz, the memory capacity expressed in megabytes. The processing power of public cloud provider j is measured by the average CPU speed. C_i^{cpu}, C_i^{ram}, denote the maximum capacity of CPU and memory, which public cloud provider j can provide for the private cloud applications. $A = (a_1, a_2, \ldots a_i)$ denote the set of private cloud applications. In cases where the private cloud's resources are not sufficient, the private cloud application will lease the resources from public cloud providers. The private cloud application' jobs are assumed to be computationally intensive, mutually independent, and can be executed at any public cloud providers. As soon as a job arrives, it must be assigned to one VM for processing. We use J to denote the set of all jobs generated by private cloud application $i, J_i = \{J_i^1, J_i^2 \ldots J_i^n\}$. Each private cloud application's job can be described as $J_i^n = (t_i^n, y_i^n, z_i^n)$, in which t_i^n is the time taken by the i-th private cloud application to complete n-th job, z_i^n is the computation task of i-th private cloud application's nth job, y_i^n is the cost of i-th private cloud application's n-th job.

In hybrid cloud hierarchical scheduling model, several utility functions are used to evaluate the performance of private cloud applications groups and public cloud providers. $U_{hybridcloud}$ is the utility of system which considers both private cloud application groups and public cloud providers. We define hybrid cloud high-level system utility of the hybrid cloud is sum of the private cloud application group utilities and public cloud provider' utility.

$U_{publiccloud}$ presents the revenue of public cloud provider. In $U_{publiccloud}$, we could have chosen any other form for the utility that increases with $x_i^{j(cpu)}, x_i^{j(ram)}$. But we chose the log function because the benefit increases quickly from zero as the provisioned VMs increase from zero and then increases slowly.

$$U_{publiccloud} = \sum_{i=1}^{N} \left(u_i^{j(cpu)} \log x_i^{j(cpu)} + u_i^{j(ram)} \log x_i^{j(ram)} \right) - energy_j \tag{1}$$

$U^s_{privatecloudgroup}$ targets at maximizing the private cloud application group's satisfaction to pay less money and complete all applications for private cloud application group g as soon as possible.

$$U^s_{privatecloudgroup} = \left(T_s - K \sum_{i=1}^{I} t_s^i \right) + \left(E_s - \sum_{i=1}^{I} t_s^i \left(u_i^{j(ram)} + u_i^{j(ram)} \right) \right) \tag{2}$$

The problem of hybrid cloud optimization is formulated as the follows:

$$\begin{gathered} \text{Max } U_{hybridCloud} \\ s.t C_j^{cpu} \geq \sum_m x_i^{j(cpu)}, \\ C_j^{ram} \geq \sum_m x_i^{j(ram)}, \\ T_s \geq \sum_{i=1}^{I} T_s^i, \\ E_s \geq \sum_{i=1}^{I} \left(u_i^{j(cpu)} + u_i^{j(ram)} \right), \\ energy_j \leq RC_j \end{gathered} \tag{3}$$

In this problem, some constraints are related with resource constraints of public cloud provider. $x_i^{j(cpu)}$ is CPU required by a VM for private cloud application i from the public cloud provider j, $x_i^{j(ram)}$

is the memory required by a VM for private cloud application i from the public cloud provider j. The constraint implies that the aggregate CPU does not exceed the total capacity $C_i^{(cpu)}$ of public cloud provider j, aggregate memory units do not exceed the total resource $C_i^{(ram)}$ of public cloud provider j. Other constraints are related with private cloud applications. Private cloud application group should complete all its applications under the deadline and certain budget. One private cloud application group needs to complete a sequence of applications within specified deadline, T_s, while the total payment cannot exceed the budget E_s, $(u_i^{j(cpu)} + u_i^{j(ram)})$ are the payments of the all private cloud applications in group s to the public cloud provider j for CPU and memory required by VMs respectively.

$$\begin{aligned}
\operatorname{Max} U_{hybridCloud} &= \sum_{s=1}^{S}\left(\left(T_s - K\sum_{i=1}^{I} t_s^i\right) + \left(E_s - \sum_{i=1}^{I}(u_i^{j(cpu)} + u_i^{j(ram)}\right)\right) \\
&+ \sum_{i=1}^{N}\left(u_i^{j(cpu)}\log x_i^{j(cpu)} + u_i^{j(ram)}\log x_i^{j(ram)}\right) - energy_j \\
s.tC_j^{cpu} &\geq \sum_m x_i^{j(cpu)}, \\
C_j^{ram} &\geq \sum_m x_i^{j(ram)}, \\
T_s &\geq \sum_{i=1}^{I} t_s^i, \qquad (4) \\
E_s &\geq \sum_{i=1}^{I}\sum_{i=1}^{J}\left(u_i^{j(cpu)} + u_i^{j(ram)}\right), \\
energy_j &\leq EC_j
\end{aligned}$$

The Lagrangian approach is used to solve constrained optimization problems. Let us consider the Lagrangian form of hybrid cloud hierarchical scheduling optimization problem:

$$\begin{aligned}
L &= \sum_{s=1}^{S}\left(\left(T_s - K\sum_{i=1}^{I} t_s^i\right) + \left(E_s - \sum_{i=1}^{I}(u_i^{j(cpu)} + u_i^{j(ram)}\right)\right) \\
&+ \sum_{i=1}^{N}\left(u_i^{j(cpu)}\log x_i^{j(cpu)} + u_i^{j(ram)}\log x_i^{j(ram)}\right) - energy_j + \lambda\left(C_j^{cpu} - \sum_m x_i^{j(cpu)}\right) \\
&+ \beta\left(C_j^{ram} - \sum_m x_i^{j(ram)}\right) + \mu\left(T_s - \sum_{i=1}^{I} t_s^i\right) \qquad (5) \\
&+ \sigma\left(E - \sum_{i=1}^{I}\sum_{j=1}^{J}\left(u_i^{j(cpu)} + u_i^{j(ram)}\right)\right) + \xi\left(energy_j - EC_j\right)
\end{aligned}$$

Where λ_i is the Lagrangian multiplier. Solving the optimization function $\operatorname{Max} U_{hybridcloud}$ requires the coordination of all private cloud applications, which is impractical in hybrid cloud environment. In order to achieve a distributed solution, we must decompose the hybrid cloud hierarchical scheduling optimization problem. Since the Lagrangian is separable, the maximization of the Lagrangian can be processed in parallel by private cloud applications and public cloud provider respectively. The hybrid cloud high-level system optimization problem leads to a decomposition of problem $\mathrm{Max}\, U_{hybridcloud}$ into two subproblems, which are respectively conducted by private cloud applications and public cloud provider as follows:

$$\begin{aligned} \mathrm{Max}\, U_{hybridCloud} &= \sum_{i=1}^{N} \left(u_i^{j(cpu)} \log x_i^{j(cpu)} + u_i^{j(ram)} \log x_i^{j(ram)} \right) - energy_j \\ s.t C_j^{cpu} &\geq \sum_i x_i^{j(cpu)}, \\ C_j^{ram} &\geq \sum_i x_i^{j(ram)}, \\ energy_j &\leq EC_j \end{aligned} \tag{6}$$

$$\begin{aligned} \mathrm{Max}\, U_{privatecloudgroup} &= \sum_{s=1}^{S} \left(\left(T_s - K \sum_{i=1}^{I} t_s^i \right) + \left(E_s - \sum_{i=1}^{I} \left(u_i^{j(cpu)} + u_i^{j(ram)} \right) \right) \right) \\ s.t T_s &\geq \sum_{i=1}^{I} t_s^i, \\ E_s &\geq \sum_i^{I} \sum_j^{J} \left(u_i^{j(cpu)} + u_i^{j(ram)} \right) \end{aligned} \tag{7}$$

Public cloud optimization problem is conducted by the public cloud provider, different public cloud providers compute optimal resource allocation for supporting the VMs and maximizing the revenue of their own. Given the payment of the private cloud applications and energy cost for hosting VMs, the public cloud providers attempt to maximize the benefit function. To host VM for private cloud users, the public cloud provider has to pay for the energy cost depending on its electricity price. $energy_j \leq EC_j$ means the public cloud provider need maximize the utility function without exceeding maximal energy consumption of public cloud provider. Private cloud optimization problem is conducted by the private cloud applications, the private cloud application gives the optimal payment to public cloud providers under the constraints to maximize the private cloud application group's satisfaction. Private cloud application i submits the payment $u_i^{j(cpu)}, u^{j(ram)}$, to the public cloud provider j for running VMs. $(E_s - \sum_{i=1}^{I}(u_i^{j(cpu)} + u_i^{j(ram)})$ represents surpluses of private cloud application group's budget, which is obtained by the budgets of public cloud providers subtracting the payments from private cloud applications. $(T_s - K \sum_{i=1}^{I} t_s^i)$ represents the deadline of private cloud application group g subtracting actual completion time.

For private cloud local optimization, a private cloud application needs to complete a sequence of jobs within the deadline, T_i, while minimizing the cost overhead and execution time. We assume that each job of private cloud application i submits Y_i^n for the cloud resource. N private cloud jobs compete for the resources of private cloud application i. The resources allocated to private cloud jobs depend on the relative payments sent by the private cloud jobs. The nth private cloud job receives cloud resources proportional to its payment proportional to the sum of the private cloud application's revenue. Given the deadline T_i for private cloud application i to complete all private cloud jobs, the low level scheduling optimization in private cloud can be formulated as:

$$\begin{aligned} \operatorname{Max} U_{privatecloudlocal} &= \left\{ \left(E_i^{privatecloud} - \sum_s y_i^n \right) + \left(T_i - \sum_s t_i^n \right) \right\} \\ s.tT_i &\geq \sum_n t_i^n \end{aligned} \tag{8}$$

4.2. Public Cloud Optimization

Public cloud optimization problem aims at finding the optimal VM allocation x^* for private cloud application groups while maximizing the utility of public cloud provider without exceeding the public cloud resource capacity and upper limit of energy cost. Aggregate allocated CPU and memory of public cloud providers does not exceed the total capacity, C_j^{cpu}, C_j^{ram}. Public cloud providers compute optimal resource allocation to maximize the revenue of their own and minimize the cost for providing VMs to private cloud application groups. The profits of public cloud provider are affected by payments of private cloud application groups and energy cost of hosting VMs. So the revenue of public cloud provider increases when the cloud resources leased to the private cloud application groups increase and private cloud application groups' the payments increase, also the energy cost decreases. $\sum \left(u_i^{j(cpu)} \log x_i^{j(cpu)} + u_i^{j(ram)} \log x_i^{j(ram)} \right)$ presents the revenue obtained by public cloud provider j from private cloud applications.

Private cloud application group adaptively adjusts cloud resource demand based on the current public cloud provider's conditions, while the public cloud provider adaptively allocates cloud resource required by the private cloud users.

$$energy_j = p_j^{electricity^*} \sum_{i=1}^{N} g_i^j \tag{9}$$

In (9), $p_i^{electricity}$ denote electricity price. The energy consumption rate of public cloud provider for hosting VMs is denoted as e_j. The energy consumption cost of public cloud provider for hosting VMs can't exceed more than EC_j, which is the maximal energy consuption of public cloud provider.

The electrical energy consumption used by public cloud provider j to run VM for private cloud application i denoted as g_i^j can be written as following:

$$g_i^j = e_j (x_i^{j(cpu)} + x_i^{j(ram)}) \tag{10}$$

We reformulate public cloud optimization problem as

$$\operatorname{Max} \sum \left(u_i^{j(cpu)} \log x_i^{j(cpu)} + u_i^{j(ram)} \log x_i^{j(ram)} \right) - p_i^{electricity} \sum_{i=1}^{N} e_j (x_i^{j(cpu)} + x_i^{j(ram)}) \tag{11}$$

We take derivative and second derivative with respect to $x_i^{j(cpu)}$:

$$U_{publiccloud}{}''(x_i^{j(cpu)}) = -\frac{u_i^{j(cpu)}}{x_i^{j(cpu)^2}} \tag{12}$$

$U_{publiccloud}$"$(x_i^{j(cpu)}) < 0$ is negative due to $0 < x_i^{j(cpu)}$. The Lagrangian for $U_{publiccloud}$ is

$$\begin{aligned} L(x^{j(cpu)_i}) &= \sum \Big(u_i^{j(cpu)} \log x_i^{j(cpu)} + u_i^{j(ram)} \log x_i^{j(ram)}\Big) - p_i^{electricity} \sum_{i=1}^{N} e_j (x_i^{j(cpu)} + x_i^{j(ram)}) \\ &+ \lambda \Big(C_j^{cpu} - \sum_i x_i^{j(ram)}\Big) + \eta \Big(C_j^{ram} - \sum_i x_i^{j(ram)}\Big) \\ &+ \psi \left(EC_j - p_j^{electricity} \sum_{i=1}^{N} e_j (x_i^{j(cpu)} + x_i^{j(ram)})\right) \end{aligned} \tag{13}$$

Where ψ, λ, η are the Lagrangian constants. From Karush-Kuhn-Tucker Theorem we know that the optimal solution is given ${\delta L(x_i^{j(cpu)})} \Big/ {\delta x_i^{j(cpu)}} = 0$ for $\lambda > 0$.

$$\delta L(x_i^{j(cpu)}) \Big/ \delta x_i^{j(cpu)} = \frac{u_i^{j(cpu)}}{x_i^{j(cpu)}} - (1 + \lambda + \psi) p_i^{electricity} e_j \tag{14}$$

Let $\delta L(x_i^{j(cpu)}) \Big/ \delta x_i^{j(cpu)} = 0$

$$x_i^{j(cpu)} = \frac{u_i^{j(cpu)}}{(l + \lambda + \psi) p_i^{electricity} e_j} \tag{15}$$

Using this result in the constraint equation, we can determine $\omega = 1 + \psi + \lambda$ as

$$EC_j = \frac{1}{\omega} \sum u_i^{j(cpu)}$$

$$\omega = \frac{\sum u_i^{j(cpu)}}{EC_j}$$

We obtain

$$x_i^{j(cpu)^*} = \frac{u_i^{j(cpu) EC_j}}{p_j^{electricity} e_j \sum u_i^{j(cpu)}} \tag{16}$$

It means that public cloud providers allocate optimal $x_i^{j(cpu)^*}$ to host VM for private cloud application i while maximizing its own profit.

Using the same method, we getto maximize public cloud provider's revenue.

$$x_i^{j(ram)^*} = \frac{u_i^{j(ram) EC_j}}{p_j^{electricity} e_j \sum u_i^{j(ram)}} \tag{17}$$

$x_i^{j(ram)^*}$ is the unique optimal memory to host VM for private cloud application i while maximizing jth public cloud provider's revenue.

4.3. Optimization of Private Cloud Application Group

We assume that private cloud application i submits payment $u_i^{j(cpu)}, u_i^{j(ram)}$ to the public cloud provider j for CPU and memory required by a VM respectively. Private cloud application i is associated with utility function U_i. The utility function for private cloud application i depends on allocated cloud resources $x_i^{j(cpu)}, x_i^{j(ram)}$ required by VM. For the private cloud application group optimization problem, the private cloud application i gives the unique optimal payment to public cloud provider under the constraints to maximize the private cloud application's benefit.

The time taken by the private cloud application group s to complete ith application is:

$$t_s^i = \frac{p_j^{cpu}}{C_j^{cpu} u_i^{j(cpu)}} + \frac{p_j^{ram}}{C_j^{ram} u_i^{j(ram)}} \tag{18}$$

The private cloud application group optimization problem can be reformulated as follows.

$$\text{Max} \left(T_s - K \sum_j \Big(\frac{p_j^{cpu}}{C_j^{cpu} u_i^{j(cpu)}} + \frac{p_j^{ram}}{C_j^{ram} u_i^{j(ram)}} \Big) \right) + \left(E_s - \sum_j \Big(u_i^{j(cpu)} + u_i^{j(ram)} \Big) \right) \tag{19}$$

$$s.t T_s \geq \sum_{i=1}^{I} t_s^j, E_s \geq \sum_{i=1}^{I} \Big(u_i^{j(cpu)} + u_i^{j(ram)} \Big)$$

Let the pricing policy, $p^{cpu} = (p_1^{cpu}, p_2^{cpu}, \Lambda, p_j^{cpu})$, denote the set of CPU prices of all public cloud providers. $p^{ram} = (p_1^{ram}, p_2^{ram}, \Lambda, p_j^{ram})$, denote the set of RAM prices of all public cloud providers. The private cloud application i receives the resource proportional to its payment relative to the sum of the public cloud provider's revenue.

The Lagrangian associated with problem $U_{privatecloudgroup}$ for the private cloud application's utility is $L(u_i^{j(cpu)}, u_i^{j(ram)})$

$$\begin{aligned} L(u_i^{j(cpu)}, u_i^{j(ram)}) &= \left(E_s - \sum_j \Big(u_i^{j(cpu)}, u_i^{j(ram)} \Big) \right) \\ &+ \left(T_s - K \sum_j \Big(\frac{p_j^{cpu}}{C_j^{cpu} u_i^{j(cpu)}} + \frac{p_j^{ram}}{C_j^{ram} u_i^{j(ram)}} \Big) \right. \\ &+ \beta \left(E_s - \sum_j \Big(u_i^{j(cpu)} + u_i^{j(ram)} \Big) \right) \\ &+ \eta \left(T_s - K \sum_j \Big(\frac{p_j^{cpu}}{C_j^{cpu} u_i^{j(cpu)}} + \frac{p_j^{ram}}{C_j^{ram} u_i^{j(ram)}} \Big) \right) \end{aligned} \tag{20}$$

Where β is the Lagrangian constant. From Karush-Kuhn-Tucker Theorem we know that the optimal solution is given $\delta L \big/ \delta u_i^{j(cpu)} = 0$ for $\beta > 0$.

$$\delta L(u_i^{j(cpu)}) \Big/ \delta u_i^{j(cpu)} = -1 + K \frac{p_j^{cpu}}{C_j^{cpu} (u_i^{j(cpu)})^2} + \frac{p_j^{cpu}}{C_j^{cpu} (u_i^{j(cpu)})^2} \tag{21}$$

Let ${}^{\delta L}\!/_{\delta u_i^{j(cpu)}} = 0$ to obtain

$$u_i^{j(cpu)} = \left(\frac{(k\eta + k)p_i^{cpu}}{(1+\beta)C_i^{cpu}}\right) \tag{22}$$

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

$$(\theta)^{-1/2} = \frac{T_s}{\sum_{j=1}^{j}\left(\frac{p_j^{cpu}}{C_j^{cpu}}\right)}$$

We substitute θ to obtain $u_i^{j(cpu)^*}$

$$u_i^{j(cpu)^*} = \left(\frac{p_j^{cpu}}{C_j^{cpu}}\right)^{1/2} \frac{\sum_{j=1}^{j}\left(\frac{p_j^{cpu}}{C_j^{cpu}}\right)^{1/2}}{T_s} \tag{23}$$

Private cloud application i wants to pay $u_i^{j(cpu)^*}$ to public cloud provider j for CPU under the constraint.

Using the same method, we can get $u_i^{j(ram)^*}$ which is the payment of private cloud application i to public cloud provider j for CPU.

$$u_i^{j(ram)^*} = \left(\frac{p_j^{ram}}{C_j^{ram}}\right)^{1/2} \frac{\sum_{j=1}^{j}\left(\frac{p_j^{ram}}{C_j^{ram}}\right)^{1/2}}{T_s} \tag{24}$$

4.4. Private Cloud Local Optimization

Private cloud local optimization is conducted by private cloud applications; the private cloud application gives the payment to public cloud provider under the deadline constraint to satisfy the private cloud application's QoS needs. $\left(E_i^{privatecloud} - \sum_n y_i^n\right)$ is all jobs of private cloud application' cost budgets subtracting the payments to public cloud providers. $\sum_n t_i^n$ represents the execution time for processing all private cloud application's jobs. The objective of low-level scheduling optimization is to minimize the cost of private cloud application and complete all jobs as soon as possible. z_i^n is the computation task of ith private cloud application's nth job. The execution time taken by the ith private cloud application to complete nth job is:

$$t_i^n = \frac{z_i^n}{x_i^j y_i^n} \tag{25}$$

We reformulate

$$\text{Max}\left\{\left(E_i^{privatecloud} - \sum_n y_i^n\right) + \left(T_i - \sum_{n=1}^{N}\frac{z_i^n}{x_i^j y_i^n}\right)\right\} \tag{26}$$

The Lagrangian for $U^{privatecloudjob}$'s utility is $L(y_i^n)$.

$$L(y_i^n) = \left(E_i^{privatecloud} - \sum_n y_i^n\right) + \left(T_i - \sum_{n=1}^{N} \frac{z_i^n}{x_i^j y_i^n}\right) + \lambda\left(T_i - \sum_{n=1}^{N} t_i^n\right) \tag{27}$$

Where λ is the Lagrangian constant. From Karush-Kuhn-Tucker Theorem we know that the optimal solution is given $\delta L(y_i^n) / \delta y_i^n = 0$ for $\lambda > 0$.

Let $\delta L(y_i^n) / \delta y_i^n = 0$ to obtain

$$y_i^n = \left(\frac{(1+\lambda)z_i^n}{x_i^j}\right)^{1/2} \tag{28}$$

Using this result in the constraint equation, we can determine $\theta = 1 + \lambda$ as

$$(\theta)^{-1/2} = \frac{T_i}{\sum_{n=1}^{N} \left(\frac{z_i^n}{x_i^j}\right)^{1/2}}$$

We substitute θ to y_i^n obtain

$$y_i^{n^*} = \left(\frac{z_i^n}{x_i^j}\right)^{1/2} \frac{\sum_{n=1}^{N} \left(\frac{z_i^n}{x_i^j}\right)^{1/2}}{T_i} \tag{29}$$

It means that nth job of private cloud application i want to pay $y_i^{n^*}$ for the cloud resource under the deadline constraint.

5. Hybrid Cloud Scheduling Algorithms for Cloud Bursting

According to the model formulation and mathematic solutions of hybrid cloud scheduling, hybrid cloud scheduling algorithms for cloud bursting are proposed, which includes the routines of public cloud optimization, private cloud application optimization and private cloud job optimization.

Algorithm 1. Hybrid Cloud Scheduling Algorithms for Cloud Bursting (SACB)

`Routine 1. Public cloud_Optimization` $(u_i^{j(cpu)^{(n)}}, u_i^{j(ram)^{(n)}})$

---

1. `Public cloud provider` j `calculates its the optimal resource allocation` $x_i^{j(cpu)^*}$, $x_i^{j(ram)^*}$ `to support VMs for private cloud applications, while maximizing` $U_{publiccloud}$

$$x_i^{j(cpu)^*}, x_i^{j(ram)^*} = \mathrm{Max}\left\{U_{publiccloud}\right\}$$

2. Computes new cloud resource price of public cloud provider j according to the following formula

$$p_j^{cpu^{(n+1)}} = \max\left\{\varepsilon,\ p_j^{cpu^{(n)}} + \eta\Big(\sum_i x_i^{j(cpu)} - C_j^{cpu}\Big)\right\}$$
$$p_j^{ram^{(n+1)}} = \max\left\{\varepsilon,\ p_j^{ram^{(n)}} + \eta\Big(\sum_i x_i^{j(ram)} - C_j^{ram}\Big)\right\}$$

3. Public cloud provider j send new prices $p_i^{cpu(n+1)}, p_i^{ram(n+1)}$ to private cloud applications which lease cloud resources for running VM.
4. $p_j^{cpu(n+1)}, p_j^{ram(n+1)}$ to private cloud application

Routine 2. Private cloud application_Optimization (p_j^{cpu}, p_j^{ram})

1. If $E_s \geq \sum_{i=1}^{I}\Big(u_i^{j(cpu)} + u_i^{j(ram)}\Big)$

   Then $u_i^{j(cpu)^*}, u_i^{j(ram)^*} = \operatorname{Max} U_{privatecloudgroup} u_j^{j(cpu)}, u_j^{j(ram)}$;

   //calculates $u_i^{j(cpu)^*}, u_i^{j(ram)^*}$ to maximize the benefit of private cloud application group;
2. Return $u_i^{j(cpu)^*}, u_i^{j(ram)^*}$ to public cloud provider j.

Routine 3. Private cloud job_Optimization (x_i^j)

1. $y_i^{n^*} \operatorname{Max} U_{privatecloudjob}(y_i^n)$;
2. If $E_i^{privatecloud} \geq \sum_{n=1}^{N} y_i^n$

   Return $y_i^{n^*}$ to private cloud application i;
   Else Return Null;
3. Return $y_i^{n^*}$.

6. Experiments

In this section, the efficiency of the proposed hybrid cloud scheduling algorithms for cloud bursting (SACB) is demonstrated through simulations. The simulation environment of the hybrid cloud was set up with 5 public cloud providers, 20 private cloud providers and 100 private cloud applications. In the experiments, the cost of public cloud resource and the electrical energy are expressed in dollar that can be defined as unit resource or energy cost. The initial price of electrical energy in public cloud is set from 1 to 100 dollars. The initial price of VM is set from 10 to 500 dollars. Private cloud applications submit their jobs with varying deadlines. The deadlines of private cloud application are chosen from 100ms to 400ms. The budgets of private cloud applications are set from 100 to 1500 dollars. Each public cloud resource had its processing power randomly taken from the interval (100, 1000). Links between resources in the private cloud had their bandwidth randomly taken from (100, 1000). The external link of the private cloud which connects to the public clouds had its bandwidth randomly taken from (100, 1000). Each experiment is repeated 6 times and 95% confidence intervals are obtained. The simulation results shown in the figures represent mean values. Simulation parameters are listed in Table 2.

Table 2. Simulation Parameters

Simulation Parameter	**Value**
Total number of private cloud applications	100
Total number of private cloud providers	20
Total number of public cloud providers	5
Initial price of VM	[10,500]
Deadline	[100,400]
Expense Budget	[100,1500]
electrical energy	[0.1,1.0]
Bandwidth	[100,1000]
Computing power	[100,1000]
RAM	[100,2000]
Energy price	[1,100]

We compare our hybrid cloud scheduling algorithms for cloud bursting (SACB) with adaptive scheduling with QoS satisfaction algorithm for hybrid cloud AsQ [25], which aims at raising the resource utilization rate of the private cloud and to diminish task response time as much as possible. In [25], they exploit runtime estimation and several fast scheduling strategies for near-optimal resource allocation, the near-optimal allocation in the private cloud can reduce the amount of tasks that need to be executed on the public cloud to satisfy their deadline. For the tasks that have to be dispatched to the public cloud, they choose the minimal cost strategy to reduce the cost of using public clouds based on the characteristics of tasks such as workload size and data size. AsQ aims at achieving a total optimization regarding cost and deadline constraints.

We adopt four metrics, some of which was also used in [25], to compare the performance of proposed hybrid cloud scheduling algorithms for cloud bursting $(SACB)$ with hybrid cloud Adaptive Scheduling with QoS Satisfaction algorithm (AsQ): QoS satisfaction rate, cost, resource utilization and profit of public cloud provider. The simulation was carried out by varying some parameters: job arrival rate and the budget. Job arrival rate is job arrival speed, which will affect the system load.

The impacts of job arrival rate on QoS satisfaction rate, resource utilization, renting cost and profit of public cloud provider were illustrated in Fig.2, Fig.3, Fig.4, and Fig.5 respectively. Fig. 2 shows that the QoS satisfaction rate decreases when job arrival rate increases. When $a = 0.6$, QoS satisfaction rate of $SACB$ is as much as 26% lower than that by $a = 0.10$. The smaller is a, enough VMs are available for private cloud applications. The requirements of the private cloud users can be processed on time and these private cloud users experience higher user satisfaction. On the other hand, the larger is a, the lower

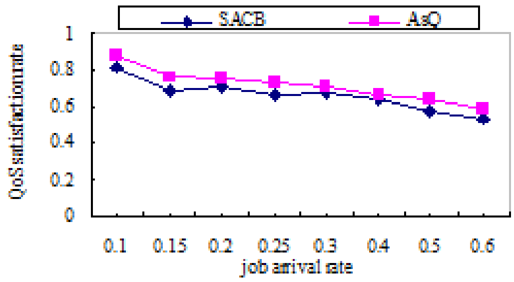

Figure 2. QoS satisfaction rate Vs job arrival rate

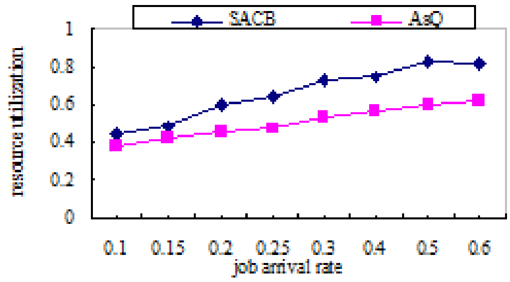

Figure 3. Resource utilization Vs job arrival rate

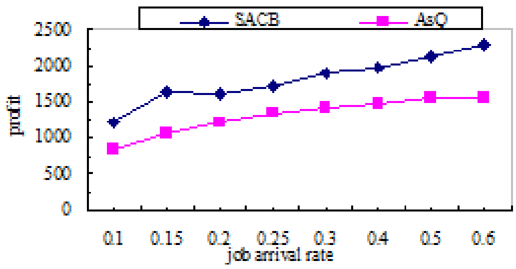

Figure 4. Profit of public cloud provider Vs job arrival rate

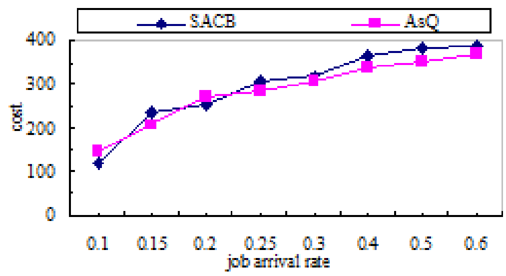

Figure 5. Cost Vs job arrival rate

is QoS satisfaction rate. When job arrival rate reaches $0.5(a = 0.5)$, QoS satisfaction rate of AsQ is as much as 13% higher than that using $SACB$. When job arrival rate increases, QoS satisfaction rate of AsQ and $SACB$ deteriorates. When the job arrival rate increases, less requests from private cloud users can be executed successfully before their deadline. But under same job arrival rate, QoS satisfaction rate of $SACB$ is not good as AsQ. AsQ algorithm doesn't consider optimization of cloud resource providers, it mainly wants to satisfy the private cloud user's applications. Fig.3 shows as job arrival rate (a) increases, resource utilization ratio increases. When $a = 0.4$, the resource utilization of $SACB$ is as much as 37% more than utilization AsQ. When the job arrival rate was very large, many jobs are sent to hybrid cloud system, cloud resources are busier. Under low job arrival rate $(a = 0.2)$, the resource utilization of $SACB$ is as much as 18% larger that AsQ. When increasing the job arrival rate by $a = 0.5$, the resource utilization of AsQ is as much as 31% less than that using the $SACB$. Fig.4 is to show the effect of job arrival rate on profit of public cloud provider. A higher job arrival rate brings out higher profit of public cloud provider. With the same job arrival rate, $SACB$ has higher profit of public cloud provider than AsQ. In $SACB$, public cloud provider can calculate suitable VMs for maximizing the revenue; so can achieve high profit of public cloud provider. When job arrival rate increases $(a = 0.5)$, the profit of public cloud provider of AsQ is as much as 22% less than $SACB$. Fig.5 is to show the effect of job arrival rate on renting cost. After the job arrival rate is more than 0.25, $SACB$ has little higher renting cost than AsQ. $SACB$ considers the benefit of the private cloud users and public cloud provider. Private cloud users calculate suitable payments for public cloud resources and maximize the revenue. The $SACB$ considers the tradeoff between profits of public cloud providers and the private cloud users. AsQ algorithm aims at raising the resource utilization rate of the private cloud and to diminish task response time as much as possible. When job arrival rate increases $(a = 0.4)$, resource cost of $SACB$ is as much as 11% more than AsQ.

How the budget effects on QoS satisfaction rate, renting cost, resource utilization and profit of public cloud provider were presented in following figures (Figs.6-9). The private cloud user budget is set from 100 to 1500. Fig.6 is to show the effect of budget on QoS satisfaction rate. When increasing budget values, the QoS satisfaction rate becomes higher. A larger budget enables private cloud user to afford more expensive public cloud services to complete the jobs before its deadline. For $SACB$, when the budget increases $(b = 1500)$, the QoS satisfaction rate is as much as 33% more than that with b=100. Under the same budget $(b = 800)$, the QoS satisfaction rate of $SACB$ is smaller than AsQ. Considering the profit of public cloud provider, from the results in Fig.7, when increasing budget values by $b = 1500$, the profit of public cloud provider of $SACB$ is 43% more than that by $b = 100$. Under same budget value $(b = 1500)$, the profit of public cloud provider of $SACB$ is 32% higher than AsQ. Because more private cloud users can pay cloud resources gotten from public cloud providers to run VMs under the budget, public cloud providers can maximize its profits. The profit of cloud provider of $SACB$ decreases when the expense budget decreases. When the budget is 1000 $(b = 1000)$, the profit of public cloud provider of $SACB$ is up to 35% than $b = 100$. Fig.8 represents the impact of different budget constraint on the renting cost. When the budgets are small, $SACB$ spends more time to complete jobs, because the private cloud user can't buy expensive cloud service with better VM configuration from public cloud provider. When the budget increases, the private cloud user can afford more expensive cloud service; one objective of $SACB$ is to maximize the utility of private cloud user under the budget and deadline constraints. When the budget is 1000 $(b = 1000)$, the cost of $SACB$ is 21% more than b=100. When increasing budget by $b = 1000$, the renting cost of AsQ is as much as 11% less than $SACB$. When $b = 1500$, the cost of AsQ increase to nearly 30% compared with $b = 250$. Considering the resource

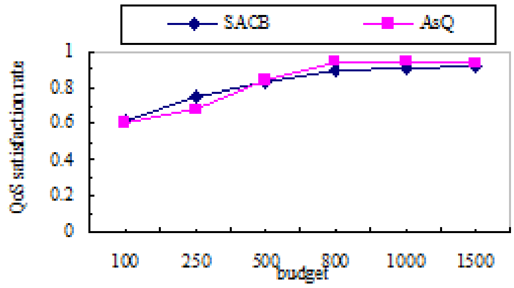

Figure 6. QoS satisfaction rate Vs. budget

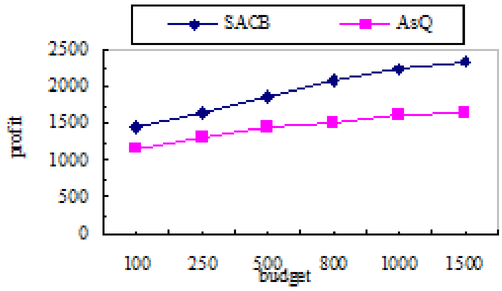

Figure 7. Profit of public cloud provider Vs. budget

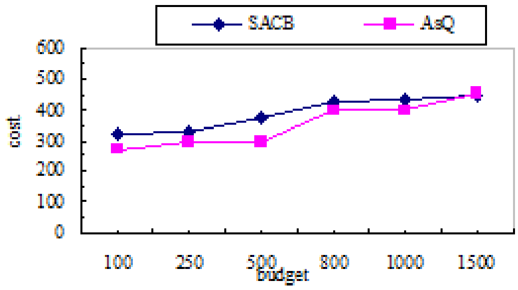

Figure 8. Cost Vs. budget

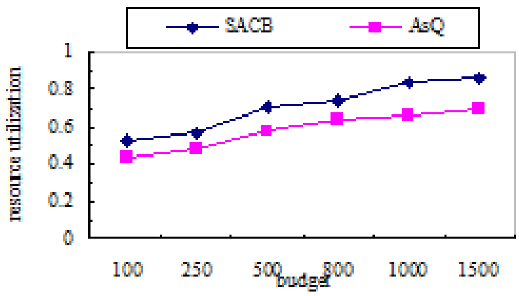

Figure 9. Resource utilization Vs. budget

utilization, from the results in Fig.9, when increasing budget values, the cloud resource utilization of $SACB$ becomes higher. A larger budget brings out higher resource utilization. When $b = 1500$, the resource utilization of $SACB$ is as much as 33% more than utilization by $b = 250$. Because when the budget decreases quickly, the private cloud user will be prevented from obtain cloud services. Under same budget $(b = 1000)$, the resource utilization of $SACB$ is 28% higher than AsQ.

7. Conclusions

The paper proposes a scheduling optimization scheme in hybrid cloud computing for cloud bursting. The scheduling optimization is conducted at different levels: hybrid cloud system level and private cloud local level. According to the model formulation and mathematic solutions of hybrid cloud scheduling, hybrid cloud scheduling algorithms for cloud bursting are proposed, which includes the routines of public cloud optimization, private cloud application optimization and private cloud job optimization. In the simulations, compared with other related algorithm, our proposed hybrid cloud scheduling algorithms achieve the better performance.

As future works, we consider other more scheduling optimization algorithms such as [34, 35] to improve the performance of our hybrid cloud scheduling algorithm.

Acknowledgements

The authors thank the editor in chief and the anonymous reviewers for their helpful comments and suggestions. The work was supported by the National Natural Science Foundation (NSF) under grants (No.61472294, No.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, Open Fund of State Key Laboratory of Software Engineering, Wuhan University (SKLSE). Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the above agencies.

References

[1] Borja Sotomayor, Ruben S. Montero, Ignacio M. Llorente, and Ian Foster: *An Open Source Solution for Virtual Infrastructure Management in Private and Hybrid Clouds*, IEEE INTERNET COMPUTING, 2009, 13(5), pp 14-22

[2] Oleksiy Mazhelis, Pasi Tyrväinen: *Economic aspects of hybrid cloud infrastructure: User organization perspective*, Information Systems Frontiers, 2012, 14(4), pp 845-869

[3] Oleksiy Mazhelis, Pasi Tyrväinen: *Role of Data Communications in Hybrid Cloud Costs*, 2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications, 2011, pp 138-145

[4] Prodromos Makris, Dimitrios N. Skoutas, Panagiotis Rizomiliotis: *A User-Oriented, Customizable Infrastructure Sharing Approach for Hybrid Cloud Computing Environments*, 2011 Third IEEE International Conference on Cloud Computing Technology and Science, 2011, pp 432-439

[5] Vecchiola, C., et al.: *Deadline-driven provisioning of resources for scientific applications in hybrid clouds with Aneka*. Future Generation Computer Systems, 2012, 28(1): p 58

[6] Ruben Van den Bossche, Kurt Vanmechelen, Jan Broeckhove: *Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds*. Future Generation Computer Systems. 2013, 29(4): pp 973-985

[7] Jiayuan Yue, Zhuo Zhang, Junhong Fu: *Extensible Architecture for High-Throughput Task Processing Based on Hybrid Cloud Infrastructure*, Electronics, Communications and Control (ICECC), 2011, pp 1452-1455

[8] Xuanjia Qiu, Hongxing Li, Chuan Wu: *Cost-Minimizing Dynamic Migration of Content Distribution Services into Hybrid Clouds*, INFOCOM, 2012 Proceedings IEEE proceeding IEEE, 2012, pp 2571-2575

[9] Faouzi Ben Charrada, Nourhene Tebourski, Samir Tata: *Risk-Aware Workload Distribution in Hybrid Clouds*. 2012 IEEE Fifth International Conference on Cloud Computing, 2012, pp 229-236

[10] Mustafizur Rahman, Xiaorong Li and Henry Palit: *Hybrid Heuristic for Scheduling Data Analytics Workflow Applications in Hybrid Cloud Environment*. 2011 IEEE International Parallel & Distributed Processing Symposium, 2011, pp 966-974

[11] Brock, M. and A. Goscinski: *Execution of Compute Intensive Applications on Hybrid Clouds (Case Study with mpiBLAST)*. in Complex, Intelligent and Software Intensive Systems (CISIS), 2012 Sixth International Conference on, 2012.

[12] Den Bossche R.V., Vanmechelen K., Broeckhove J.: *Cost-optimal scheduling in hybrid iaas clouds for deadline constrained workloads*. In: 3th IEEE International Conference on Cloud Computing, IEEE Computer Society, Los Alamitos, CA, USA, 2010, pp 228-235

[13] Linton Abraham, Michael A. Murphy, Michael Fenn, Sebastien Goasguen: *Self-provisioned hybrid clouds*, ICAC'10 Proceedings of the 7th international conference on Autonomic computing, 2010, pp 161-168

[14] Tekin Bicer, David Chiu, Gagan Agrawal: *Time and Cost Sensitive Data-Intensive Computing on Hybrid Clouds*, CCGRID '12 Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012), pp 636-643

[15] Bjorkqvist, M.: *Cost-driven service provisioning in hybrid clouds*, Service-Oriented Computing and Applications (SOCA), 2012 5th IEEE International Conference on, pp 1-8

[16] Alfonso Quarati, Andrea Clematis, Antonella Galizia, Daniele D. Agostino: *Hybrid Clouds brokering: Business opportunities, QoS and energy-saving issues*, Simulation Modelling Practice and Theory, published online 2013

[17] Daniele D'Agostino, Antonella Galizia, Andrea Clematis, Matteo Mangini, Ivan Porro, Alfonso Quarati: *A QoS-aware broker for hybrid clouds*, Computing, published online 2012

[18] Zinnen Andreas: *Deadline constrained scheduling in hybrid clouds with Gaussian processes High Performance Computing and Simulation (HPCS)*, 2011 International Conference on, pp 294-300

[19] Mohammad Mahdi Kashef, Jörn Altmann: *A Cost Model for Hybrid Clouds*, Economics of Grids, Clouds, Systems, and Services, Lecture Notes in Computer Science Volume 7150, 2012, pp 46-60

[20] Charrada, F.B.: *Approximate Placement of Service-Based Applications in Hybrid Clouds*, Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2012 IEEE 21st International Workshop on, pp 161-166

[21] Yongquan Fua, Yijie Wang, Ernst Biersack: *A general scalable and accurate decentralized level monitoring method for large-scale dynamic service provision in hybrid clouds*, Future Generation Computer Systems. published online 2012

[22] Bittencourt, Luiz F. Madeira, Edmundo Roberto Mauro; Fonseca, Nelson Luis Saldanha da: *Scheduling in hybrid clouds*, Communications Magazine, 2011, IEEE Vol. 50(9) pp 42-47

[23] Kumar B. A., Ravichandran T.: *Time and Cost Optimization Algorithm for Scheduling Multiple Workflows in Hybrid Clouds*. European Journal of Scientific Research, 2012, 89(2), pp 265-275

[24] Van Hoecke S., Waterbley T., Devos J., et al.: *Efficient management of hybrid clouds*, CLOUD COMPUTING 2011, The Second International Conference on Cloud Computing, GRIDs, and Virtualization. 2011, pp 167-172

[25] Wei-JenWang, Yue-Shan Chang, Win-Tsung Lo,Yi-Kang Lee: *Adaptive scheduling for parallel tasks with QoS satisfaction for hybrid cloud environments*, Journal of Supercomputing, published online, 2013

[26] Bittencourt L.F., Madeira E.R.M.: *HCOC: a cost optimization algorithm for workflow scheduling in hybrid clouds*. Journal of Internet Services and Applications, 2011, 2(3), pp 207-227

[27] Bittencourt L.F., Senna C.R., Madeira E.R.M.: *Scheduling service workflows for cost optimization in hybrid clouds*, Network and Service Management (CNSM), 2010 International Conference on. IEEE, 2010, pp 394-397

[28] Li Chunlin, Zhong Jin Xiu, Li Layuan: *Resource Scheduling With Conflicting Objectives In Grid Environments: Model and Evaluation*, Journal of Network and Computer Applications, Elsevier, Vol 32/3, pp 760-769, May, 2009

[29] Li Chunlin, Li Layuan: *Contexts Aware Service Provisioning in Mobile Grid*, Journal of Network and Computer Applications, Elsevier, 34(2), 2011, pp 774-782

[30] S. Subashini, V. Kavitha: *A survey on security issues in service delivery models of cloud computing*, Journal of Network and Computer Applications, Vol. 34(1), January 2011, pp 1-11

[31] Wei-Jen Wang, Yue-Shan Chang, Win-Tsung Lo, Yi-Kang Lee: *Adaptive scheduling for parallel tasks with QoS satisfaction for hybrid cloud environments*, Journal of Supercomputing, February 2013

[32] Pal R., Hui P.: *Economic models for cloud service markets*. Distributed Computing and Networking Lecture Notes in Computer Science, Vol. (7129), 2012, pp 382-396, 2012, pp 382-396

[33] D. Bertsekas: *Nonlinear Programming*, 2nd ed., Athena Scientific, 1999

[34] Yudong Zhang, Lenan Wu, and Shuihua Wang: *UCAV Path Planning by Fitness-Scaling Adaptive Chaotic Particle Swarm Optimization*, Mathematical Problems in Engineering, Vol. (2013), Article ID 705238, pp 9

[35] Zhang Y.D., Wu L.N., Wei G., Wang S.H.: *A novel algorithm for all pairs shortest path problem based on matrix multiplication and pulse coupled neural network*, DIGITAL SIGNAL PROCESSING, 2011, 21(4) pp 517