

Topology-based Workflow Scheduling in Commercial Clouds

Haoran Ji¹, Weidong Bao¹, Xiaomin Zhu¹ and Wenhua Xiao¹

¹Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology, Changsha 4100773-P.R. China
[e-mail: {hrji,wdbao,xmzhu,wenhuaxiao}@nudt.edu.cn]

*Corresponding author: Haoran Ji

*Received December 21, 2014; revised June 23, 2015; accepted August 21, 2015;
published November 30, 2015*

Abstract

Cloud computing has become a new paradigm by enabling on-demand provisioning of applications, platforms or computing resources for clients. Workflow scheduling has always been treated as one of the most challenging problems in clouds. Commercial clouds have been widely used in scientific research, such as biology, astronomy and weather forecasting. Certainly, it is very important for a cloud service provider to pursue the profits for the commercial essence of clouds. This is also significantly important for the case of providing services to workflow tasks. In this paper, we address the issues of workflow scheduling in commercial clouds. This work takes the communication into account, which has always been ignored. And then, a topology-based workflow-scheduling algorithm named Resource Auction Algorithm (REAL) is proposed in the objective of getting more profits. The algorithm gives a good performance on searching for the optimum schedule for a sample workflow. Also, we find that there exists a certain resource amount, which gets the most profits to help us get more enthusiasm for further developing the research. Experimental results demonstrate that the analysis of the strategies for most profits is reasonable, and REAL gives a good performance on efficiently getting an optimized scheme with low computing complexity.

Keywords: workflow-scheduling, cloud computing, profit-oriented, topology-based

1. Introduction

Cloud computing is an attractive solution to enable on-demand provision of computing resources, which meets the challenges of growing demands on computing, big data storage and so on [1]. Clients can enjoy the cloud computing service by a service level agreement, which defines their required Quality of Service (QoS) [2]. At present, specialization and modularity have become an important way to reduce costs for enterprises. Cloud optimizes the utilization of computing resource and decreases the risk of building a private computing center for client users. More and more enterprises do their tasks or researches on clouds to minimize the execution costs. Although, cloud operators can build large computing centers with thousands of computers and servers to satisfy the users' need for computing. However, to meet the challenges of the explosive growth of data in macro level and fluctuative demand for services, it is less efficient and high load solution of employing more computers and servers. Optimizing the service provision in service level is still one of the most efficient methods to optimize the service performance in clouds.

To research on the scheduling in cloud computing, the difference among some similar concepts should be studied first. Grid computing is also an important concept in the area of distributed computing. As workflow scheduling in grids has been well researched, so it is necessary to figure out what is special for the scheduling problem in cloud computing comparing with grid computing. The distinction of cloud computing is its service mode. Almost all current commercial clouds employ the “pay-as-you-go” pricing model that charges users based on the number of the time intervals and the amount of resource utility. The main objective of the commercial clouds is getting more profits. So, the profits can be treated as the basic motivation of clouds, especially for commercial clouds.

One type of the most important applications for cloud computing is the scientific research. Modern collaborative scientific researches usually involve a lot of works, such as structural biology, high-energy physics and astronomy, and there exist data interactions between correlative work steps. Usually we use workflows to present the correlations. The workflow will usually be described as a Directed Acyclic Graph (DAG), in which a node denotes a task and a directed edge denotes the data or control dependency between tasks. Workflow scheduling in clouds is always treated as a challenge. To the wide of our knowledge, there still lack of solutions for providing services to workflow applications. It is mainly because of the precedent constraints between tasks. And the pricing model of charging the clients by the number of time intervals and resource amount they have used, which most used in commercial clouds, also contributes to the complication. In this paper, we normally address the issues of workflow scheduling in commercial clouds in view of cloud service providers. Generally speaking, the paper concerns the scheduling problem of workflow in commercial clouds and a topology-based workflow-scheduling algorithm is proposed, which acts as a novel approach for the problem and present good performance in our experimental evaluation.

The remainder of the paper is organized as follows. The next section reviews the related work and points out the shortness of them for the issues of workflow scheduling. Section 3 formally models the workflow-scheduling problem in clouds. Section 4 analyzes the search process for optimal schedule based on a simple instance and gives the rules for fast searching for the optimum schedule. Following these work, Section 5 proposes a topology-based workflow scheduling algorithm to address the issues. The performance evaluation experiments are conducted and experimental results are given in Section 6. Finally, Section 7

concludes the work and main contributions of this paper and presents the future works we will do.

2. Related Work

Many heuristic methods have been proposed for service scheduling in homogeneous and heterogeneous distributed systems. Topcuoglu in [3] presents two novel scheduling algorithms for a bounded number of heterogeneous processors with the objective to simultaneously meet high performance and fast scheduling time, which have been called the Heterogeneous Earliest-Finish-Time (HEFT) algorithm and the Critical-Path-on-a-Processor (CPOP) algorithm. Bajaj et al. introduced a task duplication-based scheduling algorithm for network of heterogeneous systems (TANH), with complexity of $O(V^2)$, which provides optimal results for applications represented by directed acyclic graphs (DAGs). They also provided a simple set of conditions on task computation and network communication time that could be satisfied [4]. Rahman et al. proposed a dynamic critical path (DCP) based workflow scheduling algorithm that determines efficient mapping of tasks by calculating the critical path in the workflow task graph at every step [5]. Gu constructed analytical cost models and formulate the workflow mapping as optimization problems [6]. Further he developed a workflow mapping algorithm based on a recursive critical path optimization procedure to minimize the latency and conducted a rigorous workflow stability analysis. Besides he developed a layer-oriented dynamic programming solution based on topological sorting to identify and minimize the global bottleneck time. Abrishami et al. proposes a new QoS-based workflow-scheduling algorithm named partial critical paths (PCP), which makes efforts on minimizing the cost of workflow execution while meeting deadline constraints [2]. In addition, Lin et al. in [7] use queuing game model to scheduling problem in service level of cloud. Hassan et al. in [8] presents a VM resource allocation model that dynamically and optimally utilizes VM resources to satisfy QoS requirements of media-rich cloud services or applications. Mateo et al. in [9] point out that the key issue in providing fast and reliable access on cloud services is the effective management of resources and he proposes an adaptive resource management for cloud systems which supports the integration of intelligent methods to promote QoS.

For workflow scheduling in clouds, Xu introduced a multiple QoS constrained scheduling strategy of multi-workflows (MQMW) to meet the challenges of Cloud service characters [10]. Based on previous work in [2], Abrishami extended the PCP algorithm and proposed two workflow scheduling algorithms, which called Infrastructure as a Service (IaaS) Cloud Partial Paths (IC-PCP), and IaaS Cloud Partial Critical Paths with Deadline Distribution (IC-PCPD2) [11]. On the analysis of clouds, Kllapi et al. considered that the optimization criterion is at least two-dimensional and presented an optimization framework, and then they incorporated the devise framework into a prototype system [12]. Pandey et al. presented a particle swarm optimization (PSO) based heuristic to schedule workflows account of the computation cost and data transmission cost [13]. Wu et al. proposed a market-oriented hierarchical scheduling strategy in Cloud workflow systems and described the Cloud workflow scheduling with four layers, i.e. application layer, platform layer, unified resource layer and fabric layer [14]. Yu and Buyya proposed a budget constraint based scheduling [15]. The algorithm minimizes execution time while meeting a specified budget for delivering results. And then they proposed a new type of genetic algorithm to solve the optimization problem. Maciej Malawski et al. concern the efficient management of large-scale scientific workflow applications with the constraints of budget and deadline in IaaS clouds. They develop, and assess novel

algorithms based on static and dynamic strategies for both task scheduling and resource provisioning and demonstrate that an admission procedure based on workflow structure and estimates of task runtimes can significantly improve the quality of solutions [16]. Linfang Zeng et al. address the issues of combining between the task scheduling and data management in [17] and introduce an adaptive data-aware scheduling (ADAS) strategy for workflow applications. Hassan M.M. takes multimedia cloud as one important paradigm of cloud computing and makes lots of efforts on optimization of the cloud resources allocation dynamically with regards of the QoS, energy consumption and budget [18]. And Biao Song et al. in [19] follow the work of [18] to address the challenges by using a queuing based approach for task management and a heuristic algorithm for resource management. Bittencourt L F et al. follow the work in [20] and concern the scheduling problem of workflows in hybrid clouds, in which tasks can be allocated either on a private cloud or on a public cloud, and emphasize the communication capacity with a prominent importance [21].

In these work, although Pandey in [13] takes communication into account, the computing complexity of the proposed algorithm is too high to meet the challenges of big size workflows. With the efforts of [13] and [21], the importance of communications has been well studied, which presents prominent influence on the performance of clouds. Although the topologic characteristics can make help on optimize the scheduling in clouds [16], the proposed methods do not make full use of these parameters.

3. Problem Formulation

In this paper, we formulate the workflow-scheduling problem in clouds with four basic aspects: workflow model, service model, cost model and objective model.

3.1 Workflow Model

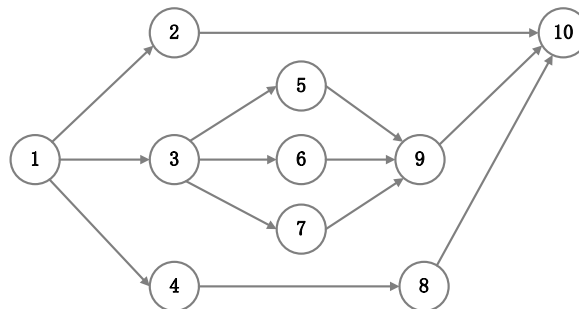


Fig. 1. DAG of a sample workflow

Normally, a workflow is a set of tasks, denoted as $T = \{t_1, t_2 \dots t_n\}$, which will be submitted to the clouds for execution. A set $E = \{parent, child\}$ represents a precedent constraint that used to indicate the dependent relationships between tasks. A workflow can be modeled with a DAG as $G = \{T, E\}$, where $E = \{e_{ij}\}$, $i, j \in [1 \dots n]$, $t_i, t_j \in T$. The DAG of a sample workflow can be described as in Fig. 1. The edge denotes the data dependence between t_i and t_j [22]. This paper normally addresses the issues of single workflow scheduling problem. For multi-workflows, it can be translated to single workflow scheduling problem by adding a virtual entry node and a virtual exit node with execution time as 0. In brief, it can well solve the workflow scheduling problem by proposing an algorithm executed on single workflow.

In G, the definition of $t_i = \text{parent}(t_j)$ or $t_j = \text{child}(t_i)$ describes that the execution of t_j depends on the completion of t_i . For this case, t_i is called the parent task of t_j and t_j is the child task of t_i oppositely. Each $t_i \in T$ has the properties of execution time et_i , start time st_i and finish time ft_i . The relation among these properties can be described as $et_i = ft_i - st_i$. And for a task, two states are used to describe its status during the scheduling: not ready and ready for scheduling. A task would be ready on condition of all its parent tasks are scheduled. Each e_{ij} will take a transmission time, denoted as tt_{ij} . So for $t_i = \text{parent}(t_j)$, there is $ft_i + tt_{ij} < st_j$.

3.2 Service Model

The most important component in clouds is the computational resource, denoted by a set of $R = \{r_1, r_2 \dots r_n\}$. There are mainly three kinds of resources, which are the computing resources, denoted by a set of $CR = \{cr_1, cr_2 \dots cr_n\}$, and storage resources, denoted by a set of $SR = \{sr_1, sr_2 \dots sr_n\}$, the network resources, denoted by a set of $NR = \{nr_1, nr_2 \dots nr_n\}$. The NRs act with growing importance in clouds for the larger cluster of computational resources [23], which have always been ignored in former researches. CR and SR are usually combined together as a virtual machine (VM). And in our research, clouds assign VM, which contains heterogeneous CRs and SRs, to the tasks.

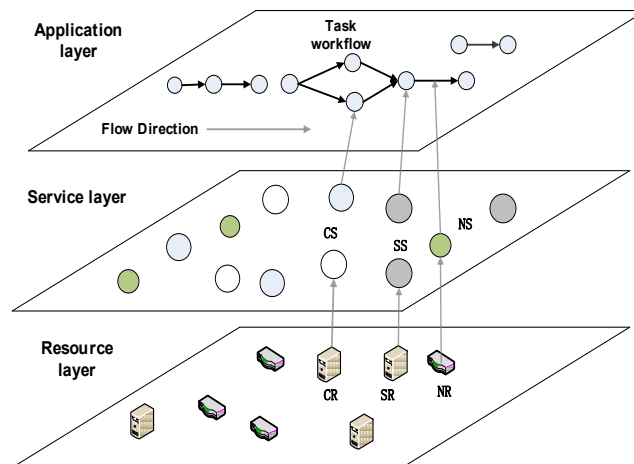


Fig. 2. Cloud services for workflows

Clouds mainly provide two kinds of services: resource services (Infrastructure as a service, IaaS) and application services (Platform as a service/Software as a service, PaaS/SaaS). Resource services provide resource as a service to remote clients. Application service allows remote clients to use their specialized applications, unlike resource services, which are capable of providing estimated service times based on the metadata of users' service requests. The work in this paper mainly concerns the issues of resource service provision, which is wide employed around the world, for instance the Amazon EC2. Normally, clouds provide services to users through offering accesses of resources and their combination. A service s_{ij} is the mapping relationship between t_i and r_j . Computing service (CS) provide applications to meet the needs for computing. Storage service (StS) offers space in the cyberspace, which is brought forward to describe the space consists of computers, data, networks and so on. Computing service will always go with temporary storage and clouds also provide the storage service, which cannot be considered separately with the network services (NS). Network

services ensure the communications between users and the data transmissions between tasks by constructing the links. The service is an access to the corresponding resource. After have paid for the service, you get the right to use the corresponding resources. Service is the key characteristic in clouds. The service providing process can be described as in Fig. 2.

Considering the importance of execution time in workflow scheduling, a developed DAG is used to describe the mapping relationships of tasks and the resources as in Fig. 3. The model is a combination of the Gantt chart and the DAG, which gives a more detailed description on the workflow tasks. A time axis is used to describe the arrival time a_i and y axis to part the different resources. In the developed DAG model, we can clearly get the knowledge of the service s_{ij} , which denotes the assignment of resource r_j and t_i . It also presents the data transition time between parent-child tasks in the same resource and the communication time for the tasks with data dependence on different resources. Time is an important element for scheduling in clouds, which is not shown in traditional DAG, while in the developed DAG, it is highlighted and directed perceived.

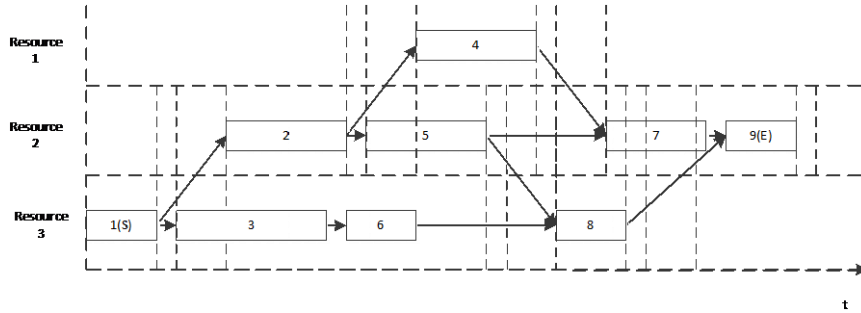


Fig. 3. Developed DAG model

The developed DAG model can describe the workflow scheduling process clearly. The major work of the scheduler is to serve the tasks with suitable resources. The match between T and R is done will present a performance. If a match, who can also be called a schedule, gives better performance under the chosen objective, the schedule will be considered as being better.

3.3 Cost Model

Most commercial clouds, for example the Amazon Elastic Computing Cloud, charge by the service time based on users' choice of the instance modes [24]. A set $SC = \{sc_1, sc_2 \dots sc_n\}$ of service cost maps to the workflow $T = \{t_1, t_2 \dots t_n\}$. Each $sc_i, i \in [1, n]$ is the execution cost of corresponding t_i . A service cost mainly consists of three parts as execution cost, storage cost and network cost. The execution cost of CPU, denoted as $CE = \{ce_1, ce_2 \dots ce_n\}$, is generated for the use of computing resource in clouds or in the data center. And the storage cost $StC = \{stc_1, stc_2 \dots stc_n\}$ is associated with the data storage. The StC differs with different data size, security level and time length. In commercial clouds, the expense is directly related with EC and StC. A set $NC = \{nc_1, nc_2 \dots nc_m\}$ of network cost arises with the communications between resources, which generated during the use of NR. For $t_j = child(t_i)$, if t_i and t_j are executed on the same resource, there will be no NC. t_j can easily get the data produced by t_i . As the transmission time of data flowing from t_i to t_j is so small that it can be nearly ignored. Furthermore, the communications between tasks executed on different resources contribute to SC. SC is mainly consisted by these three factors. So SC can be described as:

$$SC = EC + StC + NC \quad (1)$$

In equation (1), the EC and StC are fixed for a certain workflow. So, the most efficient method of optimizing the scheduling is to reduce the NC.

3.4 Objective

For the inherent essence of clouds, profits are treated as the main object in this work. The profits, denoted as p , of a cloud service can be computed by two parameters: the service price (SP) and the service cost (SC). Service price is the income and the service cost is the expending. We use a ratio as the description of p .

And also quantity of service (QoS) is an important factor, which influences the service price. Clouds will normalize the services with heterogenous capacities of resources so as to facilitate the charging. Therefore, QoS in this work is mainly decided by the service time. Extra waiting will decrease the QoS. To describe this influence, we introduce extra execution cost (EEC). It is a measure of the service quality. Reducing QoS means generating EEC. And QoS in this work is mainly decided by the execution time. So, EEC can be treated as a monetized measure of the time lateness. Providing new resources will also take extra cost. However, users will only be pleased to pay for CS. Clients will not pay for uncertain charges. Therefore, minimizing the EEC will directly contribute to minimize the SC [25]. For some certain workflow, it contributes to maximize the profits. The objective can be described as:

$$p \propto SP/SC = EC/(SC + EEC) \quad (2)$$

With equation (1) and (2), we get:

$$\max p \propto \max \sum_G EC / \sum_{G'} (EC + EEC + StC + NC) \quad (3)$$

G' denotes total resources used in the scheduling and the idle resources will also charge. So if the CR is fully used, the objective can also be described as:

$$\max p \propto \min \sum_G (StC + NC + EEC) / EC \quad (4)$$

To get higher p for fixed workflow tasks, optimizing scheduling algorithm by minimizing the NC and EEC while the EC and StC are relatively fixed will make effects. The work in this paper aims at proposing better workflow scheduling algorithm in task-service level. Furthermore, the efforts of this work are to reduce the EEC and NC. For data transmission between different resources will delay the ready time, NC can also be presented in EEC. In the experiments of this paper, EEC can be regarded as a measure of the execution time.

4. Optimizing the Workflow Scheduling

Some works have pointed out that topologic characteristics besides the critical path will facilitate the searching of optimal schedule. In this paper, three topologic parameters are introduced as the out-degree, in-degree and path length [26]. The definitions of these topologic characteristics are presented as follows.

Definition 1 Degree: the number of edges connecting to the vertex, with loops counted twice, in a directed graph.

Definition 2 Out-degree: the number of edges coming from the vertex in a directed graph.

Definition 3 In-degree: the number of edges coming to the vertex in a directed graph.

Definition 4 Path length: the longest distance an object travels in the direction of the edge to the end vertex in a directed graph.

Although some researchers have found that topologic characteristics can make help on workflow scheduling, but works on how to make full use of them is still in shortness. In the DAG, a task with bigger out-degree is more important as it influences more tasks. The lateness on parent task will lead to the *st* lateness of its child task. So, the higher out-degree a task node with, the more tasks it influences. For the nodes with bigger in-degree, *st* is decided by the latest finished task of its parent tasks. There is more risk just as any one of its parent nodes falling down will lead to its hanging up. So the parent tasks should be finished as early as possible. However, sometimes, above actions will make no effects at all, because there are sufficient. And we also introduce path length into the scheduling. Path length is a description of remaining tasks succeeding the task. For it does not contain the *tt*, it is not a precise description of the remaining time for executing following tasks. Smaller path length leads to better fault tolerance. It will make effects only when there are too many tasks later. Some other tactics for instance parallel execution is also helpful to reduce the EEC, especially for the task with big out-degree. So giving higher priority to these tasks will reduce the EEC. And the remaining time is also considered in our scheduling. A task with more remaining tasks and shorter remaining time will be executed with priority.

4.1 Problem Analysis

This section will give analysis on the proposed solution and following assumptions are made to simplify the analysis. The assumptions are not employed in the experimental evaluation. A single workflow will employ certain amount of cloud services, which means the fixed available resources for a workflow. This does not means this work is static scheduling. The assumption aims at facilitating the description of the proposed solution. This section also makes assumption that the et_i for each t_i is one time unit. In a developed DAG, we can split the t_i to several serial tasks, whose $et = 1$, with the amount as the value of et_i . So the task, whose $et = n$, can be transformed to n tasks whose $et = 1$, just as in Fig. 4. It proves that above assumptions do not exceed the practical applications of the problem and do not change the kernel of the workflow scheduling problem in clouds.

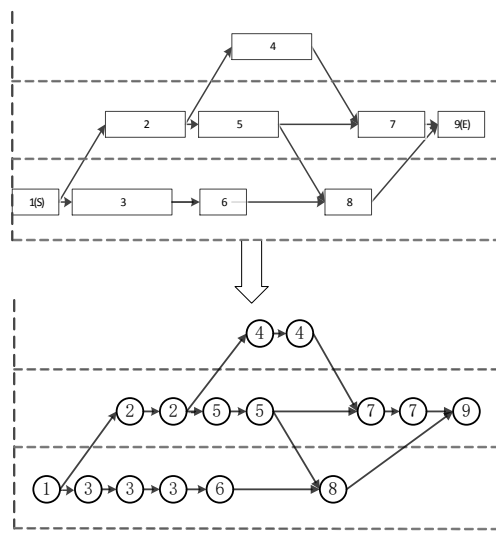


Fig. 4. Assumptions for the problem analysis.

Analysis in this section is established on these assumptions for the sake of simplicity. And in this section, we assume that the data transmission time tt is also one time unit as $tt_{ij} = 1, i, j \in [1, n]$. For the transmission data size in a workflow is pre-fixed and the environment in a computing center is generally similar. So we fix the tt as one time unit is available. It will make no difference on the performance of proposed workflow scheduling method in commercial clouds.

With above assumptions, the path length can be computed as the number of tasks from the node to the end node of workflow. For the sample workflow in Fig. 1, several possible schedules are tabled as in Fig. 5. Among all these schedules, schedule 2 is the optimum, which will take the maximal p .

Schedule 1	resource 1	1	2	3	5	6		9	10
	resource 2		4	8		7			
Schedule 2	resource 1	1	4	2	8				
	resource 2		3	5	6	7	9	10	
Schedule 3	resource 1	1	4	2	8				
	resource 2			3	5	6	7	9	10
Schedule 4	resource 1	1	4	2					
	resource 2		3	5	8	6	7	9	10

Fig. 5. Possible schedules of the sample workflow

Comparing the schedules in Fig. 5 and their topology characteristics tabled as in Table 1, we try to find the relationships between the optimal schedule and its topology characteristics. The differences between schedule 1 and schedule 2 begin with the task execution in the second step. In schedule 1, t_2 is executed on r_1 , while in schedule 2, t_3 is executed on r_2 . The main differences between schedule 2 and 3 is that schedule 2 parallel executes t_1 on two resources of r_1 and r_2 . In schedule 4, we can improve it for better performance but the NC is bigger than that in schedule 2. Compared with table 1, t_3 has bigger out-degree than t_2 . The child node of t_5, t_6 and t_7 has bigger in-degree than t_8 . To confirm our supposition, we also make analysis on many other workflows and it proves the relations between the topology parameters and optimal schedule.

Table 1. Topologic characteristics of the sample workflow

Task	In-degree	Out-degree	Path length
1	0	3	4
2	1	1	1
3	1	3	3
4	1	1	2
5	1	1	2
6	1	1	2
7	1	1	2
8	1	1	1
9	3	1	1
10	3	0	0

4.2 Scheduling Strategies

With above analysis, we set the rules of workflow scheduling as follows:

Rule 1: Execute the task node with bigger out-degree.

Rule 2: Execute the parent nodes of node with bigger input degree.

Rule 3: The node with longer path length has priority.

Rule 4: Execute the task that arrives earlier.

Rule 5: Take parallel execution of big out-degree task on idle resource.

These rules will be employed on the ready tasks and the rule with smaller number has higher priority. A task under more rules has higher priority. Since the in-degree and out-degree are both partial topology characters, it is suitable for complex workflows. If a task was scheduled, we will drop it and renew the workflow topologic information. Likewise, we schedule the sample workflow with proposed rules. To give a clear description, we take an instance on the sample workflow as in Fig. 6.

Step	Scheduling								Rule
Step 1	resource 1	1							Rule 5
	resource 2	1							
Step 2	resource 1	1	3						Rule 1, 3
	resource 2	1	4						
Step 3	resource 1	1	3	5					Rule 2
	resource 2	1	4	6					
Step 4	resource 1	1	3	5	7				Rule 4
	resource 2	1	4	6	2				
Step 5	resource 1	1	3	5	7	8		10	
	resource 2	1	4	6	2	9			
Schedule 2									
	resource 1	1	4	2	8				
	resource 2	1	3	5	6	7	9	10	

Fig. 6. Case study of proposed workflow scheduling method on the sample workflow

For the sample workflow, there is only one ready task as t_1 when the workflow arrives. And there are two free resources of r_1 and r_2 , so we take parallel execution of t_1 . After that, there are three ready tasks as t_2 , t_3 and t_4 . For t_3 has the biggest out-degree, so it is executed on r_1 . And for t_4 has longer path length than t_2 , it is executed by r_2 . When t_3 and t_4 are finished, there are t_2 , t_5 , t_6 , t_7 and t_8 waiting. They have the same out-degree and in-degree of their child tasks. Therefore, we execute t_5 and t_6 , because of their longer path length than t_2 . Then, t_7 and t_2 are executed, as t_2 arrives earlier than t_8 . The resource leisure is because we need to transmit the data to r_1 . It is obviously that the proposed method gets better p, which is the smallest for all possible schedules, than schedule 2. Therefore, the proposed method presents to be effective on the sample workflow. And some other cases are also studied, which also demonstrate that the proposed method perform well for searching the optimal solution for scheduling the workflows.

5. The REAL Algorithm

Based on above scheduling tactics, a topology-based workflow scheduling algorithm is proposed, which named as the **RE**source **A**uction **a**lgorithm (REAL) to solve the problem in commercial clouds. It is a typical implementation of heuristic method, which can well appropriate for dealing with big size workflows and practical cloud computing tasks. The inputs and output are described as in [Algorithm 1](#).

Algorithm 1. I/O of REAL

Input:

Workflow matrix m : the workflow task described as a DAG
 Resource amount N : the resource amount provided by clouds
 Cost of each cloud operations;
 Pre-set priority weight p_i for each rule

Output:

Schedule matrix sm : the schedule generated by REAL

An adjacency matrix is used to describe the workflow. For a workflow G with n tasks, an n -matrix m , the dimensions of which correspond to the tasks, is used. If there is dependent relation from t_i to t_j , we set $m_{ij} = 1$, else set it 0. With the matrix we can easily find the topologic characteristics of the DAG and operations on the workflow will be more flexible. The out-degree of t_i equals to the number of 1 in the i th row of m . And in-degree of t_i equals to that in the i th column of m . Also, we can get the path length just by scanning m . Another $m \times n$ matrix sm is used to describe the schedule. m is the number of resources and n is the maximum time span of scheduling. If we assign resource r_i to task t_j at time k , there is $sm_{ik} = t_j$. If there is no sm_{ik} , set it 0.

The REAL can be divided into three parts: tasks states scanning, resource competition and end detection. The first part aims at finding the prepared tasks. For the inherent constraints of workflow, only the ready tasks can be scheduled. Normally, the costs of CR are much bigger than the NC for the reason of CR would consumes more energy, takes bigger storage space and enjoys shorter life time. Consequently, REAL mainly focus on the fully use of CR. For each CR, all ready tasks (ready to execution on the resource) will compete for the resource. Under the proposed rules, the task with higher priority will be assigned to execute on the resource, just as the auction in which more powerful one would win the auction. In this paper, we set the service as the auction goods. And the free resource will be on the auction waiting for competing. The tasks compete for the goods by their priority generated by the topologic characteristics. If a task has been executed, it will get out of the auction. Finally, the auction will be completed when all the tasks are executed. That is what the end detection does.

First, REAL will scan the task states in real time to find which task is prepared for execution. A task will be prepared for scheduling when all its parent tasks have been completed. The process is presented as in [Algorithm 2](#).

Algorithm 2. Task states scanning

Import: topologic characteristics of workflow

Chair $\leftarrow sm_{ik}$

For each unscheduled $t_i \in T$

For each $t_j = \text{parent}(t_i)$

If in-degree of $t_i = 0$ & $tt_j + tt_{ji} < k$

```

State ( $t_i$ )=ready;
Else
State ( $t_i$ )=not ready;
PT= $t_i$ ;
End if
End for
End for

```

In REAL, each ready task will compete for the free resource. And if there are more than one resources get free, they will be in the auction by a fixed order. The detailed scheduling process of REAL is presented as in [Algorithm 3](#).

Algorithm 3. REAL

Import: prepared tasks, topologic characteristics of workflow
Chair $\leftarrow r_i$
Get the topologic characteristics of prepared tasks;
Task states scanning;
For each prepared t_j
If out-degree > 0 && there is extra free resource
Parallel execute t_j , no more than out-degree;
End if
If t_j will miss the deadline
Set sm_{ij} ;
End if
Set out-degree of t_j as $R_1(t_j)$;
Set in-degree of t_j as $R_2(t_j)$;
Set path length of t_j as $R_3(t_j)$;
Set arrival time of t_j as $R_4(t_j)$;
 $Priority = \sum_i \omega_i \cdot R_i(t_j)$;
End for
Set t_k with biggest priority to executed on r_i ;

The weight of each rule can be adjusted by experiments for different backgrounds. In this paper, we set the weight with 10 times difference to guarantee the priority of each rule. And these settings are well fitting on the problem assumptions in this paper.

6. Performance Evaluation

To evaluate the performance, this paper makes the following contributions. First, it designs a workflow generator, which generates three classic scientific workflows. Secondly, it makes analysis on the relationship between resource amount and profit under REAL. Based on these efforts, the paper evaluates the performance of REAL by employing it on different workflows. Two comparing algorithms are used separately as the critical path algorithm (CPA), which is mainly focus on the path length, and a developed CPA named ID-CPA. Former work in [2] has proved that the algorithm of CPA gives a good performance on scheduling the workflows. So rule 3 is proved to be effective. This work develops the CPA by introducing rule 2 and rule 3 to guide the scheduling named in-degree and critical path algorithm (ID-CPA). In the

experiments, we compare the REAL with ID-CP and CPA to evaluate its performance. Finally, an analysis on the computing complexity of REAL is given to describe the practicability of REAL. A laboratory cloud environments with seven PCs and structured by the Cloudstack is used in this experiments. Each PC has two 3.3 GHz CPUs and 4G RAM.

6.1 Workflow generation

Scientific workflow is one kind of important applications in clouds and especially for the workflow scheduling in commercial clouds. We take three classic scientific workflows [27], which are Montage, Cybershake and LIGO (in Fig. 7), and one instance workflow shown in Fig. 1 to evaluate the performance of proposed algorithm. The parameters of scientific workflows used in this section are given as in Table 2.

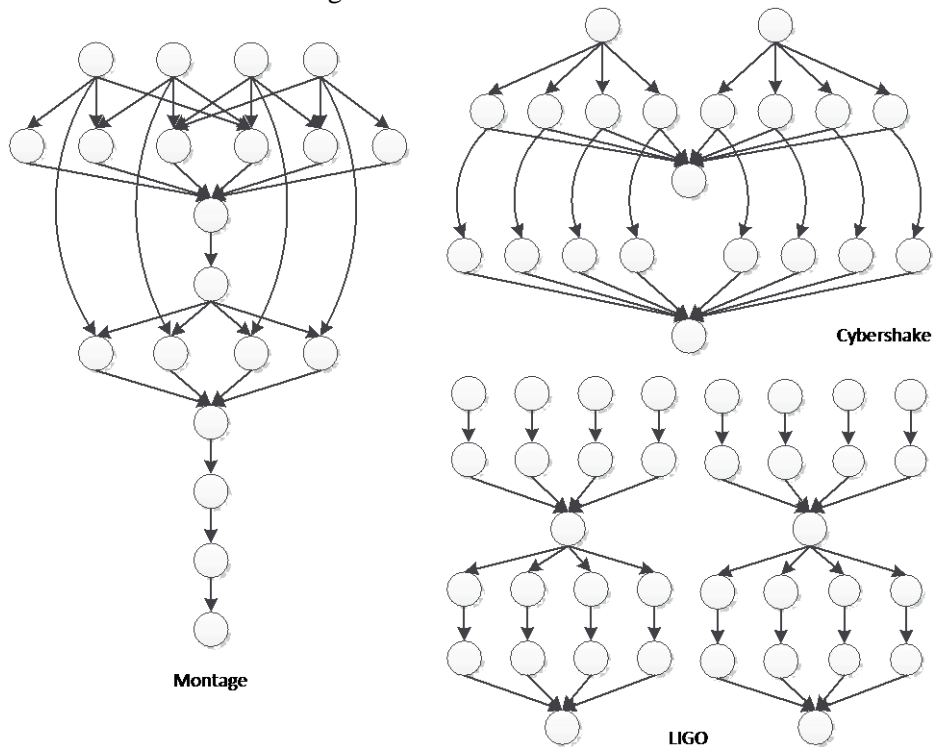


Fig. 7. Three classic scientific workflows

Table 2. Parameters of the tested workflows

Workflow	Montage	Cybershake	LIGO	Sample
Count:	203	1053	166	10
Topology	mProject:45 mDiffFit:107 degree:9	Extract-sgt:19 Seismogram_synthesis:516 PeakValCalC_Okaya:516	Tmpltbank:34 Insprial:76 Thinca:34	
Assumptions	$et = 1$ $nc = 1$	$et = 1$ $nc = 1$	$et = 1$ $nc = 1$	$et = 1$ $nc = 1$

The first three workflows are abstractions of actual workflows. For instance, NASA employs Montage for the generation custom mosaics of the sky. In this paper, we set task connection degree as the description of the tasks. Bigger degree promises tighter connection in workflow. Cybershake is used by the Southern California Earthquake Center to characterize earthquakes. LIGO is used by the Laser Interferometer Gravitational-wave Observatory to

analyze galactic binary systems. The instance workflow is abstracted from our research experience. It is the process of work assignment and results aggregation of lab administrative affairs.

6.2 The best resource amount

This experiment evaluates the performance by changing the resource amount to test the relations between resource amount and servicing cost. Some experimental assumptions are set as: st_i equals to the path length to the start node; the expected et , denoted as eet , is 1; the relationship between et and EEC can be computed as $EEC = et - eet$. And this equation is also used in the following experiments.

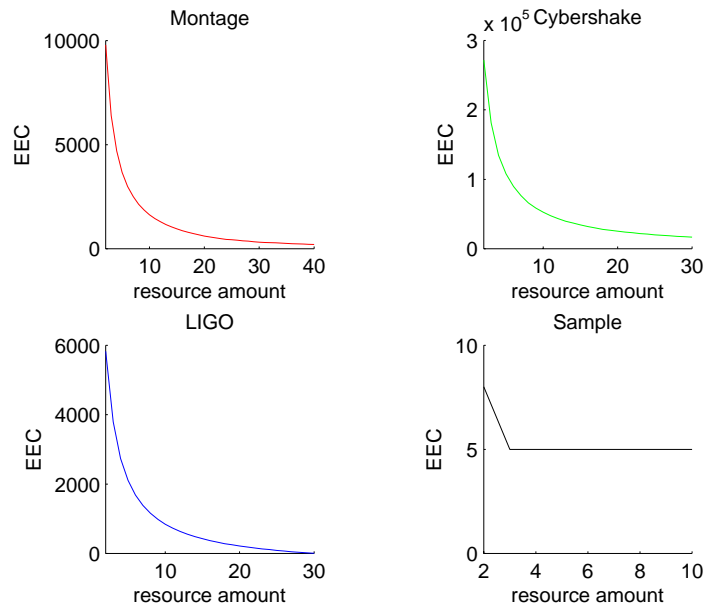


Fig. 8. Experimental results for different resource amount

The experiments on the test workflows with the REAL are conducted fifty times and the results are shown as in Fig. 8. In Fig. 8, EEC will decrease with the growing of resource amount by a negative exponential distribution for Montage, Cybershake and LIGO. Smaller EEC leads to increasing p . Meanwhile, providing extra resource will make extra cost in the process of resource allocation, data communication and resource preparing, which would lead to diminish p . This decreasing relationship between the cost and the resource amount can be described by a linear function. So, there exist fixed resource amounts, which can generate the maximum profits. However, this resource amount will differ for different workflows under different environments.

With the analysis, it is demonstrated that for a single workflow, clouds should provide a certain number of VMs, which is normally less than the biggest number that tasks requires. With this method, cloud providers will get the maximum profits.

6.3 Performance for Different Workflows

To apply in practical clouds, an effective workflow scheduling algorithm should present stable, which is considered as most important, for different workflows. In this experiment, we change the Montage parameter of degree to simulate different workflows and evaluate the

performance of proposed algorithm. And also the number of *mDiffFit* varies to generate different size workflows in the experiment. The experimental results are shown as in **Fig. 9** and **Fig. 10**.

In **Fig. 9**, the structure of Montage changes with the degree. We can see that REAL generally gives a better performance than the comparing algorithms. The ID-CPA makes little optimization as the in-degree would only make effects when there are many offspring nodes. So it demonstrates that rule 1 present more important on optimizing the performance than other rules. And all these proposed rules can make help on getting the optimal schedule.

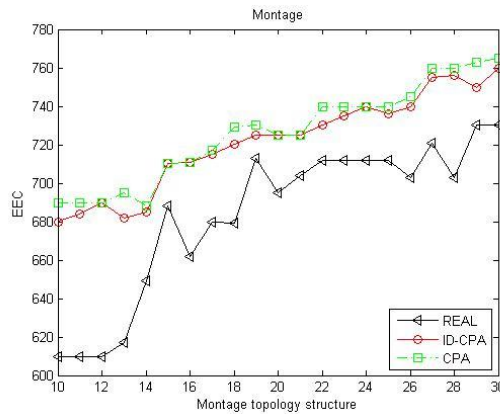


Fig. 9. Experimental results for different topologies

In **Fig. 10**, we increase the value of *mProject* in Montage. It can be seen that the EEC increases with the number by a certain linear coefficient. It shows that the REAL performs stable for different size workflows. Therefore, it well fits the practical requirements and can be applied for big size workflows. Above all, the REAL also gives a good performance for the changes of workflow size.

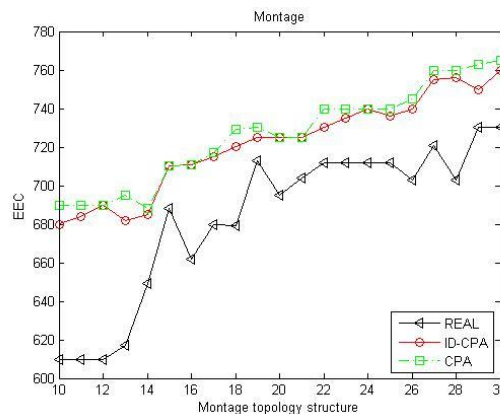


Fig. 10. Experimental results for different workflow size

On the whole, it comes to the conclusion that the changes of workflow topologic structure influence the performance of REAL more than the workflow size. We speculate that these three parameters cannot describe the topologic structure completely. So if employing more parameters to give a full description on the topologic structure, it is a novel and effective approach to get the optimal schedule. And also, the proposed REAL method performs prior than the comparing algorithms.

6.4 Performance on Computing Time

In this section, the execution time of REAL in former two experiments is studied to evaluate its performance for commercial clouds. The computing time is an important measure to evaluate a scheduling algorithm, because it also influences the QoS for clients. Algorithms with low computing complexity fit the problem of real time scheduling better, which is especially typical in commercial clouds. And high computing complexity acts like a disaster for the implementation of scheduling algorithm. It makes the algorithms lose their practical properties. So, a particular analysis on the computing time of former experiments is made as follows.

In Fig. 11, we can see that the computing time is almost linear and, for small resource amount, the computing time is very short. It indicates that the computing complexity of REAL increases linearly by the resource amount. If the resource amount is bigger, there are more possible schedules. The scanning space increases exponentially. So the REAL presents a good performance for the changing resource amount.

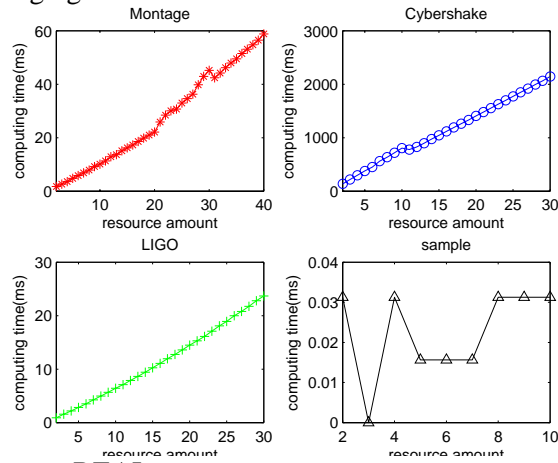


Fig. 11. Computing time of REAL in the experiments of evaluating algorithm performance for different resource amount

In Fig. 12, we can see that, for different topologies of the same size workflows, the computing time is steady. The computing time changes linearly by the workflow size. For large size workflows, it presents challenges to complete the scheduling in linear time. The results indicate that the REAL can meet to big size challenges well.

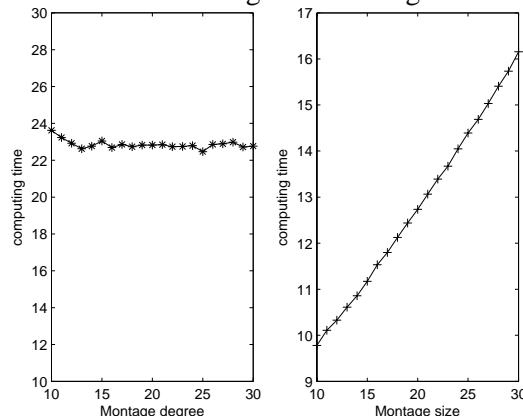


Fig. 12. Computing time of REAL in the experiments of evaluating algorithm performance for different topologies and workflow size

On the final, the REAL gives a good performance on workflow scheduling in commercial clouds. With above experiments, it comes to the conclusion that the topologic characteristics of out-degree, in-degree and path length indeed can make help on optimizing the scheduling. And the proposed algorithm, which combines the rules together, gives a good performance on workflow scheduling under the setting assumptions and shows prior than the comparing algorithms, which have already been demonstrated to be effective.

7. Conclusion

Cloud computing is going to play an important role in the future. It has been one of the hottest topics around the world and expected to change the lives of people. This paper describes the workflow scheduling problem in commercial clouds and gives a detailed formulation on these issues in cloud environments. Based on former researches, this paper makes efforts on finding the relationships between topologic characteristics and optimal schedule. To present the behavior of the proposed solution, the paper gives an instance and analyzes its available schedules. Based on the analysis, it finds the relationships and proposes a set of rules aiming at the workflow scheduling problem in clouds. After that, a Resource Auction Algorithm (REAL) is proposed to execute the workflow scheduling solutions. The algorithm employs the partial topologic characteristics of workflow to optimize the scheduling. And in the experimental evaluation, REAL performs well on the searching complexity, especially for big size workflows and successfully optimize the scheduling, especially for small size workflow. Extensive experiments are conducted to evaluate its performance by changing the workflows and monitoring the computing time. It proves that REAL can increase the searching efficiency for optimum schedule with low computing complexity.

The contributions of this paper can be summarized as follows:

- This paper gives a detailed description on the workflow-scheduling problem in commercial clouds and formulates it.
- This paper takes the communication into account and takes advantage of the precedent relation between tasks in workflow to avoid the data communication time, which has always been ignored.
- This paper proposes a topologic-based workflow-scheduling algorithm, which employs more topologic characteristics other than the critical path and gives a full analysis on the proposed solutions.
- Experiments are conducted to evaluate the performance and the experimental results show the proposed algorithm can well solve the problem. Some novel knowledge is found through the experiments that it is most profitable to provide a certain amount of computing resource.

Future works will be done specially for the implementation of workflow services in cloud computing. Many other conditions should be considered to meet the needs of practical clouds, for instance uncertain task execution time [28], task emerging, multi-objectives [29], fuzzy topologic information and so on. For the growing size of workflows, the proposed algorithm would be developed to meet the challenges. And also, further researches will be done on analyzing the influence of other topologic characteristics to further optimize the optimum schedule searching, such as clustering coefficient and so on.

References

- [1] R.Buyya, J.Broberg, I.Brandic, "Cloud computing and emerging it platforms: vision, hype and reality for delivering computing as the 5th utility," *Future Generation Computing System*, pp.599-616, 25(6), 2009. [Article \(CrossRef Link\)](#)
- [2] J. Saeid Abrishami, Mahmoud Naghibzadeh, Dick H.J. Epema, "Cost-driven scheduling of Grid workflows using Partial Critical Paths," *IEEE Transactions on Parallel and Distributed System*, pp.1400-1414, 23, 2011. [Article \(CrossRef Link\)](#)
- [3] H. Topcuoglu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE Transactions on Parallel and Distributed Systems*, pp.260-274, Mar, 2002. [Article \(CrossRef Link\)](#)
- [4] Rashmi Bajaj, "Improving scheduling of tasks in a heterogeneous environment," *IEEE Transactions on Parallel and Distributed Systems*, pp.107-118, Feb., 2004. [Article \(CrossRef Link\)](#)
- [5] M. Rahman, "A Dynamic Critical Path Algorithm for Scheduling Scientific Workflow Applications on Global Grids," in *Proc. of IEEE International Conference on e-Science and Grid Computing*, pp.35-42, Dec.10-13, 2007. [Article \(CrossRef Link\)](#)
- [6] Gu Y, "Performance analysis and optimization of distributed workflows in heterogeneous network environments," *IEEE Transactions on Computers*. 2014. [Article \(CrossRef Link\)](#)
- [7] Lin Fuhong, Zhou Xianwei, Huang Daochao, Song Wei, Han Dongsheng, "Service Scheduling in Cloud Computing based on Queuing Game Model," *KSII Transactions on internet and information systems*, pp.1554-1566, 8(5), 2014. [Article \(CrossRef Link\)](#)
- [8] Hassan Mohammad Mehedi, Song Biao, Almogren Ahmad, Hossain M. Shamim, Alamri Atif, Alnuem, Monowar Muhammad Mostafa, Mohammed, Hossain M. Anwar, "Efficient Virtual Machine Resource Management for Media Cloud Computing," *KSII Transactions on internet and information systems*, pp.1567-1587, 8(5), 2014. [Article \(CrossRef Link\)](#)
- [9] Mateo Romeo Mark A., Lee, Jaewan, "Dynamic Service Assignment based on Proportional Ordering for the Adaptive Resource Management of Cloud Systems," *KSII Transactions on internet and information systems*, pp.2294-2314, 5(12), 2011. [Article \(CrossRef Link\)](#)
- [10] Meng Xu, "A multiple QoS constrained scheduling strategy of multiple workflows for Cloud computing," in *Proc. of 2009 IEEE International Symposium on Parallel and Distributed Processing with Applications*, pp.629-643, 2012. [Article \(CrossRef Link\)](#)
- [11] Saeid Abrishami, Mahmoud Naghibzadeh, Dick H.J. Epema, "Deadline-constrained workflow scheduling algorithms for infrastructure as a Service Clouds," *Future Generation Computer Systems*, pp.158-169, 29(1), 2013. [Article \(CrossRef Link\)](#)
- [12] Herald Kllapi, Eva Sitaridi, Manolis M. Tsangaris, "Schedule optimization for data processing flows on the Cloud," in *Proc. of the 2011 ACM SIGMOD International Conference on Management of data*, pp.289-300, 2011. [Article \(CrossRef Link\)](#)
- [13] Suraj Pandey, Linlin Wu, Siddeswara Mayura Guru, R. Buyya, "A particle swarm optimization-based heuristic for Scheduling Workflow Applications in Cloud computing environments," in *Proc. of 2010 24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, April 20-23, 2010. [Article \(CrossRef Link\)](#)
- [14] J. Zhangjun Wu, Xiao Liu, Zhiwei Ni, Dong Yuan, "A market-oriented hierarchical scheduling strategy in Cloud workflow systems," *The Journal of Supercomputing*, pp.256-293, 63(1), Jan., 2013. [Article \(CrossRef Link\)](#)
- [15] Jia Yu, Rajkumar Buyya, "A budget constrained scheduling of workflow applications on utility grids using genetic algorithms," in *Proc. of WORKS '06 Workshop on Workflows in Support of Large-Scale Science*, June19-23, 2006. [Article \(CrossRef Link\)](#)
- [16] Maciej Malawski, Gideon Juve, Ewa Deelman, Jarek Nabrzyski, "Algorithms for cost- and deadline-constrained provisioning for scientific workflow ensembles in IaaS clouds," *Future Generation Computer Systems*, vol. 48, pp. 1-18, July, 2015. [Article \(CrossRef Link\)](#)

- [17] Lingfang Zeng, Bharadwaj Veeravalli, Albert Y. Zomaya, "An integrated task computation and data management scheduling strategy for workflow applications in cloud environments," *Journal of Network and Computer Applications*, vol. 50, pp. 39-48, April, 2015. [Article \(CrossRef Link\)](#)
- [18] Mohammad Mehedi Hassan, "Cost-effective resource provisioning for multimedia cloud-based e-health systems," *Multimedia Tools and Applications an International Journal*, published online, May, 2014. [Article \(CrossRef Link\)](#)
- [19] Biao Song, Mohammad Mehedi Hassan, Atif Alamri, Abdulhameed Alelaiwi, Yuan Tian, Mukaddim Pathan, Ahmad Almogren, "A two-stage approach for task and resource management in multimedia cloud environment," *Journal of Computing*, published online, June, 2014. [Article \(CrossRef Link\)](#)
- [20] Bittencourt L F, "HCOC: a cost optimization algorithm for workflow scheduling in hybrid clouds," *Journal of Internet Services & Applications*, vol. 2, pp.207-227, 2011. [Article \(CrossRef Link\)](#)
- [21] Bittencourt L F, Madeira E R M, Fonseca N L S D, "Scheduling in Hybrid Clouds," *IEEE Communications Magazine*, vol. 50, pp. 42 – 47, 2012. [Article \(CrossRef Link\)](#)
- [22] Genez T A L, Bittencourt L F, Madeira E R M, "On the Performance-Cost Tradeoff for Workflow Scheduling in Hybrid Clouds," in *Proc. of 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing (UCC)*, pp. 411-416, 2013. [Article \(CrossRef Link\)](#)
- [23] R. Lopes Gomes, L.F. Bittencourt, E.R. Mauro Medeira, "Supporting SLA Negotiation for VSDN Based On Similarity and Price Issues," in *Proc. of 2012 IEEE 13th International Symposium on Network Computing and Applications*, pp. 287-290, Cambridge, MA, 21-23 Aug., 2014. [Article \(CrossRef Link\)](#)
- [24] Available at http://zh.wikipedia.org/wiki/Amazon_EC2.
- [25] Mohammad Mehedi Hassan, M. Shamim Hossain, A. M. Jehad Sarkar, Eui-Nam Huh, "Cooperative game-based distributed resource allocation in horizontal dynamic cloud federation platform," *Journal of Information Systems Frontiers*, vol.16, pp.523-542, 2012. [Article \(CrossRef Link\)](#)
- [26] M.E.J. Newman et al., *Network: An introduction*, in Oxford University Press, 2011.
- [27] Shishir Bharathi, Ann Chervenak, "Characterization of scientific workflows," in *Proc. of Third Workshop on Workflows in Support of Large-Scale Science*, pp.1-10, Nov., 2008. [Article \(CrossRef Link\)](#)
- [28] Luiz F. Bittencourt, RizosSakellariou, Edmundo R.M. Madeira. "Using relative costs in workflow scheduling to cope with input data uncertainty," *MGC 2012*, Montreal, Quebec, Canada, Dec. 3, 2012. [Article \(CrossRef Link\)](#)
- [29] Xing Liu, Chaowei Yuan, Enda Peng, et al, "Combined Service Subscription and Delivery Energy-Efficient Scheduling in Mobile Cloud Computing," *KSII Transactions on Internet and Information Systems(TIIS)*, vol.9, pp.3191-3212, 2015. [Article \(CrossRef Link\)](#)



Haoran Ji received the M.S degree in information systems from National University of Defense Technology, China, in 2012. Currently, he is a Ph.D student in the School of Information System and Management at National University of Defense Technology. His research interests include real-time systems, workflow scheduling, and cloud computing.



Weidong Bao received the Ph.D. degree in information system from the National University of Defense Technology in 1999. He is currently a professor in the College of Information Systems and Management at National University of Defense Technology, Changsha, China. His recent research interests include cloud computing, information system, and complex network.



Xiaomin Zhu received the Ph.D. degree in computer science from Fudan University, Shanghai, China, in 2009. He is currently an assistant professor in the College of Information Systems and Management at National University of Defense Technology, Changsha, China. His research interests include scheduling and resource management in green computing, cluster computing, cloud computing, and multiple satellites. He has published more than 50 research articles in refereed journals and conference proceedings. He is also a frequent reviewer for international research journals, e.g., IEEE TC, IEEE TNSM, IEEE TSP, Journal of Parallel and Distributed Computing, Journal of Systems and Software. He is a member of the IEEE, the IEEE Communication Society, and the ACM.



Wenhua Xiao received his B.S. and M.S. degree from the International School of Software at Wuhan University, China, in 2010 and College of Information and System Management at National University of Defense Technology (NUDT), China, in 2012 respectively. He is currently a Ph.D. student in the College of Information System and Management at NUDT. His research interests include scheduling and resource management in cloud computing.