Optimized Cost-Based Biomedical Workflow Scheduling Algorithm in Cloud

N. Mohanapriya(✉) and G. Kousalya

Department of Computer Science and Engineering, Coimbatore Institute of Technology, Coimbatore, India
mohanapriyan08@gmail.com

Abstract. Owing to the data deluge of biomedical workflow applications, researchers consider cloud as a promising environment for deploying biomedical workflow applications. As the Workflow applications consist of precedence-constrained tasks, it requires high computing resources for its execution. Scheduling of pertinent resource to biomedical workflow applications is an appealing research area. The key concern is that the workflow applications should be scheduled with the appropriate resource such that the overall execution time and cost would be minimized and correspondingly resource utilization is maximized. The proposed Optimized Cost Scheduling Algorithm (OCSA) addresses this issue by scheduling the workflows to a resource in such a way that it efficiently reduces the time and cost. The proposed OCSA algorithm is simulated rigorously in WorkflowSim on real biomedical workflow application and the results are compared with the existing workflow scheduling approaches in terms of cost and time. The simulation result shows that the proposed scheduling algorithm appreciably reduces the execution time and cost than the existing scheduling algorithms.

Keywords: Biomedical workflow scheduling · Cloud scheduling · Cost based scheduling · Workflows in the cloud

1 Introduction

Biomedical workflow applications consist of numerous interdependent tasks, and there exist huge data transfer between the tasks. These applications are generally represented as Directed Acyclic Graphs (DAGs), which brings a problem of resource scheduling in distributed systems. Cloud is a distributive computing environment that dynamically delivers scalable, on demand services through virtualization of hardware and software over the internet. Cloud is based on a market-oriented paradigm, where the services consumed by the customers are charged on pay-as-you-go model [1, 2]. The prospect of running workflow applications through the cloud is made attractive by its benefits. The essential benefits includes,

- Virtualization - Cloud gives the illusion of unlimited resources and this allows the user to acquire sufficient resources at any time.
- Elasticity - Cloud providers offer scalable resources to its users so that the resources are gained and released as per the requirements.

S.M. Thampi et al. (eds.), *Advances in Signal Processing and Intelligent Recognition Systems*, Advances in Intelligent Systems and Computing 678,
DOI 10.1007/978-3-319-67934-1_40

A notion of discovering the suitable resource from the heterogeneous resource pool for the workflow task is referred as scheduling [3]. The mapping of tasks to the resources is an NP-Complete problem [4]. Scheduling of biomedical workflow applications involves substantial communicational and computational costs, which strongly emphasizes the usage of cloud computing for their execution. Cloud providers offer heterogeneous computing resources with different capabilities at various prices. Generally, high computing resources are expensive than the slower resource. Hence different scheduling is possible for the same workflow, which in turn impacts the scheduling time and cost. Therefore a special care for scheduling should be taken to avoid the unnecessary cost.

An optimized cost based workflow scheduling algorithm is proposed to schedule the biomedical workflow application in the cloud to minimize the overall execution time and cost for the execution of the workflow.

2 Related Works

Suraj pandey et al. [5] proposed the particle swarm optimization algorithm based heuristic for the scheduling of workflow applications to cloud resources with an aim to reduce the execution cost by considering the computation cost and data transmission cost. Arabnejad et al. [6] presented a Proportional Deadline Constrained (PDC) for mapping workflow tasks to cloud resources which minimizes the cost while meeting deadline constraints. The algorithm considers the execution cost and time for the selection of resource. Amandeep Verma et al. [7] proposed Budget and Deadline Constrained Heterogeneous Earliest Finish Time (BDHEFT) for workflow scheduling. The spare workflow cost and current task cost are considered for the selection of cost-efficient resource. Abrishami et al. [8] proposed QoS-based workflow scheduling algorithm based on Partial Critical Path (PCP), which tries to minimize the execution cost while meeting user defined deadline. PCP algorithm tries to schedule the critical task that is; the tasks present in the critical path to the resources that executes the task earliest in order to minimize the total cost of the path and executes all the tasks before its finish time. Su et al. [9] proposed a Pareto optimal scheduling heuristic (POSH) to schedule tasks to the cost conscious resource based on pareto dominance. It uses the execution time along with the cost factors to map the higher priority task to the cost efficient resource. Convolbo et al. [10] proposed cost aware scheduling algorithm for solving cost optimization problem for DAG scheduling on IaaS cloud. It schedules the job to the cost efficient resource by computing the execution time and resource usage cost.

3 System Model

3.1 Application Model

A workflow application is modeled by DAG is defined as W = G(T,E), where T is the set of n task$\{t_1,t_2,\ldots.t_n\}$ and E represents the set of directed edges $\{e_1,e_2\ldots.e_k\}$ between the workflow tasks. A task t_i ε T, represents a task in workflow application and each edge ($t_{i,}$ t_j) = e_1 ε E, corresponds to a precedence constraints, where t_j ε T cannot be executed till t_i ε T finishes its execution. Each task t_i ε T represents a computational workload, Wl_i which takes millions of instructions (MI) as a unit of measurement. A task with no predecessor and successor tasks is called as entry task and exit task respectively and the workflow size is determined by the number of tasks [12].

3.2 Cloud Model

Workflowsim [13] is a toolkit used in this experiment to mimic the cloud computing infrastructure. The service provider offers heterogeneous computational resources in the form of virtual machines VM {VM $_1$, VM $_2$,…VM_m} with different prices. Each resource (Virtual Machine) VM_m ε VM is capable of executing the given workflow application and its processing power is expressed in Millions of Instructions per Second (MIPS). Each VM has a different number of cores, MIPS, memory and storage configurations. Pricing is based on pay per use strategy similar to commercial clouds, where the users are charged based on the time interval and type of the resource used.

4 Optimized Cost Scheduling Algorithm

OCSA is an online scheduling algorithm, which comprises of three phases to schedule the biomedical workflow application in the cloud. The phases include Task selection, Resource Selection, and Resource allocation. Task selection phase selects the task with maximum execution time by preserving the parent-child relationship of a given biomedical workflow application. Resource selection phase is a significant phase of this proposed work, as it selects the optimal resource for the task execution and Resource allocation phase allocates the chosen resource to the workflow task for execution. The resource allocation phase in OCSA is a crucial phase where the actual scheduling occurs.

The main objective of the proposed work is to reduce the execution time and monetary cost of Biomedical Workflow applications in a cloud environment. Monetary cost includes execution cost, communication cost, storage cost and resource usage cost [14]. The following time factors are computed before resource selection which in turn is used to compute the various costs resulting in monetary cost for the selection of optimal VM.

$$CT_{t_i} = \frac{L_{t_i}}{VM_c} \tag{1}$$

where CT_{ti} is the Computation time of task t_i which calculates the Computation time of the task by the length of the task L_{ti} with the capacity of Virtual Machine $VM_{c.}$

Data transfer time between the interdependent tasks in a workflow application is calculated as

$$CMT_{t_i} = \frac{\sum_{FS=0}^{t} FS}{VM_b} \tag{2}$$

where $CMT_{ti,}$ represents the communication time of the task t_i, which computes the Communication time between the tasks by the input and output file sizes, FS with Virtual Machine bandwidth, $VM_{b.}$

Expected Execution Time of the workflow task is calculated from the Eqs. (1) and (2), as follows

$$EET_{t_i} = CT_{t_i} + CMT_{t_i} \tag{3}$$

where EET_{ti} is the Expected Execution Time of the task $t_{i,}$ which computes the Expected Execution Time of the task t_i on the VM by the computation time of the task CT_{ti} with the communication time of the task, CMT_{ti}.

Total execution cost for the workflow is calculated by using the execution time with resource usage cost, memory cost, communication cost and storage cost.

$$EC = ET_{t_i} \times C_r \tag{4}$$

where EC is the Execution Cost which computes the Cost for Execution of the task on the VM by the execution time ETt_i with the resource cost C_r.

$$MC = FS \times ET_{t_i} \times C_\mu \tag{5}$$

where MC is the Memory cost which computes the Cost for Memory Usage by the File size FS of the respective task along with its execution time and the memory usage cost, $C_{\mu.}$

$$CC = FS \times CT_{t_i} \times C_\beta \tag{6}$$

where CC is the communication cost which calculates the communication cost between the tasks by the input and output size of the files FS with the time of communication, CTt_i and Bandwidth Cost, $C_{\beta.}$ Communication cost is applicable, only when there exist a dependency between the tasks that is when $e_i,e_j > 0$. And the communication cost will be zero for the tasks executing on the same resource.

$$SC = FS \times C_s \tag{7}$$

Storage Cost, SC is computed by the size of the file stored with the Storage cost, C_s. And finally, the Minimum Execution Cost is computed from the Eqs. (4)–(7), which is used for selecting the appropriate resource from the heterogeneous resource pool.

$$MEC_w = EC + MC + CC + SC \tag{8}$$

where MEC_w is the Minimum Execution Cost required to execute the Workflow task on the VM.

```
Input: W = G(T,E)
Output: Cost Optimized Workflow schedule
Let n be the number of workflow tasks to be scheduled
        //Task Selection Phase
Task list contains number of tasks $T[n] = \{t_1,t_2,\ldots.t_n\}$
for each task $t_i$ from the tasklist T, Where i = 0 to n
        Select the task with maximum length as $t_{max}$
End for
        //Resource Selection Phase
Resource list contains m Virtual Machines $VM[m] = \{VM_1,VM_2,\ldots.VM_m\}$
for each VM $vm_j$ from the resource list VM, where j = 1 to m
        $VM_{Id}$ <- $VM_j$.getID()
        Compute ComputationTime $CT_{ti}$ of the task $t_{max}$ on $vm_j$ based on Equation (1)
        $CT_{ti}$ <- $L_{ti}/VM_c$
        Compute CommunicationTime ($CMT_{ti}$) of the task $t_{max}$ based on Equation (2)
        $CMT_{ti}$ <- $\sum_{FS=0}^{t} FS$ / $VM_b$
        Compute the ExpectedExecutionTime ($EET_{ti}$) of the task $t_{max}$ on $vm_j$ based on the
        Equation (3)
        $EET_{ti}$ <- $CT_{ti} + CMT_{ti}$
        Compute the Minimum Execution Cost (MEC) required for the execution of task $t_{max}$
        on the $vm_j$ based on the Equation (8)
        MEC <- EC+ MC + CC + SC
        MEcost.add($VM_{Id}$, MEC)
End for
        //Resource Allocation Phase
VMopt <- MEcost.getMinValue()
Set $VM_{opt}$ as BUSY
Allocate $VM_{opt}$ to the task $t_{max}$
```

Algorithm 1. Pseudo code of Optimized Cost Scheduling Algorithm (OCSA)

OCSA selects the appropriate resources for scheduling with the notion to reduce the time and monetary cost of the biomedical workflow applications. It selects the optimal cost VM by considering the execution cost, memory cost, communication cost, storage cost and resource cost so that the workflow tasks are executed in an optimal resource which is shown in the Algorithm 1.

Direct Cost of the applications is measured using the individual resource usage that is data storage cost, resource cost, resource computation cost, Network cost, I/O cost [15, 16]. As the proposed algorithm selects the Optimal VM by computing the direct cost as shown in the Eqs. (4)–(7), it significantly reduces the overall time and cost of execution of workflows in a cloud environment.

5 Experimental Setup and Result Analysis

Workflowsim toolkit is used to create a cloud environment for experimentation purposes, which consists of a Service Provider offering heterogeneous computational resources for workflow execution. The data center configuration is presented in the Table 1, while the resource characteristics and cost for using the resources are shown in Tables 2 and 3 respectively.

Table 1 Datacenter Characteristics

Ram (Host Memory)	20480 MB
Storage (Host Storage)	1000000
Bandwidth	10000
System Architecture	X_86
Operating System	Linux
VMM	Xen

Table 2 Resources characteristics

Resource type	RAM in MB	Processing elements
Small	2048	1
Medium	4096	2
Large	6144	2
XLarge	8192	4

Table 3 Cost of the resources

Resources	Usage cost (Rupees)
Virtual machine	3
Memory	0.05
Storage	0.1
Bandwidth	0.1

For the enhanced analysis and evaluation of the proposed algorithm the experiment is conducted with the real world biomedical workflow applications with a diverse structure and range of tasks varies randomly from 10, 25, 100 and 500. The traces of biomedical workflow applications are downloaded from Pegasus Workflow Generator [11]. The workflow structure of the considered biomedical workflow applications are depicted in the Fig. 1. The proposed algorithm OCSA is compared with the existing workflow scheduling algorithms in workflowsim (FCFS, MAXMIN, MINMIN and MCT) in terms of time and cost.

The execution time of the workflow application is calculated using Eqs. (1) and (2) and the results of OCSA compared with the existing scheduling algorithms are presented in Fig. 2, and the average execution time for the various workflow tasks which results from the average of 20 runs of each workflow execution is compared in Fig. 3,

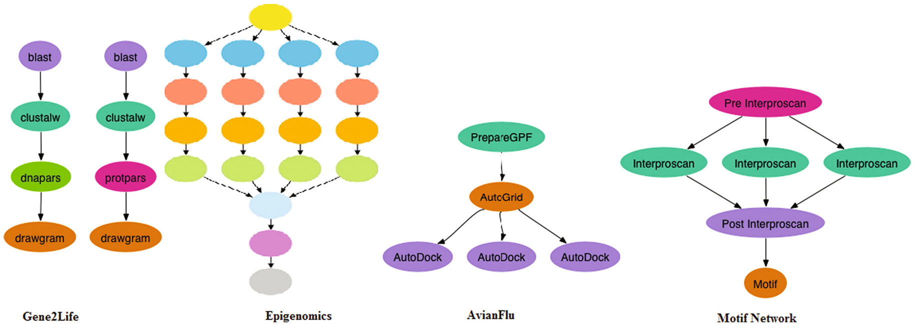

Fig. 1 Structure of the Biomedical Workflow Applications

which substantiate that the execution cost of OCSA is considerably minimum than the other scheduling algorithms.

The execution cost is computed for different biomedical workflow application based on the Eq. (4) and the results are depicted in Fig. 4 and the average cost for different workflow tasks are depicted in Fig. 5, which clearly illustrate that the proposed algorithm performance supersedes the existing approaches and reduces the overall cost.

Optimal Cost Scheduling Algorithm maps the task to appropriate VM by considering monetary costs which are computed by the summation of various individual costs involving storage cost, data transfer cost, memory cost and resource usage cost. This is turn, results in a best scheduling strategy that minimizes the overall execution time and cost of the workflow application.

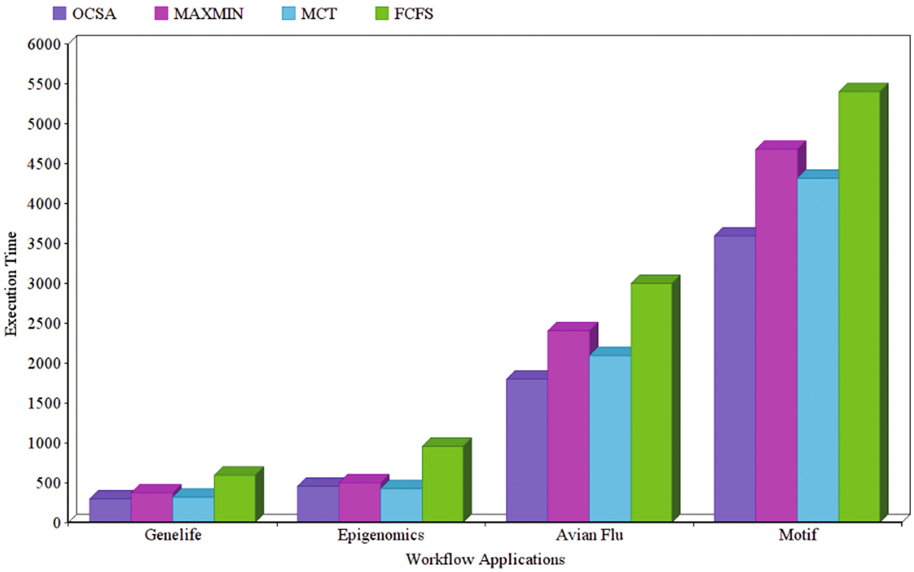

Fig. 2 Execution Time comparisons of biomedical workflow applications

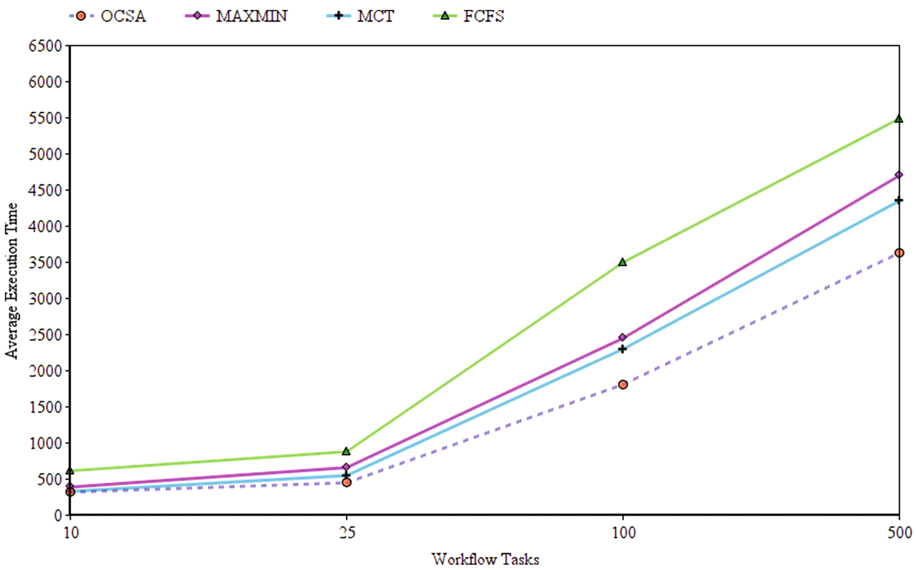

Fig. 3 Average Execution Time comparison of biomedical workflow applications

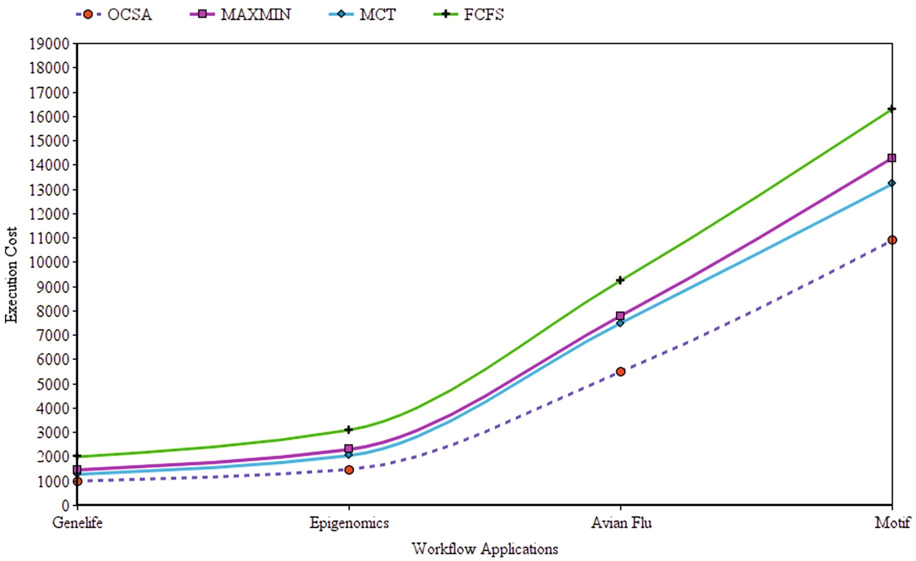

Fig. 4 Execution Cost Comparison of biomedical workflow applications

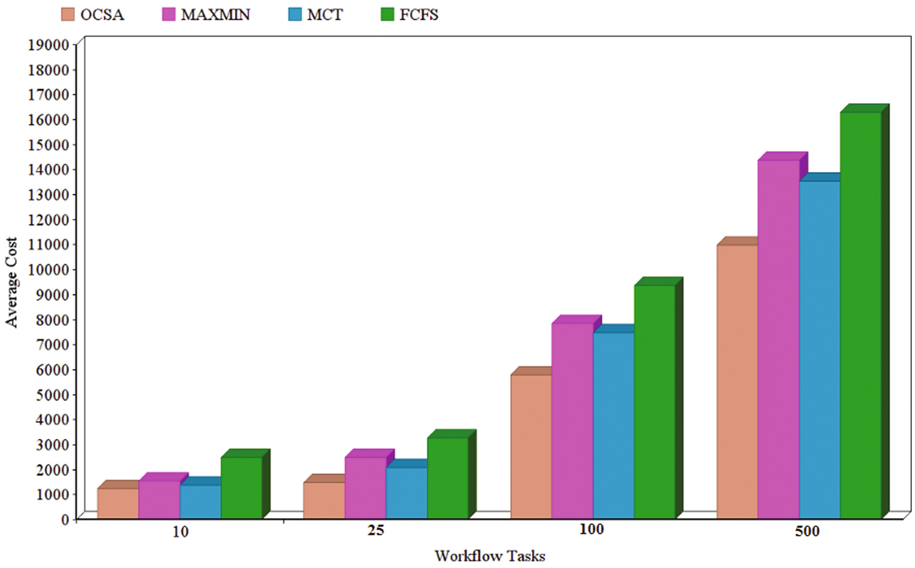

Fig. 5 Average Execution Cost Comparison of biomedical workflow applications

6 Conclusion

An Optimized Cost Scheduling Algorithm is proposed to schedule a biomedical workflow application in cloud with an aim to minimize the overall execution time and cost. The algorithm is evaluated using workflowsim toolkit for four real world biomedical workflow applications and the comparison is made with the existing scheduling approaches of workflowsim (in terms of execution time and cost). The result analysis reveals that the proposed OCSA schedules a workflow application with minimal time and cost in the cloud environment.

References

1. Buyya, R., Pandey, S., Vecchiola, R.: Cloudbus toolkit for market-oriented cloud computing. In: CloudCom 2009 Proceedings of the 1st International Conference on Cloud Computing, vol. 5931. LNCS, pp. 24–44. Springer, Germany, December 2009
2. Armbrust, M., Fox, A., Grifth, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: Above the clouds: a Berkeley view of cloud computing. Technical report, University of California, Berkeley, February 2009
3. Wang, Y., Lu, P.: DDS: A deadlock detection-based scheduling algorithm for work-flow computations in HPC systems with storage constraints. Parallel Comput. **39**(8), 291–305. http://dx.doi.org/10.1016/j.parco.2013.04.006
4. Ullman, J.D.: Np-complete scheduling problems. J. Comput. Syst. Sci. **10**(3), 384–393 (1975)

5. Pandey, S., Wu, L., Guru, S.M., Buyya, R.: A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In: 24th IEEE International Conference on Advanced Information Networking and Applications, Perth, WA, pp. 400–407 (2010). doi:10.1109/AINA.2010.31
6. Arabnejad, V. Bubendorfer, K.: Cost effective and deadline constrained scientific workflow scheduling for commercial clouds. In: 2015 IEEE 14th International Symposium on Network Computing and Applications, Cambridge, MA, pp. 106–113 (2015). doi:10.1109/NCA.2015.33
7. Amandeep, V., Sakshi, K.: Cost-Time efficient scheduling plan for executing workflows in the cloud. J. Grid Comput. **13**(4), 495 (2015)
8. Abrishami, S., Naghibzadeh, M.: Deadline-constrained workflow scheduling in software as a service cloud. Sci. Iranica **19**(3), 680–689 (2012). http://dx.doi.org/10.1016/j.scient.2011.11.047
9. Sen, S., Jian, L., Qingjia, H., Xiao, H., Kai, S., Jie, W.: Cost-efficient task scheduling for executing large programs in the cloud. Parallel Comput. **39**(4), 177–188 (2013)
10. Moise, W., Convolbo, J.C.: Cost-aware DAG scheduling algorithms for minimizing execution cost on cloud resources. J. Supercomput. **72**(3), 985–1012 (2016)
11. https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator
12. Mohanapriya, N., Kousalya, G., Balakrishnan, P.: Cloud workflow scheduling algorithms: a survey. Int. J. Adv. Eng. **VII**(III), 188–195 (2016)
13. Weiwei, C., Ewa, D.: WorkflowSim: a toolkit for simulating scientific workflows in distributed environments. In: 8th IEEE International Conference on eScience 2012 (eScience 2012), Chicago, 8–12 October 2012
14. Alkhanak, E.N., Lee, S.P., Rezaei, R., Parizi, R.M.: Cost optimization approaches for scientific workflow scheduling in the cloud and grid computing: a review, classifications, and open issues. J. Syst. Softw. **113**, 1–26 (2016). http://dx.doi.org/10.1016/j.jss.2015.11.023
15. Choudhary, V., Kacker, S., Choudhury, T., Vashisht, V.: An approach to improve task scheduling in a decentralized cloud computing environment. Int. J. Comput. Technol. Appl. **3**(1), 312–316 (2012)
16. Wu, Z., Liu, X., Ni, Z., Yuan, D., Yang, Y.: A market-oriented hierarchical scheduling strategy in cloud workflow systems. J. Supercomput. **63**(1), 256–293 (2013)