I J C T A, 9(41) 2016, pp. 1160-1170 ISSN: 0974-5572

A SYSTEMATIC REVIEW ON EFFECTIVE WORKFLOW SCHEDULING ALGORITHMS IN CLOUD UNDER DEADLINE CONSTRAINT

Jasraj Meena [1], Malay Kumar[1], Manu Vardhan[1] and Sanjeev Jain[2]

Abstract: Cloud computing is a Parallel and Distributed Computing paradigm that deliver high-performance computing resources over the internet to solve large-scale scientific problems, ex. executing scientific workflows. The scheduling of scientific workflows in the cloud is a noticeable issue of the current decade. The aim of workflow scheduling is to assign all the tasks of the workflow to the suitable resource in the cloud that minimize the performance criteria (such as execution time or execution cost) while meeting some Quality of Service (QoS) constraint (Deadline or Budget). Numerous state of the art workflow scheduling algorithms has been proposed in the literature for scheduling scientific workflows in the cloud to minimize the execution cost or execution time while meeting the QoS constraints. There is none of research paper available that surveys workflow scheduling algorithms to minimize execution cost while meeting deadline constraint. This article surveys workflow scheduling algorithms that aim to minimize the execution cost while meeting the user defined deadline. We present a taxonomy and analyze the workflow scheduling algorithms to determine their strength and weaknesses. In conclusion, we present our reflections on the analysis, discuss challenges of workflow scheduling in the cloud to minimization of the execution cost under deadline constraint.

Key Words: *Cloud Computing, Workflow Scheduling, Resource Provisioning, Deadline, QoS Constraints*

I. INTRODUCTION

The workflows are mostly uses in various domains such as Physics, Bioinformatics, Earth Science and Astronomy to model large-scale scientific and engineering applications [1] [2]. A workflow is a loosely coupled coarse-grained parallel application, and it is represented by Directed Acyclic Graph (DAGs). It consists set of node and edges where the node represents the computational tasks and edges represents data or control dependencies of the DAGs. The size of the workflows can be varying as per type of scientific and engineering applications. We need a high-performance computing environment to run scientific workflows such as grid computing, cluster computing, or cloud computing. There are lots of projects has been designed to execute large scientific workflows in the grid environment such as GrADS [3], ASKALON [4], Pegasus [5].

Nowadays, cloud computing is the most popular parallel and distributed computing paradigm that deliver high-performance computing resource over the internet to solve large-scale scientific problems, ex. Executing scientific workflows. Further, most of the cloud service providers delivers computing as a utility service, and they offer mainly three types of services such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [6] [7]. Whereas IaaS cloud provides the hardware resources (such as CPU, memory, storage, and networking) in the form of a virtual resource (examples Amazon EC2, Google Compute Engine). In PaaS cloud provides an environment for users to develop and deploy some web based applications (examples Google App Engine, Microsoft Azure, Engine Yard, and In SaaS). In SaaS, the cloud provides web

[1]*Department of Information Technology, National Institute of Technology, Raipur, Chhattisgarh, India*
Emails: {jasrajengg, malay.jaiswal88, vardhanmanu}@gmail.com
[2]*Department of Electronics & Communication Engineering, GEC Bikaner, Rajasthan, India*
Email: snjece@gmail.com

applications/software over the internet So that the service consumer can use these applications or software's (examples Google Docs, SalesForce.com). Mainly IaaS Clouds are suitable for executing the scientific workflows while PaaS and SaaS are not appropriate because they provide only an environment to design, develop and deploy applications [8]. This research paper mainly focuses on workflow scheduling in IaaS clouds that provide several cost and performance effective benefits, as compared to existing parallel and distributed paradigms such as cluster computing and grid computing.

Benefits of Cloud Computing

- Cloud provides on-demand self-service whereas the service consumers can unilaterally provision resource whenever the required without any human interaction.
- Second, a cloud is elastic in nature, therefore the scientific applications can grow or shrink their resource pool as per the need of their applications. The cloud allocates only required computing resources as per the need of service consumers from the cloud resource pool [6].
- Third, Cloud service provider's charges to service consumers based on pay-per-uses price model, in which they have to pay only for the computing resources they used.

Issues in Cloud Computing

- First, in the cloud computing environment resources are shared by multiple users, most of the resources are heterogeneous, and the hardware resource are provided in the form of virtual, so the performance of virtual machines (VMs) can be varied. For example, as per the report of research paper [9] the overall CPU performance of the VM can vary up to 24 percentages in Amazon public cloud. However, [10] reported that a typical cloud environment the performance variation can be up to 30(%) of execution time and up to 65(%) of data transfer time. So, the variations of performance are major issue in cloud computing environment that effects the total execution time/execution cost of the workflow and they can also miss their deadline constraint.
- Second, whenever a virtual machine is leased or released it takes the time to proper initialize (acquisition delay) and time in proper shutdown (termination delay). So, longer acquisition delay and termination delay can affect the execution time or execution cost of the workflows.
- Third, in the cloud if any computing resource will fail during execution of the workflow, then overall execution time, as well as the execution cost of the workflow, will increase. So, these are the major issues for workflow scheduling in the cloud.

However, by considering above-discussed benefits and issues of cloud, the researcher proposed number of heuristics and meta-heuristics workflow scheduling algorithms that try to minimize the execution cost of the workflow while meeting deadline constraint. The remainder of this paper is organized as follows Section 2 discusses workflow scheduling problem followed by the architecture of workflow scheduling such as application model, cloud resource model, and computing platform model in Section 3. Then the detailed literature survey is discussed in Section 4. The discussion and analysis are explained in Section 5. Section 6 conclude this research work and discusses some future scopes for workflow scheduling algorithms in the cloud.

1. Workflow Scheduling problem

The aim of workflow scheduling problem is to assigns each task of the scientific workflow to suitable resources that satisfies some performance criteria while meeting the QoS constraints. There is a lot of work has been done in the area of workflow scheduling in Cloud computing environment. This paper mainly focused on workflow scheduling algorithms that minimize the execution cost of the workflows while meeting deadline constraint.

In the cloud, workflow scheduling has mainly two steps, the first step is to select set of computing resources from the cloud resource pool and provisioning of those resources. Further, in the second step the schedule is generated and then the mapping of each task of the workflow to the selected computing resources that satisfy their performance criteria while meeting the QoS constraints. In this research paper, our objective is to analysis the workflow scheduling algorithms that minimize the execution cost while meeting the deadline constraint. A simple workflow (Directed Acyclic Graph) is shown in Fig. 1(a).

2. *Architecture for workflow scheduling in Cloud*

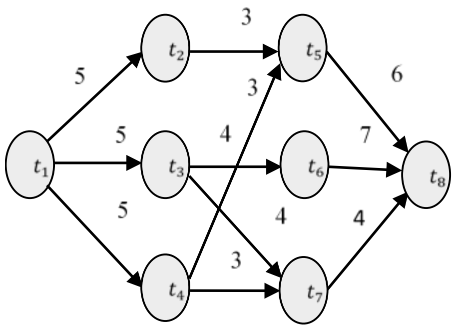
In this section, we introduce workflow scheduling models such as Application, Cloud Resource Model, and Computing Platform Model.

3.1 Application Model

The scientific workflows are represented by DAGs $W = (T, E)$ where $T = (t_1, t_2, \ldots, t_m)$, is the set of tasks and E is the set of edges. The edge $e_{i,j}$ is $t_i, t_j \varepsilon\ T$ and $t_i \neq t_j$ shows the data and control dependence from the task t_i to the task t_j. The task t_i is said to be the parent task of task t_j, and task t_j is the child of the task t_i.

This parent child relation, shows that child task can start execution after completion of parent task for example if task t_j is child task of task t_i, then execution of task t_j can start after completion of task t_i. The set of parents and children of a task t_i is represented by $Pred(t_i)$ and $Succs(t_i)$. Each DAGs has only one entry task t_{entry} and one exit t_{exit} task.

The entry task t_{entry} is the task that does not have any parents, and the exit task t_{exit} is the task that does not have any child. If there is more than one entry and exit task in the workflow we insert a dummy task (execution time and communication time is zero) and makes the graph with only one entry task and one exit task. A simple DAG Graph is shown in Fig. 1(a), with the execution time on different virtual machines in shown Fig. 1(b) and data transfer time between tasks in Fig. 1(c). Each scientific workflow (W) has a user specified deadline (D) that is associated with it, the Deadline (D) of the scientific workflow shows the time limit to complete the execution of workflow in the cloud computing environment. In the next section, we discussed cloud resource model.



	v_s	v_m	v_l
t_1	5	3	1
t_2	7	4	2
t_3	15	8	6
t_4	13	7	3
t_5	12	6	5
t_6	8	5	2
t_7	16	9	5
t_8	14	7	4

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
t_1	0	5	5	5	0	0	0	0
t_2	0	0	0	0	3	0	0	0
t_3	0	0	0	0	0	4	4	0
t_4	0	0	0	0	3	0	3	0
t_5	0	0	0	0	0	0	0	6
t_6	0	0	0	0	0	0	0	7
t_7	0	0	0	0	0	0	0	4
t_8	0	0	0	0	0	0	0	0

(a) (b) (c)

Fig. 1 (a) Simple DAG workflow, (b) ExeTime Matrix, (c) TransferTime matrix

.

3.2 Cloud Resource Model

A Cloud computing environment consists of an IaaS service provider, which delivers high-performance computing resources in the form of Virtual Machines (VMs) over the internet to execute large-scale scientific workflows. These VMs are selected from the set of VMs $(\overrightarrow{VM}) = \{vm_1, vm_2, \ldots, vm_k\}$, and each VMs have various configurations such as CPU type, memory size, and cost of per time interval. The cost of a VM is dependent on its configuration for example fast VMs means it is more costly as compare to the slower VM. Each VM of Type (VM_k) is defined as $(ET_{t_i}^{VM_k}, C_v)$, where $ET_{t_i}^{VM_k}$ is the estimated execution time of task t_i on VM of type (VM_k) is defined in Eq. (1) and C_v is the cost of VM type (VM_k) per time interval. Although, the estimated execution time of tasks on different types of VMs can be estimated by size of task Size(t_i) divided by processing capacity of virtual machine PC(VM_k) in terms of Floating point Operations Per Second (FLOPS).

$$ET_{t_i}^{VM_k} = \mathrm{Size}(t_i)\ /\mathrm{PC}(VM_k) \qquad (1)$$

$$TT(e_{ij}) = Data\,(t_i, out)/\beta \qquad (2)$$

However, if the task t_i is executed on VM of type (VM_k) and t_j is executed on VM of different type (VM_v), Then the data transfer time is represented as $TT(e_{ij})$. It is estimated by the size of the output data file $Data\,(t_i, out)$ to be transferred from task t_i to task t_j divided by the average bandwidth (β) is shown in Eq. (2). The value of $TT(e_{ij})$ will be zero when both task t_i and t_j are executed on the same VM. Further, this cloud model is similar to the IaaS service provided by Amazon such as computation service e.g. Amazon Elastic Compute Cloud (EC2) and storage service Amazon Elastic Block Store (EBS) [11] to send and receive the intermediate input/output files. In cloud if all the storage and computation services are in the same datacentre, then there may be average bandwidth to transfer data between shared storage service and VMs is roughly equal. Further, unlimited number of VMs are available in the cloud so it can provisions required number of VMs to satisfy user deadline constraints. Furthermore, most of the current cloud service providers charges to service consumers based number of time interval they used (time intervals may be hour or minutes basis). The next section defines the problem of workflow scheduling in a cloud computing environment.

3.3 Computing Platform Model

Fig. 2 depicts our computing platform model that is designed and developed in our project laboratory for workflow execution on IaaS Cloud. Our computing platform used in this study is similar to that used in [12] [13]. It has mainly two parts (Layered Architecture and Workflow Management System (WMS)). In Layered Architecture, the base layer contains the physical hardware such as CPUs, Network, Storage, and Database. To provision IaaS (Infrastructure as a Service), we used OS based virtualization technique. This virtualization technique uses the single shared kernel to run multiple instances of the virtual machine on the single host operating system. Furthermore, our IaaS cloud is designed using following open source technologies such as CentOS, CloudStack, and KVM.

CloudStack is open source software for providing Infrastructure as a Service to the end-user. It is designed to deploy and manage large networks of virtual machines. It uses virtualization technique such as KVM, Xen, VMware, However, our IaaS Cloud uses KVM. KVM (Kernel-based Virtual Machine) provides a full virtualization solution for Linux. Using KVM technology cloud service providers can create a number of virtual machines and each virtual machine has private virtualized hardware, such as CPU, storage, network and database. These three layers (base layer, CloudStack, and KVM) together provide IaaS (Infrastructure as a Service) [14] [15] [16] [17].

Further, the goal of Workflow Management System (WMS) is to execute a workflow application using the VMs provided by IaaS cloud. The workflow is submitted by the end user with its deadline and the required specification to the workflow management system (WMS). Then the WMS automatically performs resource provisioning and scheduling of tasks. Since Cloud requires users to provision the appropriate amount of computing resources and schedule each task of the workflow on computing resources that satisfy QoS constraints. Further, the WMS uses its sub-module such as Resource Capacity Estimator and Resource Acquisition Module to provide and allocate the appropriate number of VMs. Once appropriate computing resources such as VMs are allocated, the Execution Manager automatically runs the tasks on these VMs guided by the Workflow Scheduling Module. The next section discusses the literature survey in brief.

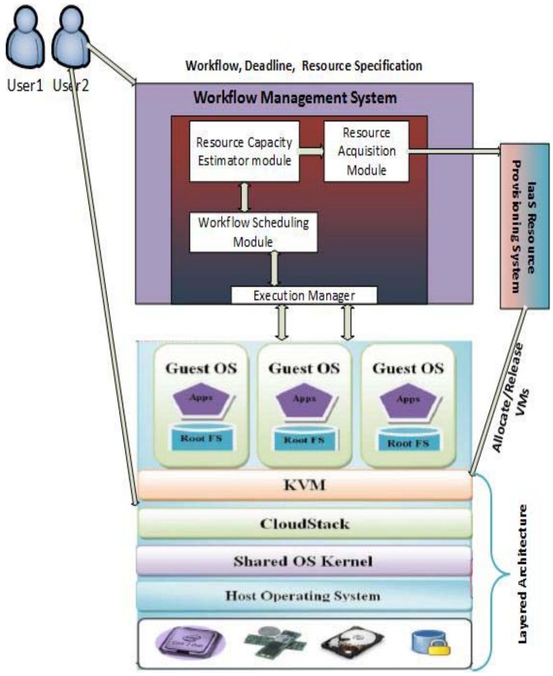


Fig. 2 Computing Platform Model for executing scientific workflows in IaaS Cloud

II. LITERATURE SURVEY

The workflow scheduling is the well-known research topic in the area of parallel and distributed systems and it is widely studied over the years. The problem of workflow scheduling is an NP-Hard problem [18], so near-optimal solutions can be provided in the polynomial time. In the parallel and distributed paradigms, such as Cluster computing and Grid computing various heuristic and meta-heuristic algorithms has been proposed to solve the workflow scheduling problem [19-26]. Nowadays, workflows are migrated from Cluster or Grid to cloud computing environment due to its various benefits. Further, the objective of the workflow scheduling in the cloud environment has been changed as compared to the cluster and grid. In grid or cluster, most of the researcher focused on a single objective such as to minimize the completion time (Makespan) of the workflow, while in cloud most of the workflow scheduling algorithms are multi-objectives. The objective may be the minimization of execution time, execution cost, energy consumption, reliability, or security while satisfying some Quality of service (QoS) constraints such as Deadline or Budget.

Further, a detailed review of workflow scheduling algorithms has been presented by some authors in [41] [42] the cloud computing environment. But to the best of our knowledge, there is no such review paper is available that analyze the workflow scheduling algorithms that minimize execution cost of the workflow while meeting deadline constraint. In this research paper, we have done a detailed survey and analysis of cost aware workflow scheduling algorithms that satisfy deadline constraint in the cloud computing environment. We have classified cost aware workflow scheduling algorithm in the cloud under deadline constraint, into two major categories Heuristic based and Meta-heuristic based scheduling algorithms shown in Fig. 3. The Heuristic based workflow scheduling algorithms have two types. The first type is based on the unlimited number of VMs available in the cloud computing environment, while in the second types the algorithms come that try to minimize the total number of computing VMs required to executed workflows. Furthermore, based on the number of workflows executed simultaneously (single or multiple workflows) the minimization of cost can be categorized such as for single workflows and for multiple workflows.

The heuristic workflows scheduling algorithms are easy to implement as compared to meta-heuristic algorithms, but meta-heuristics gives near-optimal solutions as compared to heuristics algorithms. Further, in this literature to execute multiple workflows four algorithms are proposed, in which SCS [29], DPDS [30], and WA-DPDS [30] are dynamic algorithms while the SPSS [30] is the static algorithm. However, for executing single workflows (six algorithms are proposed), in which ICPCP [31] and EIPR [32] heuristics are based on the partial critical path (PCP) of the workflow. First, they find the PCP and then assigns to VMs that minimize the execution cost while meeting deadline constraint. Further, to execute single workflows in cloud other three heuristic algorithms RCT [33], RTC [33] and Weighted [33] are based on the priority of the resources such as Robustness, execution cost and execution time. Furthermore, JIT-C (Just in Time) [12] is dynamic algorithms that first ascertains that user defined deadline is achievable or no, if not then user have to prompt deadline otherwise it finds cheapest VMs for each task and assigns to that VM that minimize execution time while meeting deadline constraint.

Another type of heuristic based workflow scheduling algorithms try to minimize the total number of VMs required to execute scientific workflows, in which BTS [13] and PBTS [34] are based on the simple idea is how much we can delayed execution of a tasks while IHM [35] algorithms first define the lower and upper bound for required number of VMs. Finally, the meta-heuristic workflow scheduling algorithms that are based on PSO (Particle Swam Optimization), and GA (Genetic Algorithm) [36-40]. Most of the Meta-heuristics gives better results in terms minimization of execution cost while meeting user defined deadline of the workflows but their time complexities are very high as compared to heuristics algorithms. In the next section, we have analyzed the each workflow scheduling algorithms in detail.

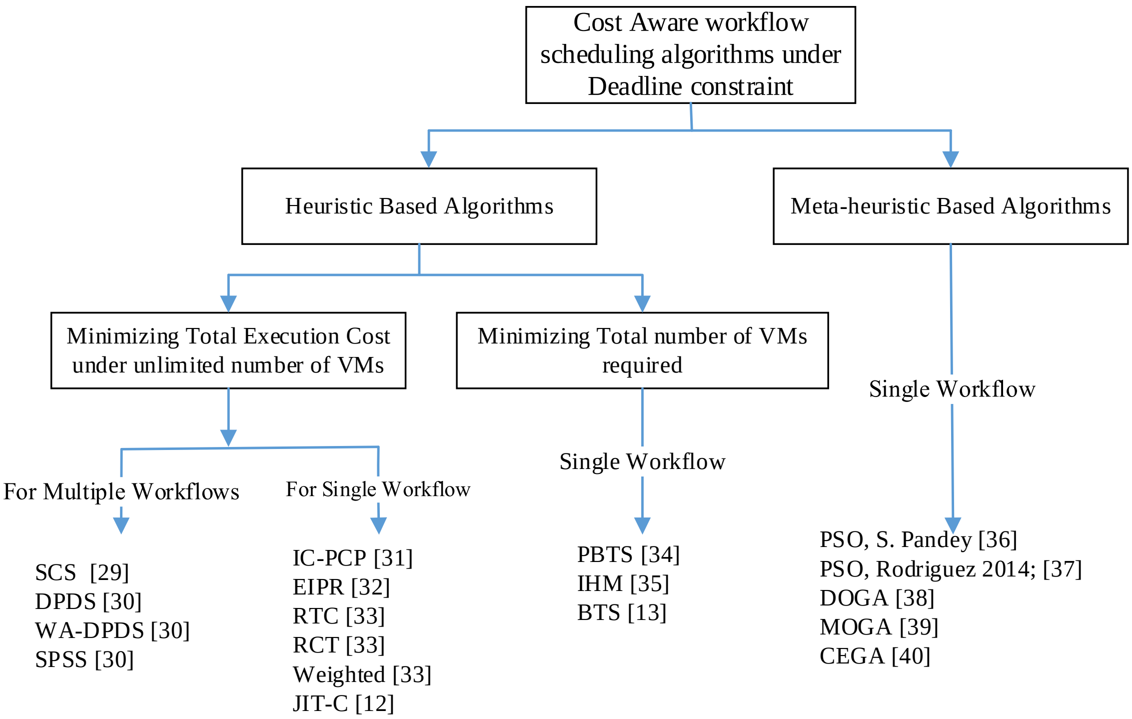


Fig. 3 Cost-aware workflow scheduling in the cloud under Deadline constraint.

III. ANALYSIS AND DISCUSSION

The most recent research work on workflow scheduling in cloud mainly focused the cost-aware workflow scheduling algorithms under deadline constraints. We have done a brief comprehensive survey that is summarized in the Table1. We have discussed that whether each workflow scheduling algorithm considers all the cloud features (such as on-demand resource provisioning, pay as you go pricing model and the heterogeneous resources) and resolves the major issues (such as Performance variation of VMs, acquisition delay, and fault tolerant). Further, we also discussed that whether they consider the data transfer time between tasks of the workflow. From Table1 it is identified that most of the heuristic algorithms considers the cloud and try to resolve the major issues of the cloud. However, it is identified that RTC, RCT, and JIT-C are the better algorithms as compared to other heuristics algorithms such as SCS, DPDP, ICPCP, EIPR, BTS, PBTS, and IHM, because RTC and RCT considers all the features as well major issues of the cloud while JIT-C does consider fault tolerance issue of the cloud but it gives better results as compared to the RTC in terms minimization of cost while meeting deadline constraint.

Table1. Analysis of cost-aware workflow scheduling algorithms in cloud under Deadline Constraints

Scheduling Method	Description	Scheduling Categories	Cloud Features considered			Cloud Issues considered			Data Transfer Time
			On-demand	Resource type	Pay-as-you-go pricing	Performance	Acquisition Delay	Fault Toleranc	
SCS (Mao & Humphrey, 2011)	Uses the concepts of bundling tasks, resource consolidation, and earliest deadline first method.	Heuristics (Dynamic)	Yes	Heterogeneous	Yes (Hour basis)	Yes	Yes	No	No
DPDS, WA-DPDA, SPSS (Malawki M. et al., 2012)	They try to maximize the number of workflows by giving numeric priorities to the workflows.	Heuristics (DPDA and DPDA are Dynamic, and SPSS is static)	Yes	Heterogeneous	Yes (Hour basis)	Yes	Yes	No	Yes (But same for all the tasks)
IC-PCP (Abrishami et al., 2013)	Allocates all the tasks of the partial critical (PCP) to a single VM	Heuristics	Yes	Homogeneous	Yes (Hour basis and 5 Min. basis)	No	No	No	No
EIPR (Calheiros, Buyya, & Membe	Improved over IC-PCP, replicate tasks on the basis of idle time of provisioned	Heuristics	Yes	Heterogeneous	Yes (hour basis)	Yes	Yes	No	Yes

r, 2014)	resources with VM								
RTC, RCT, Weighted (Poola, et al., 2014)	Selection of VMs to execute the task of workflow based on their priority (Robust, Time or Cost type) from the set of VMs in the resource pool.	Heuristics	Yes	Heterogeneous	Yes (hour basis)	Yes	Yes	Yes	Yes
JIT-C(Sahni & Vidyarthi, 2015)	Resource provisioning and Scheduling decisions are taken just before the tasks ready to execution and considers VMs performance variation and acquisition delay.	Heuristics (Dynamic)	Yes	Heterogeneous	Yes (hour basis)	Yes	Yes	No	Yes
BTS (Byun et al., 2011)	Estimates minimum resource capacity while meeting the deadline of the workflow (proposed for Gird environment).	Heuristics	No	Homogeneous	No	No	No	No	Yes
PBTS (Byun, Kee, Kim, & Maeng, 2011)	Improved over BTS, that mainly focused on cloud environment while BTS is mainly for Grid Environment.	Heuristics	Yes	Homogeneous	Yes (exactly proportional to the VMs time that may by second,	No	No	No	Yes

					minute or Hour basis)				
IHM (H. Wu et al., 2015)	Gives lower and upper bound for required VMs and using load balancing technique reduces instance hours.	Heuristics	Yes	Homogeneous	Yes (hour basis)	No	No	No	No
PSO(Rodriguez & Buyya, 2014)	A combined resource provisioning and scheduling algorithm has been proposed that use the meta-heuristic particle swam optimization (PSO).	Meta-heuristics	Yes	Heterogeneous	Yes (hour basis)	Yes	Yes	No	Yes
PSO(S. Panday et al., 2010)	They try to minimize the execution cost of the workflow while balancing the load on the available resources.	Meta-heuristics	Yes	Heterogeneous	Yes (hour basis)	No	No	No	Yes
DOGA (Chen, Z et al., 2015)	Dynamic objective strategy to with genetic algorithm when there is no any feasible solutions	Meta-heuristics	Yes	Heterogeneous	Yes (hour basis)	No	No	No	Yes
MOGA (Zhu, Z et al., 2015)	Multi-objective GA is proposed with novel encoding schemes	Meta-heuristics	Yes	Heterogeneous	Yes (hour basis)	No	No	No	Yes

CEGA (Jasraj, M. et al., 2016)	Cost effective GA is proposed with novel initialization, selection, crossover and mutation techniques	Meta-heuristics	Yes	Heterogeneous	Yes (minutes basis)	Yes	Yes	No	Yes

IV. CONCLUSION AND FUTURE WORK

In this research paper, we have done a brief survey on workflow scheduling algorithms that minimize the execution cost of the workflow while meeting the user defined deadline. We present a taxonomy and analysis of the workflow scheduling algorithms which determine what features of the cloud they have considered and also whether they resolved or not the major issues of the cloud. Further, we conclude from the analysis that most of the workflow scheduling algorithms consider all the features of the cloud and some issues. However, most of them do not focus on major issue fault tolerance in the cloud computing environment. Furthermore, none of the algorithms consider the issue termination delay of the VMs in Cloud. So, we think much work has been done that minimize the total execution cost of the workflow while meeting deadline constraint but only a few of them considers all the issues. However, our focus should be to consider other issues such as fault tolerance and termination delay of the workflow.

REFERENCES

[1] Deelman, E., Gannon, D., Shields, M., Taylor, I.: Workflows and e-Science: An overview of workflow system features and capabilities. Futur. Gener. Comput. Syst. 25, 528–540 (2009).

[2] Juve, G., Chervenak, A., Deelman, E., Bharathi, S., Mehta, G., Vahi, K.: Characterizing and profiling scientific workflows. Futur. Gener. Comput. Syst. 29, 682–692 (2013).

[3] Cooper, K., Dasgupta, A., Kennedy, K., Koelbel, C., Mandal, A., Marin, G., Mazina, M., Mellor-Crummey, J., Berman, F., Casanova, H., Chien, A., Dail, H., Liu, X., Olugbile, A., Sievert, O., Xia, H., Johnsson, L., Liu, B., Patel, M., Reed, D., Deng, W., Mendes, C., Shi, Z., YarKhan, A., Dongarra, J.: New grid scheduling and rescheduling methods in the GrADS project, (2004).

[4] Wieczorek, M., Prodan, R., Fahringer, T.: Scheduling of scientific workflows in the ASKALON grid environment. ACM SIGMOD Rec. 34, 56–62 (2005).

[5] Deelman, E., Singh, G., Su, M., Blythe, J., Gil, Y., Kesselman, C., Berriman, G.B., Good, J., Laity, A., Jacob, J.C., Katz, D.S., Gil, a, Mehta, G., Vahi, K.: Pegasus: a framework for mapping complex scientific workflows onto distributed systems. Sci. Program. J. 13, 219–237 (2005).

[6] Tim Mell, P.G.: Draft NIST Working Definition of Cloud Computing. Natl. Inst. Stand. Technol. 53, 50 (2009).

[7] Buyya, R., Buyya, R., Yeo, C.S., Yeo, C.S., Venugopal, S., Venugopal, S., Broberg, J., Broberg, J., Brandic, I., Brandic, I.: Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Futur. Gener. Comput. Syst. 25, 17 (2009).

[8] Meena, J., Kumar, M., Vardhan, M.: Efficient Utilization of Commodity Computers in Academic Institutes : A Cloud Computing Approach. Int. J. Comput. Electr. Autom. Control Inf. Eng. 9, 498–503 (2015).

[9] Schad, J., Dittrich, J., Quiané-Ruiz, J.-A.: Runtime measurements in the cloud: observing, analyzing, and reducing variance. Proc. VLDB Endow. 3, 460–471 (2010).

[10] Jackson, K.R., Ramakrishnan, L., Muriki, K., Canon, S., Cholia, S., Shalf, J., Wasserman, H.J., Wright, N.J.: Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud. 2nd IEEE Int. Conf. Cloud Comput. Technol. Sci. 159–168 (2010).

[11] Amazon elastic block store (Amazon EBS). [Online] Available: http://aws.amazon.com/ebs/.

[12] Sahni, J., Vidyarthi, D.: A Cost-Effective Deadline-Constrained Dynamic Scheduling Algorithm for Scientific Workflows in a Cloud Environment. IEEE Trans. Cloud Comput. PP, 1–1 (2015).

[13] Byun, E.K., Kee, Y.S., Kim, J.S., Deelman, E., Maeng, S.: BTS: Resource capacity estimate for time-targeted science workflows. J. Parallel Distrib. Comput. 71, 848–862 (2011).

[14] https://cloudstack.apache.org/docs/en-US/index.html

[15] http://docs.cloudstack.apache.org/en/latest

[16] A. Kivity, Y. Kamay, D. Laor, U. Lublin, "KVM: the linux virtual machine monitor, " in *Proc. Linux Symposium*, vol.1, Ottawa, Canada, 2007, pp. 225-230.

[17] http://www.linux-vm.org/page/Main_Page

[18] Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA (1990).
[19] Yu, J., Buyya, R., Tham, C.K.: Cost-based scheduling of scientific workflow applications on utility grids. Proc. - First Int. Conf. e-Science Grid Comput. e-Science 2005. 2005, 140–147 (2005).
[20] A. Afzal and J. Darlington and A. S. McGough: QoS-Constrained Stochastic Work ow Scheduling in Enterprise and Scientic Grids. In: 2006 7th IEEE/ACM International Conference on Grid Computing. pp. 1–8 (2006).
[21] Duan, R., Prodan, R., Fahringer, T.: Performance and cost optimization for multiple large-scale grid workflow applications. Supercomput. 2007. SC'07. Proc. 2007 ACM/IEEE Conf. 1–12 (2007).
[22] Garg, R., Singh, A.K.: Multi-objective workflow grid scheduling using?? -fuzzy dominance sort based discrete particle swarm optimization. J. Supercomput. 68, 709–732 (2014).
[23] Prodan, R., Wieczorek, M.: Bi-Criteria Scheduling of Scientific Grid Workflows. IEEE Trans. Autom. Sci. Eng. 7, 364–376 (2010).
[24] Yuan, Y., Li, X., Wang, Q., Zhu, X.: Deadline division-based heuristic for cost optimization in workflow scheduling. Inf. Sci. (Ny). 179, 2562–2575 (2009).
[25] Yu, J., Buyya, R.: Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms. Sci. Program. 14, 217–230 (2006).
[26] Chen, W.-N., Zhang, J.Z.J.: An Ant Colony Optimization Approach to a Grid Workflow Scheduling Problem With Various QoS Requirements. IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev. 39, 29–43 (2009).
[27] Smanchat, S., Viriyapant, K.: Taxonomies of workflow scheduling problem and techniques in the cloud. Futur. Gener. Comput. Syst. 52, 1–12 (2015).
[28] Alkhanak, E.N., Lee, S.P., Khan, S.U.R.: Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities. Futur. Gener. Comput. Syst. 50, 3–21 (2015).
[29] Mao, M., Humphrey, M.: Auto-scaling to minimize cost and meet application deadlines in cloud workflows. 2011 Int. Conf. High Perform. Comput. Networking, Storage Anal. 1–12 (2011).
[30] Malawki M., Juve G., Deelman E, N.J.: Cost and deadline-constrained provisioning for scientific workflow ensembles in iaas clouds. IEEE Int. Conf. High Perform. Comput. Networking, Storage Anal. 1–11 (2012).
[31] Abrishami, S., Naghibzadeh, M., Epema, D.H.J.: Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds. Futur. Gener. Comput. Syst. 29, 158–169 (2013).
[32] Calheiros, R.N., Buyya, R.: Meeting deadlines of scientific workflows in public clouds with tasks replication. IEEE Trans. Parallel Distrib. Syst. 25, 1787–1796 (2014).
[33] Poola, D., Garg, S.K., Buyya, R., Yang, Y., Ramamohanarao, K.: Robust scheduling of scientific workflows with deadline and budget constraints in clouds. Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA. 858–865 (2014).
[34] Byun, E.K., Kee, Y.S., Kim, J.S., Maeng, S.: Cost optimized provisioning of elastic resources for application workflows. Futur. Gener. Comput. Syst. 27, 1011–1026 (2011).
[35] Wu, H., Hua, X., Li, Z., Ren, S.: Resource and Instance Hour Minimization for Deadline Constrained DAG Applications Using Computer Clouds. IEEE Trans. Parallel Distrib. Syst. PP, 1–1 (2015).
[36] S. Pandey and L. Wu and S. M. Guru and R. Buyya: 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. pp. 400–407 (2010).
[37] Rodriguez, M.A., Buyya, R.: Deadline Based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds. IEEE Trans. Cloud Comput. 2, 222–235 (2014).
[38] Chen, Z., Du, K., Zhang, J.: Deadline Constrained Cloud Computing Resources Scheduling for Cost Optimization Based on Dynamic Objective Genetic Algorithm. 708–714 (2015).
[39] Zhu, Z., Zhang, G., Li, M., Liu, X.: Evolutionary Multi-Objective Workflow Scheduling in Cloud. IEEE Trans. Parallel Distrib. Syst. 9219, 1–1 (2015).
[40] Jasraj, M., Malay, K., Manu, V. :Cost Effective Genetic Algorithm for Workflow Scheduling in Cloud under Deadline Constraint. in IEEE Access. doi: 10.1109/ACCESS.2016.2593903
[41] Smanchat, S., Viriyapant, K.: Taxonomies of workflow scheduling problem and techniques in the cloud. Futur. Gener. Comput. Syst. 52, 1–12 (2015).
[42] Alkhanak, E.N., Lee, S.P., Khan, S.U.R.: Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities. Futur. Gener. Comput. Syst. 50, 3–21 (2015).