EXPLORING DIRECTIONS IN EFFICIENT SCHEDULING OF SCIENTIFIC WORKFLOWS IN A CLOUD ENVIRONMENT

B. R. Mandre, Vaishali Singh
Maharishi University of Information Technology, Lucknow, India
bmandre@gmail.com

ABSTRACT: *Executing scientific workflows in an elastic cloud environment is beneficial because in elastic cloud resources required for the application can be increased to speed up application execution. While trying to meet workflow execution deadlines with minimizing execution costs, scheduling workflows is the most challenging problem. This paper provides the different types of workflow scheduling algorithms and their comparison with various parameters. Previous work for clusters and grids has limitations in the capacity of resources and the existing workflow scheduling algorithm either tries to meet deadlines or tries to minimize cost. Therefore, there is a need to develop a new scheduling algorithm that completes workflow execution within the deadline and at a smaller cost.*

Keywords: cloud Computing, Scientific workflow, Task Replication, Soft deadline.

INTRODUCTION:

To provide flexible and on-demand services for a number of companies, cloud computing becomes more attractive. Cloud computing can be defined as a type of parallel and distributed system which consists of a collection of interconnected and virtualized computers that are dynamically provisioned and represented as one or more computing resources based on service-level agreements which are established between the service provider and customers.

The aim of this paper is to focus on different algorithmic workflows for scheduling. The remainder part of this paper has been arranged according to the following format. Section I focuses on workflows within cloud computing. Section II analyzes various current workflow scheduling algorithms as well as tables. Section III closes the paper with a conclusion.

Section I: Workflows in Cloud Computing

The idea of workflow comes from the concept of process within the industry of manufacturing. The processes that are described here in after were the outcome of making standards. It seeks to improve performance by focusing on the most repetitive characteristics of the task. Each procedure is defined as an assignment, role, or procedure that is frequently used during the manufacturing of specific goods on a large scale. In the past, humans performed all these procedures by manipulating physical objects. Since the advent of information technology, more processes are automated using computer software. It classifies processes within a company into:

- Material processes: Deliver the products by assembling the physical parts
- Information processes: Provide data and information utilizing process.
- Business processes: Tasks involved in business activities.

Workflow is mostly focused on the automated execution of processes. To achieve the overall goals, it creates the process more efficient, since documents and data move among parties agreeing to an established rule. The workflow can be used to organize applications within an acyclic-directed graph. This graph is composed of every node denoting the primary task, while edges denote the dependencies among the tasks. The workflow is typically comprised of a number of small tasks that could be connected to an additional task within the workflow [5].

Scientific Workflow Systems are the ones used to build and run various workflows within scientific applications.

Workflow models are widely utilized in many fields:

- Astronomy: Workflows are utilized to build custom-designed space-based mosaics using a set of images that are input.
- Bioinformatics: Workflows are utilized for automatizing the procedure of looking for encoding genes of sRNA.
- Physics: Workflows are used to detect gravitational waves.

Scientific workflows can be characterized into six different types:

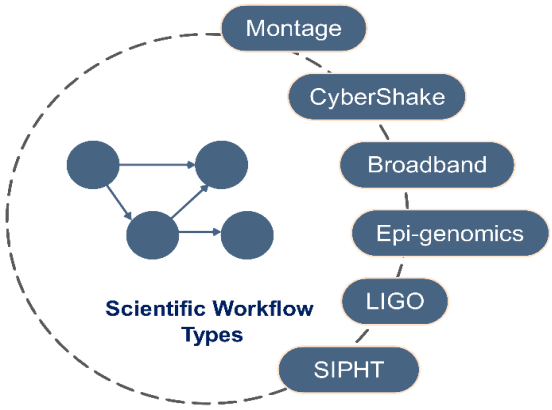

Figure 1: Scientific Workflow Types

1. Montage is developed by NASA/IPAC. It is an open-source toolkit that allows you to produce sky mosaics by utilizing images that are input. The format used to store images is called the "Flexible Images Transport System" format.
2. CyberShake: This workflow is employed for the Southern California Earthquake Center. It is used to classify earthquake threats by using an analysis of the Probabilistic Seismic Hazard method.
3. Broadband: It is a computing model that is applicable in the Southern California Earthquake Center. Broad band's goal Broadband is to incorporate an array of motion simulation algorithms and calculations that produce research

findings that can be useful for earthquake engineers.

4. Epi-genomics: This workflow is essentially an information processing pipeline. It is used to facilitate the execution of different genome sequencing procedures.
5. LIGO: It's the workflow utilized for analyzing the information derived from the merger of binary systems that are compact, like black holes.
6. SIPHT: The workflow can be utilized to computerize the discovery of sRNA- genes found in biotechnology [12].

Workflow Scheduling:

The task of scheduling workflow has to meet the process of allocating every activity load to the appropriate resources. Thus, allowing tasks to be able to meet some performance standards. A workflow is a series of linked steps. Workflows are typically focused on automated processes but also to accomplish the general target. The data and documents are shared among members as per predetermined guidelines. Workflows allow the organization of programs in an Acyclic Graph Directed form. Here, node signifies the work being performed while edge indicates interdependencies among the tasks of the application. A particular workflow is the collection of work activities. Every activity from them communicates with other activities within the workflow. It supports Workflow Management Systems. Workflow scheduling is responsible for determining the resources needed and assigning work to appropriate resources. Workflow scheduling plays an essential part in managing workflows. Proper scheduling of workflows can be a significant influence on the efficiency and efficiency of the process. To ensure that workflows are properly scheduled, there are several algorithms for scheduling.

Taxonomy for Workflow Scheduling

They can be divided into three types depending on the information available about workflow, resources, and when tasks were assigned to resources. This allows you to distinguish between different workflow scheduling methods.

Static Scheduling

The information about DAG characteristics, structure, task length and edge sizes, etc. is known in advance. All resources have instant availability and stable performance. Hence the execution time and communication time of any legal task resource assignment are definite. The scheduling is performed prior to DAG running [12].

Static Planning and Dynamic Scheduling

S-Plan-D-Sched scheduling combines dynamic and static scheduling. Static planning for all tasks is made based on an approximate estimate of activity execution time and inter-communication time. The job is automatically adjusted at runtime and rescheduled as needed. The execution time and communication time can be estimated. But tasks cannot be assigned to resources immediately. Tasks are statically planned based on estimation prior to running but are dynamically scheduled to resources at runtime [12].

Dynamic Scheduling

The execution time and communication time can only be obtained at runtime. This may be caused by incomplete DAG information or indeterminate resources. The scheduling is performed at runtime. At each scheduling step, a ready task is selected and dispatched to selected resources [12].

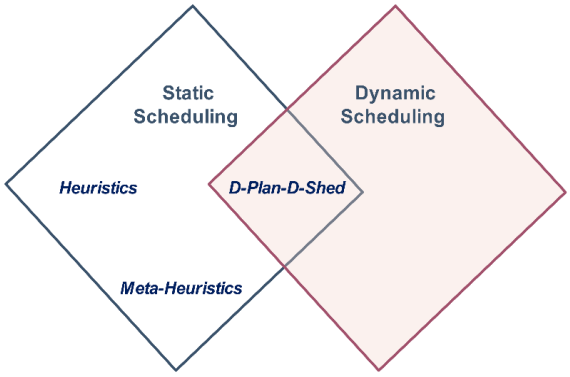

Figure 2: Workflow Scheduling Taxonomy

Critical Path-Based Scheduling

Scheduling of workflow is an essential feature of workflow job execution. Various scheduling algorithms can produce substantial differences in resource utilization and make span. Most of these algorithms use the concept of a critical path. It represents the lengthiest route in DAG that represents workflow structure. The critical path is the time it takes to complete a given DAG's workflow execution. It is calculated by adding the computing budgets for all activities within the critical path. Due to its impact on effectiveness and pricing, many solutions have been suggested in the research works. Some samples of such results include HEFT (heterogeneous earliest-finish-time) DCP (dynamic critical path), CPOP (critical path on a processor), and CPF (critical path first). Most studies are focused on either lessening make-span with the available resources or diminishing price by decreasing VM nodes required to execute the workflow job with a suitable make-span.

DCP –It is a proposal that assigns tasks to activities taking into account the tasks on the critical path. It indirectly prohibits the needless (mostly unnecessary) use of CPUs in non-CP jobs. It only studies computers that are in the workflow schedule [12].

HEFT/CPOP - [9] Advocates CPOP and HEFT for a limited count of mixed processing units for concurrently achieving high efficiency with quick scheduling. The HEFT chooses the activity or task that has the maximum increasing rank rate and allocates it to the processor. It reduces the task's earliest finish time by using an insertion-based method. For prioritizing tasks, the CPOP uses the addition of upward and downrank values. The CPOP algorithm adds additional flexibility to the processor choice stage. It schedules critical activities onto the CPU to minimize the run time. The authors created a "parametric graph generator" that generates weighted DAG with different features to compare HEFT with other workflow scheduling algorithms. The results show that HEFT can be shorter with lower costs than CPOP.

PCP – [1] The partial critical path algorithm was proposed to reduce cost and meet user-defined deadlines. This took advantage of the utility grid's negotiable pricing mechanism.

The algorithm assigns sub-deadlines for CP tasks to the tasks that are left, and then it schedules activities based on the CP task's time deadlines. The algorithm now schedules activities in a service that is the least expensive and can meet the sub-deadline constraints [8].

CPF [20] proposed the critical path first algorithm based on the assumption that there is a sufficient resource in a public cloud. It first extends the work plan to prevent the length of the critical path from being exceeded. The schedule is then compressed for resource efficiency. Tasks that are inefficient or idle slots are rearranged. It improves both resource utilization and makespan with this two-step method.

In general, the workflow scheduling problem has been studied extensively and thoroughly, with various solutions pushing both performance and cost to their limits.

Section II: Analysis of Related Work

Numerous writers have presented their work in the field of scheduling workflow algorithms. SCS [6] is a deadline-controlled system. It utilizes an automatic scaling algorithm to automatically allocate and de-allocate VMs depending on the existing state of the activities. It starts by grouping the activities in bundles to decrease the time for information transfers and assigning the overall goal to activities. Hereafter, it constructs an array of loads after identifying the maximum effective Virtual Machine kind for the activity. The load array is reviewed each time the scheduling occurs. It reveals the number of computers of the same type is required to allow the task to be completed by the agreed deadlines at the least cost. After that, the algorithm moves on to combine partial instances by combining tasks that are running across various types of instances into one [10]. This happens when VMs are idle and are able to complete additional tasks within their initial deadline. Additionally, using the "Earliest Deadline First (EDF) algorithm" is employed to assign activities to active VMs. This means the activity that has the most recent time-deadline is then planned when a VM with the appropriate kind is made vacant.

There is an algorithm that creates a preliminary allotment schedule depending on the overall optimizing algorithm and then improves it as the program runs to respond to any delays not anticipated is SCS. This improvement system permits it to reduce both the price as well as execution time alteration to ensure sufficient VMs are available within the resource pool to complete the tasks on time. An algorithm optimizes the entire system for remaining tasks every time a task is scheduled to adjust the provisioning plan. This increases the computational load and decreases the system's ability to scale with the number of tasks in the workflow.

In 2011, the Partitioned Balanced Scheduling (PBTS) algorithm [14] was developed to manage workflows in a homogeneous set of VMs by splitting its execution in such a way it can be scheduled so that decisions on scheduling are taken each billing cycle. Its primary purpose is to determine, for each scheduling cycle/partition, the minimum computing resource required to complete each workflow within the specified timeframe. PBTS determines the order of tasks in each part by using an estimated capacity that takes into account all costs. It then calculates the exact amount of resources required to complete the task within each part using BTS. These same researchers originally created Balanced Time Scheduling, (BTS) algorithm [14]. Next, the actual VMs will be assigned and tasks executed according to the BTS schedule.

Others [5] suggested two strategies for planning workflow groups. These strategies aim to maximize the number and meet deadlines and budgetary requirements while maximizing workflows. Based on budget and deadline, the "Dynamic Provisioning Dynamic Scheduling algorithm (DPDS)", first determines how many VMs you will need to start using. The VM pool is constantly updated according to usage. If utilization falls below a threshold, VMs will be shut down. If utilization surpasses the specified threshold. If budget permits, the fresh virtual machine is hired. The planning stage allocates activities depending on their precedence to randomly selected processors. This continues till all instances of the job have been completed. "WA-DPDS (Workflow aware DPDPS)" is a modified algorithm that increases efficiency by only performing tasks that fall within specified QoS limits. The system implements an admission control system that ensures that only workflows which can be finished in the given pricing are planned for completion. The author established that DPDS is capable of handling unexpected delays. These include delays in provisioning or inaccurate estimates of the time it takes to complete a task. The algorithm has one drawback. It leases more VMs than is allowed by the budget, starting at the time that the ensemble execution takes place. This can lead to VMs being jobless for a longer time while they pause for activities to be completed, which results in wasted timeslots and more billing time.

The "Static Provisioning Static Scheduling algorithm (SPSS)", allocates time limits for all activities depending upon the working schedule. This is how long the workflow can take, but still finish on time. So that the cost is low and deadlines can be met, the tasks are automatically assigned time slots in the existing VMs. If the slots do not meet these requirements, new VMs can be used to schedule the tasks. Because SPSS is a non-dynamic and fixed method, authors expressed that it can be extra delicate than DPDS to environmental modifications. However, it is superior to its dynamic counterpart in terms of quality and efficiency. It can use its knowledge of the workflow structure and examine different outputs to determine which option is best. SPPS's main problem is its static nature and inability to understand VM timings for provisioning. The authors found SPPS to be too sensitive to delays.

In 2012, Genez et al. [3], created a SaaS provider that allows customers to execute workflows. SLA contracts of basic two kinds can be created. These SLAs can be applied in renting VMs through IaaS platform, including subscription or static mechanisms. They focus on Amazon EC2's options, which include reserved instances and on-demand. It is the SaaS provider that is able to offer reserved instances which can be used to run workflows before a user has set a deadline. On-demand instances can be purchased to fill the requirements of the workflow if the reserve instance infrastructure is not sufficient. Although their method is presented in the context of a SaaS service that can serve multiple users, it is only designed to manage one workflow at a time. The mixed integer linear programming (MILP), which solves the scheduling problem, aims to reduce execution costs and

ensure that the work is completed within the given deadlines. Two heuristics are proposed by the authors that enable them to create an achievable schedule using the less restrictive MILP. They propose an algorithm that can pick the best IaaS platform with the required VMs with the main focus on maintaining service quality.

"IaaS Cloud Partial Critical Path algorithm (IC-PCP)" [2] targets to lower runtime costs while still managing to meet time constraints. The algorithm begins by identifying all tasks and partial critical paths[11]. Every task associated with each path is scheduled in the same VM. The task should be assigned to an instance already leased that can meet the current deadlines. If this is not possible, the task should be transferred to a new rented VM which is affordable to complete the tasks within time. The process continues until all activities get assigned.

The authors suggest I-PCPD2 (IC-PCP that includes a time-stamp distribution). The primary change between the two algorithms can be stated as instead of putting each task in the same VM the IC-PCPD2 assigns every task in the most affordable VM which can complete it in time. The authors claim that ICPCP surpasses ICPCPD2 in a majority of instances [13]. This is one of the major benefits of ICPCP and is an important factor to take into consideration when working with workflow execution in cloud environments the fact that data transfer times could be a significant influence on the duration and expense of executing a job. The "IC-PCP" solution reports the issue by allocating child and parent activities in the same machine, thus decreasing VM-to-VM communication.

The drawback of IC-PCP is that it doesn't consider VM provisioning delays or variations in resource performance. This means it is extremely vulnerable to the performance of CPUs and results in deadlines being delayed due to unpredicted delays. Its heuristic and static nature lets it to generate high-performance schedules quickly, which makes it appropriate for scheduling large-scale workflows that have hundreds of tasks. Therefore, IC-PCP is better suitable for scheduling larger workflows and tasks which require a low CPU to ensure that the impact of resource degradation is minimized.

In 2013, Pietri et al. propose two algorithms for scheduling workflow ensembles in cloud environments that are both founded on SPSS [5]. The authors propose two algorithms to schedule workflow ensembles in clouds [5]. An algorithm, called SPSS-ED, concentrates on meeting completion time limitations. The "SPSS-EB" concentrates on adjusting the energy as well as pricing limitations. The steps are designed to increase the number of jobs that are completed. Each workflow within the cooperative SPSS-EB designs its execution by scheduling steps in a way that the total amount of energy used is minimal. Then, it takes the plan into consideration and permits the execution of the workflow only when budget and energy constraints are fulfilled. A similar process is employed in SPSS-ED, but in lieu of budgets, deadlines are taken into consideration as the restriction.

[4] Authors have proposed the "Enhanced IPCP using Replication (EIPR)" algorithm. It is an algorithm for planning and provisioning that makes use of the vacant time of the provisioned VMs along with the excess budget to duplicate activity to decrease the impact of deviation in performance and to achieve the time deadline limitation of the application. The initial step is used in determining the quantity and type of VMs that will be used along with the order and location of the tasks that are assigned to the resources. This is accomplished by utilizing the IC-PCP's main heuristic, recognizing the fractional critical routes, and allocating their duties for the identical VM. The next process is to identify the time of beginning and ending of VMs. The EIPR takes into account both the time of start and finish of the tasks, in addition to both input and output transmission time. It also replicates activities within vacant periods in the VMs that have been provisioned when the pricing permits it. It orders the tasks' repetition with the highest proportion of execution time to the time available, then the activities with lengthy rum times, and lastly activities that have a high count of children.

The PSO-based algorithm was developed in 2014 by Rodriguez and Buyya and presented the static, cost-minimization deadline-constrained method [4] that takes into consideration aspects like flexible provisioning and heterogeneity of infinite computing resources and VM performance variability. Resource provisioning, as well as workflow planning, are combined and exhibited as a "Particle Swarm Optimization (PSO)" problem. The result will be an optimum plan which determines the amount and type of virtual machines to utilize and their lease durations and the need for map resources.

The technique of global optimization is a major benefit of the algorithm since it permits it to produce top-quality schedules. Additionally, in order to take into account, the incapability of a static schedule to changes in the environment, authors present an estimation of the decline in performance felt by VMs when they calculate time-to-run. This way an amount of acceptance of the unpredictable nature of the environmental conditions is presented. The infinite supply model is effectively recognized, moreover, the computing cost grows exponentially with the number of activities included in the job.

In 2014, Durillo and Prodan developed the "Multi-Objective, Heterogeneous Earliest Finish Time (MOHEFT)" method [4] to be an enhancement to the familiar DAG scheduler HEFT. This is the heuristic-based algorithm that calculates a set of Pareto methods so that users can choose the most suitable one. MOHEFT creates numerous transitional workflow plans, with each step analogous. The efficiency of these results is assured through dominance relationships, and their variety is guaranteed through the use of the metric called "crowding distance". This algorithm is universal in terms of the variety and number of goals it is able to handle but costs and makespan were improved when executing workflow software on Amazon cloud [15].

In 2014, Poola et al. [7] proposed an algorithm to schedule tasks based on spot and on-demand kinds of cloud instances. In particular, it takes into account one type of spot VM (the most affordable) and a variety of VMs on request. The scientists describe the idea of the "latest time to on demand" (LTO). It decides when the algorithm must change from using spot instances to on-demand instances in order to ensure that the user-defined deadline is fulfilled. The bidding

method to use the spot VMs is also suggested; the bidding begins close to the spot price at first and grows as the execution progresses so that it is closer to the price on demand as the LTO gets closer. This decreases the likelihood of out-of-bid situations close to the LTO and improves the likelihood to meet the time[16, 17].

In 2015, others proposed mathematical models that optimize how much it costs to schedule workflows when there are deadline constraints[12]. The proposed method is an overall optimization of data and task positioning by defining the scheduling issue as a "mixed integer Program". Alternate forms of this algorithm have been described. The first one is for "coarse-grained workflows" where the activities run approximately an hour. The other is for the second designed for "fine-grained workflows", with numerous tasks that are short and have deadlines of less than one hour [12, 18].

In 2015, the Security-aware and budget-aware (SABA) algorithm has been developed to plan workflows within multi-cloud environments [7]. The authors outline the concept of immovable and moveable datasets. Data that is movable does not have safety constraints, and therefore can be transferred among data centers and can be replicated when needed. Data that is not movable, on the other hand, are limited to one data center and are not able to be replicated or moved because of safety or price issues. The algorithm has three major stages. The prioritization and clustering phase is where activities and information are allocated to distinct data centers according to the workflow's irremovable datasets. Furthermore, the priority assignment is made to tasks based on the computation and I/O expenses in relation to a basic type of Virtual Machine.

The next phase is to assign activities to VMs according to performance-cost relation. In the final stage, transitional data is dynamically relocated in the course of the run time, according to the location of the tasks in the process of being executed to guide this procedure. SABA determines the price of the VM by calculating the time at which it starts the first task it is assigned and the time at which it finishes the last task that is planned for it. While the scientists do not explicitly discuss a "resource provisioning plan" however, the Virtual Machine's beginning and finish times could be determined from the corresponding time values of the activities [7].

In addition to the protection for data SABA is also able to consider tasks that may require security solutions like authentication security, integrity, and confidentiality and also the over the top of these services when they make the estimation of their time and costs. Furthermore, instead of focusing on the CPU power of virtual machines to determine the time to run, SABA also considers features such as I/O, and bandwidth in addition to memory. It is important to note that the cost for VMs is computed using the sum of the amount of time that the VM is used for, and the duration of pricing set by cloud enterprise isn't taken into consideration. This could cause greater VM expenses than anticipated in the case of using the algorithm in an actual cloud environment. Other expenses to consider include the cost of data transfer among data centers and the cloud storage that is used to store the I/O information [19, 21].

Dyna is a schedule planning tool that reflects the changing character of cloud systems from both a price and performance perspective. It uses the same resource model as Amazon EC2 and takes into account both on-demand and spot instances. Its purpose is to lower the price of execution of workflows and provide deadline assurance. This takes into consideration the variability in resource performance as well as the cost dynamics of spot instances. On-demand instances can be used to meet deadlines if Spot instances aren't able to complete tasks on time. Spot instances reduce the cost of infrastructure. A static hybrid configuration plan is created for each instance. This plan combines both on-demand and spot occurrences.

After studying and analyzing this workflow scheduling algorithms like - Ant Colony Optimization, Particle Swarm Optimization, Heuristic, Genetic, Hybrid, and Hyper-heuristic algorithms. It is clear that there is a need to explore critical path-based algorithms in a cloud environment. These algorithms should have to provide a good success rate while meeting workflows budget and deadline constraints. Table 1 shows the summary of work done and future directions for further research.

✓: Tick sign shows that work has already been done in that area

?: Question mark means that there need to explore scheduling algorithms for that domain focusing on different parameters like cost optimization, deadline-constrained, budget-constrained, reliability, resource utilization, availability, and energy efficiency.

Table 1: Analysis of different Algorithm with Parameter

Parameter → / Algorithm ↓	Makespan	Resource utilization	Deadline Constrain	Budget Constrain	Load balance	Success Rate	Cost	Energy Efficient	Time
Parameter Based	✓	?	✓	✓	?	✓	✓	✓	✓
GA	✓	✓	✓	✓	✓	?	✓	?	✓
Critical Path	✓	?	?	?	?	?	✓	?	✓
Multiple QoS	✓	?	?	?	✓	✓	✓	✓	✓
ACO	✓	✓	?	?	✓	✓	✓	✓	✓
PSO	✓	✓	✓	✓	?	✓	✓	?	✓
Market Oriented	✓	✓	?	?	?	?	✓	?	✓

Section III: Conclusion

Different existing workflow scheduling algorithms are studied and analyzed and tabulated on the basis of the nature of the scheduling algorithm, type of algorithm, and objective parameter. By analyzing these algorithms, it is clear that many authors have done a lot of work in the area of workflow scheduling but still, there are many areas that require further attention.

In this paper, we have surveyed the various existing workflow scheduling algorithms in cloud computing and summarized their various parameters along with the tools and algorithms used. From the literature reviewed, it is clear that a lot of the work is already done in this area, but still, there is a need to improve work done in critical path-based algorithms. For example, there is a need to explore resource utilization in workflow applications. These areas have been marked in table 1.

REFERENCES:

1. S. Abrishami and M. Naghibzadeh, "Deadline-constrained workflow scheduling in software as a service Cloud," Scientia Iranica, Transactions D: Computer Science & Engineering and Electrical Engineering," pp. 680-689, 2012.
2. S.Abrishami, M.Naghibzadeh, and D.Epema, "Deadline-Constrained Workflow Scheduling Algorithms for IaaS Clouds," Future Generation Computer System, vol. 29, no. 1, pp. 158 - 169, January 2013.
3. L. F. Bittencourt, and E. R. Madeira T. A. Genez, "Workflow scheduling for SaaS/PaaS cloud providers considering two SLA levels." in Proceedings of the IEEE Network Operations and Management Symposium (NOMS), pp. 906-912, 2012.
4. Rajkumar Buyya, Rodrigo, and N. Calheiros, "Meeting Deadlines of Scientific Workflows in Clouds with Tasks Replication," IEEE, vol. 25, no. 7, pp. 1787-1796, July 2014.
5. G. Juve, E. Deelman, and J. Nabrzyski M. Malawski, "Cost-and deadline- constrained provisioning for scientific workflow ensembles in IaaS clouds." in Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, IEEE, p. 22, 2012.
6. M. Mao and M. Humphrey, "Auto-scaling to minimize cost and meet application deadlines in cloud workflows." in Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC). p. 49, 2011.
7. Poola.D., S.K.Garg, R.Buyya, Y.Yang, and K. Ramamohanarao, "Robust scheduling of scientific workflows with deadline and budget constraints in clouds," IEEE 28th International Conference on Advanced Information Networking and Applications (AINA), 2014.
8. Rodriguez Sossa M. and Rajkumar Buyya, "Deadline based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds," In IEEE Transactions on Cloud Computing, pp. 222-235, 2014.
9. Salim Hariri, and Min-youWu Haluk Topcuoglu, "Performance-effective and low-complexity task scheduling for heterogeneous computing." IEEE Transactions on Parallel and Distributed Systems, vol. 13, no. 3, pp. 260–274., 2002., vol. 13, no. 3, pp. 260-274, 2002.
10. Ranjit Singh and Sarbjeet Singh, "Score Based Deadline Constrained Workflow Scheduling Algorithm For Cloud Systems," International Journal on Cloud Computing: Services and Architecture, vol. 3, December 2013.
11. J. Sonnek, J. Greensky, R. Reutiman, and A. Chandra, "Starling: Minimizing Communication Overhead in Virtualized Computing Platforms Using Decentralized Affinity-aware Migration," Proc. the 39th IEEE International Conference on Parallel Processing (ICPP), pp. 228-237, Sept. 2010.

12. Yusong Tan, Fuhui Wu Qingbo Wu, "Workflow Scheduling in Cloud: a Survey." Springer Science Business Media New York, pp. 3373-3418, May 2015.
13. Xiaofeng Wang, Chee Shin Yeo, Raj Kumar Buyya, and Jinshu Su., "Optimizing the Makespan and Reliability for Workflow Applications with Reputation and a Look-ahead Genetic Algorithm," Future Generation Computer Systems, vol. 27, no. 8, p. 1124, 2011.
14. Q.Wei, G. Xu, and Y. Li, "Research on Cluster and Load Balance based on Linux Virtual Server," Proc. Information Computing and Applications, pp. 169-176, 2011.
15. Kai Wu, Lok Kei Leong, Seungbeom Ma, and Melody Moh Jing Huang, "A TunableWorkflow Scheduling Algorithm Basedon Particle Swarm Optimization for Cloud computing," The International Journal of Soft Computing and Software Engineering, vol. 3, no. 3, p. 351, 2013
16. Y. Yang et al., "An Algorithm in swindew-C for Scheduling Transaction-Intensive Cost-Constrained Cloud Workflows," in Proceeding of 4th IEEE International Conference on e- Science, 2008, pp. 374-375.
17. C. S. Yeoa, S. Venugopala, J. Broberga, and I. Brandicc R. Buyya, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility." Future Generation Computer Systems, vol. 25, no. 6, pp. 599-616, 2009.
18. T. You, W. Li, Z. Fang, H. Wang, and G. Qu, "Performance Evaluation of Dynamic Load Balancing Algorithms," TELKOMNIKA Indonesian Journal of Electrical Engineering, vol. 12, no. 4, 2014.
19. Y. Zhao, I. Raicu, and S. Lu I. Foster, "Cloud computing and grid computing 360- degree compared," In Grid Computing Environments Workshop GCE, vol. 08, no. 2008.
20. J. Zhao, Y. Ding, G. Xu, L. Hu, Y. Dong, and X. Fu, "A Location Selection Policy of Live Virtual Machine Migration for Power Saving and Load Balancing," The Scientific World Journal, vol. 2013, no. 2013, p. 492615, Sept. 2013.
21. C. Zhou and B. He, "Transformation-based monetary cost optimizations for workflows in the cloud." IEEE Transactions on Cloud Computing, pp. 1-1, 2014.