EWSA: An Enriched Workflow Scheduling Algorithm in Cloud Computing

Shilpi Saxena
Department of Computer Science and Information Technology
Graphic Era Hill University
Dehradun, India
25shilpisaxena@gmail.com

Deepika Saxena
Department of Computer Science and Information Technology
Dayanand Mahila Mahavidhyalaya
Kurukshetra, India
13deepikasaxena@gmail.com

***Abstract*—Cloud computing environment is one of the latest research area where research is taking place at a very fast pace. This concept of computing which is based on cloud is expanding its umbrella day by day, taking almost every computing activity under its shed either it is related to resources, tasks or data storage. This hiking interest of users has made it quite challenging for the providers to meet up the consumer desires.**

Scheduling of tasks plays an important role in the performance of any cloudlet scheduling algorithm. Tasks scheduling if observed closely in realistic world is based on dependent tasks i.e., they represents a workflow. A number of scheduling algorithms have been proposed for independent tasks but very few are there which operates on dependent tasks. Here, in this paper we propose a new scheduling algorithm for workflow, which are having dependencies among tasks, taking into consideration important parameters of transfer time and bandwidth along with basic requirements of optimizing the execution time and cost. The simulation is experimented using cloudSim toolkit. Our algorithm provides better results over other existing algorithm like PSO and CSO and is more closely related to real world scenario.

Keywords—Cloud computing; Workflow; Tasks scheduling; CloudSim; Virtual Machine

I. Introduction

Cloud computing is very popular these days. The reason behind this popularity is the type of services offered by it. It has made it possible for everyone to access and use its services as per user's choice and requirements. The provision of services on rental basis is one of the major reasons for its popularity. Besides this, the services it offers is also an important reason as it not only offers ready to use applications as a service but it also does provide the platform for application development and its own infrastructure to build and run those applications on rental basis. These three services, Software-As-A-Service (SAAS), Platform-As-A-Service (PAAS) and Infrastructure-As-A-Service (IAAS) are unique in the world of computing. Cloud computing is basically a computing which is performed over a number of resources that are widely distributed at various datacenters throughout the world. Some of the common challenges of any computing are storage management, execution time, cost, latency, energy, data security, virtualization and scheduling. Scheduling of tasks is always been the primary concern for better performance of any computing scheme. Scheduling of tasks means defining the order in which tasks are going to be executed on virtual machine. Cloud computing do provide several special characteristics to workflows like:

1. Resources are available in the form of standardized services and can be availed as per the user's the choice.
2. The number and type of resources allocated to a workflow is decided as per the customer's requirement.
3. The number of resources allocated to a workflow can be changed dynamically at runtime and hence rightly said elastically scalable.

Broadly, the workflows can be categorized under two categories, a) Business workflow and b) Scientific workflow. A scientific workflow is basically represented by a Direct Acyclic Graph (DAG). Thus, the tasks are dependent in nature and while scheduling tasks this dependency needs to be taken special care. The allocation of tasks and resources at runtime and dynamically mapping resources to meet the performance expectations is to be handled with deep care.

Scientific workflows usually represent a complex and huge form of data. Handling, managing and performing calculations over such type of data is a big challenge. Computation of such type of data in an efficient way is surely going to contribute a lot in handling the data of latest emerging technologies.

There are various scheduling algorithms which takes into account Qos factors, but most of them are proposed for independent tasks which is quite impractical in consideration with the real time scenario. In this paper, we have proposed an optimized workflow scheduling algorithm which takes into account almost every kind of important QoS factors like energy, cost of execution, meeting deadline, etc. that's why the algorithm is named as "EWSA: An Enriched Workflow Scheduling Algorithm". Our algorithm shows improved performance results over existing algorithms.

The rest of the paper comprises of following sections, Second section enlists the previous work related to this. Third section describes the proposed methodology, its algorithm and flowchart. Fourth section shows the results in tabular form as well as in graphical format. Finally, the fifth section concludes our work and shows future perspectives.

II. Related Work

Mapping of tasks to machines for their execution is termed as scheduling. This scheduling can be done using various scheduling strategies. The performance of any machine depends truly on the scheduling strategy followed for directing the tasks on the machine. Again it should also be considered that which type of task is to be scheduled, as the tasks can be dependent one or independent one. It is quite easy to schedule independent tasks but scheduling of dependent tasks i.e., workflows is bit challenging, as you have to keep track of the dependency of tasks. It is to be closely observed and handled that which task is dependent on which one, and which task needs to be performed before its dependent tasks. This thing is bit complex about the workflows. As the workflows are closely related to real time tasks and are dynamic in nature. Thus more pragmatic algorithms should be designed to deal with this in an efficient way.

[7] shows scheduling of workflows depending upon different QoS requirements. It takes into consideration the type of service a ready task is demanding and based upon its requirement the scheduler checks if that service can be provided and assigns the task to that resource. Otherwise, the task have to wait for its turn. After regular intervals this availability of resources is checked, as a resource becomes available it is assigned to the waiting task. The problem with this approach is that, a lot of time is wasted in calculating the status of resources after every round. Apart from this, there is no strategy defined that in which order the waiting tasks will be assigned to the machine.[8] provides a compromised approach for time and cost based scheduling of workflows. This technique computes cost of execution and time of execution for each task on every resource, and then assigns the task to the resource which executes the task within deadline and incurs minimum cost. The major flaw of this approach is that, after every round it finds out the tasks that have failed to execute successfully and gives priority to them over new tasks to be assigned to a resource for execution that is, the new tasks in the workflow suffers starvation. Due to this, the workflows may fail to achieve their deadlines. [9] proposed a Particle Swarm Optimization (PSO) algorithm for workflows. It works on QoS features time and cost, and improves the distribution of workload and achieves cost optimization over existing algorithm Best Resource Selection (BRS). The technique works on heuristic based PSO, it should be modified for more realistic approaches. [10] Implements a workflow scheduling algorithm which surpasses many other existing algorithms in terms of execution time and scaling elastically.

[11] a three bi-criteria complementary approach has been devised which focuses on optimization of cost of execution and time of execution while scheduling of workflows. Cat Swarm Optimization (CSO) workflow scheduling technique is more close to realistic scheduling as it considers both cost of data transmission between two dependent resources as well as cost of execution of tasks on different resources [12]. [12] the technique shows improved results in terms of task to resource scheduling cost, number of iterations and load distribution among available resources. [13] a cloud based workflow architecture for Smart city is been introduced here and some of the workflow scheduling algorithms are also been reviewed to define a taxonomy of workflow scheduling and management in cloud environment. [14] an optimized scheduling scheme is proposed for highly dependent tasks to minimize the energy consumption and enhance the lifetime of the network.

Thus, various schemes have been proposed with time to schedule workflows in the cloud environment. Every scheme considers one or the other QoS factor and tries to provide better results over the existing ones. But in the proposed algorithm we have undertaken all the good points of existing algorithm with the added improvement which is provided by assigning level to each task in the workflow, while considering their parent child relationship.

III. Workflow Scheduling

The workflow is considered in form of Directed Acyclic Graph (DAG) G=(Vr,E), where Vr is defined as the set of tasks T={T1,T2,......,Tn} and E is set of edges that represents dependency between resources. It is assumed that a set of processing resources is represented by R= {R1, R2,......, Rn}. The processing resources are distributed throughout the network.

The experimental workflow assumed here consists of 11 tasks which are divided into 5 different levels to be scheduled on six different virtual machines. Fig.1 represents the workflow considered in our proposed work.

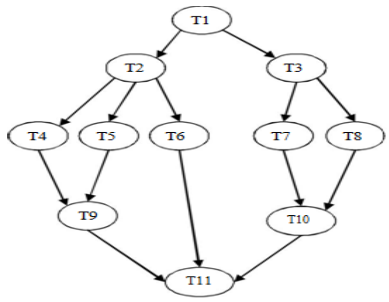

Fig.1 Workflow for Experiment

Task T1 is the initial task which is at first level, task T2 and T3 are dependent on task T1 and are placed at level 2. Task T4, T5 and T6 are dependent on task T2 and task T7 and T8 are dependent on task T3. All these are placed at level 3. Further task T9 is dependent on task T4 and T5 and task T10 is dependent on T7 and T8, it is at level 4. Finally, the workflow completes at task T11which is dependent on task T9, T6 and T10 and is at level 5.

A. *Methodology of Proposed Algorithm*

1. User submits workflows at datacenter. Sorting of all the available workflow is done in ascending order of deadline is done.
2. Take smallest deadline workflow from top of list. For scheduling each workflow follow the below steps.

a) Prepare transfer time matrix for the workflow and also assign levels to each task in the workflow.
b) Child task belonging to same parent task are at same level and total number of task at each level are counted.

3. For assigning task belonging to a workflow to resource VM for execution, do follow below steps:
   a) At each level one by one each task is checked that if it is having parent task, then first execute its parent task, and calculate its file output size. Otherwise, simply assign the task to execute on minimum turnaround machine.
   b) Now, if parent task execution is done, then all child task belonging to same parent task and at same level are executed in parallel on same or different VM.
   c) Data transfer time from parent task to its respective child task is calculated as follows:
      Transfer time = Cloudlet.outputSize / Vmlist.get(id).getBw()
      Here parent task's output size and VM's bandwidth is use for calculation of transfer time. Total execution time for each cloudlet is calculated as follows:
      Totalexecution time = (cloudlet.getCloudletLength ( ) / processing power) + transfer time
4. Cost of transfer time will be zero if child task is executed on same VM as parent task. Otherwise, cost per bandwidth is calculated.
   Transfer cost =costPerBw * getCloudletOutputSize( )
5. Finally, checks if all task levels are executed, then stop, otherwise go to step 3.

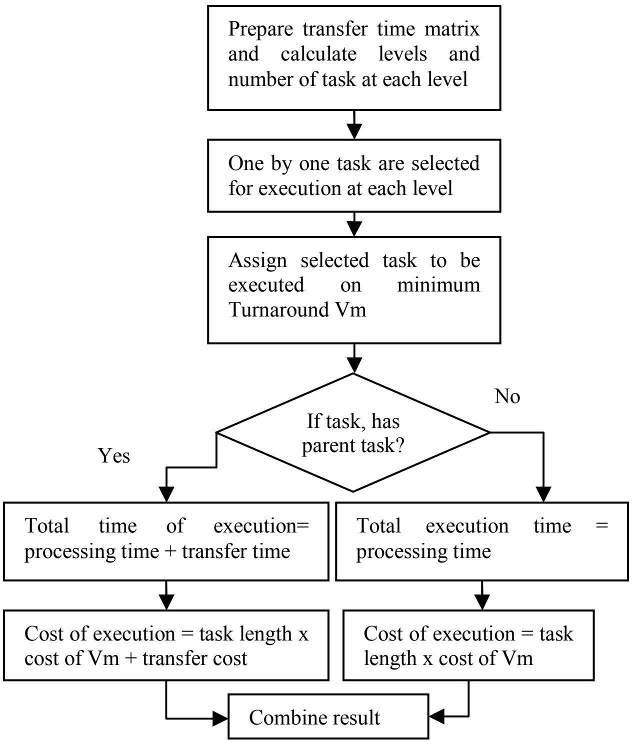

Fig. 2 Flowchart of EWSA

The proposed algorithm is clearly and concisely explained through the flowchart given in figure 2. Simultaneously multiple workflows are submitted at cloud datacenter, but which workflow is to be executed first is decided on the basis of deadline for its completion. The workflow with shortest deadline is selected and its transfer time matrix is calculated. Also, the tasks are divided into levels based upon their dependency on another task. The independent tasks are placed at level 1, the task which takes input from level 1 tasks only are placed at level 2 and the tasks dependent on either level 2 tasks or both level 1 and level 2 tasks are placed at level 3 and so on. Flowchart in Fig. 2 is showing steps for scheduling each workflow.

IV. Experimental Setup And Results

A. Experimental Data

The experiment is conducted using the CloudSim simulator. The experiment is performed for the workflows which have been scheduled on six different virtual machines having configuration as given below in TABLE I.

TABLE I. Configuration of VMs

Vms	**Vm1**	**Vm2**	**Vm3**	**Vm4**	**Vm5**	**Vm6**
RAM	5024	5024	5024	5024	5024	5024
Processing Power (MIPS)	22000	19800	15200	14400	11000	10098
Processing Element (CPU)	1	2	2	4	2	4
Bandwidth	1000	900	850	725	500	500

The transfer time between tasks calculated using above formula for the given workflow is given below TABLE II.

TABLE II. Transfer time between tasks

	T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	**T10**	**T11**
T1	0	0.2	0.5	0	0	0	0	0	0	0	0
T2	0	0	0	0.8	0.7	0.1	0	0	0	0	0
T3	0	0	0	0	0	0	0.9	0.4	0	0	0
T4	0	0	0	0	0	0	0	0	0.6	0	0
T5	0	0	0	0	0	0	0	0	0.4	0	0
T6	0	0	0	0	0	0	0	0	0	0	0.8
T7	0	0	0	0	0	0	0	0	0	0.2	0
T8	0	0	0	0	0	0	0	0	0	0.4	0
T9	0	0	0	0	0	0	0	0	0	0	0.6
T10	0	0	0	0	0	0	0	0	0	0	0.8
T11	0	0	0	0	0	0	0	0	0	0	0

B. Results Obtained

The EWSA algorithm is compared with Cat Swarm Optimization (CSO) technique [12]. The results obtained by running the simulation clearly shows that EWSA algorithm provides a more optimized solution in terms of scheduling workflow tasks in a more realistic way. The experiment is performed for different set of workflows as mentioned in table 3 and table 4. The tabular as well as graphical comparisons are shown among both the algorithms in terms of Execution time and Cost of execution.

TABLE III. Execution time Comparison

Number of task in workflow	CSO algorithm	EWSA algorithm
20	98.7135	76.5634
30	283.8921	214.5672
40	729.4284	576.3049
50	799.6712	652.8310
60	819.3914	734.2735

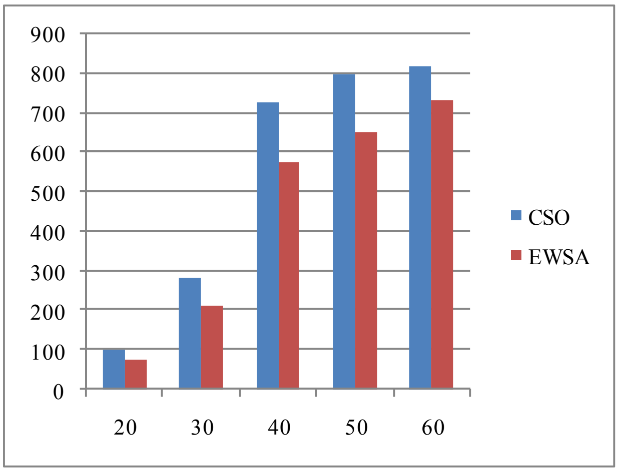

Fig. 3 Execution time Comparison

TABLE IV. Cost of Execution Comparison

Number of task in workflow	CSO Algorithm	EWSA Algorithm
20	77.842	75.325
30	129.316	110.566
40	145.001	139.458
50	211.395	188.739
60	385.568	298.526

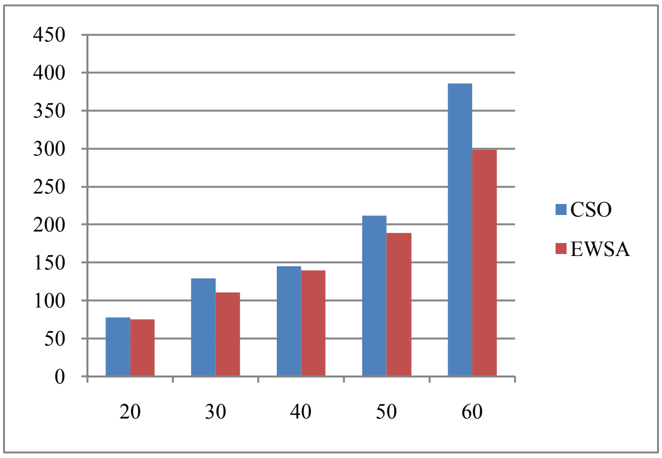

Fig. 4 Cost of Execution Comparison

We experimented by taking different number of cloudlets (different amount of data) processed by the workflow. For each number of task, we have executed the algorithm 20 times and mean value of that 20 execution was taken as final value for each respective number of task. Fig. 3 and Fig. 4 shows distribution of workload for different number of tasks with respect to execution time and execution cost respectively, which is different each time depending upon the file size, output size of file, bandwidth of Vm on which it is executed. But the results are improved each time as compared to CSO algorithm.

V. Conclusion And Future Work

The paper discusses the workflow scheduling technique which takes into consideration important QoS factors time of execution, cost of execution and the transfer time between the dependent tasks. The algorithm provides more optimized results as compared to other techniques like CSO. Thus, the paper concludes with the fact that our proposed algorithm performs better over CSO and other existing techniques. This is been clearly proven by the simulation results obtained, as each time the results are improved. In future, the work can be extended taking into consideration more factors like energy consumption which gives it a more realistic approach.

References

[1] J. Geelan, "Twenty-one experts define cloud computing," *Cloud Computing Journal,* vol. 2009, pp. 1-5, 2009.

[2] R. Mikkilineni and V. Sarathy. (2009). *Cloud Computing and the Lessons from the Past* in 18th IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises, WETICE'09,2009.

[3] S. Ma, "A Review on Cloud Computing Development," *Journal of Networks,* vol. 7, no.2, pp. 305-310, 2012.

[4] P. Mell and T. Grance, "The NIST definition of cloud computing," *National Institute of Standards and Technology,* vol. 53, no.6, 2009.

[5] I. Foster, Z. Yong, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," 2008, pp. 1-10.

[6] *CloudComputing*[Online]: http://en.wikipedia.org/wiki/Cloud_computing

[7] Meng Xu, Lizhen Cui, Haiyang Wang, Yanbing Bi, " A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing". 2009 IEEE International Symposium on Parallel and Distributed Processing with Applications.

[8] Ke Liu1,2, Hai Jin1, Jinjun Chen2, Xiao Liu2, Dong Yuan2 and Yun Yang 2, "A Compromised Time-Cost Scheduling Algorithm in SwineDe W-C for Instance-Intensive Cost-Constrained Workflows on a Cloud Computing Platform". International Journal of High Performance Computing Applications OnlineFirst, published on May 24, 2010 as doi:10.1177/1094342010369114

[9] Suraj Pandey1, LinlinWu1, Siddeswara Mayura Guru2, Rajkumar Buyya1, "A Particle Swarm Optimization based Heuristicfor Scheduling Workflow Applications in Cloud Computing Environment". 2010 24th IEEE International Conference on Advanced Information Networking and Applications.

[10] Cui Lin and Shiyong Lu, "Scheduling Scientific Workflows Elastically for Cloud Computing". 2011 IEEE 4th International Conference on Cloud Computing.

[11] Kahina∗∗Bessai ∗, Samir Youcef∗∗, Ammar Oulamara∗∗, Claude Godart and Selmin Nurcan, "Bi-crieteria Workflow task allocation and scheduling in Cloud Computing Environment". 2012 IEEE Fifth International Conference on Cloud Computing.

[12] Saurabh Bilgaiyan, Santwana Sagnika, Madhabananda Da, " Workflow Scheduling in Cloud Computing Environment Using Cat Swarm

Optimization” *2014 IEEE International Advance Computing Conference (IACC)*.

[13] Li Liu, Miao Zhang, Yuqing Lin and Liangjuan Qin, “A Survey on Workflow Management and Scheduling in Cloud Computing”. 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing.

[14] Elaine N. Watanabe and Pedro P. V. Campos and Kelly R. Braghetto and Daniel M. Batista, “Energy saving algorithms for workflow scheduling in cloud computing”. 2014 Brazilian Symposium on Computer Networks and Distributed Systems.

[15] R. Mikkilineni and V. Sarathy. (2009). Cloud Computing and the Lessons from the Past in 18th IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises, WETICE'09,2009.

[16] S. Ma, "A Review on Cloud Computing Development," Journal of Networks, vol. 7, no.2, pp. 305-310, 2012

[17] A.Fox and R. Griffith, "Above the clouds: A Berkeley view of cloud computing," Dept. Electrical Engineering and Computer. Sciences, University of California, Berkeley.

[18] S. Singh and K. Kant, "Greedy grid scheduling algorithm in dynamic job submission environment," in Emerging Trends in Electrical and Computer Technology (ICETECT), 2011 International Conference on, 2011, pp. 933-936.