

WORKFLOW SCHEDULING IN CLOUDS USING RANDOMIZED SCHEDULING ALGORITHM

G. Mohanapriya, Saikishor Jangiti and V.S. Shankar Sriram*

Center for Information Super Highway (CISH), School of Computing,

SASTRA Deemed University, India

ABSTRACT:

The provisioning of on-demand resources makes it optimal for executing scientific application workflows in cloud computing. An application starts the process with a small number of resources, and it allocates the resources when required. However, workflow scheduling belongs to NP-hard class of problems, so optimization techniques are preferred for the solution. This paper explores the effect of a Randomized scheduling algorithm in workflow scheduling for the scheduling problem. The use of Randomized scheduling algorithm in comparison with other scheduling algorithms increases the efficiency of workflow scheduling in various scientific workflows and simulators. The experimental result confirms that the Randomized scheduling algorithm well performed than other scheduling approaches and provides better scheduling with reduced makespan.

KeyWords: Cloud computing, Scheduling, Scientific workflows, Randomized scheduling algorithm.

1. INTRODUCTION

Cloud computing is a platform for various kind of applications with different Quality of Service (QoS). Cloud services can be classified as Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a service (IaaS). SaaS is a browser-based interface that allows users to alter the documents online (e.g. Web-based email, Dropbox, or Netflix). PaaS provides different components to develop applications in the cloud (e.g. Google app engine). IaaS provides the resources such as virtualization components, networks, and servers (e.g. Windows Azure, Amazon web services). These services make execution of scientific applications easier in cloud computing [6]. Usually, Scientific Workflow Management System (SWfMS) software is used to handle scientific applications and also reduce complexity to execute data-intensive applications. These workflow management systems are obtained from some Grid projects (Pegasus, ASKALON, and GrADS) [8].

Workflow scheduling is a very important task for representing complex applications in cloud computing. Workflows are a sequence of tasks related to data and control flow dependencies, and it can be represented by Directed Acyclic Graph (DAG). These workflows are applied in different scientific areas such as biology and astronomy [10]. However, size of scientific workflows is exponentially increasing leads to difficulty in managing big data, more effective and scalable infrastructures are required to execute complex workflows within a time.

Workflow scheduling is the procedure of mapping tasks on appropriate resources to achieve performance criteria like QoS constraints. Initially, in a grid environment, some scheduling techniques attempt to reduce the execution time without measuring the cost of resources. But, in a cloud environment, different capabilities are provided by the service provider at varied cost. Thus, same workflows using distinct resources result in different cost and execution time [5]. Since scheduling problem is NP-hard problem and it can be very difficult to obtain an optimal schedule because huge communication and computation cost is required for scheduling of workflows.

In distributed systems, Workflows create a common model for a wide area of scientific applications. The main issue of Distributed and parallel systems environment is the efficient utilization of resources. For this reason, main parameters considered in cloud computing is execution time and cost. Commonly, faster resources are costlier than slower one [3]. Therefore, Scheduling algorithms and provisioning of resources are necessary for minimizing the makespan, cost and also improves utilization of resources. Different techniques like Ant colony optimization [13], Shuffled frog leaping algorithm [14], Particle Swarm optimization [5] and Genetic algorithm [3] is used to solve the problem of workflow scheduling. These techniques try to reduce the makespan (execution time) and cost of workflows, but still more work needs for efficient scheduling. This paper presents a randomized scheduling algorithm for the optimal scheduling problem in workflows. Randomized scheduling algorithm selects random VM for executing tasks in the workflow. Further, schedule generated from this algorithm ensures to meet a specified deadline.

The remaining article is organized as follow: Section 2 presents literature survey on scheduling techniques in workflows. Section 3 describes proposed Randomized scheduling algorithm. The experimental results are discussed in Section 4. Finally; Section 7 concludes the article.

2. RELATED WORK

According to Parmeet Kaur and Shikha Mehta, workflow scheduling is a major problem on distributed systems, So Optimization techniques are used to improve the system performance based on execution time and cost of workflows [14]. Hence, Different optimization techniques are applied in cloud computing to solve the problems of scheduling in workflow applications. In

the recent years, multi-objective optimization techniques are designed for a better optimal solution (i.e. cost, transfer rate, execution time, security, energy consumption) in workflow scheduling. In general, optimization techniques are widely applied in various research domains such as medical applications [16], engineering and feature selection[18].Further, several researchers focused on applying scalarization method for minimizing optimization problem. Table 1 shows some recent optimization based workflow scheduling and explains challenges of scheduling techniques.

Table .1 Related work

Authors	Technique proposed	Datasets	Validation
Amandeep Verma et al.[5]	Hybrid Particle Swarm	• Montage • Epigenomics • CyberShake • LIGO • SIPHT	Execution Cost and Execution time
Henrique et al.[3]	PSO+GA	• CyberShake • SIPHT • Epigenomics	Cost, Security, and Deadlines
Luan Taylor et al[12]	Hybrid Evolutionary Algorithm	• CyberShake • Epigenomics • Montage • Inspiral • SciPhy	Execution time
Yongsheng Ding et al.[11]	Fault tolerant elastic scheduling	• CyberShake • Montage • Epigenomics • Inspiral	Fault tolerance
Saima Gulzar Ahmad et al.[2]	Hybrid Genetic Algorithm	• Montage • CyberShake • Gaussian Elimination • Random	Makespan
Xin Ye et al.[15]	Knee point was driven algorithm	• Business workflow • Scientific workflow	Reliability, Cost and Execution time

Israel casas et al. [17]	Genetic Algorithm with efficient tune –in of resources	• Montage • CyberShake • LIGO • Epigenomics • SIPHT	Makespan and Monetary cost
Raza Abbas Haidri et al.[4]	Cost effective deadline aware scheduling	• Montage • CyberShake • SIPHT • LIGO	Execution time and cost
Vahid Arabnejad et al[1]	Proportional Deadline Constrained (PDC)+Deadline Constrained Critical Path	• Montage • CyberShake • SIPHT • LIGO • Epigenomics	Cost, Throughput

3. PROPOSED METHODOLOGY

The current solutions for resource utilization and optimal scheduling still face some efficiency problems, and there is a demand to examine better solutions[14]. Based on this objective, the paper presents randomized scheduling algorithm for the problem of resource provisioning and optimal scheduling. Further, it results in a cost-efficient and deadline constrained workflows.

3.1.RANDOMIZED SCHEDULING ALGORITHM

In Randomized scheduling algorithm, tasks in the workflows are represented as meta-task and resources are represented as R_j . First, randomly select the VM from the idle VM list and allocate the tasks for execution. The tasks are allocated to the VM based on arrival time in queue, which job arrived first in the queue is first executed. Evaluate the completion time of every task and update the execution time. (fig 1) shows the randomized scheduling algorithm of workflows.

Schedule 1 and Schedule 2 are scheduling of tasks in the workflow using randomized scheduling. Perform the Maximum iterations (I_{max}) until the makespan of schedule 2 is better than the schedule 1. If the schedule 2 is better, update the schedule 1 is schedule 2. Every iteration Minimum Execution time is updated in the schedule. Finally, update the makespan (overall execution time) of workflows. Randomized scheduling algorithms are more efficient than other scheduling algorithms.

```

Function Randomized Scheduling(meta-task)
{
While meta-task is not empty
  Choose a random VM ( $R_j$ ) from the idle VMs.
  Assign the next independent task  $T_m$  to the resource  $R_j$ 
  Remove the task  $T_m$  from meta-tasks set
  Update Makespan and schedule
End while
Return Schedule
}
Schedule1 = call Randomized Scheduling(meta-task)
Initialize maximum number of iterations ( $I_{max}$ )
While  $I < I_{max}$  do
Schedule2 = call Randomized Scheduling(meta-task)
If(Makespan(Schedule2) < Makespan(Schedule1))
{
Schedule1 = Schedule2;
}
End while
Print Schedule1 and Makespan

```

Fig .1 Randomized scheduling algorithm

4.EXPERIMENTAL RESULTS

In this section presents the performance of Randomized scheduling algorithm in workflow scheduling. The Randomized scheduling algorithm is validated using workflow sim, and the results are compared with existing techniques. For all the experiments, different resources were chosen ($vm=5$), and therefore, cost and makespan are varied from others. For every workflow execution, the average execution time of minimum and maximum possible time is considered as a deadline constraint

4.1. MONTAGE AND CYBERSHAKE WORKFLOWS

Montage is used to create a mosaic of the sky and is an astronomical image developed by NASA. The input images are mixed to form the output mosaic. The input image validates the geometry of output mosaic, and it can be denoted as a workflow. Montage consists of tasks with small and largest executions times. CyberShake is used to find the earthquake hazards, and it can be used by SCEC (Southern California Earthquake Center). This workflow is relatively simple, but it can

be used to manage large datasets. CyberShake is both data and computes intensive workflow. Therefore, different characteristics of these workflows make them highly sufficient for the validation.

We have executed simulations using Montage and CyberShake workflows to validate the performance of Randomized scheduling algorithm. The Randomized scheduling algorithm well performed than another scheduling algorithm. Similar experiments are conducted using Montage (25,50 and 100) nodes and CyberShake (50,100) nodes. These experiments produce results with better performance, compared to the previous Scheduling techniques. Fig(2) shows the experimental results of four scheduling algorithm in terms of makespan. Makespan Of Proposed Randomized Scheduling is better than other three algorithms. Table 2 provides the makespan of scheduling algorithms; Randomized scheduling algorithm produce good results than other algorithms

Table.2 Comparison of Makespan using Montage_25,50,100 and Cybershake_50,100

Algorithm	Montage 25	Montage 50	Montage 100	CyberShake 50	CyberShake 100
MCT SA*	63	908	4363.05	666.97	7944.12
RR SA*	1221.06	702.19	2259.03	6705.22	7547.39
Minmin SA*	75.43	817.68	684	3186.22	1734.67
Maxmin SA*	63.54	1469.96	605.18	504.28	5328.61
Data aware SA*	415.19	1154.54	2814.17	2436.47	5746.95
FCFS SA*	521.1	833.37	1348	6685.84	9634
Randomized SA*	58.69	180.98	330.78	452.4	1036.79

Scheduling Algorithm –SA *

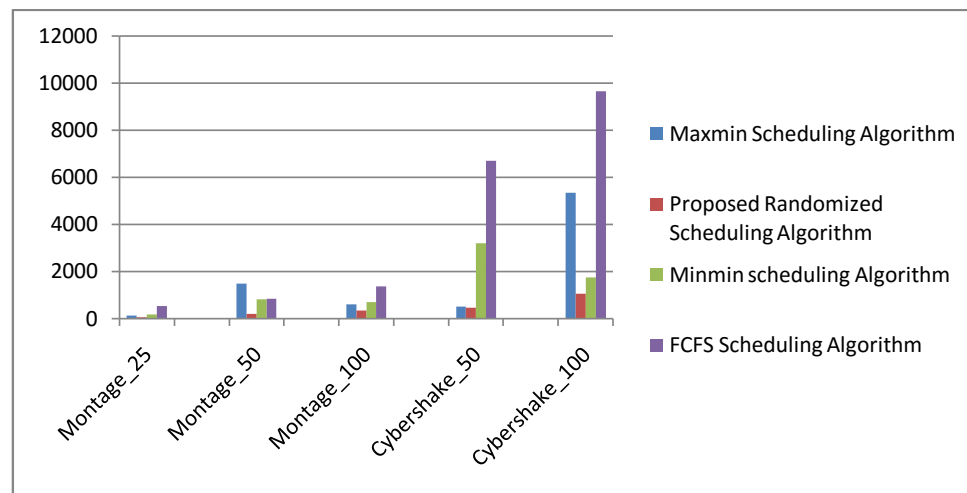


Fig. 2 Makespan of four Scheduling algorithms

5. CONCLUSION

In this article, the Randomized scheduling algorithm is applied in workflow scheduling for resource provisioning problem. Hence, the objective of this algorithm is to improve the system performance, and these studies are conducted using scientific workflows. The validity and efficiency of Randomized scheduling algorithm over existing techniques were exposed using simulation environment in terms of Makespan. Extension of the proposed algorithm in the directions like trust management, reliability, security and minimizing execution cost can be taken as future work.

REFERENCES

1. Vahid Arabnejad, Kris Bubendorfer, Bryan Ng: Scheduling deadline constrained scientific workflows on dynamically provisioned cloud resources. *Futur.Gener.Comput.Syst.* 75, 348-364 (2017)
2. Saima Gulzar Ahmad, Chee Sun Liewa, Ehsan Ullah Munirb, Ang Tan Fonga, Samee U. Khanc: A hybrid genetic algorithm for optimization of scheduling workflow applications in a heterogeneous computing system. *Journ.Paral.Distr.Comput.* 87, 80- 90(2016)
3. Henrique Yoshikazu Shishido, Júlio Cezar Estrella, Claudio Fabiano Motta Toledo, Marcio Silva Arantes: Genetic-based algorithms applied to a workflow scheduling algorithm with security and deadline constraints in clouds. *Comput.Electri.Engi.* 1- 17(2017)

4. Raza Abbas Haidri, Chittaranjan Padmanabh Katti, Prem Chandra Saxena: Cost effective deadline aware scheduling strategy for workflow applications on virtual machines in cloud computing. *Journ.Kin.Saud.univ.Comput.Inform.Sci.*
5. Amandeep Verma, Sakshi Kaushal: A hybrid multi-objective Particle Swarm Optimization for scientific workflow scheduling. *Paral.Comput.*62, 1-19(2017)
6. Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopala, James Broberg, Ivona Brandic: Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Futur.Gener.Comput.Syst.*25, 59-616(2009)
7. Zhongjin Li, Jidong Ge, Hongji Yang, Liguang Huang, Haiyang Hu, Hao Hu, Bin Luo: A security and cost aware scheduling algorithm for heterogeneous tasks of scientific workflow in clouds. *Futur.Gener.Comput.Syst.*65, 140-152(2016)
8. Mohammad Masdari, Sima ValiKardan, Zahra Shahi, and Sonay Imani Azar: Towards workflow scheduling in cloud computing: A comprehensive analysis. *Journ.Netw.Comput.App.*66, 64-82(2016).
9. Ehab Nabil Alkhanaka, Sai Peck Leea, Reza Rezaeia, Reza Meimandi Parizi: Cost optimization approaches for scientific workflow scheduling in the cloud and grid computing: A review, classifications, and open issues. *Journ.Syst.Soft.*113,1-26(2016)
10. Fuhui Wu¹, Qingbo Wu¹, Yusong Tan: Workflow scheduling in the cloud: a survey. *Journ.Supe.Comput.*71, 3373–3418(2015)
11. Yongsheng Ding, Guangshun Yao, Kuangrong Hao: Fault-tolerant elastic scheduling algorithm for workflow in Cloud systems. *Inform.Sci.*393, 47-65(2017)
12. Luan Teylo, Ubiratam de Paula, Yuri Frota, Daniel de Oliveira, Lucia M.A. Drummond: A hybrid evolutionary algorithm for task scheduling and data assignment of data-intensive scientific workflows on clouds. *Futur.Gener.Comput.Syst.*76, 1- 17(2017)
13. Saad Ghaleb Yaseen, Nada.M.A.AL-Slami: Ant Colony Optimization. *Intern.Journ.Comput.Sci.Netw.Secur.*8 (2008)
14. Parmeet Kaur, Shikha Mehta: Resource provisioning and workflow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm. *Journ.Paral.Distr.Comput.*101, 41-50(2017)

15. Xin Ye, Sihao Liu, Yanli Yin, Yaochu Jin: User-oriented many-objective cloud workflow scheduling based on an improved knee point driven evolutionary algorithm. Know. Base. Syst. 135, 113-124(2017).
16. Mingjing Wang, Huiling Chen, Bo Yang, Xuehua Zhao, Changfei Tong: Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses
17. Israel Casas, Javed Taheri, Rajiv Ranjan, Albert Zomaya GA-ETI: An enhanced genetic algorithm for the scheduling of scientific workflows in cloud environments.
18. Ahmed Ibrahim Hafez, Hossam M. Zawbaa, E. Emary, Aboul Ella Hassanien: Sine Cosine Optimization Algorithm for Feature Selection

