ISSN XXXX XXXX © 2019 IJESC

IJESC

Research Article Volume 9 Issue No.3

Cost-Effective Workflow Scheduling Algorithm for Cloud

Lakshay Kukreja[1], Chitvan Jindal[2], Anmol Jyoti[3], Dr. Abhishek Swaroop[4]
B.Tech Student[1, 2, 3], HOD & Professor[4]
Department of Information Technology
Bhagwan Parshuram Institute of Technology, Delhi (GGSIPU), India

Abstract:
Cloud computing is the future to support the upcoming and promising inventions and various applications that are establishing revolution among all the industries that involve technology. As, cloud computing offers vast facilities to store and compute data over the internet, its network is predicted be a major hardware and software resource provider to certain developing areas like Artificial Intelligence, Quantum Computing, etc. Due to this, workflow scheduling in cloud computing is a major concern to the developers as well as consumers. Therefore, we analyzed the various existing workflow algorithms that are used to schedule tasks for various existing scientific workflows like Montage, LIGO, and Epigenomics. Most of the simulation of these workflows was done using CloudSim on NetBeans using Java. The existing workflow scheduling algorithms were simulated and compared to the proposed algorithm that considers VM performance variation and acquisition delay, other than time and cost.

Keywords: Cloud Computing, Workflow Scheduling, CloudSim, Cost-Effective, Acquisition Delay, VM (Virtual Machine), VM Performance Variation.

I. INTRODUCTION

Cloud Computing is a large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet [1]. Cloud Computing is the use of computing resources (hardware and software) that are delivered as a service over a network [2]. It has been developed over the past 10 years as a new computing paradigm where applications, data and IT services are provided over the Internet. The on-demand, self-service, pay-by-use nature of cloud computing is also an extension of established trends. From an enterprise perspective, the on-demand nature of cloud computing helps to support, the performance and capacity aspects of service-level objectives. The self-service nature of Cloud Computing allows organizations to create elastic environments that expand, and contract based on the workload and target performance parameters. And the pay-by-use nature of Cloud Computing may take the form of equipment leases that guarantee a minimum level of service from a Cloud provider [3]. Workflow is defined as the automation of a business process, in whole or in part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules [4]. When we talk about cloud, the major advantage of cloud is its application scalability or elasticity. This elastic nature of cloud facilitates changes of resource and characteristics at run time. This ability of cloud enables the workflow management systems to readily meet the quality of service requirements of application [5]. Workflow scheduling is the problem of mapping each task to appropriate resource and allowing the tasks to satisfy some performance criterion. A workflow consists of a sequence of concatenated steps. Workflow mainly focuses with the automation of procedures and in order to achieve an overall goal thereby files and data are passed between participants according to a defined set of rules. A workflow enables the structuring of applications in a directed acyclic graph form where each node represents the task and edges represent the dependencies between the nodes of the applications. A single workflow consists of a set of tasks and each task communicate with another task in the workflow. Workflows are supported by Workflow Management Systems. Workflow scheduling discovers resources and allocates tasks on suitable resources. Workflow scheduling plays a vital role in the workflow management. Proper scheduling of workflow can have an efficient impact on the performance of the system. For proper scheduling in workflows various scheduling algorithms are used.

II. RELATED WORK

Various algorithms were studied and analyzed on different platforms like Amazon EC2, CloudSim, etc. to find out the key characteristics as well as the pros and cons. The following are the algorithms that were studied and are talked about in brief:

1. A Particle Swarm Optimization based Heuristic for Scheduling Workflow Applications: Pandey et al. [6] presented a particle swarm optimization (PSO) based algorithm to schedule the applications to cloud resources that takes both computation and data transmission cost. This heuristic involves variation in the computation and communication cost. After evaluation, PSO is proven to be successful in distributing the workload onto the resources and minimize the cost as well.

2. For the execution of a single workflow instance in the cloud environment, the heuristics-based workflow scheduling algorithms are described in [7] – [8]. The work in [7] – [9] are based on the partial critical path (PCPs) of the workflow. IaaS Cloud Partial Critical Path (IC-PCP) was proposed in [7] which is a static algorithm that initially considers and estimates the latest finish time for each task and then the determination of each partial critical path (PCP) takes place one by one, with consideration of the exit node of the workflow. Traversing the partial critical path, each task is provided the cheapest VM instance, which can execute before its finish time. Failing to

find any active VMs for allocation, that meets the finish time constraint of the tasks, a new instance of the cheapest VM from the cloud is selected, which will complete all tasks before its finish time. Finally, PCP is assigned to it.

3. In [9] robust and fault tolerant algorithms are presented for the execution of a single workflow. However, in various research works, [7] – [9] focused on all the characteristics of the cloud. While, paper [7] does not consider the issues (VMs performance variation, and acquisition delay) of the cloud. Major issues of cloud were solved by the work that was proposed in [9]. The performance of their proposal is estimated based on the propriety order of these policies that are a) Robust Cost-Time (RCT) and b) Robust Time-Cost (RTC) that allow users to define their own function. These scheduling algorithms cater to the performance variation and the acquisition delay issue of the VM, hence, our proposed work was compared. As our baseline algorithms for the comparison, we selected RTC and RCT resource selection policies.

III. PROBLEM DEFINITION

In the target cloud environment, a workflow system needs to be established which gets its input from the user and maintains each task efficiently.

The number of resources or slots are to be specified by the user for provisioning. Selection of the important number of resources, that are to be provisioned, is important. Overall, the execution time and the monetary cost of the workflow depend on the number of resources used. For this selection procedure, estimation of the execution time using a set of resources is important. Each task has a different runtime and resource needs running on various cloud environments, but it is possible to predict the runtimes under a specific system setup.

This estimation model can also be used to determine the number of resources that are required to achieve a certain level of performance and finally the cost can also be calculated with the use of the resources. Further, it is necessary for each VM that is provisioned to transfer all the output data files obtained by executing tasks scheduled on it, to the local storage of the VMs on which the corresponding children tasks are scheduled, before these VMs to de-provision. So, there is no need to active the VMs, where the child tasks are executed during the output data file transfer. A PV parameter was used which kept record of the variation generated in the performance of the virtual machines. The assumed average time for allocation of a VM is around 60 seconds (acquisition delay is 60 seconds). Based on the above-defined terms the problem of workflow scheduling is defined as follows:

The problem is to find a feasible schedule (S) for a given workflow (W) such that total execution cost (TEC) of the schedule is minimized, considering the total execution time (TET) and deadline of each task in the workflow.

Based on the parameters considered, Minimized TEC was defined as:

Min. TEC = $\sum Cost_{type}\ vm_k \times [ET_k - ST_k] \times 1/$ [avg. time interval]

$C_{type(vmk)}$ is the cost of VM type vmk per time interval
ET_k is the lease end time of VM of type VMk
ST_k is the lease start time of VM of type VMk

IV. PROPOSED ALGORITHM

The workflow application is defined as Workflow, W = (T, E) where T is defined as the set of tasks and E is defined as the set of edges. We start by generating the set of tasks and ranking them based on their priority and deadline. Each task in the set has its predecessor and successor list of tasks that may or may not have an impact on the current task to be executed.
Start Time ST VM_k and End Time ET VM_k:

The start time of a VM of type VM_k is time at which the VM_k is ready to execute tasks and the end time of VM type VM_k is the time at which VM_k is de-provisioned from the resource pool (set of active resources). Minimum Execution Time of Task MET (t_i): It is the execution time of task ti on the VM of type $VM_k \varepsilon VM_{set}$ that has minimum execution time among all types of VMs available in cloud and its minimum execution is defined as follow:

$$MET\ (t_i) = \min_{VMk} \{ET\ (t_i \text{ to } VM_k)\}$$

Each task of the available set is allocated a unique index which helps in keeping track of all the information for the task itself. The next step includes arranging the tasks in an ordered list manner, based on their deadline constraints. These tasks will also be arranged according to their need of resources, execution time and cost.

Then, each task will be allocated with a virtual machine according to the type that is suitable for the task. The Total Execution Time (TET) and Total Execution Cost (TEC) of the workflow is estimated by scheduling each task of the workflow to the cheapest virtual machine. For the scheduling of each task, t_i, on VM of type VM_k, Start and Finish time of the task is recorded for the calculation of the minimum total execution time and thus, the calculation of minimum total execution cost.

Tournament-based selection approach [10] was used for the selection of the entity that includes the task, VM, time and cost. The entity that satisfies the chosen deadline constraints and is recorded to have the minimum Total Execution Cost is selected. The least possible deadline constraint (μ) was selected for the scheduling of the workflow in the cloud system.

$$\mu \leq 1.1 \text{ (Hard Deadline)}$$

$$\text{Deadline, } D = MET_W * (\mu)$$

Minimum Execution Time of Workflow,

$$METW = \sum FT(t_{i \text{ to } VMk})$$

We use the constraints handling strategy proposed in literature [11]. If two schedules are feasible, then we select the schedule having the minimum execution cost. However, if one schedule is feasible, and second schedule is not feasible, then we ignore the second schedule and choose the first schedule. If both the schedules are not feasible, then we select the schedule that is close to the deadline.

V. SIMULATION ENVIRONMENT

Three real scientific workflows were used in the experiments, namely Montage [12], Epigenome [13] and LIGO [14]. The basic structure of the three workflows is shown in Figure 3.1.

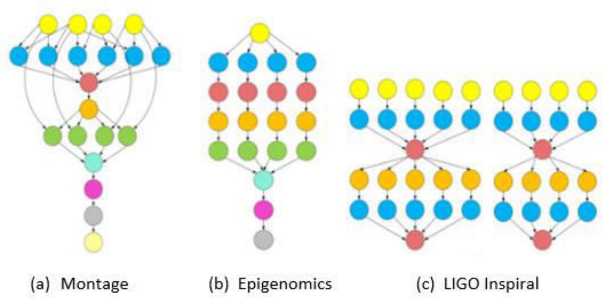

Figure .3.1. Work flow structures :(a) Montage, (b) Epigenomics, and (c) LIGO.

Montage is a scientific workflow that includes generation of image mosaics of the sky is described as I/O intensive [15], [16]. The jobs that run here have relatively low runtime requirements, comparatively low CPU utilization and their execution time is mainly spent on I/O operations. The number of clusters and the workflow generated decide the number of jobs present at each level. A Montage workflow with 10429 tasks was generated and used in the experiments. The tasks were clustered to create a total of 31, 55, 103-104, 152, 248, 440 and 823 jobs when using 4, 8, 16, 32, 64, 128 and 256 clusters per level, respectively. The total number of jobs were divided into 13 levels considering the dependency of data between them. The Epigenome workflow is generally categorized as CPU-bound application which involves mapping of the epigenetic state of human cells [15], [16] with several parallel jobs that are being operated in independent data files, as shown in Figure 3.1. (b). The jobs that split or convert the input files into multiple output files tend to have low CPU utilization. Most of the jobs are found to be computationally intensive. These jobs result into spending most of the runtime in the CPU and very small amount of time is allocated to other operations. A workflow of 529 tasks was used in the experiments, with the tasks clustered to create a total of 50-51, 83, 147, 275 and 529-531 jobs using 8, 16, 32, 64 and 128 clusters per level, respectively. The final workflow build consists of 11 job levels. The ''Laser Interferometer Gravitational Wave Observatory (LIGO)'' scientific workflow [14] is used for the identification of the gravitational waves that are produced by vast range of events that occur in the universe. The tasks involved generally require high computing capacity with large memory. For the allocation of VMs, we selected four types of VMs with the following characteristics:

Table .5.1. Various types of VMs

Type	Memory (GB)	Core Speed	Cores	Cost per hour ($)
l	7	4	2	0.16
m	14	8	4	0.32
n	7	20	8	0.8
o	1.7	1	1	0.04

The simulation was performed on CloudSim using Java with the help of Netbeans IDE. CloudSim is a library for the simulation of cloud scenarios. It provides essential classes for describing data centre, computational resources, virtual machines, applications, users, and policies for the management of various parts of the system such as scheduling and provisioning. Using these components, it is easy to evaluate new strategies governing the use of clouds, while considering policies, scheduling algorithms, load balancing policies, etc. It can also be used to assess the competence of strategies from various perspectives such as cost, application execution time, etc. It also supports the evaluation of Green IT policies. It can be used as a building block for a simulated cloud environment and can add new policies for scheduling, load balancing and new scenarios. It is flexible enough to be used as a library that allows you to add a desired scenario by writing a Java program.

VI. RESULT AND ANALYSIS

Following are the graphs for analysis of Cost in relation with the hard deadline constraint. The graph shows the performed analysis on the various baseline algorithms with the existing scientific workflows and our predicted values for the same analysis for the proposed algorithm:

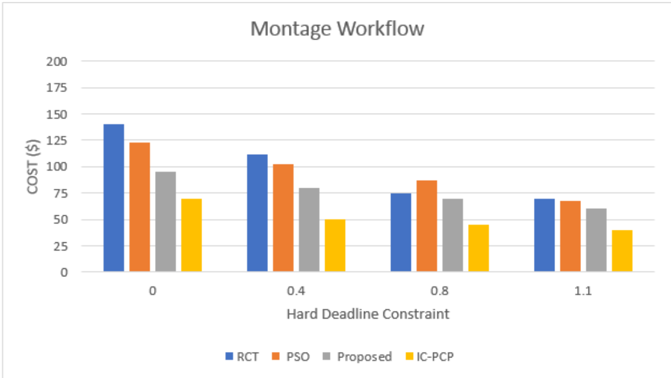

Figure. 6.1. Montage Workflow Analysis

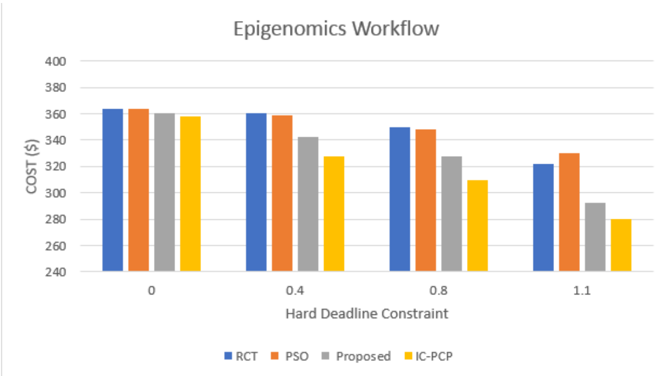

Figure.6.2. Epigenomics Workflow Analysis

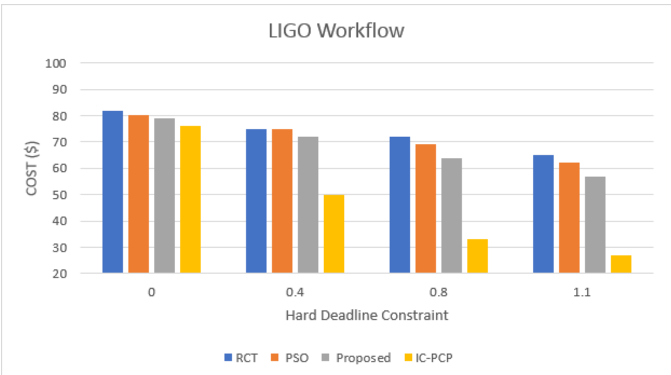

Figure .6.3. LIGO Workflow Analysis

The proposed algorithm for Montage workflow under deadline constraints give average 30% lower execution cost than RCT and average 21% lower execution cost than PSO. Also, the proposed algorithm for LIGO workflow under deadline constraints give average 14% lower execution cost than RCT and average 9% lower execution cost than PSO. The proposed algorithm for Epigenomics workflow under deadline constraints gives average 11% lower execution cost than RCT and average 9% lower execution cost than PSO.

VII. CONCLUSION AND FUTURE WORK

Workflow scheduling is one of the major issues in cloud computing environment. Different successful and existing workflow scheduling algorithms, in cloud computing, were surveyed and comparison was done based on various characteristics. Some of the existing workflow algorithms do not consider the execution time. Therefore, there was a need to implement a new scheduling algorithm that can minimize the execution time in cloud environment. To increase the usage of various cloud services for workflow execution Moving workflows to a cloud computing environment enables the use of various cloud services to facilitate workflow execution. Cloud computing enables efficient provisioning and scheduling of required resources in such a way that the deadline constraint is met, and execution cost is minimized. An algorithm is proposed based on existing workflow algorithms that resulted into improving the scenario of the workflow scheduling parameters in the cloud environment. This algorithm considers all the characteristics of the cloud such as heterogeneity, on-demand resource provisioning and pay-as-you-go price model. Also, issues and parameters such as VM performance, time required for allocation of VM and de-allocation time. Methods were created to build each step of the algorithm and then implemented using CloudSim via NetBeans. The simulation experiments conducted on three scientific workflows show that in comparison to state-of-art algorithms, such as PSO, RCT, RTC and IC-PCP. The proposed algorithm exhibits the highest hit rate for deadline constraint. Although, the algorithm will be having a lower execution time than IC-PCP and RCT, and lower execution cost than RTC and PSO with considering the deadline constraint.

Acknowledgment

We would like to show our gratitude to our institution, Bhagwan Parshuram Institute of Technology, Delhi, for providing us the opportunity and motivating us to work on this project and research work.

VIII. REFERENCES

[1]. SYS-CON Media Inc ,“Twenty-one Experts Define Cloud Computing”, October 2012. [Online], Available: http:// cloudcomputing .sys-con.com/read/612375_p.htm

[2]. Wikipedia, “Cloud computing”, September 2012. [Online] Available: http://en.wikipedia.org/wiki/Cloud_computing

[3]. Sun Microsystems,”Introduction to cloud computing architecture”, June, 2006.

[4]. Liu, k.. “Workflow Management Coalition”, Workflow Management Coalition Terminology & Glossary, 1999.

[5]. R. Buyya, J. Broberg, A. Goscinski , ”Cloud Computing: Principles and paradigms”, March 2011. [E-book] Available: Kindle.

[6]. Pandey, Suraj & Wu, Linlin & Guru, Siddeswara & Buyya, Rajkumar. (2010). A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments. Proceedings - International Conference on Advanced Information Networking and Applications, AINA. 400-407. 10.1109/AINA.2010.31.

[7]. S. Abrishami, M. Naghibzadeh, and D. H. J. Epema, ‘‘Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds’’ Future Generat. Comput. Syst., vol. 29, no. 1, pp. 158–169, 2013.

[8]. J. Sahni and P. Vidyarthi, "A Cost-Effective Deadline-Constrained Dynamic Scheduling Algorithm for Scientific Workflows in a Cloud Environment," in IEEE Transactions on Cloud Computing, vol. 6, no. 1, pp. 2-18, 2018.

[9]. D. Poola, S. K. Garg, R. Buyya, Y. Yang, and K. Ramamohanarao, ‘‘Robust scheduling of scientific workflows with deadline and budget constraints in clouds’’ in Proc. Int. Conf. Adv. Inf. Netw. Appl. (AINA), 2014, pp. 858–865, 2014.

[10]. D. Whitley, ‘‘A genetic algorithm tutorial,’’ Statist. Comput., vol. 4, no. 2, pp. 65–85, Jun. 1994.

[11]. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, ‘‘A fast and elitist multi-objective genetic algorithm: NSGA-II,’’ IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182–197, Apr. 2002.

[12]. D. S. Katz, J. C. Jacob, E. Deelman, C. Kesselman, G. Singh, M.-h. Su, G. Berriman, J. Good, A. Laity, and T. A. Prince, “A comparison of two methods for building astronomical image mosaics on a grid,” in Proceedings of the IEEE International Conference on Parallel Processing Workshops (ICPPW). IEEE Computer Society, 2005, pp. 85–94, 2005.

[13]. USC Molecular Genomics Core, Epigenome Centre. [Online] Available: http://epigenome.usc.edu

[14]. D. A. Brown, P. R. Brady, A. Dietz, J. Cao, B. Johnson, and J. McNabb, ‘‘A case study on the use of workflow technologies for scientific analysis: Gravitational wave data analysis,’’ in Workflows for e-Science. London, U.K.: Springer, 2007, pp. 39–59, January, 2007.

[15]. G. Juve, E. Deelman, K. Vahi, G. Mehta, B. Berriman, B. P. Berman, and P. Maechling, “Data sharing options for scientific workflows on Amazon EC2,” in Proceedings of the ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (SC10). IEEE Computer Society, 2010, pp. 1–9.

[16]. G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi, “Characterizing and profiling scientific workflows,” Future Generation Computer Systems, vol. 29, no. 3, pp. 682–692, 2012.