International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-11, September 2019

i- Rosa, Improved Uncertainty Aware Workflow Scheduling Algorithm

Namrata Bulchandani, Shikha Agrawal, Uday Chourasia, Priyanka Dixit, Smita Sharma

Abstract*: *Many applications running in cloud computing environment are workflow applications which contain large number of precedence specific tasks and so require proper schedule in order to complete successfully. Efficient scheduling of workflow applications is a challenging task. In workflows, the uncertainties like 'uncertain data transfer time' among dependent tasks and the uncertain task execution time, if ignored may lead to deadline violation. The proposed workflow scheduling algorithms so far unconcerned these uncertainties. This paper presents an improved unceRtainty aware workflow scheduling algorithm abbreviated as i-ROSA, that considers the uncertainties of scheduling workflows such as the uncertain running time of tasks in distributed environment when focused upon gave the superior outcome in way of cost and resource utilization, for DAG when compared with the original algorithm. The compared task scheduling algorithms are implemented in Workflowsim.

Keywords* : *Scheduling, Tasks, Uncertainties, Workflows.

I. INTRODUCTION

Fields like biology, chemistry, physics, finance, mathematics etc come under scientific computing areas. The scientific applications are required to be run timely and speedily [2]. Scientific applications produce a number of workflow tasks. Such tasks which are related (such applications are called workflow applications) have to be appropriately sorted. In general, a DAG(Dynamic acyclic graphs) is used to represent a workflow. Figure 1 shows the example of a DAG.

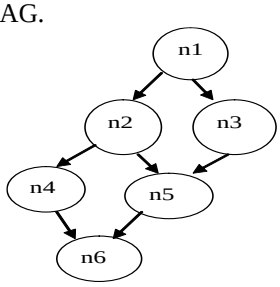


Fig.1. An example of DAG

Manuscript published on 30 September 2019.
*****Correspondence Author(s)
Namrata Bulchandani*, Computer Science & Engineering, University Institute of Technology, RGPV, Bhopal, India. Email: namratabulchandani11@gmail.com
Dr. Shikha Agrawal, Computer Science & Engineering, University Institute of Technology, RGPV, Bhopal, India. Email: shikha@rgtu.net
Prof. Uday Chourasia, Computer Science & Engineering, University Institute of Technology, RGPV, Bhopal, India. Email: uday_chourasia@rediffmail.com
Priyanka Dixit, Computer Science & Engineering, University Institute of Technology, RGPV, Bhopal, India. Email: priyanka.dixit@yahoo.com
Smita Sharma, Computer Science & Engineering, University Institute of Technology, RGPV, Bhopal, India. Email: sharmasmita34@gnmail.com

Workflow scheduling means planning the allocation of workflow tasks to virtual machines based on the scheduling algorithms[3]. Other than makespan being the primary objective for effectiveness in scheduling, cost factor also matters in mutual benefit of both the users and the cloud providers. This paper compares the uncertainty aware algorithm with traditional algorithm on the basis of cost. All these factors only matter when a particular workflow completes on time. So the purpose of the algorithm presented here, is to complete the subsequent dependent tasks on time alongwith optimizing the cost. The broker or scheduler decides the virtual machines to which the tasks are to be allocated on the basis of specified QOS constraints like deadline and cost.

The task scheduling in such applications is a complex process since the processing of all the tasks depends on the output data of their predecessor tasks. There are many historical approaches in workflow scheduling.

Their one of the most prominent weakness is that they all assumed that the tasks running time and data transfer time are pre-known. However, in actual cloud systems the execution time of the tasks in any workflow cannot be predetermined. In this paper we compare the investigated Improved Uncertainty Aware Online Workflow Scheduling Algorithm abbreviated as i- ROSA with the original ROSA and a traditional algorithm FCFS.

II. MOTIVATION

The uncertain factors in dependent tasks' scheduling, the 'uncertain task execution time', the 'uncertain data transfer time' among tasks and the 'sudden arrival' of new workflows may result in non optimized allocation. Hence these uncertainties result in over utilization of resources which can in turn increase the users' expense. Keeping too short time for tasks may not be right, as the actual execution time may be longer, it can vary according to the network bandwidth(to transfer data among tasks) or due to different capabilities of virtual machines in distributed environment. This further can postpone the waiting tasks and the successor tasks and in turn result in deadline invasion. Whereas, reserving large amount of time too may be inappropriate since its real running time may be too less based on resources availability. Hence, running time of jobs and data transfer time among the tasks are 'uncertainties'. To solve the above problem, the sub problems are formulated in investigated paper: (1) how to diminish these uncertainties to give baseline schedules, (2) how to diminish the free time slots of the instances to reduce renting costs, while meeting deadlines.

The paper ahead is in the sequence: Section 2 is dedicated to Related Work. Section 3 gives the proposed methodology.

Section 4 explains the implementation and experimental results. As such experimental settings are introduced and performance results of algorithms are analyzed. Finally, in Section 5 paper is concluded with future work.

III. RELATED WORK

We start our analyses by the traditional algorithm FCFS. First Come First Serve (FCFS) algorithm is usually considered for parallel processing. It selects the instances having least waiting tasks . The disadvantage of FCFS is that it is non- preemptive. In non-preemptive scheduling the processing of tasks occurs according to their incoming order. The tasks having long makespan may finish before shorter tasks [4].Min-Min algorithm arranges the tasks according to their length. The shorter length tasks are first given to the machines having least expected completion time. Drawback of this procedure is that it chooses small tasks to be executed firstly, this in turn delays the longer tasks [5].

Max-Min algorithm is like Min-Min, but it arranges the tasks according in descending order of their length. Drawback of the algorithm is that it firstly chooses large tasks to be executed, this in turn delays the shorter tasks. Both max-min and min-min keep updating the available time of machine, i.e. suppose task 1 takes 80 seconds to complete , the execution time of next tasks are expected to be 80 seconds [6].Priority scheduling algorithm considers the priority of jobs for scheduling. It is based on multiple criteria decision making model. The list of the jobs is sorted on basis of priority. The task with greatest priority is chosen and it assigns it to resource that has minimum expected completion time. Drawback of the algorithm low priority processes may wait long time [7].Round Robin algorithm applies the time slicing technique to schedule the tasks. Each task is given a pre- decided time slice. After it operates for specified time slot, the next waiting task on that particular instance is allocated the memory, CPU and storage of that instance, again for that fixed time slot [8]. The next subsection tells about the uncertainty- aware architecture to execute workflows in cloud service environment.

There has been rapid research in workflow scheduling. The latest research in the field includes lot of successful algorithms described ahead. Lagrange relaxation aggregated cost algorithm [20] found out scheduling decision arrangement of every task to reduce the utilization of power on the mobile while focusing on meeting deadline. Design of this workflow scheduling algorithm saved energy utilization when differentiated to local execution and came out to be more optimal than remote execution in terms of flexibility.

Numerous workflow designed applications are stored in cloud. The proposed algorithm [21], Extended dynamic constraint algorithm (add-on of multiple choice knapsack problem- MCKP) was compared with prevailing scheduling algorithm - Extended Dynamic Constraint Algorithm (EDCA). It guaranteed that monetary cost is optimized along with secure and reliable operation. It reduced 25% failures while generating the cheapest solutions among three algorithms.

Resource allocation for workflow scheduling always persists as a problem. Next is the study of a novel hybrid algorithm CR-AC. It came on combining the chemical reaction optimization and ant colony optimization algorithms which were proposed [22] to optimize the workflow scheduling. When compared with traditional CRO, ACO and recent PSO and CEGA; it is observed that the new algorithm gives finer results in terms of makespan and cost of planning the schedule of three workflows, on a number of machines. It achieves a high-standard optimal schedule with negligible cost meeting the deadline constraint. Outperforms given algorithms in all cases (different tasks on different no of VMs).

Workflow scheduling is a main issue in cloud computing, execution of inter dependent tasks take place along with considering Quality of Service (QOS) requirements. Quality of Service (QOS) aware workflow scheduling algorithm[23] compared 20-30 of heuristic, meta-heuristic, and hybrid algorithms with their numerous QOS constraints. It was concluded that most algorithms took makespan and cost as the reduction motives, so future work in workflow scheduling must consider fault tolerance along with load balancing, workflows security as well as protection of cloud recourses.

There are many workflow scheduling algorithms designed, neither of which considered uncertainties associated with workflows. The uncertain execution time on a machine, and the random uncertain arrival of workflows resulted in the coming of unceRtainty aware Online Scheduling Algorithm [1] abbreviated as ROSA. Being aware of uncertainties, this algorithm optimizes service renting cost, resource utilization, schedule deviation and resource utilization fairness.

IV. PROPOSED METHODOLOGY

A. Service Instance Modelling

Differently parameterised service instances are provided in cloud platform [9], [10], [11]. Let us use the symbol s_u^k as the type u of k- th service instance; m denotes the number of obtainable service types, u: {1,2,…,m} refers service type index. Distinct types of service instances in {1,2,….,m} refer to different configurations and prices. Price (u) symbolize for cost of service type ‘u’ and the configurations like CPU, memory sizes and network bandwidth vary. The instances in cloud platform are priced by integer time units. The instances can be released at any point of time [12]. In this paper, it is supposed that the service instances will only be released if they have processed all the allocated task and transferring of data.

B. System Model and Architecture

To diminish the effect of uncertainties introduced, the explored approach, the uncertainty aware algorithm develops a novel architecture for executing workflows in cloud service environment depicted in Figure 2. The service platform can be split into three components, i.e., cloud clients, a stock of instances and a scheduler. The buyers can give in the complex applications to the cloud platform no matter when. The cloud platforms then supply a pool of service instances which can be increased or reduced in number on the basis of demand, dynamically. The scheduler acts as the mediator which maps the incoming tasks from user area to service instances considering the predefined objectives and meeting the constraints specified in applications and by platform providers.

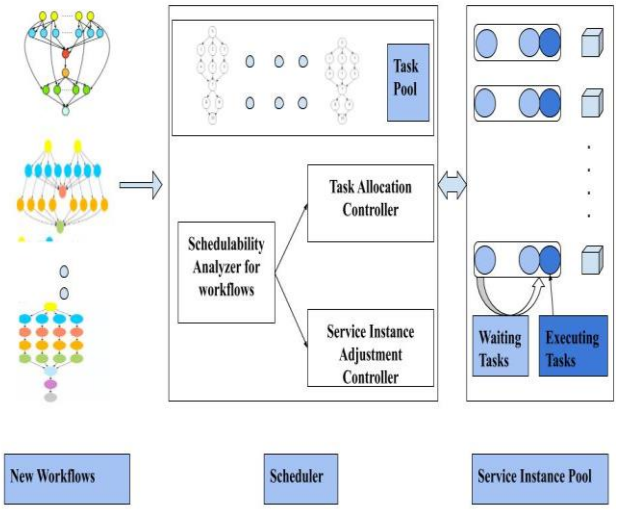


Fig. 2. The uncertainty -Aware Scheduling Architecture

The scheduling strategy in this work of the scheduler is as follows. The units of scheduler are: task pool (TP), workflows schedulability analyzer, task allocation controller, and instance adjustment controller. The task pool contains the waiting workflow tasks, these waiting tasks are allocated to service instances by schedulability analyzer, it also makes the plan for service instances adjustment. This plan consists of when to lease new instances of distinct types, and the execution of this plan is done by adjustment controller. Furthermore, the task controller performs the foremost responsibility of allocating the tasks to selected service instances according to the specified plan. The next sub-section tells the algorithm to implement this procedure.

C. Algorithm Description

For the above depicted scheduling architecture, the implemented working process can be explained in following way-

- When new workflows enter, the ranking of tasks is the prime step in this algorithm. Then, immediately, all the ready tasks are given to the machines as running tasks or the waiting tasks. If the ready for use instances are deficient, then new service instances are released. Furthermore, the remaining tasks, non-ready tasks are placed in the task pool.
- Case two can be the second uncertainty considered, i.e., when the workflow arrives even when the machines are busy. As soon as this task is completed, the waiting task on that instance is executed provided, all data from its predecessor task is received. Furthermore, the successor tasks of the completed task become ready. So the next appropriate action is to allocate these tasks that are ready to the instances instantly. In the same way, if the ready for use service instances are deficient, then new service instances are released.

The interesting feature of this setup is that only the ready tasks can be mapped or can wait directly on the service instances, and the waiting tasks are kept in the task pool.

In order to bring the above strategy into existence, these rules are followed.

Rule 1. Only one workflow task can run on a service instance at a time.

Rule 2. Tasks waiting on the service instance can run as soon as they receive the data from their predecessor tasks.

Rule 3. Only when the tasks allocated and the data transfer for all the dependent tasks executed by a machine are fully completed, then only the instances are freed.

The Figures 3 and 4 diagrammatically explains the proposed approach through flowchart.

With contrast to the traditional scheduling schemes [9], [12], [13], on the arrival of new workflows (in Figure 3), the ranking of tasks is the prime step in this algorithm. Then, immediately, all the ready tasks are allocated to the machines as executing tasks or the waiting tasks and all the waiting tasks are placed in the task pool. See Figure 3 and Figure 4 for understanding this process diagrammatically. The other case, in Figure 4, is when an instance completes a workflow task. As soon as this task is completed, the waiting task on that instance is executed provided, all data from its predecessor task is received. Furthermore, the successor tasks of the completed task become ready and are sorted according to their $plst_{ij}$. Then, all these ready tasks are allocated to service instances instantly and are removed from the task pool. The mapping is done by function mapReadyTask() guaranteeing that the predicted finish time (i.e., $plft_{ij}$) of the workflow task is lower than its deadline.

For the better optimization of cost and resource utilization, when the second uncertainty is looked upon, on the sudden completion of ready tasks, if the machines do not have any waiting tasks, they are disabled. The figure 4 shows this that on searching for other tasks to execute, if there are no waiting tasks, plus the next workflow tasks are'nt ready too, the service instance is turned off using the function optimizeVm(). Now in the next section we'll see through experiments how algorithm gives us considerable performance.

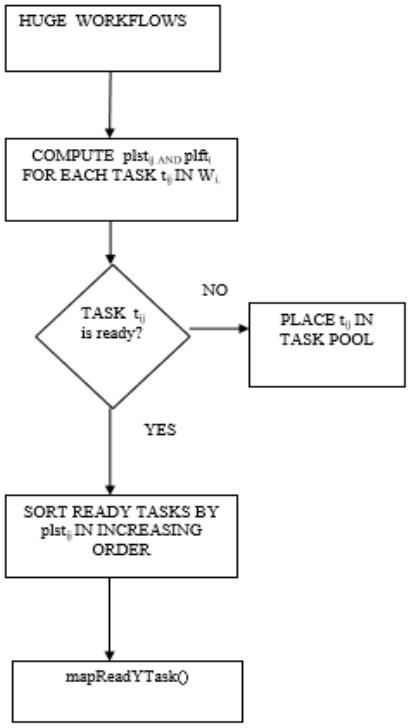


Figure 3. Algorithm process followed when new workflows arrive

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

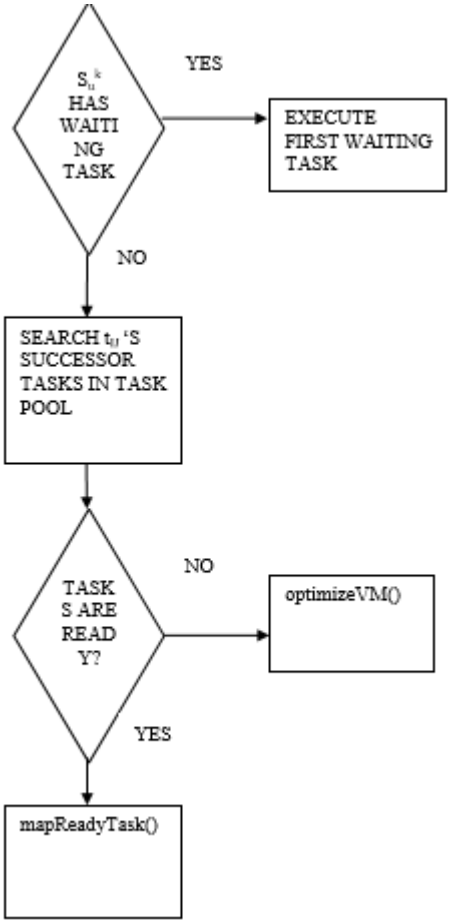


Fig. 4. Algorithm process followed when a task completes suddenly

V. RESULT AND DISCUSSION

A. Experimental Setup

The software used for the experiments is Eclipse Java Neon and WorkflowSim1.0. The configuration of the system used are 64-bit Windows 10 operating system, Intel(R) Core(TM) i5 CPU 2.20 GHz, 8GB RAM. The proposed algorithm is developed in Java language. Then Workflow Sim simulator is used to test the developed algorithm. It offers workflow level support during the simulation.

WorkflowSim is the toolkit used for simulation of scheduling algorithms. It is the advanced version of CloudSim by offering better workflow management and accurate evaluation. CloudSim [14] is used for simulation of cloud services and infrastructure. CloudSim allows executing only single workload. It does not consider failures and overheads. It does not support clustering and job dependencies. Unlike other simulators, failures and overheads occurred in the heterogeneous system are considered into the WorkflowSim. It also supports clustering. Figure 5 shows the WorkflowSim Architecture. WorkflowSim contains multiple layers such as Failure monitor, Failure generator, Clustering engine, Workflow engine and Workflow mapper along with the Workflow scheduler which is present into the CloudSim. Workflow mapper has used for mapping non-concrete workflows to the actual workflows which are reliant on execution sites. Data dependencies are managed by Workflow engine. Tasks are scheduled to the resources with the help of Workflow Scheduler. Small jobs are combined into a large one by using Clustering engine [15].

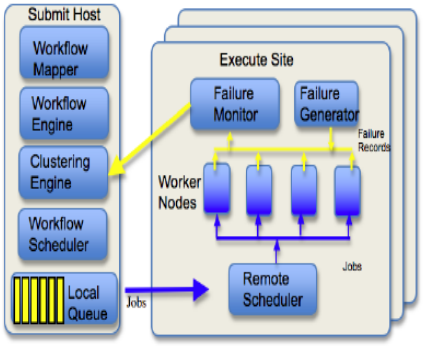


Fig. 5. Workflowsim Architecture

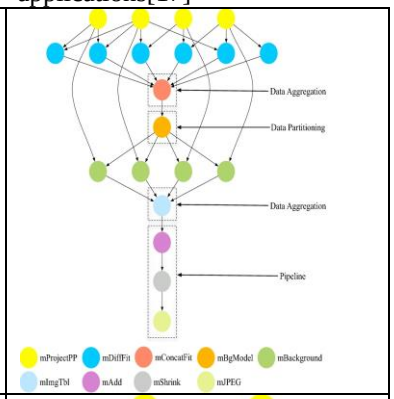
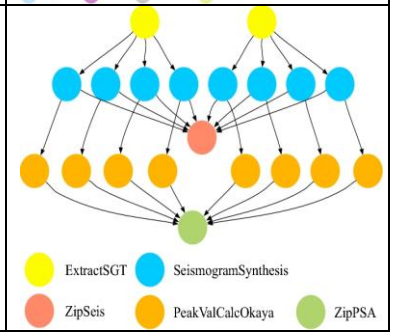
To test the proposed algorithm, following configuration of virtual machines are used(shown in below table)

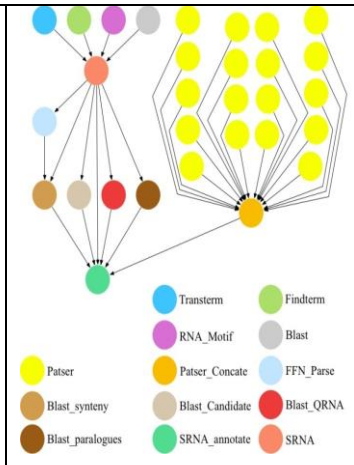
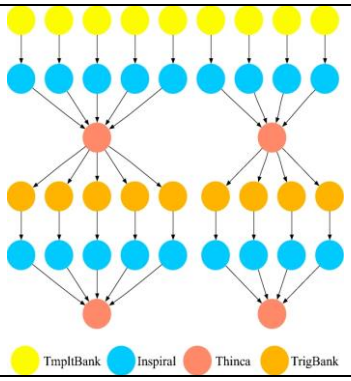
Table-I: Virtual Machines configurations used in simulation.

Service type(u)	Type name	Price($)	F(u)
1	Small	0.023	8
2	Medium	0.464	4
3	Large	0.0928	2
4	xLarge	0.1856	1

For experimenting, these four scientific workflow applications are used: Montage (used in astronomy), Cyber Shake (used in earthquake science), SIPHT (used in bioinformatics), LIGO Inspiral (used in researching gravitational physics) [16]. For each workflow class, the structure of the workflow is shown in Table 2, which is taken from the Pegasus Workflow repository [17]. It gives the description of the four applications which will be used in experiment in the next chapter.

Table-II: The structure of four real world workflow applications[17]

(1) **Montage**:	
(2) **Cybershake:** .	

(3) **Sipht :**	
(4) **Inspiral** :	

B. Simulation Results

- **Service renting cost evaluation**

The first experimental result is carried out by comparing the proposed i- ROSA with ROSA [1] and a traditional algorithm FCFS fluctuating the parameter deadline base to consider the service renting cost constraint. Table 4 shows the service renting costs that are read at different deadline base for the three algorithms. It is seen in figure 7 that the service renting cost of all the algorithms decrease slowly with increasing deadline base. The reason being that, as we increase the deadline, the schedulers look for the cheaper machines since it has got longer time to execute the same task. It is clearly visible that the i-ROSA algorithm outperforms the other two algorithms and gives the reduced cost.

Table-III: Service renting costs for different workflow algorithms at different deadline base.

Deadline base	I-ROSA	ROSA	FCFS
2	1600	1780	2500
4	1400	1787	2520
6	1453	1726	2550
8	1339.78	1710	2573
10	1100	1780.7	2571

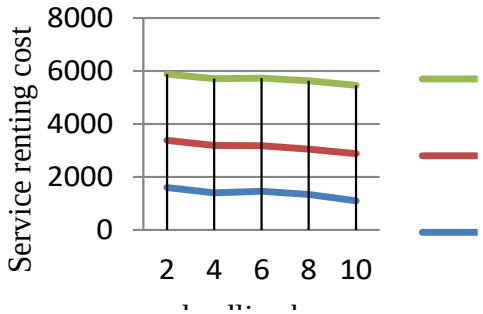


Fig. 7. Comparison of algorithms by single workflow application with increasing deadline base.

The second experimental result is carried out by comparing the proposed i-ROSA, ROSA and algorithm FCFS to optimize the considered QOS constraint- service renting cost constraint. The i-ROSA algorithm outperforms than ROSA as it gives the reduced cost with different types of scientific applications having same number of tasks(100). Table 4 shows what the service renting costs come for different workflow applications using all three scheduling algorithms. Figure 8 shows the clear comparison. It is clearly seen that the i- ROSA outperforms all the other algorithm and gives least service renting costs and the FCFS gives highest cost

Table-IV: Service renting costs for different workflow algorithms at different deadline base.

Deadline Base	i-ROSA	ROSA	FCFS
2	1600	1780	2500
4	1400	1787	2520
6	1453	1726	2550
8	1339.78	1710	2573
10	1100	1780.7	2571

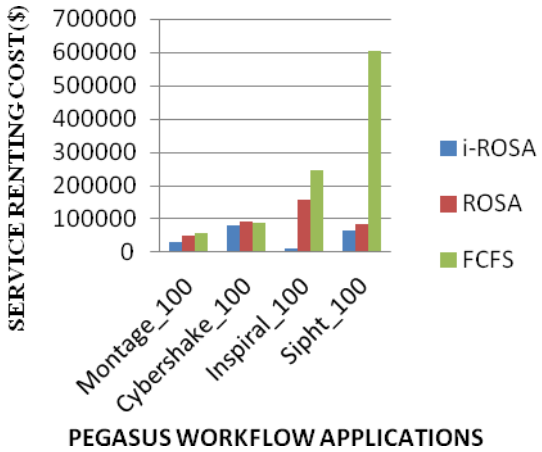


Fig. 8. Comparison of algorithms by different pegasus workflow applications with same number of tasks

- **Resource Utilization evaluation**

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The third experimental result is carried out by comparing the proposed i-ROSA with ROSA[1]

and a traditional algorithm FCFS fluctuating the parameter deadline base to consider the second constraint of resource utilization constraint. It is seen that the resource utilization of all the algorithms individually increase slowly with increasing deadline base. The reason being that, when the deadline is early, the scheduler maps the task to a machine having high processing speed and when the deadline is more, the slower machines too get the tasks. Table 5 gives the resource utilization readings in simulation cloud using different scheduling algorithms on deploying a single application. Looking comparatively, it is clearly visible in figure 9 that the i-ROSA algorithm gives higher resour5e utilization than the other two algorithms.

Table-V: Service renting costs for different workflow algorithms at different deadline base.

Deadline base	i-Rosa	Rosa	Fcfs
2	0.6291	0.5231	0.34
4	0.698	0.5322	0.378
6	0.7231	0.6487	0.432
8	0.778	0.6772	0.491
10	0.891	0.7623	0.541

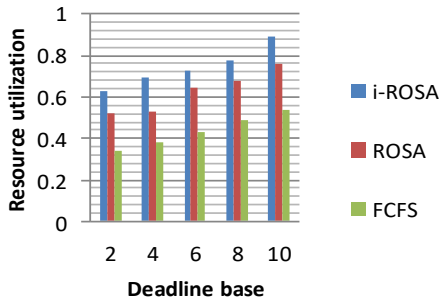


Fig. 9. Comparison of algorithms by single workflow application with increasing deadline base

VI. CONCLUSION AND FUTURE WORK

This study strives to propose a better uncertainty aware workflow scheduling algorithm concerning the service renting cost and resource utilization by considering the uncertainties in cloud service environment. Experimental results convey that the approached architecture for cloud service platforms outperforms all the other algorithms. In the context of real-workflow traces, three experiments were carried out to prove the superiority of the proposed algorithm. To be precise, the i -ROSA performs approximately 50% and 30% better in terms of service renting cost and resource utilization respectively.

Presently, cloud service is chanced to have problem of resource failure [18], [19]. The expensive commodities maintained by cloud providers need to be protected for their effectiveness in case a failure occurs. Hence, the research can be further carried out considering the fault tolerance and robustness in scheduling workflows considering these uncertainties.

REFERENCES

1. H. Chen, X. Zhu, G. Liu and W. Pedrycz, "Uncertainty-Aware Online Scheduling for Real-Time Workflows in Cloud Service Environment," in IEEE Transactions on Services Computing.
2. Gideon Juve, Ann Chervenak, Ewa Deelman, Shishir Bharathi, Gaurang Mehta, and Karan Vahi. 2013. Characterizing and profiling scientific workflows. Future Gener. Comput. Syst. 29, 3 (March 2013), 682-692. DOI= http://dx.doi.org/10.1016/j.future.2012.08.015
3. F. Fakhfakh, H. H. Kacem and A. H. Kacem, "Workflow Scheduling in Cloud Computing: A Survey," 2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, Ulm, 2014, pp. 372-378.
4. Dharma P. AgrawalW. Zhao and J. A. Stankovic, "Performance analysis of FCFS and improved FCFS scheduling algorithms for dynamic real-time computer systems," [1989] Proceedings. Real-Time Systems Symposium, Santa Monica, CA, USA, 1989, pp. 156-165.
5. Anousha S., Ahmadi M. (2013) An Improved Min-Min Task Scheduling Algorithm in Grid Computing. In: Park J.J..H., Arabnia H.R., Kim C., Shi W., Gil JM. (eds) Grid and Pervasive Computing. GPC 2013. Lecture Notes in Computer Science, vol 7861. Springer, Berlin, Heidelberg
6. Kobra Etminani, "A Min-Min Max-Min Selective Algorithm for Grid Task Scheduling," 3rd IEEE/IFIP International Conference in Central Asia on Internet 2007 (ICI 2007), Iran, September 2007.
7. "Efficient Workflow Scheduling Algorithm for Cloud Computing System: A Dynamic Priority-Based Approach " Gupta, I., Kumar, M.S. & Jana, P.K. Arab J Sci Eng(2018) 43: 7945. https://doi.org/10.1007/s13369-018-3261-8
8. C. Zhou and S. K. Garg, "Performance Analysis of Scheduling Algorithms for Dynamic Workflow Applications," 2015 IEEE International Congress on Big Data, New York, NY, 2015, pp. 222-229.doi: 10.1109/BigDataCongress.2015.39
9. Z. Cai, X. Li and J. N. D. Gupta, "Heuristics for Provisioning Services to Workflows in XaaS Clouds," in IEEE Transactions on Services Computing, vol. 9, no. 2, pp. 250-263, 1 March-April 2016.
10. R. N. Calheiros and R. Buyya, "Meeting Deadlines of Scientific Workflows in Public Clouds with Tasks Replication," in IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 7, pp. 1787-1796, July 2014.
11. M. A. Rodriguez and R. Buyya, "Deadline Based Resource Provisioningand Scheduling Algorithm for Scientific Workflows on Clouds," in IEEE Transactions on Cloud Computing, vol. 2, no. 2, pp. 222-235, 1 April-June 2014.
12. Scheduling Precedence Constrained Stochastic Tasks on Heterogeneous Cluster Systems IEEE TRANSACTIONS ON COMPUTERS, vol. 63, no. xx, 2014
13. X. Zhu, J. Wang, H. Guo, D. Zhu, L. T. Yang, and L. Liu, "Fault tolerant scheduling for real-time scientific workflows with elastic resource provisioning in virtualized clouds," IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 12, pp. 3501–3517, 2016.
14. A. and Buyya, R. (2011). Cloudsim: a tool kit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Software: Practice and Experience 41(1):23–50.
15. Chen, W. and Deelman, E. (2012). Workflow sim: A toolkit for simulating scientific work- flows in distributed environments, E-Science (e-Science), 2012 IEEE 8thInternational Conference on, IEEE, pp.1–8.
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, 2013.
17. https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator

18. R. N. Calheiros and R. Buyya, “Meeting deadlines of scientific workflows in public clouds with tasks replication,” IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 7, pp. 1787– 1796, 2014.
19. Z. Wen, J. Cala, P. Watson, and A. Romanovsky, “Cost effective, reliable and secure workflow deployment over federated clouds,”IEEE Transactions on Service Computing, vol. 10, no. 6, pp. 929–941, 2017.
20. W. Zhang and Y. Wen, "Energy-Efficient Task Execution for Application as a General Topology in Mobile Cloud Computing," in IEEE Transactions on Cloud Computing, vol. 6, no. 3, pp. 708-719, 1 July-Sept. 2018.
21. Cost-Effective Algorithm for Workflow Scheduling in Cloud Computing Under Deadline Constraint Nasr, A.A., El-Bahnasawy, N.A., Attiya, G. et al. Arab J Sci Eng (2019) 44: 3765. https://doi.org/10.1007/s13369-018-3664-6
22. Z. Wen, J. Cała, P. Watson and A. Romanovsky, "Cost Effective, Reliable and Secure Workflow Deployment over Federated Clouds," in IEEE Transactions on Services Computing, vol. 10, no. 6, pp. 929-941, 1 Nov.-Dec. 2017.
23. Kaur, S., Bagga, P., Hans, R. et al. Arab J Sci Eng (2019) 44: 2867. https://doi.org/10.1007/s13369-018-3614-3

AUTHORS PROFILE

Namrata Bulchandani is a research scholar, pursuing Masters in Technology in Department of Computer Science & Engineering at University Institute of Technology, Rajiv Gandhi Proudyogiki Vishwavidyalaya ,Bhopal (MP) India. She obtained B.E. in Computer Science & Engineering from Rajiv Gandhi Proudyogiki Vishwavidalaya Bhopal. Her area of interests includes cloud computing, machine learning and web engineering.

Dr Shikha Agrawal is an Assistant Professor in Department of Computer Science & Engineering at University Institute of Technology, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (MP) India. She obtained B.E., M.Tech. and Ph.D in Computer Science & Engineering from Rajiv Gandhi Proudyogiki Vishwavidalaya Bhopal. She has more than fifteen years of teaching experience. Her area of interest is Artificial Intelligence, Soft Computing and Particle Swarm Optimization and Database. She has published more than 40 research papers in different reputed international journals and 10 chapters. For her outstanding research work in Information Technology, she has been awarded as “Young Scientist” by Madhya Pradesh Council of Science and Technology, Bhopal. Her other extraordinary achievements include “ICT Rising Star of the Year Award 2015” in International Conference on Information and Communication Technology for Sustainable Development (ICT4SD - 2015), Ahmedabad, India and Young ICON Award 2015 in Educational category by Dainik National News Paper Patrika, Bhopal, India. She got recognition of IEEE as a senior member. She is also member of various academic societies such as IEEE, ISTE, CSI, ACM & CSTA.

Prof. Uday Chourasia is an Assistant Professor in Department of Computer Science & Engineering at University Institute of Technology, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (MP) India. His area of interest is Cloud Computing. He has published more than 10 research papers in different reputed international journals and 10 chapters. He is also member of various academic societies.

Priyanka Dixit received the B.E. degree in information technology and Masters in Computer Science & Engineering from RGPV University. She is now working as Assistant Professor in Computer Science & Engineering department of U.I.T RGPV, Bhopal. She has published papers and research articles in international conferences, journals (IEEE, SPRINGER and ELSEVIER) and the member of IEEE and ACM professional societies. Her research interests include Cyber Security, AI, Machine Learning and Image Processing.

Smita Sharma has completed her B. Tech and M.Tech in Computer Science & Engineering from Rajiv Gandhi Technical University, Bhopal (M.P.). She is currently a PH.D scholar in the Department of Computer Science and engineering in SSSUTMS. Her interests of research are cloud security , artificial intelligence and image processing. She has published more than 10 journals and conference papers related to these research areas. She has about six years of teaching experience. She is a member of ACM. She is currently working as an Assistant Professor in Computer Science & Engineering department in University of Information Technology, Rajiv Gandhi Technical University , Bhopal (M.P.).