ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

Rank-Based Task Scheduling Algorithm for Scientific Workflows in Infrastructure as a Service Cloud Computing

Shilpa Rana[1], Dr. K.J.Mathai[2]

M. Tech Student, Department of Computer Engineering and Applications, NITTTR, Bhopal, India

Assistant Professor, Department of Computer Engineering and Applications, NITTTR, Bhopal, India

ABSTRACT: The Cloud Computing is the emerging trend in distributed computing which focus on delivering computing power to the end-user directly, no matter when and where. It delivers software, platform and infrastructure as services i.e., SaaS, PaaS and IaaS to the users. Infrastructure as a Service model of cloud computing is a desirable platform for the execution of workflow of scientific applications such as astronomy, bioinformatics, geophysics etc. Therefore, various researches are carrying on for executing scientific workflows in Infrastructure as a Service cloud computing environment. In this paper an algorithm named as 'Rank-Based Task Scheduling Algorithm' (RBTSA) is designed for scheduling the tasks of scientific workflows to maximize the utilization of resources in IaaS cloud computing environment. The above research work is experimentally simulated on 'WorkflowSim-1.0' and results of proposed RBTSA Algorithm are compared with the existing 'Enhanced Max-Min' Algorithm (EMMA) for executing scientific workflow. The experimental result shows that proposed RBTSA schedules the task of workflows with better resource utilization as compared to existing EMMA and it therefore reduces the makespan of workflow.

KEYWORDS: Cloud computing; Task Scheduling; Scientific Workflow; Infrastructure as a Service; Resource Provisioning; Resource Utilization; Makespan

I. INTRODUCTION

Cloud Computing is the emerging trend in distributed computing that delivers infrastructure, platform and software as services i.e., IaaS, PaaS and SaaS to the users. The users can consume these services based on a Service Level Agreement (SLA) which describes their essential Quality of Service (QoS) parameters on a pay as you go basis. Cloud computing paradigm offers tremendous opportunities to execute large scale scientific applications. Scientific communities in areas such as astronomy, bioinformatics, geophysics, high-energy physics, weather monitoring, earthquake science and gravitational wave physics are utilizing cloud environment to share, manage and process large data sets. These scientific applications consist of hundreds and thousands of compute and data intensive tasks and hence it becomes increasingly complex because it requires large number of computations and huge data transfer. These scientific applications are modeled as workflows and are called as scientific workflow.

Workflow is defined as coarse-grained parallel applications that are represented by a Directed Acyclic Graph (DAG) in which each task that is responsible for computation is presented by nodes and each data/control dependency between tasks is represented by edge between corresponding nodes (Juve et al. 2011) [1]. Montage, CyberShake, Epigenomics, LIGO (Laser Interferometer Gravitational-Wave Observatory) and SIPHT (sRNA Identification Protocol using High-throughput Technology) are well-known examples of scientific workflow applications. IaaS cloud computing offers many advantages for execution of scientific workflows. It gives the illusion of infinite resources to the users. Thus, the users can demand for any number of resources any time. As workflows are loosely-coupled applications this model is ideal for scientific workflow applications. However, the execution of scientific workflows in IaaS cloud computing environment presents the challenges of efficient scheduling of workflow tasks. If a schedule is properly planned it will lead to increase in resource utilization and reduces 'makespan'. But due to the interdependencies among the tasks of scientific workflows it is difficult to attain maximum resource utilization and minimum 'makespan' for scientific workflows. Thus, scheduling of scientific workflows in IaaS cloud environment is a significant research issue.

Current Time based workflow scheduling algorithms focuses only on 'makespan' and has not considered the resource utilization. Selection of resources for scheduling the tasks of scientific workflows in an efficient manner helps in improving the resource utilization along with the 'makespan'. For scheduling of scientific workflows there is a need of such algorithms which may increase the resource utilization and further reduces the 'makespan'. Scheduling of workflow tasks has a crucial impact on execution time and utilization of resources in IaaS cloud environment and hence it must be done optimally.

So motivated by the above mentioned challenges in this paper, the authors have proposed an algorithm which is named as '**R**ank-**B**ased **T**ask **S**cheduling **A**lgorithm' (RBTSA) to schedule the scientific workflow applications on Infrastructure as a Service (IaaS) cloud service model so that the 'makespan' of workflow is reduced which results in efficient resource utilization because lesser the value of 'makespan' of algorithm, the better is the resource utilization. The proposed algorithm is based on the 'Enhanced Max-Min Algorithm' (Upendra Bhoi et al. 2013) [7]. In the proposed algorithm resources for scheduling tasks of workflow are selected based on their assigned rank values. The proposed algorithm states that "Select a task whose completion time is maximum and assign it to the resource which has highest rank among all the resources".

II. RELATED WORK

Various researchers have proposed different workflow scheduling algorithms. One of the well-known heuristic for scheduling scientific workflows is Heterogeneous Earliest Finish Time (HEFT) (Topcuoglu et al. in 2002) [2]. HEFT is a static scheduling algorithm that aims to minimize makespan only. An extension of [2] is proposed in [3] (Cui Lin et al. in 2011) which is called as Scalable Heterogeneous Earliest Finish Time (SHEFT) algorithm. In the proposed algorithm resources are grouped into clusters based on their data transfer rate and tasks of workflows are ordered based on their priority. The resources can be scaled-in and scaled-out as per the requirements. The proposed SHEFT algorithm outperforms HEFT while scheduling the tasks of scientific workflows.

Similar to [3] an algorithm is proposed by named as Partitioned Balanced Time Scheduling (PBTS) (E. K. Byun et al. in 2011) [4] that considers the elastic adjustment of cloud resources but at runtime. The primary goal of the research was to minimize the cost not makespan of workflow. Hence, there may be scenarios where it can violate deadlines. Moreover, this work only considers the single type of resources available, ignoring the heterogeneous nature of IaaS cloud resources.

The algorithm proposed by (M. Mao et al. in 2011) [5] considered Cost Optimization and Deadline Constraint as an objective while scheduling the workflows. The authors have proposed a dynamic approach for scheduling workflow ensembles in cloud in a way such that execution cost is minimized within the user defined soft deadlines. Similar to (M. Mao et al. in 2011) [5] the authors in [6] (M. Malawski et al. in 2012) have developed static and dynamic scheduling algorithms for provisioning and scheduling of resources for ensemble of workflow applications. The author have assigned a priority to each workflow in ensemble and the goal of the scheduling and provisioning algorithm was to complete the execution of maximum number of high priority workflows within the given deadline and budget constraints.

In contrast to [4] the algorithms proposed by (M. Mao et al. in 2011) [5] and (M. Malawski et al. in 2012) [6] consider the heterogeneous nature of resources while scheduling the tasks of scientific workflows in IaaS cloud environment. However, these algorithms do not focus on the efficient utilization of IaaS cloud computing resources. Hence, results in the inefficient utilization of resources.

The algorithm proposed in [7] named as 'Enhanced Max-Min' (Upendra Bhoi et al. in 2013) is an individual task scheduling algorithm. The primary goal of the proposed algorithm is to reduce the 'makespan' and balance load across resources. However, it is not suitable for executing scientific workflow applications.

From the review of various workflow scheduling algorithms, it can be analyzed that these workflow scheduling algorithms have the advantage of reducing the makespan of workflow but they suffer from the shortcomings of not considering the resource utilization of IaaS cloud computing resources. Resource utilization in IaaS cloud computing is an important performance measure for accessing the efficiency of a scheduling algorithm for scientific workflows. To overcome the above identified shortcoming the authors in this paper will design a scheduling algorithm names as '**R**ank **B**ased **T**ask **S**cheduling **A**lgorithm' (RBTSA) which is developed by improving the 'Enhanced Max-Min Algorithm' (EMMA) [7] for scientific workflow. The proposed RBTSA algorithm not only considers the maximization of resource utilization of IaaS cloud environment but will also minimize the execution time of individual tasks of the workflow by

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

scheduling them on the virtual machine selected based on their rank. The rank of virtual machine is calculated by considering the characteristics of virtual machines.

III. Proposed Methodology

In proposed RBTSA algorithm, firstly the rank values are calculated for each resource based on two of the important parameters - the CPU speed and memory requirements of VM resource. Then, a task is selected among all the tasks of workflow whose execution time is greater than or equal to the average execution time of all the tasks of workflow. The selected task is the first task to be scheduled and is scheduled on the resource which results in the minimum completion time. After scheduling the first task of the workflow the scheduling of rest of the tasks of workflow is done. In every cycle the task whose completion time is maximum among all the tasks of scientific workflow is selected and is scheduled to the highest rank resource available at that time. In the proposed RBTSA algorithm for task scheduling of scientific workflows, three scheduling parameters are considered for optimization- Resource utilization, Execution time of each task and Makespan. The flow diagram for the proposed RBTSA algorithm is given in Fig.1 and the steps for developing the RBTSA for scientific workflow scheduling are as follow:

Step 1: The proposed algorithm for the task scheduling of scientific workflow starts from giving a scientific workflow as input to the scheduler.

Step 2: The execution time and completion time of each task is calculated on each resource and the minimum value of execution time is taken into consideration. The execution time and completion time is calculated based on the following formula:

Execution timeij = File Sizei / MIPSj eq. (1)

Completion timeij = Execution timeij + Waitj eq. (2)

Where, i is the index of task.

j is the index of resource

After calculating the execution time of each task, now the average of the execution time of each task is calculated. This average execution time helps in the selection of first task for scheduling.

Step 3: ***Phase-I: Task Selection*** –In this, task selection phase; a task of workflow is selected for scheduling based on some criteria. For the selection of tasks of workflow there are two cases:

Case i) If the task that is to be selected for scheduling is the first task of the workflow then select that task of workflow whose execution time is just greater than or equal to the average execution time (calculated in step 2) of all the tasks. After selecting the task go to step4.case-i for scheduling the task.

Case ii) If one or more tasks of the workflow are already scheduled, then select the task whose completion time (calculated in step 2) is maximum among all the tasks of workflow. After selecting the task go to step 4.case-ii for scheduling the task.

Step 4: ***Phase-II: Resource Selection*** –In this resource selection phase; a resource is selected for allocating it to the task selected in step 3. For the selection of resources there are two cases:

Case i) If the task is selected from step 3.case-i i.e., the selected task is the first task of the workflow to be scheduled then choose the resource which gives minimum completion time on scheduling the task.

Case ii) If the task is selected from step 3.case-ii i.e., the selected task is other than the first task then choose the highest rank resource available at that time among all the resources. The rank of the resource is calculated based on the following formula given by (Soni et al. 2014) [8].

$Rank\ (j) = a * MIPS_j + b * RAM_j$ eq. (3)

Step 5: ***Phase-III: Task Submission*** – In this task submission phase; the mapping of task and resource is done which is obtained in step 3 and step 4. Now, the task that is selected in step 3.case-i or step 3.case-ii is submitted to the resource chosen in step 4.case-i or step 4.case-ii for execution. The scheduled task is deleted from the meta-task list of workflow and the resource list is updated.

Step 6: Repeat step 3 to step 5 until the meta-task list of workflow is not empty.

Step 7: All the tasks of workflow are now scheduled and have completed their execution. Print the results and exit.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

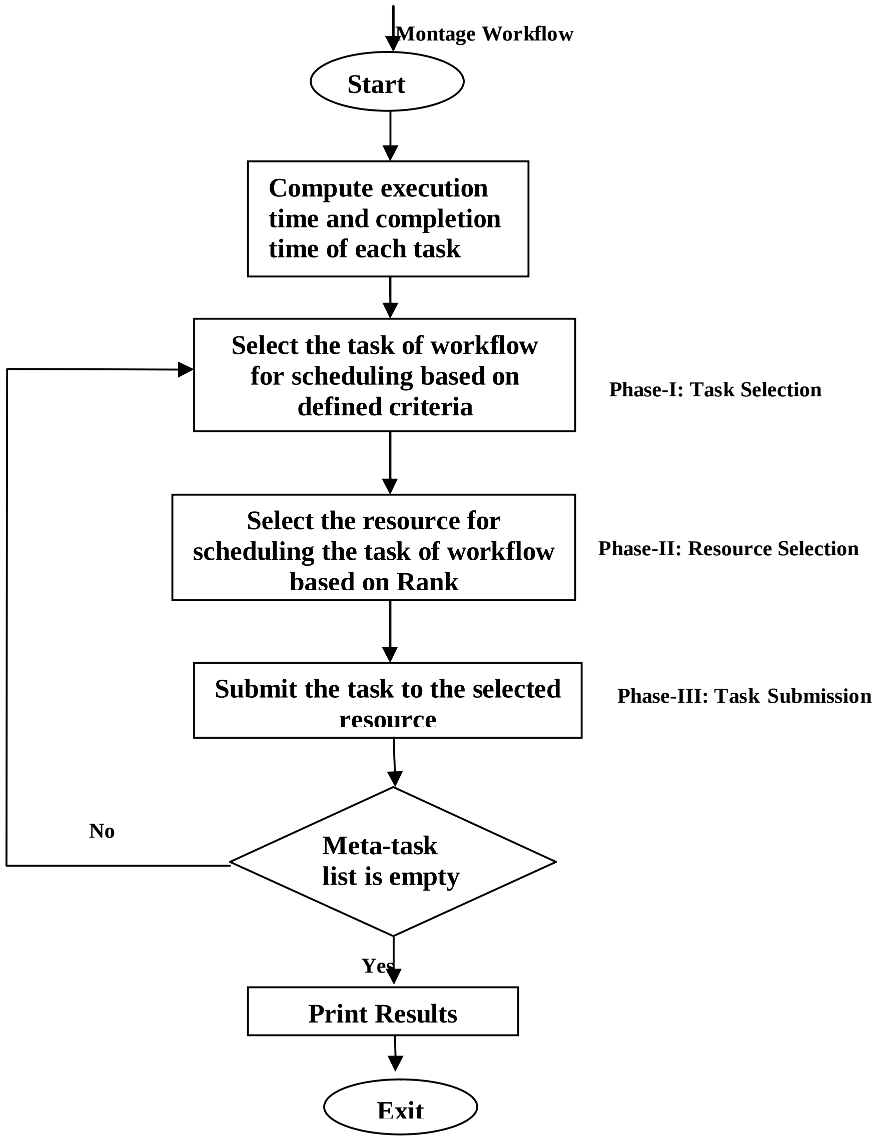

Fig.1 Flow Diagram of Rank-Based Task Scheduling Algorithm

IV. IMPLEMENTATION

The authors have used the WorkflowSim simulator for implementing the proposed RBTSA algorithm. It is an open source simulation toolkit for scientific workflows in cloud computing environment. WorkflowSim provides the workflow level support of simulation by extending CloudSim simulator. It accepts the directed acyclic XML files of workflows and executes the tasks by considering the parent child interdependencies of workflows. Different Simulation parameters considered for the simulation are defined in Table I The virtual machines are created with the parameters specified in the simulated environment of WorkflowSim and then implement the proposed RBTSA scheduling algorithm for scientific workflows. The aim of the proposed RBTSA algorithm is to minimize the total execution time (Makespan) of the workflow and to maximize the utilization of virtual machine. After completing the simulation for both approaches, the output of the existing and proposed rank based algorithm is analyzed to check the efficiency of proposed algorithm.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

TABLE I.
TABLE II. SIMULATION PARAMETERS AND THEIR VALUE

Virtual Machine Parameters	
Parameter Name	***Value***
Number of Virtual Machines	10
RAM	Varying size from 256 to 2048 Minimum 256 MB to Maximum 2048 MB
MIPS	Varying size from 1000 to 2000 Minimum 1000 MIPS to Maximum 2000 MIPS, Varying by 100 MIPS per Virtual Machine
Bandwidth	1000
Number of PES	1
Physical Memory	10000 MB

V. PSEUDO CODE

Rank Based Task Scheduling Algorithm
Input: tasks of scientific workflow
Output: Scheduled list of tasks

```
1. set size ← number of tasks in scientific workflow
2. set vmsize ← number of vm resources
3. set index ← 0
4. set task ← null
5. set vm_resource ← null
6. for i ← 0 to size - 1
        a) task ← task whose execution_time >= avg_execution_time
   end for
7. for j ← 0 to vmsize – 1
        a) vm_resource ← resource which gives minimum completion time
        b) index ← j
   end for
8.
      a) set task.vmid ← index
      b) Remove task from task list of workflow
      c) Set vm_resource_status ← BUSY
9. for i ← 0 to size – 1
      a) for k ← 0 to size – 1
            i) if task is already scheduled
                  continue
               end if
            ii) else
                  task ← task which has maximum    completion time
         end for
      b) for j ← 0 to vmsize – 1
            i) if vm_resource_status ← BUSY
                  continue
            ii) else
                  vm_resource ← highest ranked resource
            iii) index ← j
         end for

      c)
            i) Set task.vmid ← index
            ii) Remove task from task list of workflow
```

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

iii) Set vm_resource_status ← **BUSY**

end for

10. Print Results

11. Exit

VI. FINDINGS AND RESULTS

The results and the performance of the proposed '**R**ank **B**ased **T**ask **S**cheduling **A**lgorithm' (RBTSA) for the task scheduling of scientific workflow are explained in this section. The scientific workflow application used in the performance evaluation is Montage (astronomy applications) [10]. The algorithm '**E**nhanced **M**ax-**M**in **A**lgorithm' (EMMA) proposed by Upendra Bhoi et al.[7] is used to compare the performance of proposed '**R**ank-**B**ased **T**ask **S**cheduling **A**lgorithm' (RBTSA) algorithm. The performance is evaluated for following performance parameters for scheduling scientific workflow tasks for infrastructure as a service cloud environment:

- Makespan of Scientific workflow Montage.
- Number of tasks Vs MIPS of Virtual machine.
- Utilization of Virtual Machine.

A.Makespan Comparison of Scientific workflow Montage

Makespan of scientific workflow is a critical performance metric for computing the efficiency of a workflow scheduling algorithm. It is defined as the total time the tasks of the workflow takes to be executed right from the submission of first task to the execution of last task of the workflow. For computing the performance of EMMA and RBTSA with the performance metric makespan, Montage workflow with different number of tasks is executed. Table II shows the value of makespan obtained for the scientific workflow Montage with number of tasks 25, 50, and 100 for EMMA and RBTSA. Fig.2 represents graph corresponding to the data in Table II

TABLE III. EMMA VIS-À-VIS RBTSA: MAKESPAN COMPARISON

Execution Time (makespan)		
Number of tasks	***Enhanced Max-Min Algorithm (ms)***	***Rank Based Task Scheduling Algorithm (ms)***
25	41.35	30.42
50	60.76	49.95
100	108.28	93.28

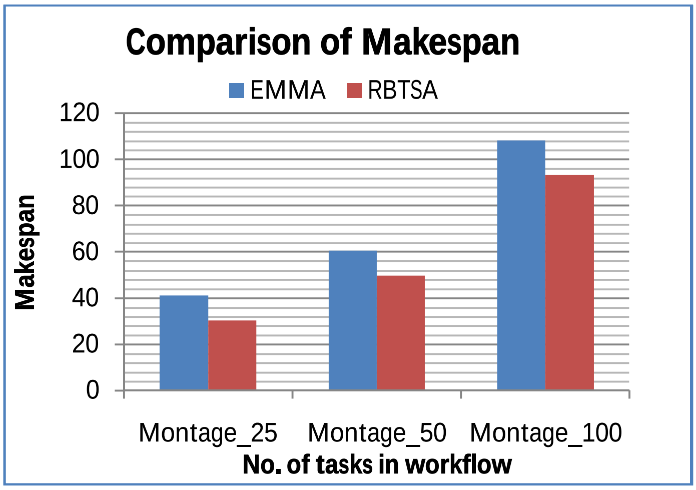

Fig.2. EMMA vis-à-vis RBTSA: Makespan Vs Number of Tasks

It can be analyzed from the graph that the proposed RBTSA achieves less makespan as compared to the existing EMMA for each of workflow Montage_25, Montage_50 and Montage_100. It can be concluded from results of makespan that the RBTSA algorithm reduces the total execution time of scientific workflow to meet their deadlines and hence is more efficient in scheduling the tasks of scientific workflow compared to EMMA scheduling algorithm.

B. Number of tasks Vs MIPS of Virtual machine

The performance of execution of workflow is enhanced if a virtual machine with higher processing power (MIPS) executes more number of tasks. To measure the performance of algorithm from this perspective the Montage_100 is executed in the simulation environment and the number of tasks executed by each virtual machine is analyzed. Table III shows the data for this empirical study. The fig.3 represents the graph corresponding to the data in Table III.

It can be analyzed from the graph that in the RBTSA algorithm the virtual machine with low MIPS i.e. processing power and CPU memory i.e., RAM executes less number of tasks and it increase linearly with the increase in processing power and memory of virtual machine. This task distribution on virtual machines achieves better

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

performance because tasks spend less time waiting in the waiting queue to be picked by the scheduler to be executed. This balances the load better among the virtual machines of simulation environment. On the contrary to the RBTSA algorithm, in EMMA the virtual machine with low MIPS and RAM executes more tasks and hence the waiting time of tasks increase which degrades the performance of workflow execution. From the preceding analysis of empirical data; it can be concluded that as compared to EMMA the RBTSA scheduling algorithm improves the performance of workflow execution by better distribution of tasks on the virtual machines. In the same manner Number of Tasks Vs MIPS of Virtual Machine between Montage_25 and Montage_50 can also be tabulated and graphically represented. Thus concluding improved performance of workflow execution between EMM and RBTSA Algorithms too.

TABLE IV. EMMA VIS-À-VIS RBTSA: NUMBER OF TASKS PERFORMED VS MIPS OF VIRTUAL MACHINE

Number of Tasks		
MIPS and RAM of Virtual Machine	*Enhanced Max-Min Algorithm*	*Rank Based Task Scheduling Algorithm*
1100 , 256	11	7
1200, 256	9	7
1300, 256	7	8
1400, 512	8	8
1500, 512	9	9
1600, 512	10	10
1700, 1024	12	10
1800, 1024	10	11
1900, 1024	12	12
2000, 2048	13	19

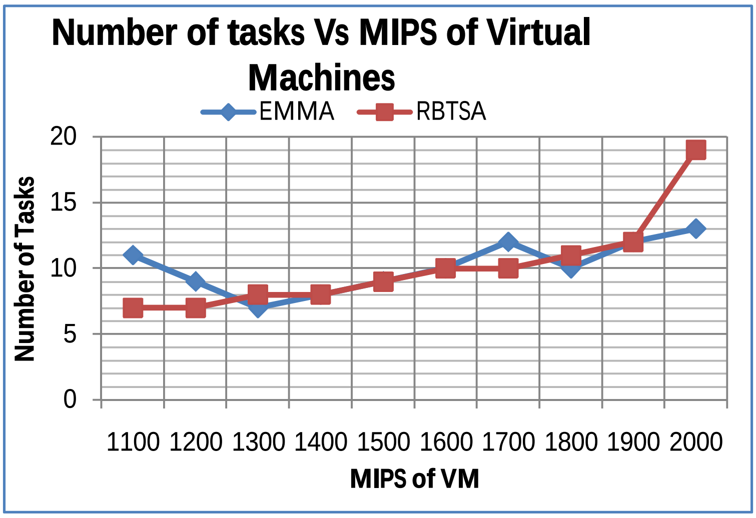

Fig.3. EMMA vis-à-vis RBTSA: Number of Tasks Vs MIPS of Virtual Machine

C. Utilization of Virtual Machine

An efficient scheduling algorithm should increase the utilization of virtual machine resources allocated to it to execute the tasks of scientific workflows. The utilization of virtual machines may be defined as the ratio of total execution time of virtual machine to the total execution time of the workflow. In this dissertation work following formula is used to calculate the utilization of virtual machines (Salehi et al. in 2014) [9].

$$\textit{Percentage Utilization of VM} = \left({}^{\textit{VM Execution Time}}/_{\textit{Workflow Total Execution Time}} \right) \times 100$$

To calculate the utilization of virtual machine the total execution time of all the tasks executed by the virtual machine is calculated and a ratio with the total execution time of workflow is taken. The simulation data for this experiment corresponding to scientific workflow Montage_100 is shown the Table IV. The fig.4 represents the graph corresponding to the data in Table IV.

It can be analyzed from the graph that when the scientific workflow is scheduled with the RBTSA algorithm the utilization of each of the virtual machine increases. The virtual machine utilization in the case of RBTSA algorithm is increased as compared to EMMA algorithm. It can also be observed from the graph that in case of EMMA algorithm the virtual machine whose MIPS is minimum i.e., VM with ID 0 has maximum utilization while the VM whose MIPS is maximum i.e., VM with ID 9 has utilization minimum. It menas that EMMA algorithm s not utilizing the processing power of virtua machine resources efficiently. While in the proposed RBTSA algorithm VM with maximum MIPS i.e., VM with ID 9 has maximum utilization among all the virtual machines. Thus, it can be concluded that the RBTSA algorithm achieves better utilization of resources as compared to EMMA algorithm. In the same manner Virtual Machine Utilization between Montage_25 and Montage_50 can also be tabulated and graphically represented. Thus concluding improved performance of workflow execution between EMMA and RBTSA Algorithms too.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

TABLE V. EMMA VIS-À-VIS RBTSA: VIRTUAL MACHINE UTILIZATION

Utilization		
Virtual Machine ID	***Enhanced Max-Min Algorithm(%)***	***Rank Based Task Scheduling Algorithm(%)***
0	95.85	75.83
1	59.74	69.19
2	62.35	72.58
3	58.22	67.17
4	63.27	73.25
5	64.57	74.37
6	66.98	70.68
7	63.49	73.91
8	60.35	75.02
9	62.01	91.26

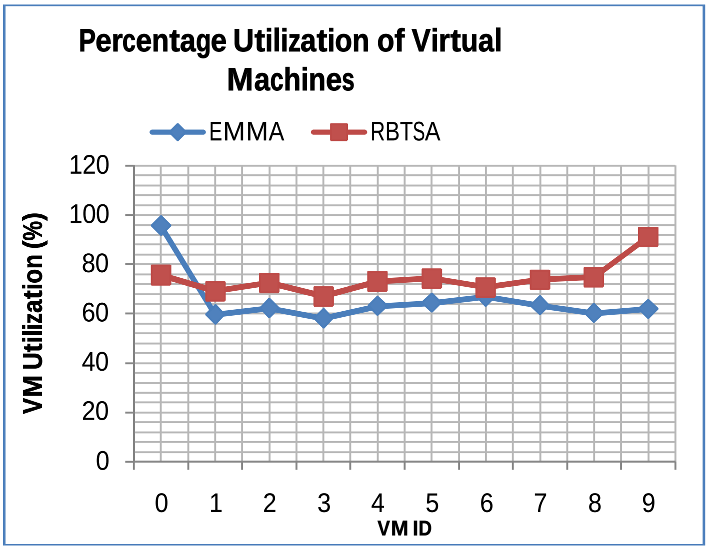

Fig.4. EMMA vis-à-vis RBTSA: Virtual Machine Utilization

VII. CONCLUSION

Scientific workflows are used to model scientific application of various domains like astronomy, physics, and bioinformatics. Scientific Workflows have huge computation and storage demand for their successful execution and hence are executed in parallel distributed environments. Infrastructure as a service model of cloud computing is an ideal platform for executing large scientific workflows because of its elasticity, scalability and virtually unlimited supply of computing and storage resources as required by scientific workflows. This paper focuses on optimizing the task scheduling of scientific workflows on IaaS cloud computing environment. The main goal of this paper is to design a scheduling algorithm for scientific workflows in IaaS cloud environment to minimize the makespan and increase the resource utilization.

The performance of the designed algorithm is evaluated on the basis of performance metrics of makespan of workflow, MIPS of virtual machines and utilization of virtual machines. The results of comparison of performance of the '**R**ank-**B**ased **T**ask **S**cheduling **A**lgorithm' (RBTSA) scheduling algorithm with the 'Enhanced Max-Min' (EMMA) scheduling algorithm shows that the RBTSA scheduling algorithm reduces the makespan of scientific workflow, have good task distribution on the virtual machines, reduces the execution time of individual workflow tasks and increases the utilization of virtual machine of the IaaS cloud computing environment.

REFERENCES

1. Juve, Gideon, and Ewa Deelman. "Scientific workflows in the cloud." Grids, Clouds and Virtualization. Springer London, 2011, pp. 71-91.
2. H. Topcuoglu, S.Haririand M. Wu, "Performance Effective and Low-Complexity Task Scheduling For Heterogeneous Computing", IEEE Transactions on Parallel and Distributed Systems, Vol. 3,2002, pp. 260-275.
3. Lin, C. and Lu, S., 2011, July. "Scheduling scientific workflows elastically for cloud computing", IEEE, 2011, pp. 746-747
4. Byun, Eun-Kyu, Yang-Suk Kee, Jin-Soo Kim, and Seungryoul Maeng. "Cost optimized provisioning of elastic resources for application workflows." Future Generation Computer Systems, 2011, pp. 1011-1026
5. Mao, Ming, and Marty Humphrey. "Auto-scaling to minimize cost and meet application deadlines in cloud workflows." ACM 2011 [International Conference for High Performance Computing, Networking, Storage and Analysis] p. 49.
6. Malawski, Maciej, Gideon Juve, Ewa Deelman, and Jarek Nabrzyski. "Cost-and deadline-constrained provisioning for scientific workflow ensembles in iaas clouds." IEEE Computer Society Press 2012 [In Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis] p. 22.
7. Bhoi, Upendra, and Purvi N. Ramanuj. "Enhanced Max-min task scheduling algorithm in cloud computing." International Journal of Application or Innovation in Engineering and Management, 2013, pp. 259-264.
8. Soni, G. and Kalra, M., 2014, February. A novel approach for load balancing in cloud data center. IEEE In *Advance Computing Conference (IACC), IEEE International,* 2014, pp. 807-812

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

9. Salehi, M.A., Javadi, B. and Buyya, R., "Resource provisioning based on preempting virtual machines in distributed systems." *Concurrency and Computation: Practice and Experience*, 2014, pp.412-433.
10. Juve, G., Chervenak, A., Deelman, E., Bharathi, S., Mehta, G. and Vahi, K., "Characterizing and profiling scientific workflows." *Future Generation Computer Systems*, 2013, pp.682-692.