Resource-Aware Load Balancing Scheme using Multi-objective Optimization in Cloud Computing

Kavita Rana , Vikas Zandu
Swami Vivekanand Institute of Engineering and Technology
Department of Computer Science and Engineering
Chandigarh-Patiala Highway, Rajpura
Kavita.rana107@gmail.com

Abstract : Cloud computing is a service based, on-demand, pay per use model consisting of an inter-connected and virtualizes resources delivered over internet. In cloud computing, usually there are number of jobs that need to be executed with the available resources to achieve optimal performance, least possible total time for completion, shortest response time, and efficient utilization of resources etc. Hence, job scheduling is the most important concern that aims to ensure that use's requirement are properly and correctly satisfied by cloud infrastructure. Most of the task scheduling algorithm developed in cloud computing target single criteria which fails to provide efficient resource utilization. So, to enhance the system performance and increase resource utilization it is must to consider multiple criteria. This paper proposes a multi- objective scheduling algorithm that considers wide variety of attributes in cloud environment. The paper aims to improve the performance by reducing the load of a virtual machine (VM) by using Load Balancing Method. Finally, it optimizes the resource utilization by using Resource Aware Scheduling Algorithm involving combination of Min- max and Max-Min strategy.

Keywords: VM, QoS, Non- dominated sorting, Multi-objective optimization, RASA Algorithm

I. INTRODUCTION

A rapid change has been observed in the dispersed computing with time. Initially beginning with desktop computing, then grid computing and further moving into cloud computing has completely changed the whole computing definition. Since last few years, cloud computing has been widely adopted into business institutions, research institutions, industries and in academics. The rapid development of cloud computing has brought a bright prospect and more economic benefits to the commercial industries. Cloud Computing is intended to enable the computing across largely and diverse resources and dynamic stability. Cloud computing is a computing method that uses internet to provide the user with infrastructure, software and other IT services as per the user's demand with minimum expenditure in minimum time effort.In this computing model, users use the needed services without being bothered to know how these services are delivered or where services are hosted. With the increasing number of users in cloud computing, the volume of data is also expanding. The resource demands for different jobs keep on fluctuating over time. One of the major contribution of cloud computing is to avail all the resources at one place in the form a cluster and to perform the resource allocation based on request performed by different users.

With it, the user does not need to buy the software, but the rent service business operation computing application infrastructure on the cloud. This classification enables the users to easily understand and choose the appropriate and suitable cloud services compatible with their business needs [1].

Task Schedule is an NP-hard problem. There are generally two levels of Task scheduling in cloud computing, first level is user level that schedule task between service provider and user while the second level is system level that schedule management resource within data centre.

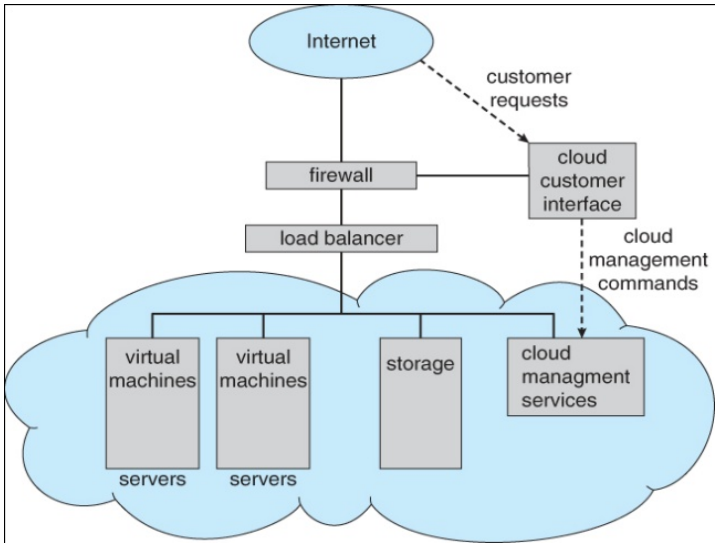


Figure 1: Job scheduling in cloud

Numerous algorithms have been proposed and implemented till date such as First Come First Serve, Data-aware algorithm, Min-Min algorithm, Round-Robin algorithm etc., but the optimized scheduling of the individual tasks in cloud is still an issue to solve. Many of the task scheduling algorithms in cloud computing use a single criteria which does not define efficient resource utilization.

II. REVIEWED WORK

There are numerous algorithms proposed by scholars for scheduling that provide better resource scheduling in cloud environment. Scheduling is assigning fundamentally number of resources to the tasks in a manner that there will be maximum utilization of resource, minimum time for total processing and minimum time of waiting. In this section study and review of various load balancing and scheduling algorithms of various authors has been done on the basis of various performance parameters like bandwidth, cost, deadline, execution time, make span, priority, reliability, scalability, time, and throughput.

A.I.Awad et.al [1] addresses the dynamic Multi-objective task scheduling in Cloud Computing using an optimized Particle Swarm optimization and presents efficient allocation of tasks to available virtual machine in user level base on different parameters. The paper proposes a mathematical model multi-objective Load Balancing Mutation particle swarm optimization (MLBMPSO) to schedule and allocate tasks to resource. The proposed algorithm considers two objective functions to minimize round trip time and total cost. First objective function is to minimize Expected Round Trip Time (ERTTij) of task i in vm_j and the second objective function is to minimize Expected Total Cost (ETCij) of task i in vm_j . The weighted sum approach is used to solve multi-objective problem.

A multi-objective task scheduling algorithm is proposed by Ekta S. Mathukiya et.al [2] that performs non-dominated sorting for ordering of tasks. Task size, make span, and deadline are considered as the criteria in the proposed algorithm. The task's priority here is allocated in accordance with the QoS value and QoS are assigned to the VMs on the basis of their Millions Instructions per Second (MIPS) value. The list of VMs possessed by cloud broker is updated after fixed time interval. Based on MIPS the list of VMs is sorted in descending order starting from high QoS VM to low QoS VM and non- dominated sorting is performed on the list to generate non-dominated task's set. These tasks are bounded to VMs sequentially and the process of allocation is repeated for all tasks.

Atul Vikas Lakra et.al [3] explained the underutilization of cloud resources due to poor scheduling of tasks in cloud and came up with a multi- objective task scheduling algorithm in this regard to optimize the throughput, cut the execution expenses at the same time keeping the SLAs intact. The proposed scheduling technique prioritizes the tasks according to the QoS such that low QoS valued tasks are assigned higher priority and vice versa. Cloud broker assigns QoS to VMs on receiving list of VMs from cloud service providers.

Sandeep Singh Brar et.al [4], came up with a Max-Min algorithm for optimizing workflow scheduling. In this approach, different types of workflows are assigned as an input. The Four wide used scientific workflows used as input are: Montage, Cyber Shake, and SIPHT. Then, different scheduling algorithms like FCFS, Cyber Shake, Sipht and Max Min are applied and executed. The results showed that Max- Min produced different results for different inputs.

A modification of Improved Max-min task scheduling algorithm has been proposed by Upendra Bhoi [5]. The proposed enhanced version of Max-min also includes the expected execution time as a selection basis just like the old improved Max-Min method but there is a slight difference in their working that differentiates them vividly.

One such effort is also done by Navdeep Kaur et.al [6] to optimize conventional Max- Min algorithm. It involves the principle of sorting jobs (cloudlets) based on completion time of cloudlets. The traditional Max-Min allocates the longer jobs to better resources and reduces the overall task runtime. So, the overall focus is on time parameter, but factor like storage capacity lacks the consideration. Unlike conventional Max- min algorithm, the improved algorithm works on multiple parameters. In addition to the completion time parameter, the improved algorithms also considers the storage requirement into focus as this algorithm is being optimized for data intensive specific application. The new algorithm designs weight to each parameters based on its proportion of contributing as a resource for excluding a particular jobs.

The completion time statistics are based on a particular virtual machine.

$P=T^{\alpha} / R^{\beta}$, where T is the payload storage rate potentially achievable for a particular virtual machine, R denotes the heuristic average of payload storage of a particular virtual machine and α, β denotes the time the fairness schedule. The balance between serving best virtual machines can be achieved by changing the α, β values. The completion time can be calculated along with equation:

$$P=T^{\alpha} /R^{\beta} +CT$$

Hong Sun et.al [7] emphasized on task scheduling algorithms based on comprehensive QoS. The paper considers the new Berger's Model under dynamic cloud computing environment and relates to benefit- fairness algorithm. New Berger game theory model apply the theory of social distribution and game theory on task scheduling in cloud computing environment. In cloud ,task scheduling is achieved by assigning independent task to m virtual machine resource to fully utilize the resources in minimum finishing time. If FTi is the finish time of task i, then span is defined as

$$FT_{max}= \max(FTi, i=1,2, 3…..n)$$

III. JOB SCHEDULING MODEL IN CLOUD

When more than one processes starts to run in a computer simultaneously, competing with each other for the CPU resources then Operating System has to decide in that situation that which process to run next. This process of taking decision that which process will get resources is called Scheduling. Similarly, in cloud computing too, task scheduling is a vital part. It is achieved by scheduling subtasks of workflows by arranging the tasks in queue and providing them with the suitable resources that can execute these tasks. Cloud Computing has many features like virtualization and flexibility. With the help of technology of virtualization, all the resources that are physically available can be made virtualized and transparent for users [4].

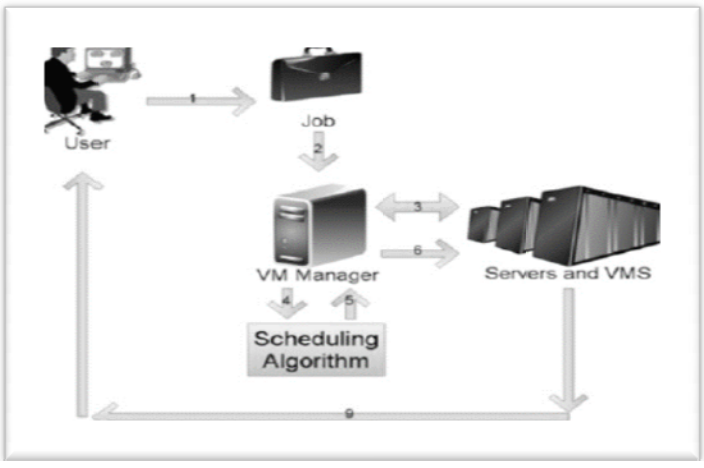


Figure.2. Architectural Framework

The main motive of task scheduling is to attain better cloud performance in terms of better throughput, load balancing among resources, quality of service (QoS), economic feasibility and the optimal operation time. Task scheduling can be viewed from two directions- from the cloud resources users' view, users have to identify which cloud computing resource can meet their job QoS requirements for computing and how much amount to be paid for the cloud computing resources.

Load Balancing is a technique which divides the workload across multiple computing resources such as computers, hard drives and network. In this fair allocation of resources of client request tried to achieve in the best way to ensure proper utilization of resource consumption. In a Cloud computing environment different load balancing scheduling exists among which first, is the Batch mode heuristic scheduling algorithm where jobs are queued in a set and collected as batches as they arrive in the system after which they get started after a fixed time period. Example: First Come First Serve (FCFS), Min-Max algorithm, Min-Min algorithm and Round Robin (RR) algorithm. Second, one is On-line mode heuristic scheduling here jobs are scheduled individually as they arrive in the system.

IV. PROBLEM FORMULATION

The primary goal of Cloud Computing is to provide efficient access to remote and geographically distributed resources with the help of Virtualization in Infrastructure as a Service (IaaS). Various virtual machines (VM) are needed as per the requirement and cloud provider provides these services as per the Service Level Agreement (SLA) to ensure QoS. For managing enormous amount of VM requests, the cloud providers require an efficient resource scheduling algorithm. In the existing approach, cloudlets were being scheduled according to First Come First Serve algorithm after sorting them according to Non-dominated sort technique, thereby mapping the tasks to the virtual machines, according to their arrival times. But the concept of balancing the load of virtual machines was not there, which was a big drawback to the existing system.

The paper aims to study and analyze the processing time of low level scheduling algorithms and to develop a multi- objective task scheduling using quality of service parameters of resource nodes. It also attempts to improve the performance of CPU, memory and network operations by reducing the load of a virtual machine (VM) by using Load Balancing Method. Finally, it optimizes the resource utilization by using Resource Aware Scheduling Algorithm.

V. PROPOSED WORK

In the proposed architecture, after performing the Non-dominated sort, the concept of Task-Based Load Balancing is being taken into consideration. Non-dominated sorting is used to solve the multi-objective problems which target multiple objective functions. In the proposed work, the main goal is to minimize the processing time of a task. As balancing the load of machines helps in distributing the work load of a task onto multiple computers, and it also provides reliability in particular tasks.

1) Min-Min

The Min-Min algorithm is simple many cloud scheduling algorithms are based on it. It begins with a set S of all unmapped tasks. Then the resource R which has the minimum completion time for all tasks is found. Next, the task T with the minimum size is selected and assigned to the corresponding resource R. Last, the task T is removed from set S and the same procedure is repeated by Min-Min until all tasks are assigned. This algorithm fails to utilize the resources efficiently which lead to a load imbalance. And without use-priority aware during scheduling, the VIP users are not guaranteed with better services which lead to VIP users' dissatisfaction. For a set of i tasks (T1, T2, T3… Ti) to be scheduled onto j available resources (R1, R2, R3 …Rj).

2) Max-Min

The Max-min algorithm is commonly used in distributed environment and it begins with a set of unscheduled tasks. Then expected execution matrix and expected completion time of each task on the available resources is calculated. Further the task with overall maximum expected completion time is chosen and assigned to the resource with minimum overall execution time. Finally recently scheduled task is removed from the meta-tasks set, updating all calculated times, then repeating until meta-tasks set become empty. Max-min algorithm losses some of its major advantages as load balance between available resources in small distributed system configuration and small total completion time for all submitted tasks in large scale distributed environment.

3) RASA

RASA is a combined approach of Max-Min and Min-Max algorithms that performs efficient resource utilization, further optimizing the resources in terms of accuracy and efficiency. This algorithm uses both the algorithms Max- min and min- max to use the pros of both and eliminate the drawbacks. To realize this optimization, it first estimates the completion time of the tasks on each of the available cloud resources and then

applies the Max-Min and Min-Min algorithms, alternatively. Small tasks are executed by using Min-Min strategy before the large ones. Max- Min strategy comes into play to avoid delays in the execution of large tasks, to support concurrency in the execution of large and small tasks.

The key objective of the proposed algorithm is to obtain better results than existing algorithm to reduce the execution time of jobs.

The evaluation parameters considered for performance evaluation are:

a) Average Waiting Time
b) Total Processing time
c) Total Processing cost

VI. METHODOLOGY

1. To develop multi objective task scheduling. In this, tasks are sorted using non-dominating sort and then the sorted tasks are mapped to virtual machines in order to optimize the processing and average waiting time.

2. Load balance of a virtual machines is achieved by first mapping tasks to VM's and then all the VM to host resources, using the Task-Based System Load balancing method. This algorithm ensures the system load balancing through only transferring extra tasks from an overloaded VM instead of migrating the entire overloaded VM. The loads are formulated as:

By taking these parameters for calculating the load, we will get the better balanced load virtual machine for the task to be performed on cloud.

$$\mathbf{Fitness1_{ij}} = \frac{\sum_{i=1}^{n} \mathbf{cloudlet\_length_{ij}}}{\mathbf{Vm_j\_mips}}$$

where Vm_j_mips is defined by millions of instructions per second for each processor of Vm_j, n is the total no of scout foragers, fit_{ij} defines the fitness function of machines (i) for Vm_j or say capacity of Vm_j with i^{th}virtual machine number, cloudlet_length is defined as the task length that has been submitted to Vm_j.

The virtual machine (Vm_j) capacity is being calculated using the following parameters

$$\mathbf{Capacity\_Vm_j} = \mathbf{Vm_j\_cpu} * \mathbf{Vm_j\_size} + \mathbf{Vm_j\_bandwidth}$$

$$\mathbf{Fitness2_{ij}} = \frac{\sum_{i=1}^{n} \mathbf{cloudlet\_length_{ij}}}{\mathbf{Vm_j\_Size}}$$

VMload =Mean of (Fitness1+Capacity_vm+Fitness2)

LowloadedVM = maxcap- (load/capacity)

OverloadedVM = (load/capacity) – maxcap

3. Resource utilization is achieved by using RASA a combined approach of Max-Min and Min-Max algorithms) to further optimize the resources in terms of accuracy and efficiency. To achieve this, it first estimates the completion time of the tasks on each of the available cloud resources and then applies the Max-Min and Min-Max algorithms, alternatively. Small tasks are executed by using Min-Max strategy before the large ones. To avoid delays in the execution of large tasks to support concurrency in the execution of large and small tasks Max-Min strategy is used.

4. Scheduling Strategies Considering Parameters are:

Parameters	Formulation
Total Processing Time	CloudletLength / vmMips*vmNumberOfPes
Total Processing cost	characteristics.getCostPerMem * vm.getRam
Average Waiting Time	cloudlet.WaitingTime

VII.FLOWCHART

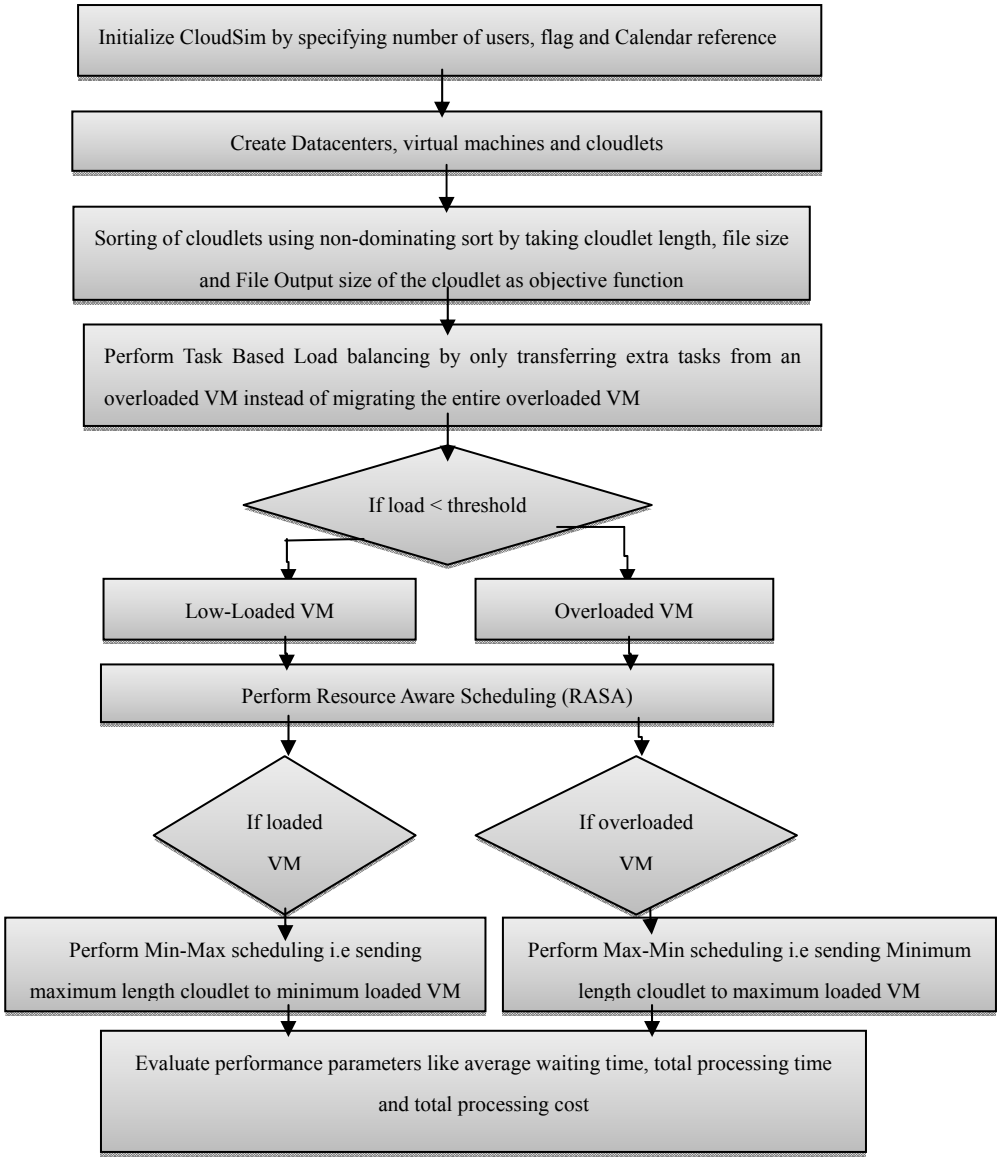


Figure 3 : Detailed Flow diagram for the proposed technique

VIII. RESULTS AND DICUSSIONS

This section presents the simulation results of the proposed methodology implemented with the help of Cloudsim and Net beans IDE 8.0. The proposed scheme is evaluated using different scenarios where varying number of jobs are assigned to virtual machines. In different scenarios the load balancing capacity is analyzed. Following table presents the summarized view of the different scenarios taken into consideration along with the parameters of performance analysis. Many VMs and tasks with different task size have been created. Task size ranges from 100 to 500.

Table 1: Showing the end result analysis of the proposed and the previous scheduling model in different scenarios

Virtual Cloud Environment		**Performance Analysis**					
Virtual Machines	Jobs	Total Processing Time (ms)		Total Processing cost ($)		Average Waiting Time(ms)	
		Previous	Proposed	Previous	Proposed	Previous	Proposed
10	100	463899.418	411078.6237	2725.17	2194.35	0.51082	0.44874
20	200	1761519.5914	1589568.981	6563.264	5075.199	0.53455	0.4814
30	300	3750152.908	3563512.389	11514.2849	7625.685	0.50084	0.47642
40	400	6826202.1468	6173982.811	14701.026	11094.626	0.50919	0.4743
50	500	1.03E+07	1.00E+07	20265.911	16935.242	0.52104	0.2449

The performance analysis is further illustrated graphically:

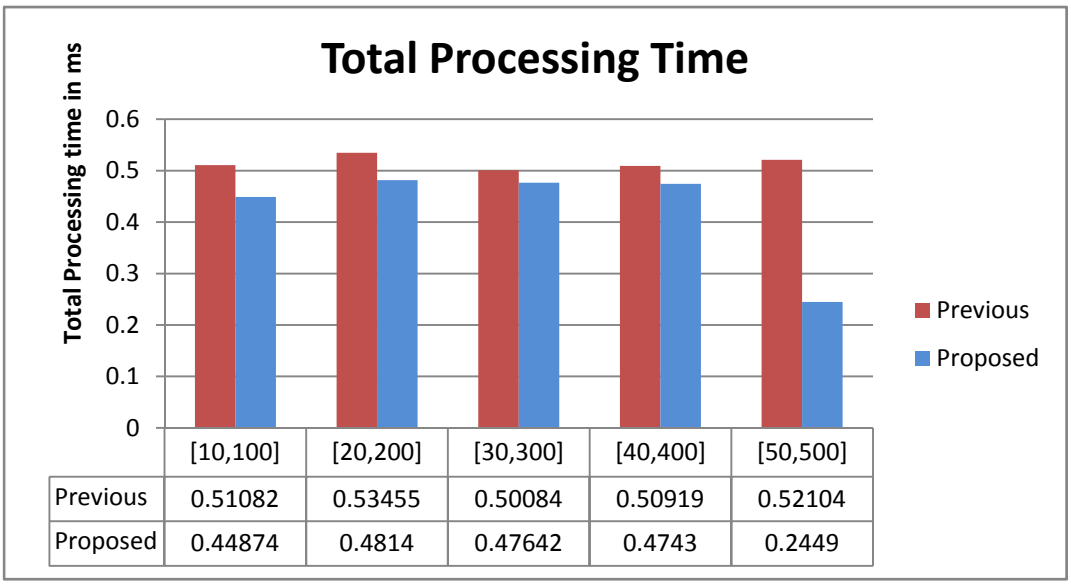


Figure 4: Graphical Depiction of Processing Time of Previous and Proposed model in different scenarios

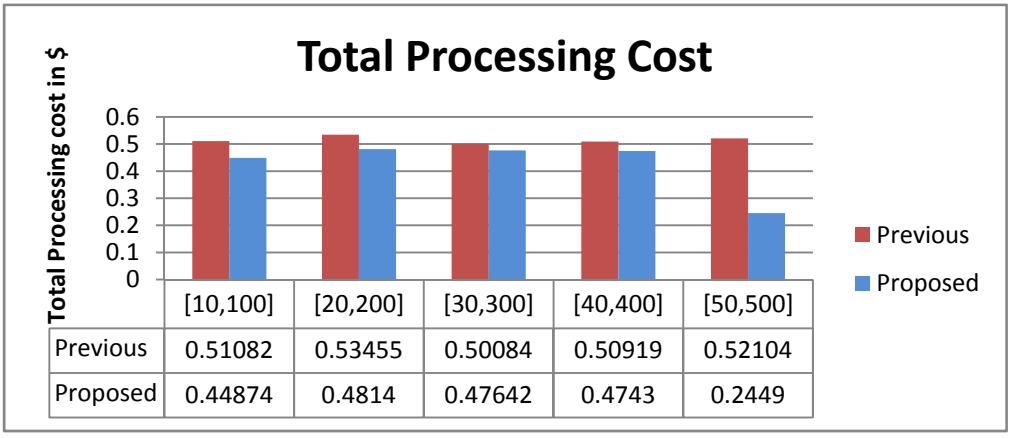
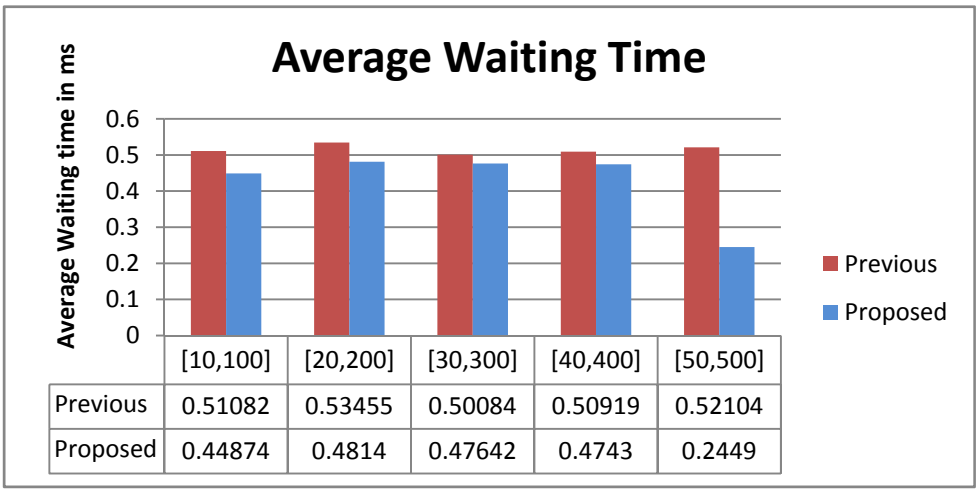


Figure 5: Graphical Depiction of Processing Cost of Previous and Proposed model in different scenarios



	[10,100]	[20,200]	[30,300]	[40,400]	[50,500]
Previous	0.51082	0.53455	0.50084	0.50919	0.52104
Proposed	0.44874	0.4814	0.47642	0.4743	0.2449

Figure 6: Graphical Depiction for AWT of Previous and Proposed model in different scenarios

IX. CONCLUSION AND FUTURE SCOPE

In this paper, Multi- objective task scheduling is emphasized and to achieve the same a multi- objective scheduling algorithm is proposed considering the user's Quality of Service requirements. The proposed algorithm considers three parameters i.e. Total processing cost, total processing time and average waiting time. Load balancing technique is used to guarantee balancing of load and RASA is used for proper resource utilization. RASA is hybrid approach of Max-Min and Min-Min algorithms. The parameters considered in this research work are tested with various input values and the results depicts that the proposed method achieves better results than the existing method.

The work can be further extended in future aiming to achieve more efficient performance results. The proposed work is using RASA for resource allocation. Also, resource allocation can be performed with the improved versions and evolutionary algorithms with implementation in real time scenario.

REFERENCES

[1] Tarek Z, Zakria M,.Omara F A, "PSO Optimization algorithm for Task Scheduling on The Cloud Computing Environment", ISSN 2277-3061, International Journal of Computers And Technology Vol. 13, No. 9
[2] Mathukiya E.S, Gohel P.V, "Efficient Qos Based Tasks Scheduling using Multi-Objective Optimization for Cloud Computing", International Journal of Innovative Research in Computer and Communication Engineering Vol. 3, Issue 8, August 2015
[3] Lakra.A.V,Yadav.D.K," Multi-Objective Tasks Scheduling Algorithm for Cloud Computing Throughput Optimization ," International Conference on Intelligent Computing , Communication & Convergence , 2015
[4] [4] Brar S. S., Rao S., "Optimizing Workflow Scheduling using Max-Min Algorithm in Cloud Environment", International Journal of Computer Applications (0975 – 8887)Volume 124 – No.4, August 2015
[5] Bhoi U, Ramanuj P.N, "Enhanced Max-min Task Scheduling Algorithm in Cloud Computing", International Journal of Application or Innovation in Engineering and Management, Volume 2, Issue 4, April 2013
[6] Kaur N., Kaur K, "Improved Max-Min Scheduling Algorithm", IOSR Journal of Computer Engineering (IOSR-JCE)e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 3, Ver. 1 (May – Jun. 2015), PP 42-49
Sun H, Chen S.P, Jin C, Guo K, "Research and Simulation of Task Scheduling Algorithm in Cloud Computing", TELKOMNIKA, Vol.11, No.11, November 2013, pp. 6664~6672e-ISSN: 2087-278X
[7] Chen H, Wang F, Dr Helian N, Akanmu G, "User-Priority Guided Min-Min Scheduling Algorithm for Load Balancing in Cloud Computing"
[8] Etminani K, Naghibzadeh M, "A Min-Min Max-Min selective algorithm for grid task scheduling", DOI: 10.1109/CANET.2007.4401694 · Source: IEEE Xplore
[9] Liu J, Luo X. G, Zhang X.M, Zhang F and Li B.N, "Job Scheduling Model for Cloud Computing Based on Multi-Objective Genetic Algorithm", IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 3, January 2013
[10] Pandey S, Wu L, Guru S M, Buyya1R, "A Particle Swarm Optimization-based Heuristic for Scheduling Work flow Applications in Cloud Computing Environments"
[11] Tripathy L, Patra R.R, "Scheduling In Cloud Computing", International Journal on Cloud Computing: Services and Architecture (IJCCSA) Vol. 4, No. 5, October 2014
[12] Agarwal A, Jain S, "Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment", International Journal of Computer Trends and Technology (IJCTT) – volume 9 number 7– Mar 2014
[13] Ghanbaria S, Othman M, "A Priority based Job Scheduling Algorithm in Cloud Computing", International Conference on Advances Science and Contemporary Engineering 2012(ICASCE 2012)
[14] Vijayalakshmi M, Muthusamy K, "An Efficient Study of Job Scheduling Algorithms with ACO in Cloud Computing Environment", Volume 3, Special Issue 3, March 20142014 International Conference on Innovations in Engineering and Technology

[15] B. Kanani and B. Maniyar," Review on Max-min Task Scheduling Algorithm for Cloud Computing ," Journal of Emerging Technologies and Innovative Research ,vol. 2 , pp. 781-784 ,2015
[16] A.Bhatia and R.Sharma, "An Analysis Report of Workflow Scheduling Algorithm for Cloud Environment," International Journal of Computer Applications, vol.119, pp. 21-25 ,2015
[17] Kowsik , K. RajaKumari ," A Comparative Study on Various Scheduling Algorithms in Cloud Environment , " International Journal of Innovative Research in Computer and Communication Engineering , vol.2 , 2014