PBVMLBA: Priority Based Virtual Machine Load Balancing Algorithm for Cloud Computing

D. Suresh Kumar[1] and Dr. E. George Dharma Prakash Raj[2]

[1] Research Scholar , School of Computer Science ,Engineering and Applications, Bharathidasan University,
Tiruchirappalli, Tamil Nadu, Indian

[2] Assistant Professor, School of Computer Science ,Engineering and Applications, Bharathidasan University,
Tiruchirappalli, Tamil Nadu, Indian

[1]dskumar07@gmail.com, [2]georgeprakashraj@yahoo.com

ABSTRACT

Cloud computing is a fast growing area in computing research and industry today that offers vast benefits such as reduced time, unlimited computing power and flexible computing capabilities. It is a model that provides an on-demand network access to a shared pool of computing resources It comprises a huge number of concepts primarily Load Balancing, Scheduling, etc. The number of users in cloud computing is growing in an exponential rate at every day. The number users access cloud server by sending enormous request for many application which lead to high load on cloud server. To reducing the heavy load on server, the virtual machines are allocated for resources based on priority. Allocating the resources on virtual machines based on priority achieves the better response time and processing time. The main aim of the system is to minimize the makespan and to maximize the resource utilization and user satisfaction. This proposed algorithm is implemented using Microsoft azure real time cloud and Microsoft azure as a deployment tool.

Keywords: *Cloud Computing, Load balancing, Virtual Machine.*

1. INTRODUCTION

Cloud computing a relatively new technology, which has been gaining immense popularity over the last few years where user can rent software, hardware, infrastructure and computational recourse as per user basis. It is an entirely internet-based approach where all the applications and files are hosted on a cloud which consists of thousands of computers interlinked together in a complex manner. These are emerging distributed systems which follows a “pay as you use” model.[2] The number of cloud users has been growing exponentially and apparently scheduling of virtual machines in the cloud becomes an important issue to analyze. Users can submit their jobs into cloud for computational processing or leave their data in cloud for storage. Cloud scheduler must be able to schedule the task properly.[1]Load balancing is use to balance load between multiple resources to get minimum makespan, improve performance, reduce response time and optimal resource utilization.

This paper has been orchestrated as follows. Section 1 gives introduction on cloud computing. Section 2 specifies the literature study relevant to this research focus. Section 3 comes out the proposed load balancing algorithm. Section 4 gives out the implementation of this study using tools. Section 5 deals with the conclusion and future directions.

2. RELATED WORK

In cloud computing, load balancing is a most important issue. Many approaches had been proposed to handle the load balancing issues in cloud computing systems. All these works seems designed to improve the process of distributing the workload among cloud nodes and try to reach optimal resource utilization, minimum data processing time, minimum average response time, and overload avoidance. .

The distributed dynamic priority based algorithm is used for balancing the load on instances effectively and to improve the system consistency, minimum response time and increase the throughput. Allocating the resources on virtual machines based on priority achieves the better response time and processing time. Load balancing ensures all instances in a node in the networks to do the equal amount of work at any instant of time. Priority based resource provision to improve the utilization of resources and reducing response time of cloud services.[1]

The VMARLB virtual machine based random load balancing algorithm assign the task to all available machines and balance the system using efficient load balancing technique. The focus is to allocate the request based on random allocation and minimum completion time. The algorithm compare with different scenarios and min-min algorithm [3].

This algorithm, sort the tasks by their running deadline, after specifying the number of available virtual machines, map task sets to the virtual machines by their running deadline with shuffled frog leaping algorithm. Therefore, tasks by lower running deadline time have higher priority to allocation. As a result, the rate of user satisfaction will increase. The algorithm compared with existing task scheduling algorithms in different scenarios, and the results are presented. Simulation results with Cloudsim framework display that the proposed algorithm has better results than previous algorithms in the makespan time, average response time and load imbalance degree parameters.[4]

A new Priority based job scheduling algorithm (PJSC) in cloud computing. This algorithm is based on multiple criteria decision making model and mathematical model called Analytical Hierarchy Process (AHP) provides scheduling with minimum makespan, high throughput and reasonable complexity.[5]

A VM scheduling approach which assign the priority for each VM based on their cost and then place the VM first that has the highest priority. To create the cloud environment for measuring the performance of the proposed approach and compare with the existing VM scheduling approach. Experiments result shows that proposed approach minimize the energy consumption and simulation time [6].

The algorithm classified and grouped all tasks as deadline based and minimum cost based constraints and after dynamic optimization, priority of fairness is applied. Here different priority queue (high, mid, low) is implemented in round-robin fashion as per weights assign to them. Recompile the CloudSim and simulate the proposed algorithm and results of this algorithm is compared with sequential task scheduling and simple constraints (cost and deadline) based task scheduling algorithm. The experimental results indicate that proposed algorithm is, not only beneficial to user and service provider, but also provides better efficiency and fairness at priority level. [7]

A priority based load balancing algorithm, where in execution length of cloudlets is input as the workload to the system and is used to assign priority to the tasks. A comparison of the execution times of cloudlets is also being made using the algorithm and the existing Round Robin load balancing algorithm. [8]

Load balancing is essential to improve the performance significantly, to provide a tendency of fault tolerance in case the system failures, to maintain the system constancy and to accommodate prospect variation in the system. Load balancing is the mechanism to distribute equal workload and a resource across all the nodes i.e. no single node is over loaded and others remains idle. It guarantees that each and every node has equal load. Now day's product has been increased rapidly in the cloud computing environment, an algorithm which performs the load balancing on the basis of product priority.[9]

A systematic review of various priority based job scheduling algorithms is presented. These algorithms have different perspective, working principles etc. The study concludes that all the existing techniques mainly focus on priority of jobs and reduces service response time and improving performance etc. There are many parameters that can be mentioned as factor of scheduling problem to be considered such as load balancing, system throughput, service reliability, service cost, service utilization [10].

Due to increasing demand of cloud computing, the growing number of tasks affects the system load and performance. Scheduling of multitasks with respect SLA (Service Level Agreement) can face serious challenges. In order to overcome this problem as well as provide better quality of service, the tasks have to be scheduled in optimal way. Address the problem of the priority task scheduling through proposing a global strategy over distributed data-center in cloud computing basing on three parameters: tasks deadline, task age and the task length [11].

(HTV Dynamic Load Balancing Algorithm) in which continuous monitoring of the resources are done to know the status of each and every node and queue is maintained in which the weight factor will be stored and update whenever continuous monitoring is done . When request comes, the resources will be allocated from the information present in the queue dynamically to balance the load on nodes. But still there is some problem which needs to be worked out. The problem of time limit of the requests (tasks) that needs to be executed. Some tasks have lower time limit that the other ones. Such task needs to be serviced prior than the other ones.[12]

PA-LBIMM separates the tasks into G1 and G2 groups. The tasks submitted by VIP user's or high priority users are considered as group G1 and tasks submitted by low priority users are considered as group G2. Tasks are scheduled to the resources on the priority basis. Firstly all the tasks in G1, each task is assigned to the VIP category resource by using Min-Min. Then each task in G2 group is assigned to all the resources by using Min-Min. The load balancing function of LBIMM algorithm is executed to load balance all the resources. In this way, an optimal load balanced schedule is generated.[13] and recovery policy applied RPA_LBIMM in which helps the cloud scheduler to reschedule the tasks if a resource fails at the time of execution to achieve the minimum makespan [13].

This paper provides efficient load balancing algorithm for cloud environments, then it combines the merits of divisible load balancing algorithm and weighted round robin algorithm. The simulation result shows that this method is efficient and the values of processing time and response time yields low values compared to other methods. Additionally this method also removes the drawbacks of traditional round robin methods.[14]

The researchers attempt to find newer ways for Workflow scheduling which could work well in the cloud environment. Another important element to be considered about cloud computing is load balancing. This paper recommends firefly criteria for effective fill controlling in reasoning processing. This criterion is based on the travel behavior of the fireflies which go looking for the closest possible maximum alternatives. It employ Firefly algorithm to schedule the jobs and thereby evenly distribute the load and in turn reduce the overall completion time (makespan) [15].

A variety of load balancing algorithms in the cloud computing environment have been surveyed. The most important issues are response time, resource utilization, User satisfaction. The existing algorithms take more response time and less important to optimum utilization of resources. This has become motivation factor to propose the PBVMLBA algorithm. The PBVMLBA distribute the load among all the available resources at the same time to minimize the makespan with the effective utilization of resources and also reduce the time delay.

3. PROPOSED WORK

The proposed PBVMLBA is a load balancing algorithm in which all the allocation and decision of scheduling are completed by a special node called as Load Balancer (LB). This node is responsible for storing knowledge base of entire cloud network and can apply dynamic approach for load balancing. The Data Center Controller (DCC) receives all the requests from the users from all around the world, which is one of the major components of Cloud. Data Center Controller forwards the request to the Load Balancer to assign the request to the available virtual machines. It handles a table which contains the job id of the user request (priority or no priority), completion time of the virtual machine and the state of the virtual machine. Initially, checks the jobs priority, if any priority, allocate the VM and update the status or allocate the VM based on the condition of the completion time of that task is less than to makespan of RPA_LBIMM. To handle further request, this algorithm will search the table and repeat the above procedure until all the tasks get completed.

PROPOSED ALGORITHM (PBVMLBA)

```
Step1: Request from user to DCC.
Step2: DCC forwarded the request to LB if DCC=Null go to stpe8
Step 3: Compute the completion time for all tasks for a VM.
 Step4: Check if the request is priority or not and check the CT < MP.
Step5: If status = idle
Step 6: Allocate the VM and update VM status.
        Otherwise
        Wait for signal until the job gets completed.
Step 7: Repeat step 4 and 9 till some user request exist.
Step 8: if user request complete then stop the allocation process
```

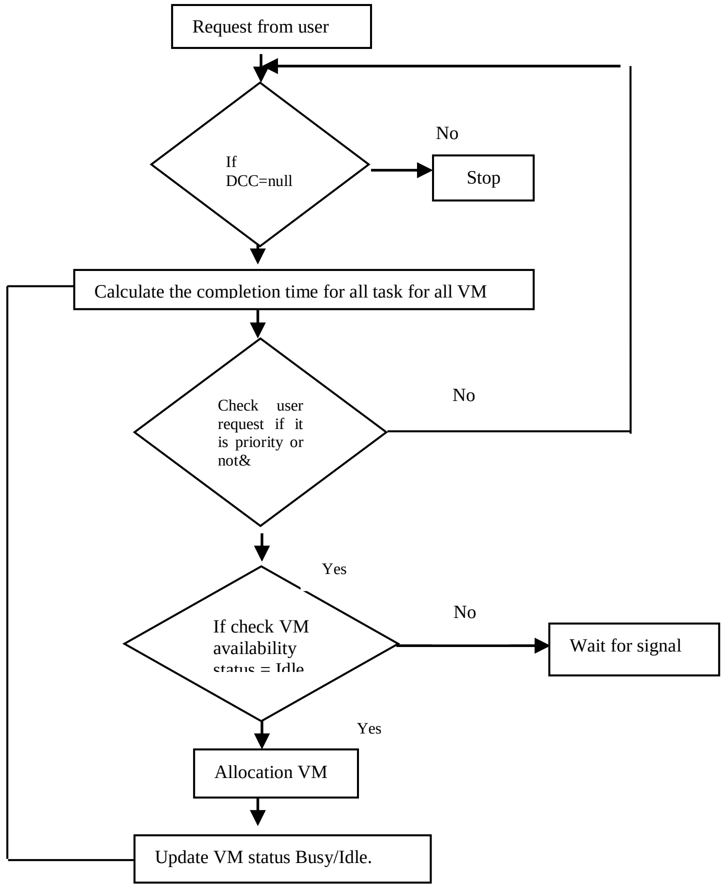
Request from user
If DCC=null
No
Stop
Calculate the completion time for all task for all VM
Check user request if it is priority or not&
No
Yes
If check VM availability status = Idle
No
Wait for signal
Yes
Allocation VM
Update VM status Busy/Idle.

Fig.1. shows the flowchart for Proposed PBVMLBA.

4. RESULT AND DISCUSSION

The proposed algorithm PBVMLBA is implemented using Microsoft azure real time cloud which runs on ASP, SQL database. And Microsoft azure as a deploying tool. Assume we have a setup of three available resources (VM) to which various users can submit their tasks. Suppose five tasks have been submitted by users. Table1 represents the id, size and the user group of each task. Table 2 represents the id, processing speed and service type of each resource, data present in Table3 is the completion time for all task for all VMs.

Table1: Task Parameter

Task ID	Task Size (MB)	User Group
T1	100	Ordinary
T2	150	Ordinary
T3	200	Ordinary
T4	250	Priority
T5	500	Ordinary

Table2: Resource Speed

Resource (VM) Id	Resource (VM) Speed(Mbps)	Type
VM 1	20	Priority
VM 2	16	Ordinary
VM 3	10	Ordinary

Table3: Computed Completion time (CT)

Task	Resource		
	Priority VM 1	VM 2	VM 3
T1	5	6.25	10
T2	7.5	9.375	15
T3	10	12.5	20
T4	12.5	15.625	25
T5	25	31.25	50

.

Based on the results of above table 1,2,3 & 4 the given fig.2 represents (PA_LBIMM) the user priority awared load balance improved min min algorithm produce the makespan 38.375

Table4: Execution time of PA_LBIMM

Task	Resource		
	Priority VM 1	VM 2	VM 3
T4(P)		19.625	
T1		25.875	
T2			15
T3		38.375	
T5	25		

.

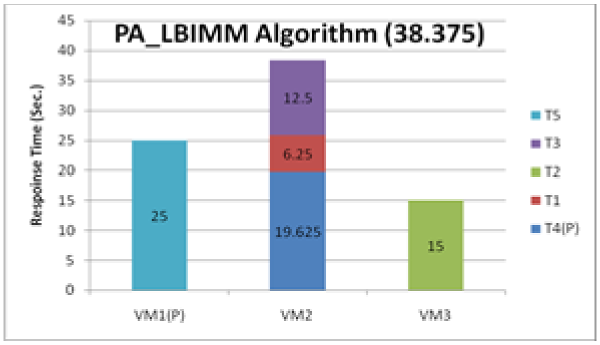

Fig. 2. PA_LBIMM

Table5: Execution time of VMARLB algorithm

Task	Resource		
	Priority(R1)	R(2)	R(3)
T3			20
T1		6.25	
T5	10		
T4		31.25	
T2	7.5		

The results of above table 5 and the fig.:3 represents (VMARLB) the virtual machine based random load balancing algorithm produce the makespan 37.5

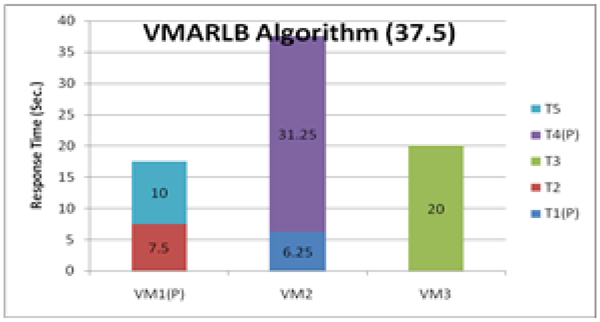

Fig. 3. VMARLB

Table6: Execution time of RPA_LBIMM

Task	Resource		
	Priority(R1)	R(2)	R(3)
T4(P)		19.625	
T2			15
T5	25		
T3	35		
T1			25

The results of above table 6 and the fig.:4 represents (RPA_LBIMM) algorithm produce the makespan 35.

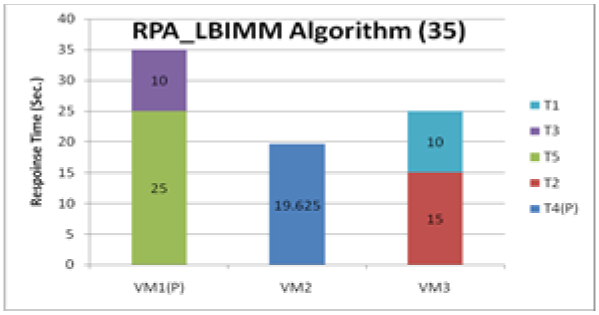

Fig. 4. RPA-LBIMM

Table7: Completion time of Proposed PBVMLBA algorithm

Task	Resource		
	Priority (R1)	R(2)	R(3)
T4 (P)	12.5		
T5		31.25	
T3			20
T2	7.5		
T1			10

.

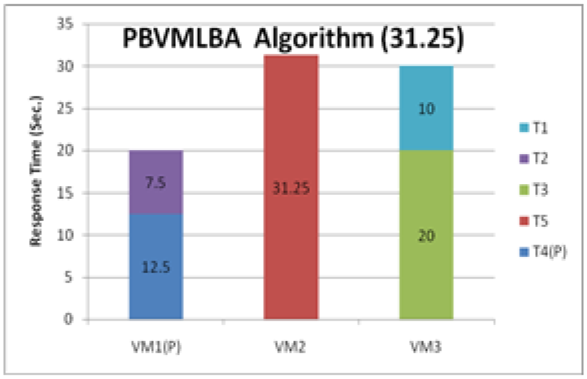

Fig. 5. Proposed PBVMLBA

The Proposed algorithm shows a reduction in execution time of tasks. The PA-LBIMM algorithm execute 5 tasks with time utilization is 38.375 seconds , the VMARLB algorithm execute the same tasks with time utilization is 37.5 , the RPA_LBIMM algorithm executes 5 tasks with time utilization is 35 seconds the proposed algorithm executes the same tasks 31.25 seconds. This shows that the PBVMLBA algorithm used minimum time for execution of the task. This is due to the use of priority based approach which reduces the waiting time and increases the utilization time of the resources. Further, subsequent allocations will be handled based on the idle state of the VMs. This idle state will be known once the VM completes the execution of the job.

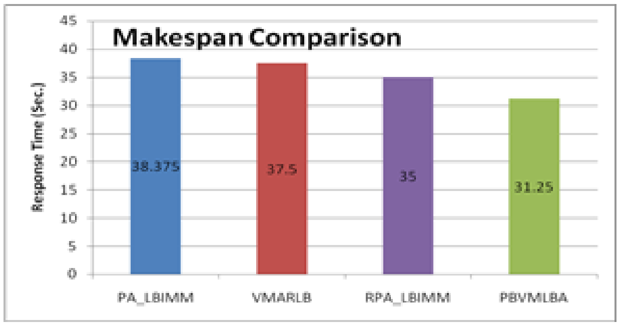

Fig. 6. Makespan Comparison

Based on the above results the following points are concluded.

(i) The resources utilization of PBVMLBA is increased compared with other algorithms.

(ii) Makespan = max (rt_j), Maximum execution time in a node(VM)

Table 8: Makespan

Method	Makespan
PA_LBIMM	38.375
VMARLB	37.5
RPA_LBIMM	35
PBVMLBA	31.25

Average resource utilization (U_a)

$$Ua = \frac{\sum_{i=1}^{N} Ci * 100}{Nm} \qquad [1]$$

N = Number of nodes, m= makespan, C_i = Completion Time

Table9: Resource Utilization

Algorithmas	Resource Utilization (%)
PA_LBIMM	68.07
VMARLB	66.66
RPA_LBIMM	75.83
PBVMLBA	86.66

.

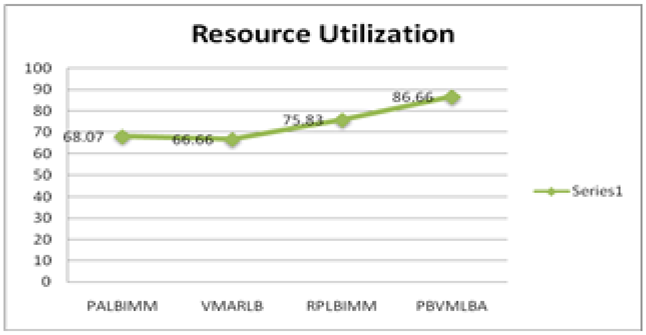

Fig. 7. Resource Utilization

The above result says that the proposed PBVMLBA algorithm is better than other algorithms. It produce the best result for resource utilization.

5. CONCLUSION AND FUTURE WORK

This study deals with job completion in shorts span of time with the help of efficienct utilization of resources such as Vitual Machines. The proposed algorithm PBVMLBA handles the request in priority manner it allots job and resources to the virtual machine using minimum completion time of virtual machine. In this proposed work, the load balancer maintains a table priority of virtual machines as well as their VM availability states (Busy/ Idle). It considers the job allocation by priority basis the virtual machine which helps in the fair allocation of the jobs and efficient user utilization. This work can be further improved by including other factors of the virtual machines. Further, the load balancing can be developed by considering the memory and processor utilization.

REFERENCES

[1] G.Suryadevi,D.Vijayakumar,R.SabariMuthuKumar, Dr. K .G. Srinivasagan , An Efficient Distributed Dynamic Load Balancing Algorithm for Private Cloud, International Journal of Innovative Research in Science, Engineering and Technology, Volume 3, Special Issue 3, March 2014.

[2] Khushbu Zalavadiya, Dinesh Vaghela, Honey Bee Behavior Load Balancing of Tasks in Cloud, International Journal of Computer Applications April 2016.

[3] D.Suresh Kumar, Dr. E. George Dharma Prakash Raj, VMARLB – Virtual Machine Based Algorithm for Random Load Balancing in Cloud Computing, 2017 International Conference on Advanced Computing and Communication Systems (ICACCS – 2017), Coimbatore, INDIA,Jan. 06 – 07,2017.

[4] Ghafour Alipour, Mohammad Hossinzadeh Mogaddam, Soheila Alipour, A New Task Scheduling Algorithm in Cloud Computing, Journal Of Current Research In Science ISSN 2322-5009 Coden (USA) J. Curr. Res. Sci. Vol., S (1), 883-889, 2016.

[5] Kamalakar.M, Moulika.T, A Priority Based Job Scheduling Algorithm in Cloud Computing, International Journal of Innovative Technologies Volume.03, IssueNo.01, May-2015.

[6] Prafulla Chakankar, Nishchol Mishra, Sanjeev Sharma,An Improved Priority Based VM Placement Approach for the Cloud Environment, IJCSN International Journal of Computer Science and Network, Volume 5, Issue 2, April 2016.

[7] Divya Garg, Urvashi Saxena,Dynamic Queue Based Enhanced HTV Dynamic Load Balancing Algorithm In Cloud Computing, International Journal of Innovative Research in Science, Engineering and Technology Vol.5, Issue 1, January 2016.

[8] Harmandeep Singh Brar, Vivek Thapar,An EfficientPriority Based Load Balancing Algorithm forCloud Environment, International Journal of Computer Science Trends and Technology (IJCST) – Volume 2 Issue 4, July-Aug 2014.

[9] Rahul Rathore, Bhumika Gupta, Vaibhav Sharma, Kamal Kumar Gola,Implementation and Result Analysis of Priority Based Load Balancing In Cloud Computing, International Journal of Scientific & Engineering Research, Volume 5, Issue 12, December-2014.

[10] Swachil Patel, Upendra Bhoi,Priority Based Job Scheduling Techniques In Cloud Computing: A Systematic Review, International Journal Of Scientific & Technology Research Volume 2, Issue 11, November 2013.

[11] Faouzia Benabbou,Naoufal Er-raji, Priority Task Scheduling Strategy for Heterogeneous Multi-Datacenters in Cloud Computing, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 2, 2017.

[12] Divya Garg, Urvashi Saxena,Dynamic Queue Based Enhanced HTV Dynamic Load Balancing Algorithm in Cloud Computing, International Journal of Innovative Research in Science,Engineering and Technology Vol.5, Issue 1, January 2016.

[13] Er. Rajeev Mangla , Er. Harpreet Singh, Recovery and user Priority based Load Balancing in cloud Computing, International Journal of Engineering Sciences & Research Technology, Mangla, 4(2):February, s2015.

[14] S. Sankara Narayanan, M. Ramakrishnan and 3Murtaza Saadique Basha, “Efficient Load Balancing Algorithm For Cloud Computing Using Divisible Load Scheduling And Weighted Round Robin Methods”, 2017/Advances in Natural and Applied Sciences. 11(1) January 2017.

[15] Amanpreet Kaur, Sonia Sharma,” Workflow Scheduling in Cloud Computing Environment Using Firefly Algorithm”, International Journal of Computer Sci ence And Technology, IJCST Vol. 8, Iss ue 2, ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print) April - June 2017.