International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-10, August 2019

Performance Analysis of Load Balancing Algorithms using Microsoft Windows Azure Cloud Platform

S. Thirumurugan, D. Suresh Kumar, K. Sankar

Abstract: ***As we are living in technology dominating era, the role of cloud computing as one of the rapidly emerging technologies is highly indispensable. This cloud computing holds load distribution concept, which is usually addressed as a challenging phenomenon, to handle multiple requests which are coming from geographically differentiated locations. There are various load balancing mechanisms exits for the effective distribution of the work load. This load balancing mechanism takes the advent of the virtualization approach on resources to optimize the utilization of the resources. In virtualization, the creation of virtual machine and mapping the same to the suitable request makes the cloud computing a prominent technology. This paper work considers various load balancing mechanisms for the allocation of resources and also shows a comparative study on them with the help of an example scenario. These load balancing algorithms are applied on applications like real-time systems and prioritized medical field related data storage system. This research work is implemented in Windows Azure Platform as Cloud Operation System under load balancing component.***

Index Terms: **VMARLB, PBVMLB, EVMLBA.**

I. INTRODUCTION

Cloud computing is one of the emerging technology comes under utility computing. Load balancing [1] is the challenging phenomenon where an effective method of distributing the jobs must be ensured. This helps to reduce the waiting time of the jobs in the queue and also speed up the execution. There are various algorithms exist to improve the load balancing and thereby increases the throughput. This load balancing should observe the status and privilege level of the virtual machines to properly distribute the work load. Virtualization provides illusion to user where a single hardware can serve multiple user requests.

Virtualization is an abstraction layer between the client and cloud resources. Using the concept of virtualization, we can create a virtual world where a broad range of resource is available. Virtualization is a mechanism which exists in the form of software. This software helps to devise virtual machines with applications and operating system as guest bundled in it. Virtualization makes the cloud resources to be available for various users with different requirements, improves availability of service, performance and scalability factors in cloud computing. VM Ware and Hypervisor – V are existing virtualization tools of Microsoft cloud service provider.

Virtual machine: VM provides abstraction layer to share the computing resources for network level user requests. Isolation plays major role in VM's concept, where each machine has own environment to run process separately. Multiple processes running on single host is separated by instances of machine. Each process has their own virtual machine and it does not affect another process running on same host. Each virtual machine has operating system which is devised from server is called as guest operating system. The status of the virtual machines is maintained by intelligent monitoring process runs in cloud. Each virtual machine can provide an environment to run the applications or may allow the user to develop the application from the scratch. Each virtual machine can play two roles namely web-role and worker-role.

Hypervisor: Hypervisor is also called as software layer because it manages and monitors the virtualization of resources to complete the user requirements. It is an interactive layer that works between operating system and hardware resources. Hypervisor act as a host machine which manages multiple users called as guest machines. Native and host operating systems are two types of hypervisor. In native hypervisor there is no need of software abstraction because it can directly run over the hardware. A host operating system has software rules that required performing virtualization of resources in a proficient way.

Emulation: Emulation is a type of virtualization technique where the behavior of hardware can be translated into a software package. By using technique of emulation better flexibility can be achieved in a cloud model.

This paper is orchestrated as follows. Section I contains the introduction of load balancing and virtualization. Section II describes the related works. Section III contains the existing approaches. Section IV contains results and discussion. Section V concludes research work with future directions.

II. RELATED WORKS

A. Round Robin VM Load Balancing Algorithm

This is a simple approach where the resources are allocated in a time quantum based circular manner. This approach comes with the prerequisite of maintaining the historical data about allocated tasks and resources. Further, this will not have monitoring system to keep track of maintaining the status of the system.

Manuscript published on 30 August 2019.
***Correspondence Author(s)**
S.Thirumurugan, Dept. of Computer Science, Vivekananda College, Chennai, India.
D.Suresh Kumar, Research Scholar, Bharathidasan University, Trichy, India.
K.Sankar, Dept. of Computer Science, Vivekananda College, Chennai, India.

B. Throttled VM Load Balancing Algorithm

This is a dynamic approach where the user requests the data center controller (DCC). Of late, DCC forwards the user request to load balancer to fix the VM machine. As this approach maintains the status (available/busy) of the VMs, the suitable VM will be allotted for the user after ensuring the resources availability as per user request. If the VMs are busy then the client will be put in a queue till the VMs are becoming idle. The uniqueness of this approach is maintaining the present state of all VMs in data center. The downside of this approach is ensuring homogenous VMs in terms of hardware configuration [2].

C. ESCE VM load Balancing Algorithm

Equally Spread Current Execution load balancing algorithm is one of the well-known load balancing approach where workloads are effectively distributed among the VMs present in data center. It maintains a list of VMs and their status. If there is a request then idle VM will be allocated from the list.

At the same time, VMLB inspects the overloaded VMs. The overloaded VMs are relieved from the burden by moving the workload to the idle VM. The main drawback of this approach is high computational overhead [2].

D. An enhanced priority based HTV load balancing algorithm

This algorithm performs an effective and reliable resource allocation of the tasks on the servers in cloud computing environment. This algorithm considers the three parameters such as load on the server, current performance of server and time limit of the tasks. This algorithm computes the load and performance factor of each virtual machine and then allocates the incoming task to various virtual machines according to their time limit and stand-by time to increase the throughput and performance [3].

E. A novel Minimum makespan algorithm

This algorithm produces higher throughput by migrating resource to unallocated node. This algorithm produces minimum makespan when it is compared with other load balancing algorithms like Min-Min, Max-Min and RASA [4].

F. VM and PM Categorization Algorithm

Load balancing approach increases resource utilization and also reduces the energy consumption. This is understood when the existing load balancing algorithm is compared with energy consumption and resource utilization. Experiment results say that VM and PM categorization approach proves to get better result when it is compared with the previous load balancing algorithms [5].

G. Composite algorithm

Load Balancing is essential to correctly manage the resources of the service contributor. It is a technique to dispense the workload among many virtual machines in a Server over the network to attain optimum utilization of resources, reduction in data processing time, diminution in average response time, and avoid overload. The essence of an effective and enhanced composite scheduling algorithm is used to manage the load across the servers of datacentre and providing efficient resource allocation techniques. This Composite approach is useful for load balancing using Equally Spread Current Execution (ESCE) and Throttled algorithms [6].

H. Distributed dynamic priority based algorithm

This algorithm reduces the response time, improves the throughput and also stimulates the system consistency. It considers priority on allocation of resources in order to improve the response time and achieve better processing time. Load balancing guarantees all instances in a node in the networks to do the equivalent amount of work at any instant of time [7].

I. Burstness-aware load balancing algorithm

This algorithm makes use of two algorithms namely burst RR and Non-burst Random to handle the request in a fast pace. RR works when request burst occurs and Random works when the requests are gradually rising in non-burst state. These requests are assigned to VM under fuzzy logic approach. This algorithm has been assessed and related with other algorithms using Cloud Analyst simulator. Implementation of this algorithm shows that this algorithm improves the average response time and average processing speed [8].

J. Honey Behavior Load Balancing Technique

The waiting time of the tasks are reduced by allocating high priority tasks to under loaded machines by considering least numbers of same priorities to those tasks, cost effective virtual machine, and least expected completion time which also balances the loads of dependent tasks in pre-emptive manner. The least expected completion time, cost and priority at submission time of that task supports to make lowest completion time, decreases waiting time of the tasks and ultimately achieves better resource utilization [9].

K. Weighted based optimized load balancing

The jobs are distributed effectively among the servers based on the weight. Experimental results have confirmed that this algorithm has distributed the load evenly among virtual machines [10].

L. New enhanced load balancing algorithm

The evaluation algorithm has been devised to provide an improved solution to load distribution issue. The enhanced load balancing strategy is added with the setting of other parameters like fitness and the selection of the initial resource pool to provide the significant impact on the performance of the algorithm. This enhanced algorithm shows improved results than the existing genetic algorithm [11].

M. Two –level global load balancing framework

Two-level load balancing approach is a framework for global server load balancing of the Web sites in a cloud. This framework is planned for accepting an open-source load-balancing system and the framework permits the network service provider to install a load balancer in geographically differentiated data centers dynamically while the customers demand for more load balancers for ensuring service availability [12].

N. Dynamic load balancing algorithms

Load balancing algorithms play important role in equalizing load among data centers and in efficient use of computing resources. In this paper, performance of a dynamic load balancing algorithm has been evaluated by dividing data-centers in different zones. It has been shown that the proposed algorithm improves the computing efficiency of data-centers and minimizes the response time of user's applications [13].

Retrieval Number: I8458078919/19©BEIESP
DOI: 10.35940/ijitee.I8458.0881019
Journal Website: www.ijitee.org

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

III. LOAD BALANCING ALGORITHMS

Algorithm: Virtual Machine Random Load Balancing (VMARLB)

1: Request from user to DCC.
2: DCC forwarded the request to LB. Initial Stage: random allocation by LB.
3: Update VM availability status (Busy/idle)
4: If there is some more request than forwarded to DCC and check the VM availability status with the help of LB.
5: If VM availability status equal to idle then allocate and update VM availability status.
Else wait for VM availability status equal to idle.
7: Repeat steps 3 and 4 till some user request exist.
8: Stop the process there is no user request.

VMARLB is a load balancing algorithms which handles the users request by appropriate distribution of the load. The component which does this functionality is called as load balancer. This is responsible for storing knowledge base of entire cloud network. Data Center Controller is a component which handles group of servers housed under single roof. It forwards the user request to load balancer for the work allotment among the VMs. Further, load balancer maintains table which contains the task id of the user request, shortest completion time of the virtual machine and the state of the virtual machine. At the initial stage, the tasks are allocated in a random manner to the available VMs. Later, this algorithm searches the table to find the idle virtual machine with shortest completion time. If idle VM is found, the algorithm will reply back to the Data Center Controller with the id of that machine (VM id) and the Data Center Controller allocates the task of that VM, otherwise waits for the signal.

Algorithm: Priority Based Virtual Machine Load Balancing (PBVMLBA)

1: User request reaches DCC.
2: LB accepts the request forwarded by DCC.
If DCC doesn't receive request then go to step8.
3: For each VM find the task completion time of all tasks.
4: Check for the priority of task and check CT < MP.
5: If VM status is found as idle then
Allocate the VM and update VM status. Otherwise Wait for signal until the Job gets completed.
7: Repeat steps 4 and 9 till some user request exist.
8: if there is no request then stop the allocation process.

PBVMLBA is a priority based load balancing algorithm. The DCC receives the request from user which comes with priority tag on it. Later, Data Center Controller forwards the request to the Load Balancer to allocate the request to the available virtual machines. It handles a table which contains the task id of the user request (priority or no priority), completion time of the virtual machine and the state of the virtual machine. If the tasks has priority then depute 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 minimum Makespan. To handle subsequent request, this algorithm will search the table and repeat the above procedure until all the tasks get completed.

Algorithm: Enhanced Virtual Machine Load Balancing (EVMLBA)

1: Request from user to DCC.
2: if DCC=Null Go to step7.otherwise step 3
3: Calculate the completion time for all Tasks for all VM
CT=T/N CT- completion time, T- Task, N-Virtual Machine
4: To search the table and compare all the alternative solutions. To find the Minimum Makespan VM
5: Check the availability, if VM status is found to be idle,
a. Allocate VM
b. Update VM
else wait for signal until all jobs get over.
6: Repeat steps 3 to 5 till some user request exist.
7: if there is no request then stop the allocation process.

EVMLBA is an improved load balancing. The Data Center Controller (DCC) accepts the request from difference places in the world. Data Center Controller forwards the request to the Load Balancer. It handles a table which contains the task id of the user request, completion time of all the virtual machines and the state of the virtual machine. Based on the information available on the table.LB compute the minimum Makespan VM and allocate the task to that VM and update the status. The algorithm will reply back to the Data Center Controller with the id of that machine (VM id), Otherwise waits for the signal.

IV. RESULTS AND DISCUSSION

The implementation scenario takes ten sample task sets and five virtual machines using Microsoft windows azure real time cloud.

A. Implementation of VMARLB

Table 1. VMARLB Task Parameters

Task id	Task Size (mb)
T1	105
T2	200
T3	66
T4	123
T5	155
T6	356
T7	459
T8	512
T9	445
T10	635

It is assumed that there are ten tasks from the users. Table 1 represents the task id, task size. Table 2 represents the VM id, processing speed, data present in Table 3 is the completion time for all task for all VMs. Table 1 shows task parameters namely, task size and task id. It considers ten tasks.

Table 2.VMARLB Resource Parameters

Resource id (VM)	Resource Speed (VM) (mbps)
VM1	35
VM2	40
VM3	45
VM4(P)	80
VM5	100

Table 2 shows the resource parameters, resource (VM) id and resource (VM) speed (mbps). It considers five resources (VMs).

Table 3.VMARLB Task Completion Time of Virtual Machine

VM / Task	VM1	VM2	VM3	VM4	VM5
T1	3	2.62	2.33	1.31	1.05
T2	5.71	5	4.44	2.5	2
T3	1.88	1.65	1.46	0.82	0.66
T4	3.51	3.07	2.73	1.53	1.23
T5	4.42	3.87	3.44	1.93	1.55
T6	10.17	8.9	7.91	4.45	3.56
T7	13.11	11.47	10.2	5.73	4.59
T8	14.62	12.8	11.37	6.4	5.12
T9	12.71	11.12	9.88	5.56	4.45
T10	18.14	15.87	14.11	7.93	6.35

Table 3 shows the results of task completion time for all tasks for all VMs.

Table 4.Task Completion Time of Virtual Machines for Min-Min

VM / Task	VM1	VM2	VM3	VM4	VM5
T1	3	2.62	2.33	1.31	***1.05***
T2	5.71	5	4.44	2.5	***2***
T3	1.88	1.65	1.46	0.82	***0.66***
T4	3.51	3.07	2.73	1.53	***1.23***
T5	4.42	3.87	3.44	1.93	***1.55***
T6	10.17	8.9	7.91	4.45	***3.56***
T7	13.11	11.47	10.2	5.73	***4.59***
T8	14.62	12.8	11.37	6.4	***5.12***
T9	12.71	11.12	9.88	5.56	***4.45***
T10	18.14	15.87	14.11	7.93	***6.35***

Table 4 shows the results of resource allocation time using Min-Min algorithm. The Makespan time of existing Min-min algorithm is 30.56seconds.

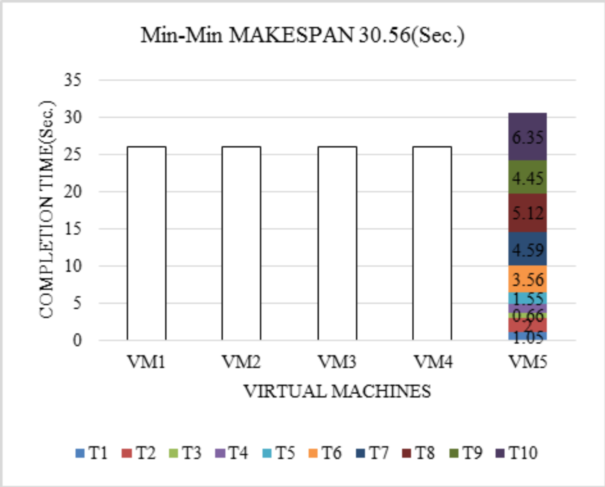

Figure 1. Makespan Time of Min-Min

Figure 1 clearly reveals the resource completion time of Min-Min algorithm. VM5 takes 1.05s, 2s,0.66s,1.55s,3.56,4.59s,5.12s,4.45s, and 6.35s for handling the tasks. VM1,VM2,M3 and VM4 are not allocated. These measurements clearly show that the Makespan time of Min-Min is 30.56seconds and puts the VM1,VM2,VM3 and VM4 to be in ideal state.

Table 5. Task Completion Time of Virtual Machines for VMARLB

VM / Task	VM1	VM2	VM3	VM4	VM5
T1	3	2.62	2.33	1.31	1.05
T2	5.71	5	4.44	2.5	2
T3	1.88	1.65	1.46	0.82	0.66
T4	3.51	3.07	2.73	1.53	1.23
T5	4.42	3.87	3.44	1.93	1.55
T6	10.17	8.9	7.91	4.45	3.56
T7	13.11	11.47	10.2	***5.73***	4.59
T8	14.62	12.8	11.37	6.4	***5.12***
T9	12.71	11.12	***9.88***	5.56	4.45
T10	***18.14***	15.87	14.11	7.93	6.35

Table 5 shows the results of resource allocation time using VMARLB algorithm. The task allocation time for proposed VMARLB algorithm is 21.14 seconds.

Retrieval Number: I8458078919/19©BEIESP
DOI: 10.35940/ijitee.I8458.0881019
Journal Website: www.ijitee.org

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

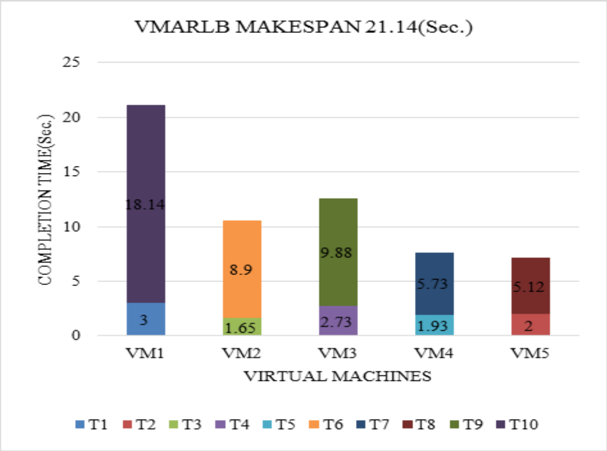

Figure 2. Makespan Time of VMARLB

Figure 2 clearly reveals the resource allocation time of VMARLB algorithm. VM1 takes 3s and 18.14s, VM2 takes 1.65s and 8.9s, VM3 takes 9.88s and 5.73s VM4 takes 1.93s and 5.73s,VM5 takes 2s and 5.12s for resource allocation. These measurements clearly show that the Makespan time of VMARLB is less than that of Min -Min algorithm.

Table 6. Comparisons of Makespan Time for Min-Min vs VMARLB

Algorithms	**Makespan Time (Sec.)**
MIN-MIN	30.56s
VMARLB	21.41s

Table 6 shows the results of Makespan time for Min-Min and VMARLB algorithm for all cases.

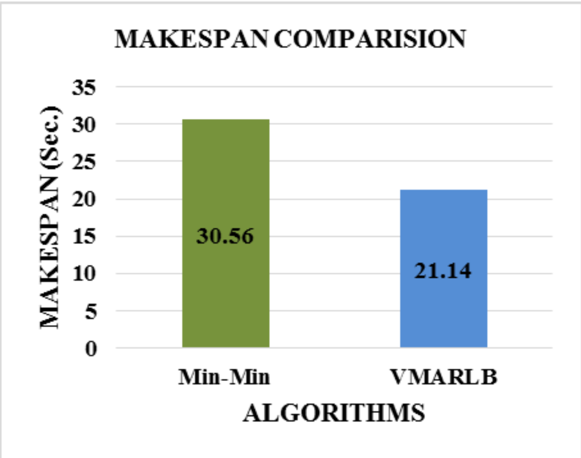

Figure 3. Makespan Comparisons

Figure 3 clearly reveals the Makespan time of Min-Min and VMARLB algorithm. These measurements clearly show that the Makespan time of VMARLB is less than that of existing Min-Min algorithm. Based on the above results the following observation is made. The resources utilization of VMARLB is increased compared with other algorithms.

Average resource utilization

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

N = Number of nodes, m= Makespan, Ci = Completion Time

Table 7. Comparisons of Resource Utilization for VMARLB

Algorithms	**Resource Utilization (%)**
Min-Min	20 %
VMARLB	55.89 %

Table 7 shows the results of resource utilization for Min-Min and proposed VMARLB algorithm.

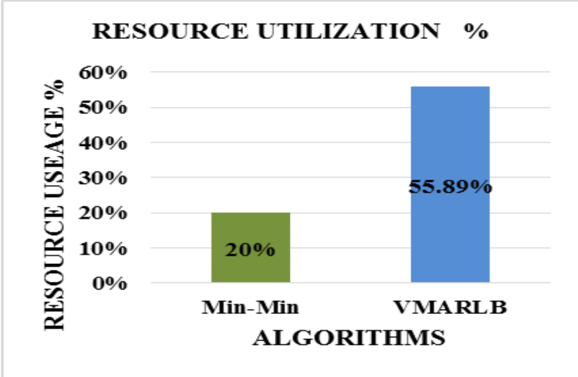

Figure 4. Comparisons of Resource Utilization for Min-Min vs VMARLB

Figure 4 clearly reveals the resource utilization of Min-Min and proposed VMARLB algorithm. These measurements clearly show that the resource Utilization of VMARLB is increased than that of existing Min-Min algorithm.

B. Implementation of PBVMLB:

This analysis assumes ten tasks by the users. Table 8. represents the task id, task size and the user group of each task. Table 9 represents the VM id, processing speed and service type of each resource and data of Table 10 represents the completion time for all tasks.

Table 8. PBVMLBA Task Parameters

Task id	**Task Size (mb)**	**User Group**
T1	105	Ordinary
T2	200	Ordinary
T3	66	Ordinary
T4	123	Priority
T5	155	Ordinary
T6	356	Ordinary
T7	459	Ordinary
T8	512	Ordinary
T9	445	Priority
T10	635	Ordinary

Retrieval Number: I8458078919/19©BEIESP
DOI: 10.35940/ijitee.I8458.0881019
Journal Website: www.ijitee.org

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

Table 8 shows task parameters the task size and task id. It considers ten tasks for PBVMLB algorithm.

Table 9. PBVMLBA Resource Parameters

Resource id (VM)	Resource Speed VM (mbps)	Type
VM 1	35	Ordinary
VM 2	40	Ordinary
VM 3	45	Ordinary
VM 4	80	Priority
VM 5	100	Ordinary

Table 9 shows resource parameters resource (VM) id and resource (VM) speed (mbps). It considers five resources (VM) for PBVMLB algorithm.

Table 10. PBVMLBA Task Completion Time of Virtual Machine

VM / Task	VM1	VM2	VM3	VM4(p)	VM5
T1	3	2.62	2.33	1.31	1.05
T2	5.71	5	4.44	2.5	2
T3	1.88	1.65	1.46	0.82	0.66
T4(p)	3.51	3.07	2.73	1.53	1.23
T5	4.42	3.87	3.44	1.93	1.55
T6	10.17	8.9	7.91	4.45	3.56
T7	13.11	11.47	10.2	5.73	4.59
T8	14.62	12.8	11.37	6.4	5.12
T9(p)	12.71	11.12	9.88	5.56	4.45
T10	18.14	15.87	14.11	7.93	6.35

Table 10 shows the completion time for the given tasks. It considers ten tasks and five virtual machines for PBVMLB algorithm.

Table 11. Task Completion Time of Virtual Machines for PA_LBIMM

VM / Task	VM1	VM2	VM3	VM4(p)	VM5
T1	3	2.62	2.33	1.31	1.05
T2	5.71	5	4.44	2.5	2
T3	1.88	1.65	1.46	0.82	0.66
T4(p)	3.51	3.07	2.73	1.53	1.23
T5	4.42	3.87	3.44	1.93	1.55
T6	10.17	8.9	7.91	4.45	3.56
T7	13.11	11.47	10.2	5.73	4.59
T8	14.62	12.8	11.37	6.4	5.12
T9(p)	12.71	11.12	9.88	5.56	4.45
T10	18.14	15.87	14.11	7.93	6.35

Table 11 shows the results of resource allocation time using existing PA_LBIMM algorithm. The Makespan time for PA_LBIMM algorithm is 24.88 seconds.

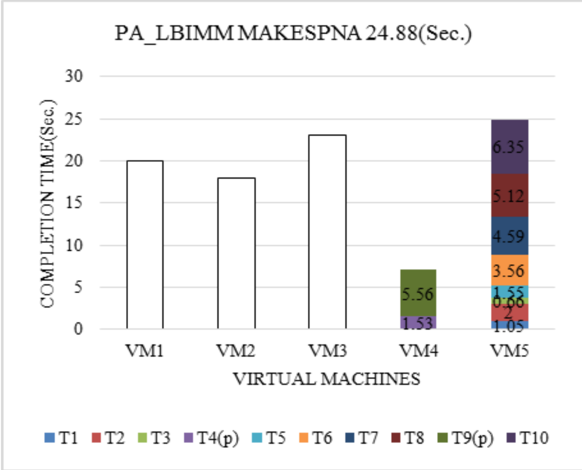

Figure 5. Makespan Time of PBVMLBA

Figure 5 clearly reveals the resource completion time of PA_LBIMM algorithm. VM4 takes 1.53 and 5.56 and VM5 takes 1.05, 2, 0.66, 1.53, 3.56, 4.59, 5.12 and 6.35 for resource allocation. The resources VM1, VM2 and VM3 are not used. These measurements clearly show that the Makespan time of proposed PA_LBIMM algorithm is 24.88 seconds.

Table 12. Task Completion Time of Virtual Machines for PBVMLBA

VM / Task	VM1	VM2	VM3	VM4(p)	VM5
T1	3	2.62	2.33	1.31	1.05
T2	5.71	5	4.44	2.5	2
T3	1.88	1.65	1.46	0.82	0.66
T4(p)	3.51	3.07	2.73	1.53	1.23
T5	4.42	3.87	3.44	1.93	1.55
T6	10.17	8.9	7.91	4.45	3.56
T7	13.11	11.47	10.2	5.73	4.59
T8	14.62	12.8	11.37	6.4	5.12
T9(p)	12.71	11.12	9.88	5.56	4.45
T10	18.14	15.87	14.11	7.93	6.35

Table 12 shows the results of resource allocation time using proposed PBVMLB algorithm. The Makespan time for PBVMLB algorithm is 19.04 seconds.

Retrieval Number: I8458078919/19©BEIESP
DOI: 10.35940/ijitee.I8458.0881019
Journal Website: www.ijitee.org

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

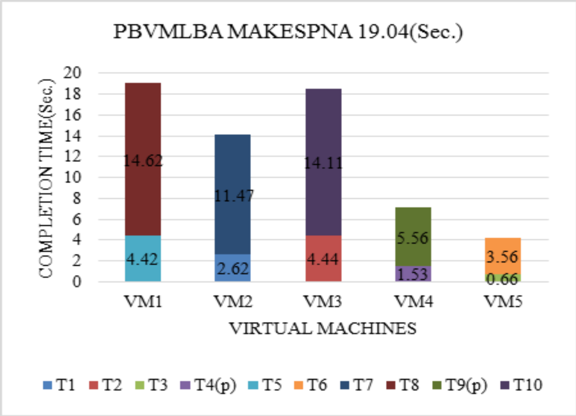

Figure 6. Makespan Time of PBVMLBA

Figure 6 clearly reveals the resource completion time of PBVMLB algorithm. VM1 takes 4.42s and 14.62s, VM2 takes 2.62s and 11.47s, VM3 takes 4.44s and 14.11s, VM4 takes 1.52s and 5.56s, VM5 takes 0.66s and 3.56s for resource allocation. These measurements clearly show that the Makespan time of proposed PBVMLB algorithm is less than that of previous load balancing algorithm.

Table 13. Comparisons of Makespan Time for PA-LBIMM vs PBVMLBA

Algorithms	Makespan Time (Sec.)
PA_LBIMM	24.88s
VMARLB	21.14s
PBVMLBA	19.04s

Table 13 shows the results of Makespan time for PA_LBIMM, VMARLB and PBVMLB algorithms.

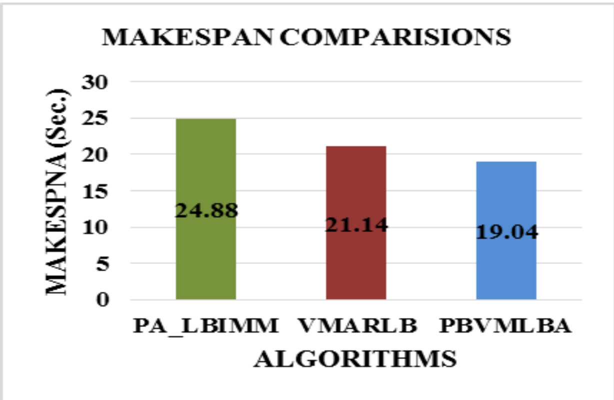

Figure 7. Makespan Comparisons for PBVMLBA

Figure 7 clearly reveals the Makespan time of PA_LBIMM, VMARLB and PBVMLB algorithms. These measurements clearly show that the Makespan time of PBVMLBA is less than that of existing PA_LBIMM and VMARLB algorithm.

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 (rtj), Maximum execution time in a node(VM)

Average resource utilization (Ua)

$$Ua = \sum_{i=1}^{N} Ci * 100$$

Nm

N = Number of nodes, m= Makespan, Ci = Completion Time

Table 14. Comparisons of Resource Utilization for PA-LBIMM vs PBVMLBA

Algorithms	Resource Utilization (%)
PA_LBIMM	68.07 %
PBVMLBA	86.66 %

Table 14 shows the results of resource utilization for PA_LBIMM and proposed PBVMLB algorithm.

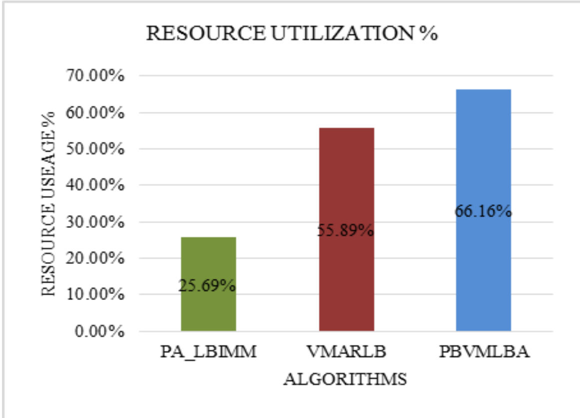

Figure 8. Resource Utilization of PBVMLBA

Figure 8 clearly reveals the resource utilization of PA_LBIMM, VMARLB and proposed PBVMLB algorithm. These measurements clearly show that the resource Utilization of PBVMLBA is increased than that of existing PA_LBIMM and VMARLB algorithms.

C. Implementation Results of EVMLB:

This analysis assumes ten tasks by the users. Table 15 represents the task id, task size. Table 16 represents the VM id, processing speed and data present in Table 17 represents the completion time for all tasks.

Table 15. EVMLBA Task Parameters for Scenario 2

Task id	Task Size (mb)
T1	105
T2	200
T3	66
T4	123
T5	155
T6	356
T7	459
T8	512
T9	445
T10	635

Retrieval Number: I8458078919/19©BEIESP
DOI: 10.35940/ijitee.I8458.0881019
Journal Website: www.ijitee.org

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

Table 16. EVMLBA Resource Parameters

Resource id (VM)	Resource Speed (VM) (mbps)
VM1	35
VM2	40
VM3	45
VM4	80
VM5	100

Table 17. EVMLBA Task Completion Time of Virtual Machine

VM \ Task	VM1	VM2	VM3	VM4	VM5
T1	3	2.62	2.33	1.31	1.05
T2	5.71	5	4.44	2.5	2
T3	1.88	1.65	1.46	0.82	0.66
T4	3.51	3.07	2.73	1.53	1.23
T5	4.42	3.87	3.44	1.93	1.55
T6	10.17	8.9	7.91	4.45	3.56
T7	13.11	11.47	10.2	5.73	4.59
T8	14.62	12.8	11.37	6.4	5.12
T9	12.71	11.12	9.88	5.56	4.45
T10	18.14	15.87	14.11	7.93	6.35

Table 18. Task Completion Time of Virtual Machines for MMA

VM \ Task	VM1	VM2	VM3	VM4	VM5
T1	3	2.62	2.33	1.31	1.05
T2	5.71	5	4.44	2.5	2
T3	1.88	1.65	1.46	0.82	0.66
T4	3.51	3.07	2.73	1.53	1.23
T5	4.42	3.87	3.44	1.93	1.55
T6	10.17	8.9	7.91	4.45	3.56
T7	13.11	11.47	10.2	5.73	4.59
T8	14.62	12.8	11.37	6.4	5.12
T9	12.71	11.12	9.88	5.56	4.45
T10	18.14	15.87	14.11	7.93	6.35

Table 18 shows the results of resource completion time using existing MM algorithm. The Makespan time for existing MM algorithm is 12.01seconds.

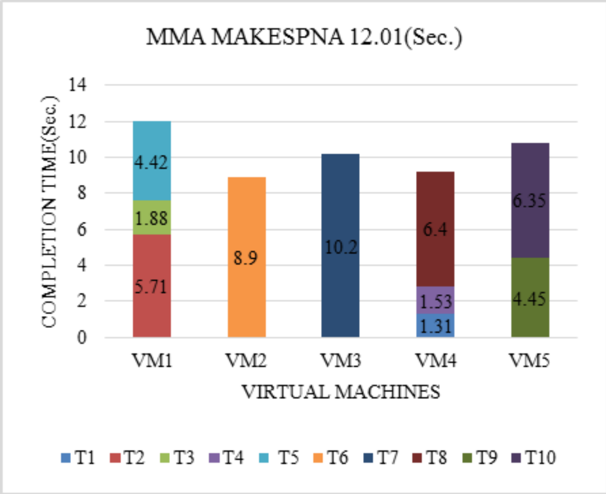

Figure 9. Makespan Time of MMA

Figure 9 clearly reveals the resource allocation time of MMA. VM1 takes 5.71s, 1.88s and 4.42s, VM2 takes 8.9s, VM3 takes 10.2s, VM4 takes 1.31s, 1.53s and 6.4s VM5 takes 4.45s and 6.35s for resource completion. These measurements clearly show that the Makespan time of existing MM algorithm is 12.01 seconds.

Table 19. Task Completion Time of Virtual Machines for EVMLBA

VM \ Task	VM1	VM2	VM3	VM4	VM5
T1	3	2.62	2.33	1.31	1.05
T2	5.71	5	4.44	2.5	2
T3	1.88	1.65	1.46	0.82	0.66
T4	3.51	3.07	2.73	1.53	1.23
T5	4.42	3.87	3.44	1.93	1.55
T6	10.17	8.9	7.91	4.45	3.56
T7	13.11	11.47	10.2	5.73	4.59
T8	14.62	12.8	11.37	6.4	5.12
T9	12.71	11.12	9.88	5.56	4.45
T10	18.14	15.87	14.11	7.93	6.35

Table 19 shows the results of resource completion time using proposed EVMLB algorithm. The Makespan time for proposed EVMLB algorithm is 10.62 seconds.

Retrieval Number: I8458078919/19©BEIESP
DOI: 10.35940/ijitee.I8458.0881019
Journal Website: www.ijitee.org

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

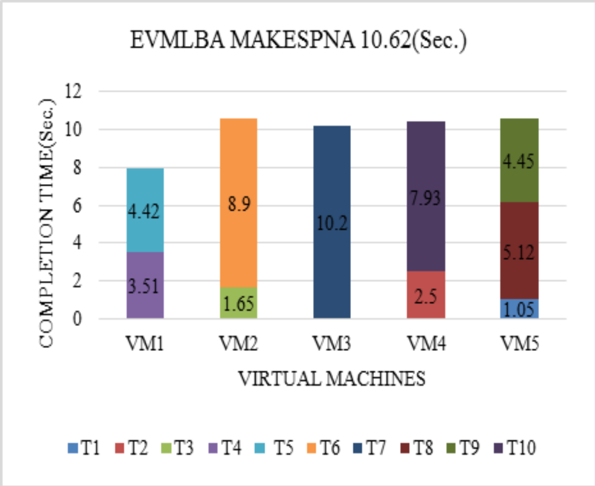

Figure 10. Makespan Time of EVMLBA

Figure 10 clearly reveals the resource completion time of EVMLB algorithm. VM1 takes 1.88s, 3.51s and 4.42s, VM2 takes 1.65s and 8.9s, VM3 takes 10.2s, VM4 takes 7.93s, and VM5 takes 1.05s, 5.12s and 4.45s for resource completion. These measurements clearly show that the Makespan time of proposed EVMLB algorithm is less than that of previous load balancing algorithm.

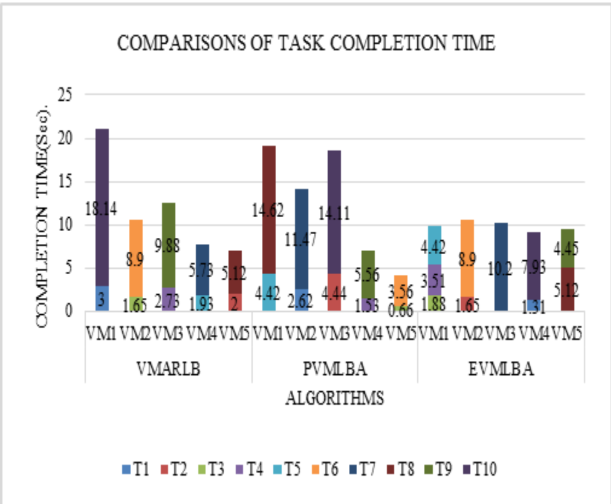

Figure 11. Comparisons of Task Completion Time

Figure 11 clearly reveals the comparisons of task completion time of VMARLB algorithm. VM1 takes 3s and 18.14s, VM2 takes 1.65s and 8.9s, VM3 takes 9.88s and 5.73s VM4 takes 1.93s and 5.73s,VM5 takes 2s and 5.12s for resource allocation. resource completion time of PBVMLB algorithm. VM1 takes 4.42s and 14.62s, VM2 takes 2.62s and 11.47s, VM3 takes 4.44s and 14.11s, VM4 takes 1.52s and 5.56s, VM5 takes 0.66s and 3.56s for resource allocation. resource completion time of EVMLB algorithm. VM1 takes 1.88s, 3.51s and 4.42s, VM2 takes 1.65s and 8.9s, VM3 takes 10.2s, VM4 takes 7.93s, VM5 takes 1.05s, 5.12s and 4.45s for resource completion. These measurements clearly show that the Makespan time of proposed VMARLB, PBVMLBA and EVMLB algorithm is less than that of previous load balancing algorithm.

Comparative Analysis of Proposed Algorithms:

Table 20. Makespan Comparison

Algorithms	Makespan
VMARLB	21.14s
PBVMLBA	19.04s
MMA	12.01s
EVMLBA	10.62s

Table 20 shows the results of Makespan time comparison algorithms. The Makespan time for proposed EVMLB algorithm is 10.62 seconds. VMARLB algorithm completes the same tasks with 21.14s as time utilization factor. PBVMLBA algorithm executes the same by taking 19.04 seconds. MM algorithm executes the same tasks with time utilization of 12.01 seconds and EVMLBA algorithm completes the same tasks within 10.62 seconds. This shows that EVMLBA algorithm consumes minimum time for completion of the task. This is due to the use of enhanced approach, which reduces the waiting time and increases the utilization time of the resources. In the random approach the tasks are randomly allocated then allocation continues based on state of VM. In Priority approach, the high priority task is assigned to the priority virtual machine. In the Enhanced approach, initially, it computes completion time for all tasks; among them one which has the minimum execution time is selected and utilized.

Based on the above results the following points are concluded.

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

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

Average resource utilization (Ua)

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

N = Number of nodes, m= Makespan, Ci = Completion Time

Table 22. Comparisons of Resource Utilization

Algorithmas	Resource Utilization (%)
VMARLB	55.89%
PBVMLBA	66.16%
MMA	85.47%
EVMLBA	92.97%

Table 22 shows the results of resource utilization comparison of proposed EVMLB algorithm. The resource utilization for proposed EVMLB algorithm is 92.97%. It shows that the EVMLBA algorithm is better than other two algorithms VMARLB and PBVMLBA. It produces the best result for resource utilization.

Retrieval Number: I8458078919/19©BEIESP
DOI: 10.35940/ijitee.I8458.0881019
Journal Website: www.ijitee.org

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

V. CONCLUSION

This study describes the existing load balancing algorithms. Further, the performance of those algorithms are analysed with the help of an example scenario. The results show that the load balancing algorithms are performing better in terms of resource utilization, based on the approach and nature of request. It is realized that out of three load balancing algorithms, Enchanced load balancing gives better results on resource utilization over the other algorithms. These algorithms are suitable for high speed computing, privileged on line services and secured services. This study considers few parameters for the implementation of the algorithms. In the future, some more parameters like geographical distance of virtual machines are identified on allocation of VMs.

ACKNOWLEDGMENT

I would like to thank the management of Ramakrishna Mission Vivekananda College for their constant support on carrying out this research work.

REFERENCES

1. Reena Panwar ,Bhawna Mallick, "A Comparative Study of Load Balancing Algorithms in Cloud Computing", ,*International Journal of Computer Applications ,Vol. 117, No. 24, May 2015 .*
2. Mamta Khanchi, Sanjay Tyagi, "An Efficient Algorithm for Load Balancing in Cloud Computing*", International Journal of Engineering Sciences & Research Technology*, June, 2016.
3. 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, No. 1, January 2016.
4. Harish Chandra, Pradeep Semwal, Sandeep Chopra," load balancing in cloud computing usinga novel minimum makespan algorithm", International Journal of Advanced Research in Computer Engineering & Technology ,Vol. 5, No. 4, April 2016.
5. Sachin Soni, Praveen Yadav, "A Load Balancing Approach to Minimize the Resource Wastage in Cloud Computing", *International Advanced Research Journal in Science, Engineering and Technology*, Vol. 3, No. 3, March 2016.
6. Navtej Singh Ghumman, Rajesh Sachdeva," an efficient approach for load balancing in cloud computing using composite techniques", *International Journal of Research in Engineering and Applied Sciences*, Vol. 6, No. 2 February, 2016.
7. G.Suryadevi, D.Vijayakumar, R.SabariMuthuKumar, Dr. K .G. Srinivasagan, "An Efficient Distributed Dynamic Load Balancing Algorithm for Private Cloud Environment", *International Journal of Innovative Research in Science, Engineering and Technology*, Vol. 3, No. 3, March 2014.
8. 8. Sally F. Issawi, Alaa Al Halees, Alaa Al Halees, "An Efficient Adaptive Load Balancing Algorithm for Cloud Computing Under Bursty Workloads", *Engineering, Technology & Applied Science Research,* Vol. 5, No. 3, 2015.
9. Khushbu Zalavadiya,Dinesh Vaghela , "Honey Bee Behavior Load Balancing of Tasks in Cloud Computing", *International Journal of Computer Applications, Vol. 139, No.1, April 2016.*
10. B. Nithya Nandhalakshmi , Mahalingam, "Efficient Load Balancing in Cloud Computing Using Weighted Throttled Algorithm", *International Journal of Innovative Research in Computer and Communication Engineering* ,Vol. 3, No. 6, June 2015.
11. Er. Pooja Er. Vivek Thapar,"An Enhanced Virtual Machine Load Balancing Algorithm for Cloud Environment", *, International Journal of Emerging Research in Management &Technology, Vol. 5, No. 5,May 2016.*
12. Po-Huei Liang1 and Jiann-Min Yang. "Evaluation of two-level Global Load Balancing Framework in cloud environment", *International Journal of Computer Science & Information Technology* , Vol. 7, No. 2, April 2015.
13. Navtej Singh Ghumman, Rajesh Sachdeva," an efficient approach for load balancing in cloud computing using composite techniques", *International Journal of Research in Engineering and Applied Sciences*, Vol. 6, No. 2, February 2016.
14. Harshada Raut , Kumud Wasnik, " a survey on load balancing algorithms in cloud computing, *International Journal of Advanced Technology in Engineering and Science* ,Vol. 03, No. 1, March 2015.

AUTHORS PROFILE

Dr.S.Thirmurugan Vivekananda College, Assistant Professor, Department of Computer Science. He has 16 publications in his credit and holding around 16 years of experience in teaching and industry. His focus includes wireless network, computing and sensor networks.

D.Suresh Kumar National College, Assistant Professor, Dept. Computer Science. He has 8 publications in his credit. He has 15 years of teaching experience at Various Colleges, Tamil Nadu, India.

K.Sankar Vivekananda College, Assistant Professor, Department of Computer Science. He has 2 publications in his credit and holding around 20 years of experience in teaching. His focus includes advanced operating system, data structure and algorithms.

Retrieval Number: I8458078919/19©BEIESP
DOI: 10.35940/ijitee.I8458.0881019
Journal Website: www.ijitee.org

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