An Enhanced R-R Load Balancing Algorithm for VM Scheduling in Cloud Computing Environment

Y.Geetha[1] Dr.K.Venkataramana[2]
[1]Student [2]Professor
[1,2]Department of Computer Applications
[1,2]KMM Institute of PG Studies, Tirupati, India

Abstract— Cloud computing is the next generation of computation and comprises web-based tools and applications as opposed to a direct connection to a server. Cloud providers typically use a "pay-as-you-go" model, which can lead to unexpected operating expenses if administrators are not familiarized with cloud-pricing models. The availability of high-capacity networks, low-cost computers and storage devices as well as the widespread adoption of hardware virtualization, service-oriented architecture, and autonomic and utility computing has led to growth in cloud computing. 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 make span and to maximize the resource utilization and user satisfaction.

Key words: Cloud computing, VM Scheduling, R-R Load Balancing Algorithm

I. INTRODUCTION

Cloud computing is collection of distributed servers which provides services on demand that means storing and accessing data and program over the internet. These resources can be dynamically assigned and released with minimal management effort or service provider interaction. It provides 3 services such as Software as Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Different physical and virtual resources are provided to the users on demand. In Cloud computing, access to the resource is based on Virtualization [1]. Virtualization is an abstraction of real machines. Virtual Machine has ability to run applications like any real machine. Virtualization provides facilities similar to real machines. We can create larger number of less powered servers through virtualization, which in turn reduces the overall cost in space, power, and infrastructure. Cloud resources can be scaled rapidly using virtualization technique. Cloud resources are dynamically allocated to users on demand. As the number of user increases, the available resources decrease dynamically. Allocation of cloud resources to users on demand gives rise to the problem of load balancing. If workload is not distributed properly, then some nodes in cloud will be heavily loaded and some nodes will be under loaded. In the same way if the resources provided by the cloud are not allocated efficiently, it leads to delay in providing service to the users [2].

II. LITERATURE SURVEY

G.Suryadevi, D.Vijayakumar, R.SabariMuthuKumar, Dr. K .G. Srinivasagan [3] Priority based resource provision to improve the utilization of resources and reducing response time of cloud services.

R. Kanakala; V. K. Reddy; K. Karthik, [4] Static algorithms are best in homogeneous and stable environments. However, static algorithms are not flexible and cannot consider the dynamic changes to the attributes. While assigning tasks to the nodes, static load balancing algorithms will not check the state and functionality of the node in previous tasks.

R. Kanakala; V. K. Reddy; K. Karthik, [4] Dynamic algorithms provide better results in heterogeneous and dynamic environments. These algorithms are more flexible. Dynamic algorithms can consider the dynamic changes to the attributes. However, these algorithms are more complex.

Prafulla Chakankar, Nishchol Mishra, Sanjeev Sharma,[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.

III. VM SCHEDULING

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 [5].

Load balancing is a process for distributing the workload dynamically and uniformly across the all available nodes in the cloud. In cloud computing system it is possible, that some nodes to be heavily loaded and other is lightly loaded. This situation can lead to poor performance. The goal of load balancing is distributes the load among nodes in cloud environment. Load balancing is a central issues in cloud computing.

For better resource utilization, it is desirable for the load in the cloud system to be balanced evenly. Thus, a load balancing algorithm tries to balance that total system load by transparently transferring the workload from heavily loaded nodes to lightly loaded nodes in an attempt. When we considering the performance from point of view, the metric involved is often the response time of the processes.

All rights reserved by *www.ijsrd.com*

However, when performance is considered from there source point of view, the metric involved the total system throughput. In contrast to response time, throughput is concerned with seeing that all users are treated fairly and that all are making progress. To improve the performance of the system and high resource allocation ratio we need load balancing in cloud computing. The characteristics of load balancing are:[6]

- Distribute load evenly across all the nodes.
- To achieve a high user satisfaction.
- Improving the overall performance of the system.
- To reduce response time.
- To achieve resource utilization ratio.

Suppose if hundreds of users are using this application at the same time from single machine and we did not apply to the load balancing approach to our application. This time every particular server is very busy to execute the user's tasks and other servers are lightly loaded or idle. The users didn't satisfy because of low response and slow performance of the system. If we apply load balancing on our application, we can distribute some user's tasks to other nodes and we will get the high performance and faster response time. In this way we can achieve some of above characteristics in load balancing. Hence, the load balancing will be needed to maximize the throughput by minimizing the responsetime.

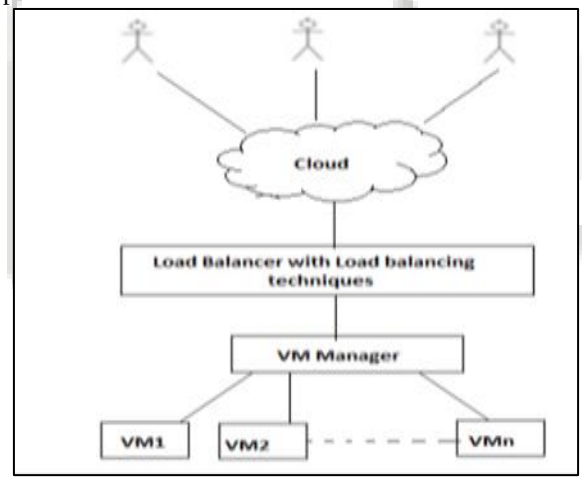


Fig. 1: General Structure of Load balancing in Cloud Environment [6]

A. Load balancing Measurement Parameter

There are some of the measurement parameters to evaluate the load balancing techniques [7].

1) Throughput: It is the amount of work to be completes in the given amount of time.
2) Response time: It is the amount of time used to start manages the demand of the user after registering the request.
3) Scalability: It is the ability of the algorithm to plate itself according to required conditions.
4) Performance: It is the overall checks the algorithms and working by considering accuracy, cost and speed.
5) Resource utilization: It is used to keep and checks on the utilization of various resources.

B. Classification of Load Balancing Algorithm

There are many load balancing algorithms. Generally Load balancing algorithms are classified into two types based on the present system state:

- Static Algorithm: Static Algorithms are good for similar and stable environment.
- Dynamic Algorithm: Dynamic Algorithms are good for diverse environment.

C. Static Algorithm

Static algorithms are the best in homogeneous and stable environments. But, static algorithms are not flexible and cannot consider the dynamic changes to the attributes. At the same time assigning tasks to the nodes, static load balancing algorithms will not check the state and functionality of the node in previous tasks.[4] Some Static Algorithms are:

1) Round Robin Load Balancing Algorithm (RR)
2) Load Balancing Min-Min Algorithm (LB Min-Min)
3) Load Balancing Min-Max Algorithm (LB Min-Max)

A. Round Robin Load Balancing Algorithm

In this algorithm, the quantum time is fixed to the given job. It allocates the jobs to all nodes in a circular fashion. Processors are assigned in a circular order and thus there is no starvation. This algorithm gives a faster response in the case of equal workload distribution among processes. However, some nodes may be over loaded while others remain idle and under-utilized.

1) MIN-MIN Load Balancing Algorithm

A list of task is maintained the minimum completion time is calculated for all the available nodes. A task with least completion time is assigned to the machine. Hence the name of the algorithm is min-min [7]. Update the file and running time of the machine. It gives the good results when small tasks are more.

2) MIN-MAX Load Balancing Algorithm

A list of task is maintained the minimum completion time is calculated for all the available nodes. A task with least completion time is assigned to the machine. Hence the name of the algorithm is min-max [7]. Update the files and running time of the machine.

D. Dynamic Algorithms

Dynamic algorithms provide effective results in heterogeneous and dynamic environments. These algorithms are more flexible. Dynamic algorithms can examine the dynamic changes to the attributes. However, these algorithms are more complex [4]. Main advantage of this selection of task is based on current state and this will help to improve the performance of the system. Dynamic algorithms can be implemented by two forms:

1) Distributed System

In this all the nodes interact with each other and load balancing algorithm is executed by all the nodes in the system. The task of load balancing is distributed between all the nodes. Interaction among nodes can be cooperative or non-cooperative. If any node fails in the system, it will not finish the functionality.

1) In cooperative distributed system, all node works with each other.
2) In non-cooperative distributed system, each node works separately.

All rights reserved by www.ijsrd.com

2) Non-distributed System

Non-distributed can be centralized or semi-distributed.

Centralized system: central node is responsible for load balancing of the entire system. The other nodes interact with this central node. If central nodes fail, then it will stop the functionality. In case of failure, recovery of functionality will not be easy.

Semi-distributed system: All nodes are grouped to form a cluster. A central node of each cluster performs the load balancing of whole system. If central node of cluster fails, it will be stop the functionality of that cluster only. Multiple central nodes are manages the load balancing. Thus more accurate load balancing. Some dynamic algorithms are:

1) Honeybee Foraging Behavior Load Balancing Algorithm
2) Throttled Load Balancing Algorithm
3) Ant Colony Load Balancing Algorithm

a) Honeybee Foraging Behavior Load Balancing Algorithm

This algorithm was obtained from the behavior of real honey bees in finding their food sources. After finding the food sources, the honey bees come back to the bee hive to inform the food source. They do this by performing group movement. This group an act of moving is also known as "waggle Dance". They perform the waggle dance to inform other bees of the exact location of the food source. This waggle dance shows the quality and quantity of the food and the distance of the food source from the bee hive [1].

b) Throttled Load Balancing Algorithm

Throttled load balancing algorithm is suitable for virtual machines. Load balancer maintains the list of whole virtual machines in the system. When the load balancer can receive a request, it scans the indexing table. If the virtual machine is available, then the job is assigned to that machine. Load balancer updates the indexing table after each allocation and de-allocation of resource.

c) ANT COLONY Load Balancing Algorithm

Real ant selects a shortest path in search of its food. This algorithm is from on the behavior of real ants. When request is initiated ant starts its movement. Ant continuously checks whether the node is overloaded or under loaded. If ant finds any overloaded node, it turns back. And if ant finds any under loaded node, it proceeds. In this method the behavior of ant is used to collect the information from different nodes.

IV. PROPOSED R-R SCHEDULING ALGORITHM

In this algorithm, to fixed quantum time is given to the job. It allocates jobs to all nodes in a priority fashion. Processors are assigned in a descending order and hence there is no starvation. In this algorithm provides faster response in the case of equal workload distribution among processes. However, some nodes may be over loaded while others remain idle and under-utilized. These 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 gets 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.[3] 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 creates the cloud environment for measuring the performance of the proposed approach and compare with the existing VM scheduling approach. The Experiments of result shows that proposed approach minimize the energy consumption and simulation time. 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 high response time and less important to optimum utilization of resources. This has become motivation factor to propose the Priority Based Virtual Machine Load Balancing Algorithm. The PBVMLBA distribute the load balance 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.

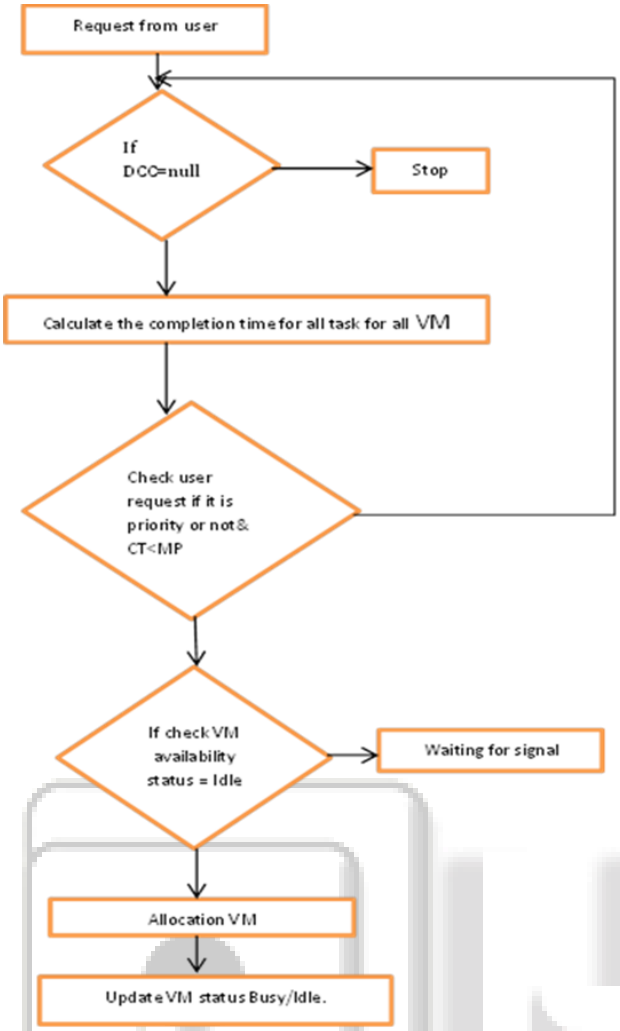
V. WORKING OF R-R SCHEDULING ALGORITHM

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). These nodes are 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 manages 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, examine 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 manage further request, this algorithm will search the table and repeat the above procedure until all the tasks get completed.

A. PROPOSED ALGORITHM (PBVMLBA)

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

All rights reserved by www.ijsrd.com



VI. Results Of R-R Scheduling Algorithm With An Example

The proposed algorithm PBVMLBA is implemented by using the Microsoft azure real time cloud which runs on ASP, SQL database. And Microsoft azure as a deploying tool. Assumes 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.

Task id	**Task size(MB)**	**User group**
T1	100	Ordinary
T2	150	Ordinary
T3	200	Ordinary
T4	250	Priority
T5	500	Ordinary

Table 1: Task Parameter

Resource (VM) id	Resource (VM) speed (Mbps)	Type
VM1	20	Priority
VM2	16	Ordinary
VM3	10	Ordinary

Table 2: Resource Speed

Task	Resource		
	Priority VM1	VM2	VM3
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

Table 3: Computed Completion time (CT)

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

	Resource		
Task			
	Priority VM1	VM2	VM3
T4(P)		19.625	
T1		25.875	
T2			15
T3		38.375	
T5	25		

Table 4: Execution time of PA_LBIMM

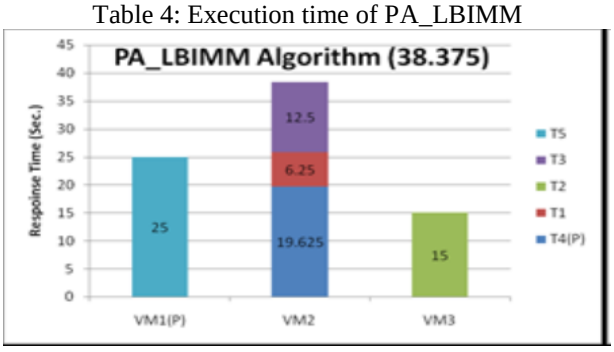


Figure 2: PA_LBIMM

All rights reserved by www.ijsrd.com

Task	Resource		
	Priority R(1)	R(2)	R(3)
T3			20
T1		6.25	
T5	10		
T4		31.25	
T2	7.5		

Table 5: Execution time of VMARLB algorithm

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.

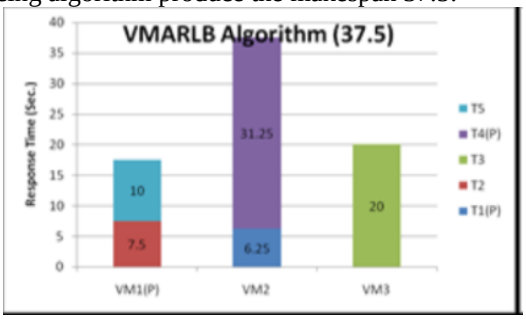
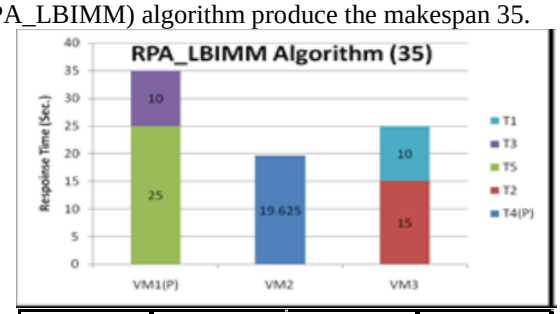


Fig 3: VMARLB

Task	Resource		
	Priority R(1)	R(2)	R(3)
T4(P)		19.625	
T2			15
T5	25		
T3	35		
T1			25

Table 6: Execution time of RPA_LBIMM

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



Task	Resource		
	Priority (R)1	R (2)	R (3)
T4(P)	12.5		
T5		31.5	
T3			20
T2	7.5		
T1			10

Table 7: Completion time of Proposed PBVMLBA algorithm

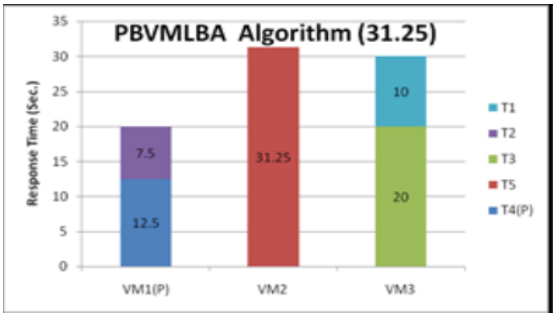


Fig. 5: Proposed PBVMLBA

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.

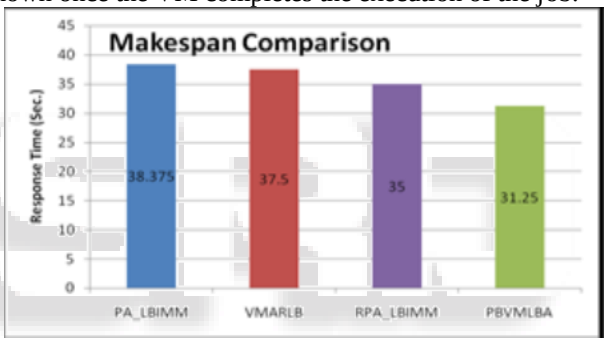


Fig. 6: Makespan Comparison

Based on the above results the following points are concluded.

1) The resources utilization of PBVMLBA is increased compared with other algorithms.
2) Makespan = max (rtj), Maximum execution time in a node(VM)

Method	Makespan
PA_LBIMM	38.375
VMARLB	37.5
RPA_LBIMM	35
PBVMLB	31.25

Table 8: Makespan

Average resource utilization (Ua)

$$Ua=\Sigma iN_{NM}=1\ ci*100 \quad [1]$$

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

All rights reserved by *www.ijsrd.com*

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

Table 9: Resource Utilization

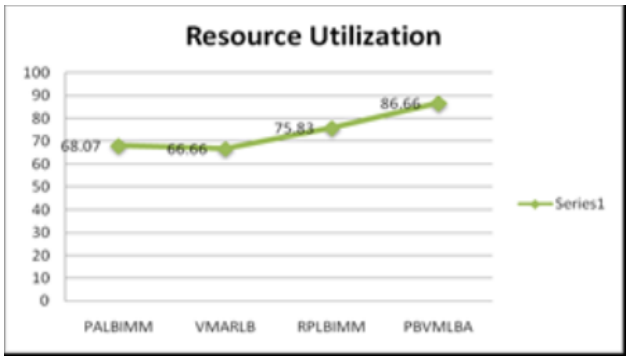


Fig. 7: Resource Utilization

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

VII. Conclusion

This study deals with job completion in shorts span of time with the help of efficient utilization of resources such as Virtual 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 other improved by including further factors of the virtual machines. Further, the load balancing can be developed by considering the memory and processor utilization.

Acknowledgment:

We would like to express our deep sense of gratitude towards Prof. D r.K.Venkataramana of KMM Institute of PG Studies for providing us constant motivation and support as and when required.

References

[1] Rajwinder Kaur and Pawan Luthra," Load Balancing in Cloud Computing", Association of Computer Electronics and Electrical Engineers, 2014, DOI: 02.ITC.2014.5.92.
[2] Ashalatha R; J. Agarkhed,"Dynamic load balancing methods for resource optimization in cloud computing environment ", 2015 Annual IEEE India Conference (INDICON), Pages: 1 - 6, DOI: 10.1109/INDICON.2015.7443148.
[3] 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.
[4] R. Kanakala; V. K. Reddy; K. Karthik, " Performance analysis of load balancing techniques in cloud computing environment", 2015© 2017, IJCSE All Rights Reserved 100 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Pages: 1 - 6, DOI: 10.1109/ICECCT.2015.7226052
[5] 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.
[6] Garima Rastogi, Dr Rama Sushil, "Analytical Literature Survey on Existing Load Balancing Schemes in Cloud Computing", 2015 International Conference on Green Computing and Internet of Things (ICGCloT), pages:1506-1510
[7] R. Kanakala; V. K. Reddy; K. Karthik, " Performance analysis of load balancing techniques in cloud computing environment", 2015 International Journal of Computer Sciences and Engineering Vol.-5(1), Jan 2017, E-ISSN: 2347-2693 © 2017, IJCSE All Rights Reserved 100
[8] IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Pages: 1 - 6, DOI: 10.1109/ICECCT.2015.7226052