DOI: 10.21917/ijct.2013.0107

IMPROVING RESOURCE UTILIZATION USING QoS BASED LOAD BALANCING ALGORITHM FOR MULTIPLE WORKFLOWS IN IAAS CLOUD COMPUTING ENVIRONMENT

L. Shakkeera[1], Latha Tamilselvan[2] and Mohamed Imran[3]
School of Computer, Information and Mathematical Sciences, B.S. Abdur Rahman University, India
E-mail: [1]shakkeera841@gmail.com, [2]latha.tamil@bsauniv.ac.in and [3]mohamedimran31@gmail.com

Abstract
Cloud computing is the extension of parallel computing, distributed computing and grid computing. It provides secure, quick, convenient data storage and net computing services through the internet. The services are available to user in pay per-use-on-demand model. The main aim of using resources from cloud is to reduce the cost and to increase the performance in terms of request response time. Thus, optimizing the resource usage through efficient load balancing strategy is crucial. The main aim of this paper is to develop and implement an Optimized Load balancing algorithm in IaaS virtual cloud environment that aims to utilize the virtual cloud resources efficiently. It minimizes the cost of the applications by effectively using cloud resources and identifies the virtual cloud resources that must be suitable for all the applications. The web application is created with many modules. These modules are considered as tasks and these tasks are submitted to the load balancing server. The server which consists our load balancing policies redirect the tasks to the corresponding virtual machines created by KVM virtual machine manager as per the load balancing algorithm. If the size of the database inside the machine exceeds then the load balancing algorithm uses the other virtual machines for further incoming request. The load balancing strategy are evaluated for various QoS performance metrics like cost, average execution times, throughput, CPU usage, disk space, memory usage, network transmission and reception rate, resource utilization rate and scheduling success rate for the number of virtual machines and it improves the scalability among resources using load balancing techniques.

Keywords:
Cloud Computing, Infrastructure as a Service (IaaS), Load Balancing, Assignment Approach, Round Robin Scheduling, KVM

1. INTRODUCTION

Cloud computing [1]-[3] is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the use of a cloud-shaped symbol as an abstraction for the complex infrastructure. Cloud computing entrusts remote services with a user's data, software and computation. Users access cloud-based applications through a web browser or a light-weight desktop or mobile application while the business software and user's data are stored on servers at a remote location. Cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand. As computers became more prevalent, scientists and technologists explored ways to make large-scale computing power available to more users through time sharing, experimenting with algorithms to provide the optimal use of the infrastructure, platform and applications with prioritized access to the CPU and efficiency for the end users.

In cloud computing, load balancing [4], [5] is another important technology which comes under the cloud computing platform. As the number of user increases for the cloud resources the cloud resources should have the capability to servicing to the 'n' number of users request. For the better availability of cloud resources and for the shorter response time for the user request, load balancing plays a main role in IaaS virtual cloud computing environment.

Virtualization [6] is a term that refers to the abstraction of computer resources. Virtualization can be applied to many types of computer resources: storage, network, and computing (CPU/Memory etc.) We concentrate on server virtualization for the purposes of this paper. There are several approaches to virtualizing servers, including GRID approaches (where discrete workloads are distributed among multiple physical servers, and the results eventually collected), OS –level virtualization, sometimes called containers where multiple instances of an application can run in isolation from one another on a single OS instance, and hypervisor-based virtualization which currently appears the most widespread. Within hypervisor virtualization there are several sub-approaches to achieve the same goal; to run multiple workloads (defined here as Operating Systems, such as Windows or Linux along with applications) on a single physical host. For high-availability, multiple hosts can be "pooled" together to form "clusters" or "farms" often sharing the storage on which the virtual machines themselves reside.

Amazon Elastic Compute Cloud (EC2) is a central part of amazon's cloud computing platform, Amazon Web Services (AWS). EC2 allows users to rent virtual computers on which to run their own computer applications. EC2 allows scalable deployment of applications by providing a Web service through which a user can boot an Amazon machine image to create a virtual machine, which Amazon calls an "instance", containing any software desired. A user can create, launch, and terminate server instances as needed, paying by the hour for active servers, hence the term "elastic". EC2 provides users with control over the geographical location of instances that allows for latency optimization and high levels of redundancy.

2. PROBLEM DEFENITION

In Cloud Computing technology, there have been some issues considering the maintenance of the load (like CPU load, memory capacity and delay or network load) between the cloud computing resources. The case of the organization that operates a local cluster managed by virtual machine technology to supply its user with resource required by their application is considered. The cloud computing environment is established by making the virtual resource of a machine and sharing the virtual resource as per the user specification. If the number of user to the particular virtual machine exceeds the load balancing server will redirect the new incoming user's request to the other virtual machines in node controller. But, this is common mechanism which does not concentrate on time, throughput and efficiency. Meanwhile the random arrival of load in such an environment can cause some server to be heavily loaded while other server is idle or only lightly loaded. Equally load distributing improves the performance by transferring load from heavily loaded server. So we need an emerging requirement to balance the multiple workflows in cloud environment.

3. LOAD BALANCING

Load balancing [7]-[9] is a computer networking methodology to distribute workload across multiple computers or a computer cluster, network links, central processing units, disk drives, or other resources to achieve optimal resource utilization, maximize throughput, minimize response time, and avoid overhead. Using multiple workflows with load balancing, instead of a single workflow may decrease efficiency of the system through scalability. The load balancing service is usually provided by dedicated software or hardware, such as a multilayer switch or a System server.

In this paper we have proposed the some of the load balancing strategies which aims the efficient use of the cloud computing resources [10]. First one is the common load balancing algorithm in which if more number of users or requests (for e.g., n = 50) to the particular virtual machine exceeds then the load balancing server redirect the incoming request to the other virtual machine. The other type of load balancing algorithm is to balance the load based on the available memory space or disk space on the virtual machine. Most of the cloud users use the cloud resources either to store or retrieve data or for some computation. Here if the memory space of one virtual resource exceeds there may be slow down on computation in virtual resource and there may be high delay in response time for other users. In this condition the load balancer redirect the new incoming request to the other virtual machine based on the available space on different virtual machine.

The above said approach balances the load irrespective of time and cost for the tasks. But in assignment approach the tasks are assigned to the specified virtual machine based on the task and its time of execution and throughput. Here we consider threshold value for every virtual machine. If the threshold value of one virtual machine exceeds its send the status to the status update phase which is present in load balancing server and then the load balancer server identify the other virtual machines for its client request.

4. ARCHITECTURAL FRAMEWORK MODEL

Load balancing is a process of reassigning the total load to the individual nodes of the collective system. The load balancing concept is to make resource utilization effectively and improve the response time of the jobs. The framework model of our proposed load balancing system for multiple workflows is shown in Fig.1. It describes the working mechanism of load balancing technique in IaaS cloud environment [11]. This architecture consists of three modules. (1). "Task Selection". In this module, first the clients submit their tasks to the server which is present in the IaaS cloud environment; the tasks are placed in the queue and send to the server. The server gets the task which is present in the queue and it navigates to the load balancer which is present in the server where it also has the details and information about resource connected to it. (2). "Resource Selection and Monitoring". This module selects the virtual resources for the appropriate tasks by the assignment approach. Once it enters into the load balancer it will check whether it have the sufficient resource as per the incoming task. If the number of task is equal to the to the number of the resource then the task is submitted to the respective resource as per the cost matrix table in order to utilize the virtual resource efficiently and to finish the task in the minimum execution time. If the number of tasks is not equal to the number of resources then the load balancer identify other virtual machines for each task as per the cost matrix table by assignment approach and apply the round robin scheduling approach to the task which are waiting in the queue for the assigned virtual machine. (3). "Status update phase". In the status update phase we have to update the status of the virtual machines i.e., how it took to execute the task on which virtual machine.

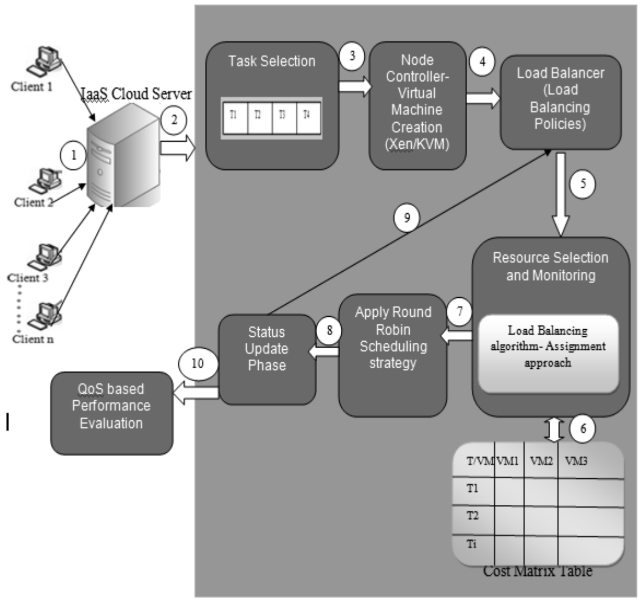

Fig.1. Architectural Framework Model

4.1 TASK SELECTION

The first module of this paper is selecting the task. In this module we have to divide the application into different tasks and place it onto task pool and also we have to specify the weightage of the task i.e., execution time of each task. For example consider the web application; this application can be divided into different tasks. First the clients submit their tasks to the server which is present in the cloud environment; the tasks are placed in the queue and send to the server. The server gets the task which is present in the queue and it navigates to the load balancer which is present in the server where it also has the details and information about resource connected to it. In this module we have considered a web service which has different task and this web service application are placed in the server and made the user to access these applications from the client machine. Each task of our applications is specified as "t_i" where "t" represents the task and "i" represents the number of the tasks.

4.2 RESOURCE SELECTION AND MONITORING

In this module we have to select the available resources depending upon the specification of our task by using load balancing condition. To maintain load balancing, load balancing policies simultaneously checks the resource limitations within the cloud and selects corresponding virtual resources. First check for number of tasks equal to the number of available resources. If so, assign the task to the respective resources as per the cost matrix table. If the number of task is not equal to the number of resource then apply the load balancing condition using assignment approach as per the cost matrix table.

Then, we have installed the LINUX environment and created the virtual machines using KVM virtualization and loaded Windows 7 operating system on each virtual machine and loaded MongoDB database application on it. When the user sends the request to server, the server identifies the task and directs it to the particular virtual machine. The virtual machine processes the request and sends the corresponding response to the client request. If any one of virtual machine have overloaded apply load balancing policies and we can redirect the incoming request to the other virtual machines to balance the load among cloud environment. Here in different virtual machines databases are loaded related to the application. If more number of user points to the database of the single virtual machine, hence the virtual machine load will get increased and hence there may be a slowdown of virtual machine or may even shutdown. Hence to minimize the load of virtual machine we can point the incoming request to the other virtual machine as per assignment approach. By this technique we are maintaining the load balancing among different virtual machines.

After applying assignment approach tasks must be assigned to the resources based on the round robin scheduling condition. Then implement the round robin approach to execute the tasks which is present in the queue based on specified time slots on the respective virtual machines.

4.3 STATUS UPDATE PHASE

In the status update phase we have to update the status of the virtual machine i.e., how long it took to execute the task on each virtual machine. The status update phases are represented in the form of table to represent the status of CPU speed, memory and disk of different virtual machines. The status update phase is used to represent the availability of the virtual machines whether it is in busy state or in the idle state so that it can be allocated for other task if it is free. The expected output of this architecture is to make use of available resource efficiently and to finish the task in the minimum cost. Finally we obtain the performance analyze graph for various performance metrics which use to show the performance of different virtual machines.

5. RESOURCE SELECTION AND MONITORING APPROACH

5.1 DESIGN MODEL OF LOAD BALANCING MECHANISM

Efficient load balancing [12] is used to find out the effective solution for a prticular problem. In this paper for the load balancing condition we have proposed the assignment approach for the effective and efficient utilization of available virtual machine.The assignment approach [13] is mostly used for the cost minimization and effective utilization of cloud resources. The following algorithm explains the load balancing condition among different virtual resources which are available in the cloud environment.

Algorithm: Resource Selection and Monitoring using Assignment approach.

Input: Task 't_i' in the task pool, $t_i \in T = \{1, 2,\ldots.n\}$, Resource '$r_j$' in the cloud data center, $r_j \in R = \{1,2,\ldots..m\}$, Cost Matrix ($C_{n \times m}$) table.

Begin

for $t_i \in T$ **do**
  for $r_j \in R$ **do**
  Consider the Cost Matrix table ($C_{n \times m}$)
  Check the resource availability for all the tasks
  for each task t_i calculate min(e_k) between resources **do**
    $L_a \leftarrow t_i(e_k) - min(e_k)$
for each resource r_j calculate min(e_k) between the tasks **do**
    $L_b \leftarrow t_i(e_k) - min(e_k)$
Find $r_j(e_k) = 0$
    for each task t_i re-compute $min(e_k)$ among resources **do**
    Prepare List (L_c) with execution times in resources except $t_i(e_k)$ is zero
    Prepare List (L_d) with execution times consider by both tasks and in resources
    Calculate $L_c - min(e_k)$
    Calculate $L_d + min(e_k)$
    Reconstruct cost matrix ($C1_{n \times m}$)
If tasks are in ready queue **then**
    task scheduler re-compute all tasks and then repeatedly do: (apply Round Robin scheduling algorithm)
    specify the time quantum value based on execution time of tasks upon assigned resources
    Allocate the task to the resources which is best fitted based on assignment mechanism
    Insert the task into next round of queue, if the time quantum value is expired
    Maintain slot table for local mapping to record execution schedule of resources
else
    break
Endif
Calculate average execution time of tasks
Calculate throughput
Calculate resource utilization rate
Calculate scheduling success rate
endfor
endfor

31. **endfor**
32. **endfor**
End

The main objective of the above algorithm is to develop optimized load balancing algorithm in IaaS cloud environment that aims to utilize the cloud resources of the virtual machine efficiently. This approach is to identify the virtual cloud resources that must be suitable for all the applications and minimize the cost of applications. The algorithm minimizes the average execution time, maximizes the throughput, increases the resource utilization rate and increases the scheduling success rate. The final results are measured using the following performance metrics Eqns. (1), (2), (3) and (4).

Average execution time (Te_k): Average execution time of tasks with resources is calculated using Eq.(1).

$$Te_k = \sum_{i=1}^{n} \left(t_i\left(e_k\right)/n\right) \quad (1)$$

Here n is the number of tasks and e_k is the execution time of tasks.

Throughput: Throughput (T_t) is calculated based on number of tasks and average execution time of each tasks.

$$T_t = \text{No. of tasks} \times \text{Average execution time of each tasks } (Te_k) \quad (2)$$

Network Transmission and Reception Rate: The network transmission and reception rate is based on the total number of data bytes (kbps) are sent and received from the virtual machines on each and every request and response to the user.

Resource Utilization Rate (ru_j): Resource utilization rate is the subtraction of starting execution time from finishing execution time of tasks on each resources.

$$ru_j = \Sigma t_i \text{ where } t_i \text{ will be executed on } r_j\ (te_i - ts_i) \quad (3)$$

where, te_i is the finishing time and ts_i is the start time of task t_i on resource r_j.

Scheduling Success Rate ($SSR_{i,j}$): It is based on effective resource utilization of resources by the tasks.

$$SSR_{i,j} \rightarrow \Sigma t_{i=1}(ru_j/m) \quad (4)$$

where, ru_j is the resource utilization rate of resource r_j and m is the number of tasks in each jobs.

5.2 LOAD BALANCING OPERATIONAL WORKFLOW

The assignment approach is a mathematical pragramming techniques which are used to find the optimal solution for a requested problems. Fig.2 gives the operational workflow for proposed mechanism.

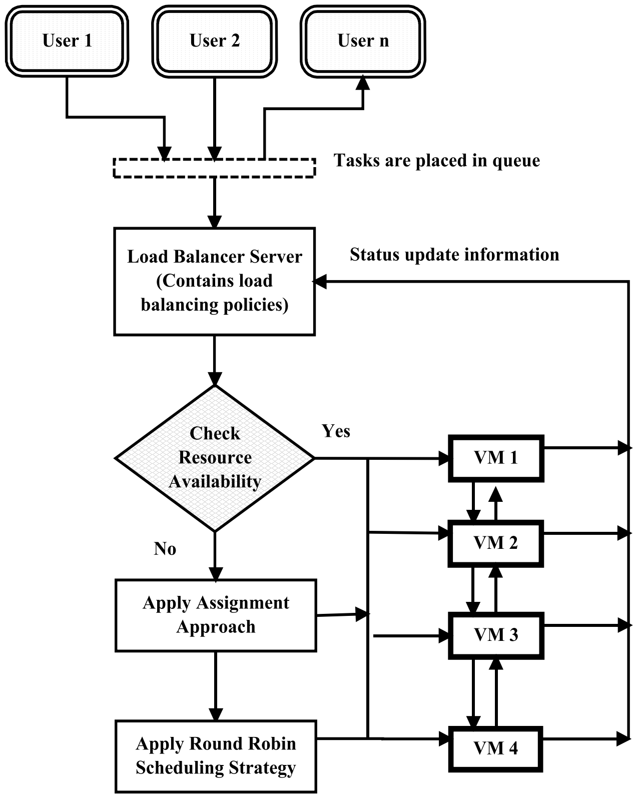

Fig.2. Load Balancing Operational Workflow

The above diagram explains the entire operation or workflow of this paper. For this paper we have created a web application in which ‘n’ number of users is allowed to access the virtual machines efficiently from the client side. First the different users uses the application and submit the task to the server. The users request are placed in the task queue and it is sent to the load balancer. The load balancer checks for the available virtual resource which are connected to it and it also has the status information of every virtual machine which are connected to it. The status information represents the status of every virtual machine, such as whether it is in busy state or in available state. Based on the status of virtual machines the task can be allocated to the virtual resources. If number of tasks are available to number of available virtual resources the tasks are loaded to the appropriate virtual machine as per the cost matrix table. If the number of incoming tasks is not equal to the number of available virtual resources then we apply the assignment approach algorithm to find the virtual machine for the requested application and implement the round robin approach to execute the task which are present in the queue based on specified time slots of the respective virtual machines.

6. EXPERIMENTAL RESULTS

6.1 EXPERIMENTAL SETUP

The Ubuntu operating system is installed to support the virtualization of the resources using the hypervisors like KVM which provides the KVM virtualization for creating the virtual machines. The required OS is installed in the virtual machines and MongoDB database for the application also loaded in the virtual machines. The server side program is loaded into Ubuntu OS which act as a server and also act as a load balancer. The server side program consists of load balancing policies. By default the user are allowed to access the first virtual machine when the capacity of the first virtual machine get exceeded then the status update phase make note of that virtual machine information and it will redirect the incoming request to the another virtual machine as per the cost matrix table. If there are more tasks assigned for particular virtual resource then the task are placed in the queue and virtual machine executes these task which are present in the queue as per the round robin scheduling approach. Meanwhile the performance (like CPU Speed, Memory, Disk Space, Network transmission and reception rate) of all the virtual machines are monitored using performance analysis graphs.

6.2 RESULTS

The Fig.3 shows the virtual machine resource with MongoDB database and Fig.4 shows two virtual machines with windows 7 OS. Fig.5 and Fig.6 shows the web application for load balancing mechanism and database status on each cloud resources.

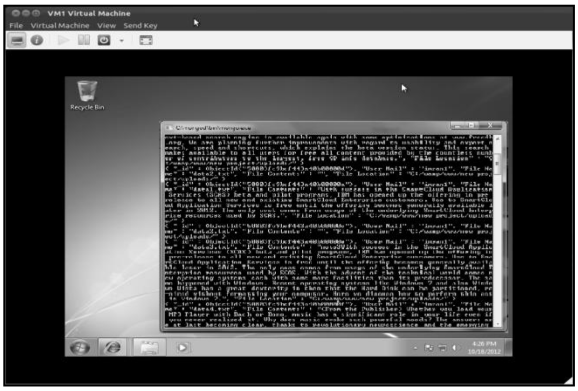

Fig.3. Cloud Resource 1 (VM1)

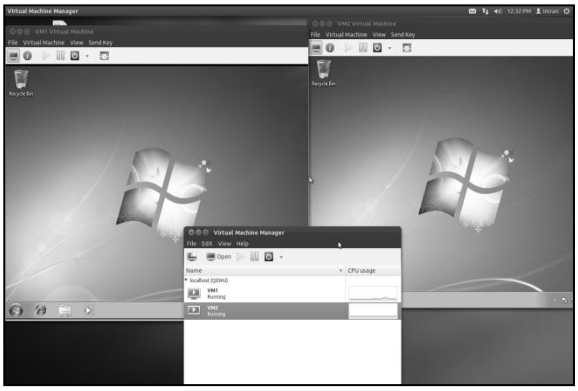

Fig.4. Windows Virtual Machines

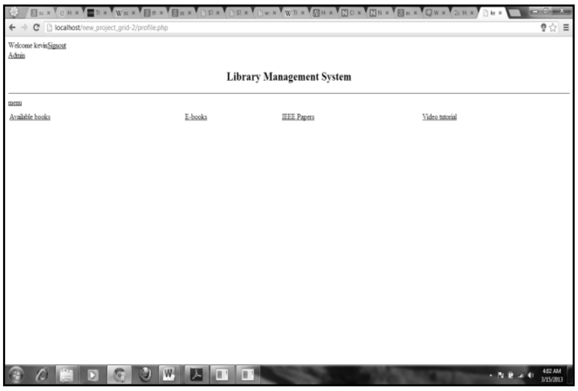

Fig.5. Load Balancing Web Application

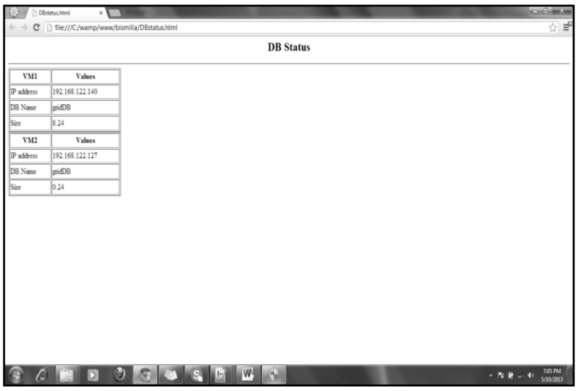

Fig.6. DB Status on Cloud resources

6.2.1 Resource CPU Usage:

The Fig.7 depicts the performance of CPU usage of VM1, VM2 & VM3. In this graph the first virtual machine VM1 is accessed by the application and then the VM2 and the VM3 respectively. At beginning the VM1 has 20% of CPU usage. After some time the CPU usage of VM1 is increased to 55% at that time the new VM is identified (VM2) and new incoming requests are redirected to VM2. At certain time the VM2 load also meets the threshold value of 50% and again new VM is identified (VM3) and further incoming request are directed to VM3. By this proposed approach the different virtual machines are equally balanced at some point and idle virtual cloud resources are used efficiently.

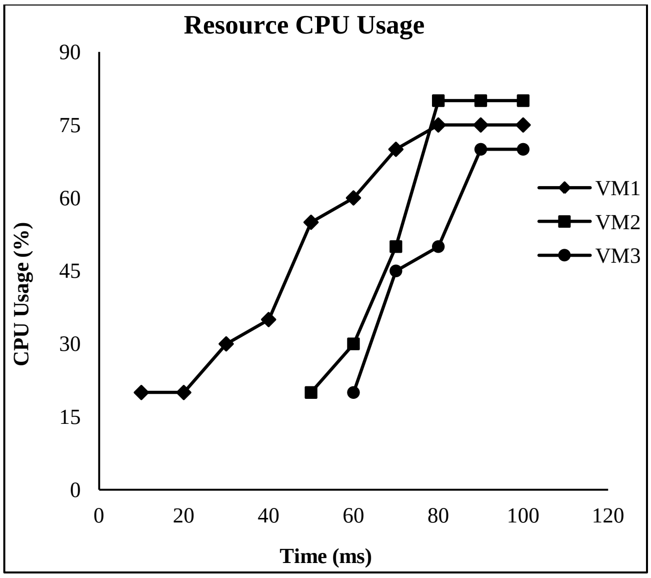

Fig.7. CPU Usage

6.2.2 *Resource Memory Usage*:

The Fig.8 represents the memory usage among different virtual machines (VM1, VM2 & VM3). At beginning every virtual machine has reasonable memory performance. Later the first virtual machine is get accessed and if the performance of first virtual machine goes down or if it attains the threshold value then the load balancer point the request to the other virtual machine which has high memory space. At some point, by this process the load is equally balanced among different virtual resources.

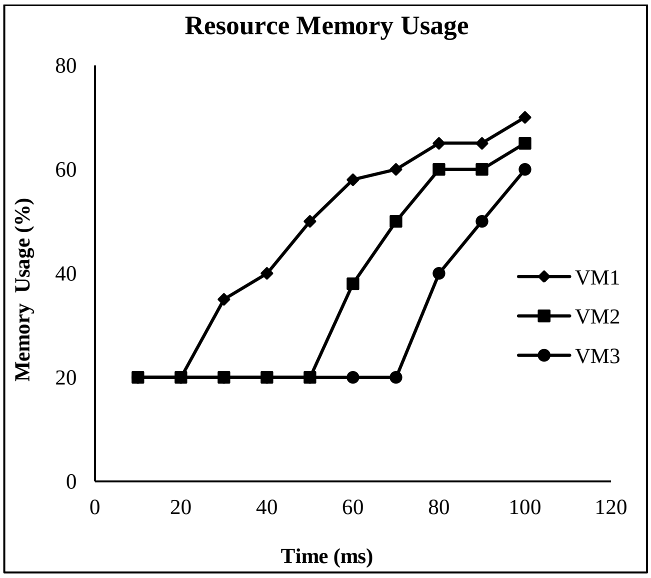

Fig.8. Memory Usage

6.2.3 *Average Execution Time*:

The execution time (in sec) of each tasks on corresponding virtual machines are taken and those values are plotted in the cost matrix table. Average execution time value is calculated using the Eq.(1). Fig.9 shows the average execution time of tasks on each cloud resources. So from the graph, the average execution time of tasks in VM2 is less than the execution time of tasks in VM1. Because tasks are running in VM2 effectively utilizes the cloud resource compare to VM. And also both the virtual resources are instantiated with different configurations within the node controller.

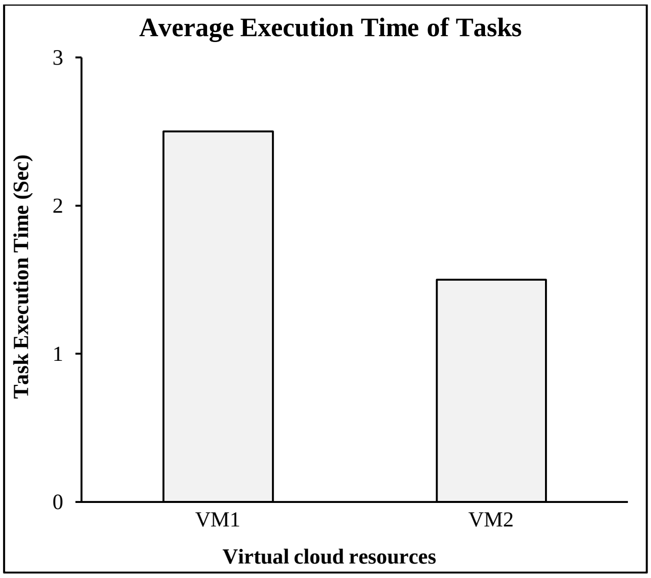

Fig.9. Average Execution Time of Tasks

6.2.4 *Throughput*:

The throughput of virtual resources on different tasks are calculated by the product of number of tasks and average execution time of each tasks using the Eq.(2). Fig.10 shows the throughput value of virtual cloud resources. Based on the average execution time calculation, compare to VM1, VM2 maximizes the throughput value.

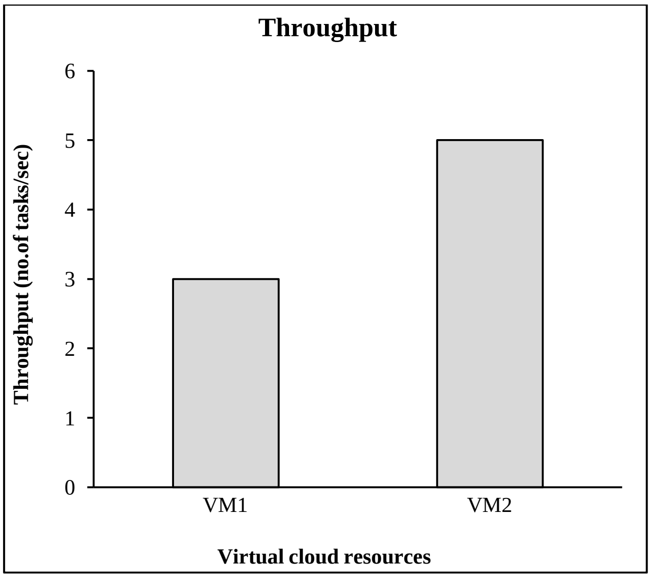

Fig.10. Throughput of virtual resource

6.2.5 *Network Transmission and Reception Rate*:

The network transmission and reception rate is to find the total number of bytes received and sent from the virtual machines on each and every request and response to the user who accessing the virtual machines from the client side. Here the request and response are based on the amount of user interact and how much data they upload or download from the virtual resources. Fig.11 shows probably the transmission and reception rate of VM1 is high based the number of tasks when compare to VM2 resource.

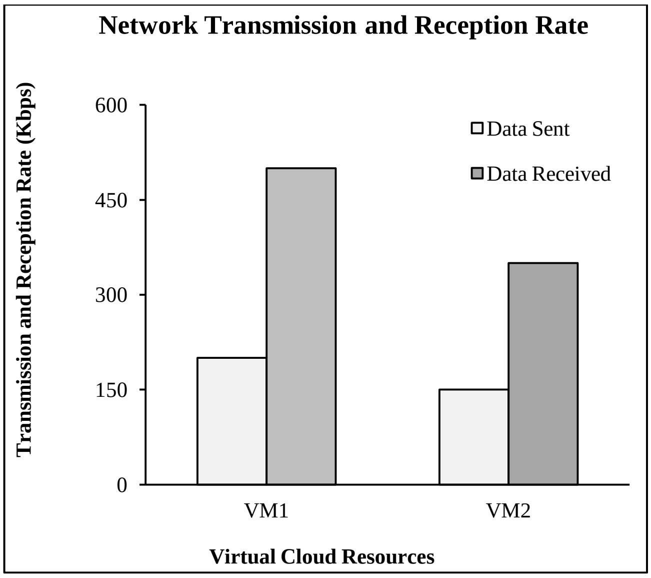

Fig.11. Network Transmission and Reception Rate

6.2.6 Resource Utilization Rate:

The resource utilization rate of tasks are calcualted based on task execution time values on resources using the Eq.(3). Resource utilization rate is the subtraction of starting execution time from finishing execution time of tasks on each resources. Fig.12 shows the resource utilization rate of all the tasks on each virtual cloud resources. From the graph, resource VM1 is effectively utilized by tasks when compare to VM2. So, the resource utilization rate is high in VM1.

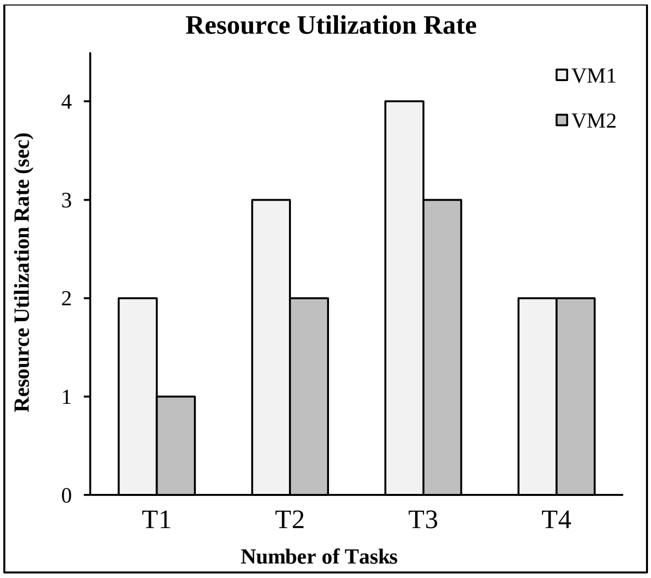

Fig.12. Resource Utilization Rate

6.2.7 Scheduling Success Rate (SSR):

The scheduling success rate of tasks are based on effective utilization of resoucres by the tasks and it is calculated using the Eq.(4). Fig.13 shows the scheduling success rate of tasks on each resource. When compare to VM2, the scheduling success rate is high in VM1, because the tasks are effectively utilized the VM1 resources.

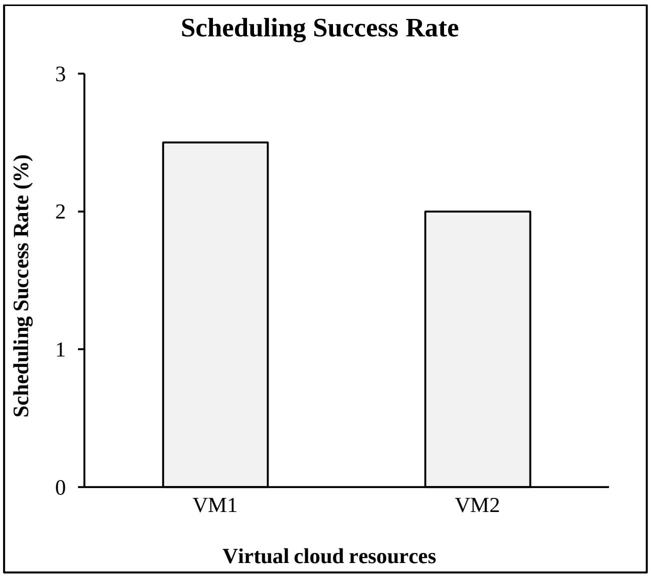

Fig.13. Scheduling Success Rate

7. CONCLUSION

In this paper we discuss QoS based load balancing mechnaism for multiple workflows in IaaS cloud environment. We have considered load balancing conditions for independent tasks. Some load balancing approaches based only on the CPU and Memory usage and not concenterated at execution time of tasks. So we have proposed the assignment approach based on cost matrix table which balance the load by redirecting the task to the assigned virtual machines. This approach is based on the execution time of tasks on each virtual machines. Finally the CPU, memory usages, average execution of tasks, throughout time, resource utilization rate and scheduling success rate of each virutal machines are monitored based on load balancing conditions. This work is based on load balancing among the independent tasks on virtual cloud environment and the future work considers all the dependent tasks in IaaS cloud environment.

REFERENCES

[1] Cary Landis and Dan Blacharski, "*Cloud Computing Made Easy*", Version 0.3.

[2] Bhaskar Prasad Rimal, Eummi Choi and Ian Lumb, "A Taxonomy and Survey of Cloud Computing System", *Proceedings of the 5th International Joint Conference on INC, IMS and IDC*, pp. 44-51, 2009.

[3] Tharam Dillon, Chen Wu and Elizabeth Chang, "Cloud Computing: Issues and Challenges", *Proceedings of 24th International Conference on Advanced Information Networking and Applications*, pp. 22-33, 2010.

[4] Ioannis Psoroulas, Ioannis Anagnostopoulos, Vassili Loumos and Eleftherios Kayafas, "A Study of the Parameters Concerning Load Balancing Algorithms",

IJCSNS International Journal of Computer Science and Network Security, Vol. 7, No. 4, pp. 202-214, 2007.

[5] Sandeep Sharma, Sarabjit Singh and Meenakshi Sharma "Performance Analysis of Load Balancing Algorithms", *World Academy of Science, Engineering and Technology 38*, pp. 269- 272, 2008.

[6] B. Sotomayor, R.S. Montero, I.M. Llorente and I. Foster, "Virtual Infrastructure Management in Private and Hybrid Clouds", *IEEE Internet Computing*, Vol. 13, No. 5, pp. 14-22, 2009.

[7] Martin Randles, David Lamb and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", *Proceedings of IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*, pp. 551-556, 2010.

[8] Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanzadeh and Christopher Mcdermid, "Availability and Load Balancing in Cloud Computing", *International Conference on Computer and Software Modeling*, Vol. 14, pp. 134-140, 2011.

[9] Jaspreet Kaur, "Comparison of load balancing algorithms in a cloud", *International Journal of Engineering Research and Applications*, Vol. 2, No. 3, pp. 1669-1673, 2012.

[10] Mousumi Paul, Debabrata Samanta and Goutam Sanyal, "Dynamic Job Scheduling in Cloud Computing based on Horizontal Load Balancing", *International Journal of Computer Technology Applications*, Vol. 2. No. 5, pp. 1552-1556, 2011.

[11] Aobing Sun, Tongkai Ji, Qiang Yue and Feiya Xiang, "IaaS Public Cloud Computing Platform Scheduling Model and Optimization Analysis", *International Journal of Communications, Network and System Sciences*, Vol. 4, No. 12, pp. 803-811, 2011.

[12] Meenakshi Sharma, Pankaj Sharma and Dr. Sandeep Sharma, "Efficient Load Balancing Algorithm in VM Cloud Environment", *International Journal of Computer Science and Technology*, Vol. 3, No. 1, pp. 439-441, 2012.

[13] Shakkeera L and Latha Tamilselvan "Task Scheduling System for Multiple Workflows in Virtual Cloud Environment using Load Balancing Techniques", *European Journal of Scientific Research*, Vol. 94, No. 1, pp. 120-131, 2013.

[14] Cloud Computing and distributed computing. Available at, http://www.cncloudcomputing.com/.