IJCTA, 9(2) 2016, pp. 1023-1034

Enhanced Scheduling Approach Using Heuristics Flow Equilibrium Based Load Balancing Algorithm In Cloud

S. Selvi* and B. Kalaavathi**

ABSTRACT

Scheduling based on load balancing is a very big riddle in Cloud computing. Various techniques are available to optimize cloud scheduling so as to effectively utilize the cloud resources and thereby improving the response time of the users. Users on request for resources to execute their job, providers either allocate them a new Virtual Machine (VM) or an ideal existing Virtual Machine or schedule accordingly. Response time varies based on the distribution of load to the VMs. Though the load sharing is balanced statically, there may be unavoidable circumstances dynamically due to which load distribution becomes uneven. Many existing technologies are available to balance the load among the VMs, but still there are drawbacks and hence it is still in research. This paper proposes Heuristic Flow Equilibrium based Load Balancing (HFEL) algorithm to improve scheduling in the Cloud. HFEL algorithm would reduce the latency, reduce the response time and increase the throughput than compared to the existing systems

1. INTRODUCTION

Cloud is a massively using computing model in the internet world. Nowadays customers prefer cloud for doing their work, since it is a metered service. As the number of user requests increases, the challenges also gets increased which has to be solved by both the cloud providers and cloud users [1]. When the user submits the job to the providers, the provider analyzes the job, and checks the availability of resources. Then the provider assigns the tasks to the new virtual machines or to the existing one based on their load in a static manner [2].

Virtualization helps the providers to schedule the incoming tasks based on their needs. Each physical machine encompasses many virtual machines. Load balancing plays a significant role in the optimization of scheduling in Cloud. Scheduling gets optimized both in static and dynamic manner. Improved response time, increased resource utilization and achieving maximized QoS are the parameters to be concentrated in scheduling [3]. Though incoming loads are balanced among the virtual machines initially, there will be a high degree of unpredictability of resource utilization by the running tasks and there may be the over dump of new tasks to the overloaded machines which is a serious issue to be monitored and which may be the root cause for the deviation in the submission of results to the user, that leads to increase in the response time to the users than what he/she is expected. Balancing the loads helps in solving the above issue [4].

Load Balancing is achieved by migration of running tasks or infrastructure from heavy loaded machines to lightly loaded machines which will reduce the response time of the user. Migration is majorly classified into Process Migration and Live Migration [5]. Process Migration is further divided into Operating System migration, User level migration and Object based migration. Process Migration concentrates on the transferring of running tasks from higher tasks machine to a lower tasks machine, whereas Live Migration

* Assitant Professor/CSE, Angel College of Engineering and Technology, Tirupur, *E-mail: selviresearch@gmail.com*

** Professor & Head /CSE, K.S.R. Institute for Engineering and Technology, Tiruchengode.

transfers the virtual infrastructure from one machine to another machine. Our work focuses on process migration. Heuristic Flow Equilibrium based Load Balancing algorithm is the proposed work of our paper to balance the load dynamically, thereby achieving the QoS of the user. HFEL uses indexing based mechanism to find out the heavy loaded and least loaded machines. Number of virtual machines included in the physical machine depends upon its level of configuration. Our paper considers on memory, bandwidth, speed, I/O access to determine the load of virtual machines.

2. RELATED WORK

Achievement of an efficient scheduling using load balancing is one of the approaches to provide an expected Qos for the users. Various researches have been undergone on load balancing among the virtual machines and it has to be done as a continuous process in order to reduce the response time of the users.XuChaoqun *et al.* [6] proposed a load balancing approach with key resource relevance(KRR) and they considered number of tasks, average waiting time and assignment of loads to the VMs as the primary factors for the optimization of scheduling in Cloud. The authors formerly divided the virtual resources in to groups by creating VMs and categorized on the basis of relevance between the user tasks and the priorities of the tasks are being calculated dynamically and then assigned to their appropriate VMs. The paper [7] sketched out Artificial Bee Colony(ABC) for balancing the load in Grid. The ABC algorithm tried to optimizes the algorithm based on the behavior of honey bee swarm. The makespan and deadlines are the constraints considered to achieve load balancing. Omer et. al. In his paper [8] took the challenge of running High Performance Computing (HPC) jobs in the virtual machines on the grid environment. The authors aimed at better provisioning of underlying resources and customization of execution environment at run time.

The paper [9] developed a deadline aware algorithm to reduce the job execution delays and thereby optimizing jobs dynamically to meet their job obligations. The algorithm encompasses training phase and steady phase in the scheduling model that not only maximizes the success rate of the jobs but also concentrating on the high utilization of virtual resources. The authors divided the user jobs based on unlimited deadlines and tight deadlines. Jobs with unlimited deadlines are those which are aimed to finish their jobs with no time constraints and not bothering about the virtualization overhead. Jobs with tight deadlines are those that are strictly adhered for the completion of their jobs with the prescribed deadline. The authors in [10] projects on two approaches for efficient resource management in a grid environment. First approach attempts on sub optimal scheduling based on prior information about local schedules and resource reservation status. Cyclic Scheduling Scheme was used for this approach. Backfilling approach was undertaken for the second approach to dynamically perform rescheduling with an aim to improve the overall resource utilization.Genetic algorithm based load balancing was undergone in Koushik et.al's paper[11] to make an optimal utilization of resources in a cloud environment. But the authors considered all the user's jobs were all of the same priority which would be impossible for all the cases ina real time environment. A survey paper of Karthik *et al.* [12] gave an overview on various background techniques of offloading computations in mobile systems.A monitoring tool [13] was proposed for KVM based virtual machines to identify bottlenecks among virtual machines and resource profiling with a motto of measuring the performance of KVM hosted web servers.

Grain size was the parameter considered to check the load balancing in the heterogeneous systems both in a static and dynamic environme nt. The authors in [14] presented a performance model to find out an optimal value of grain size and improve the performance in the context of sparse grid interpolation. A soft computing based load balancing was developed by the authors in [15] and they allocated the resources to the virtual machines based on the analysis of the properties of the tasks. Branko Radojevic [16] propose a decision making based load balancing approach for dividing the whole tasks into sub tasks using the session switching process. The session switching is done with the help of the round robin process. Ratan Mishra

[17] uses the Ant Colony optimization technique for distributing the whole tasks into the different sub tasks which avoids the drawback of the heavy load in the network.

Martin Randles [18] uses the round robin techniques for dividing the process between processors. These divided processes are equal, but the job processing time is different from process to processors. This algorithm mostly used in the web servers and the services are distributed manually and equally. Yi Lua, [19] uses the power of medium approach for reducing the system overloads by applying the join idle queue process. Then the process manages the tasks and resources in the virtual machine.

3. DISCUSSION ABOUT DIFFERENT ALGORITHMS

In this section discusses the various discussions about the cloud load balancing algorithm like Join-idle Queue, Ant Colony Optimization algorithm and the decision making based queue process.

Join-Idle Queue based Load Balancing Algorithm

This algorithm uses the single shared queue and the jobs are waiting in the queue while one job completed in task it transfer into the next queue. If the server process one job then the other requested jobs need to wait in the idle queue. After completing the process of the particular job the next job is chosen according to the arrival time of the job. Then the arrival time is calculated as follows,

$$\frac{\lambda n}{m} . \rho . \frac{1}{r(1-\lambda)} = \frac{\lambda \rho}{(1-\lambda)} \tag{1}$$

Where λn is the job arrival rate in the queue.

$(1-\lambda)$ is the job proportional rate of the job in the queue.

Based on the job arrival rate it distributes to the different server for managing the load in the cloud environment.

Ant Colony Optimization based Load Balancing algorithm

Ant Colony Optimization is worked based on the behavior of the insects which have the limited amount of memory and large random components. The behavior of the insects used to estimate the shortest path while making the large trail in the optimization process. During the load balancing process, each task should be maintaining the probability table, source and destination point to manage the task while distribution. Initially the tasks are analyzed and the variables and the table has been initialized. Then the time of each task is calculated and the optimized solution has been determined using the minimum execution time of the task. The new task value is updated as follows,

$$X_{new} = \min \{f_{obj} (P_k)\}$$

Where k=1, 2, 3… K

After estimating the new task value the probability table value is updated simultaneously for assigning the tasks to each machine.

Round Robin Techniques based Load Balancing Algorithm

The next approach is Round Robin based load balancing technique which is applied in Honey Bee Foraging load balancing Algorithm. The algorithm uses the time for executing the particular task in which each task time is divided into the multiple slice based on the time scheduling process. The time is managed by using the bee optimization algorithm and the resources are allocated to the particular job. Based on the execution time the load is balanced in the distributed manner. These methods have some of the drawbacks like single queue management system, time exists and so on. So the proposed system uses the Heuristic Flow Equilibrium based Load Balancing (HFEL) algorithm based approach for scheduling task in the cloud environment.

4. PROPOSED WORK

This section discusses the proposed scheduling algorithm based load balancing in the cloud environment. In cloud, load balancing placing an important role because several users request services simultaneously during that time collusion or some issues like response time increasing, the cost may be increased [20]. For this above issue scheduling based load balancing is used to distribute the resources with efficient manner. So, the proposed system uses the Heuristic Flow Equilibrium based Load Balancing (HFEL) approach for scheduling and distribute the work in the cloud environment. Then the proposed system architecture is shown in the figure 1.

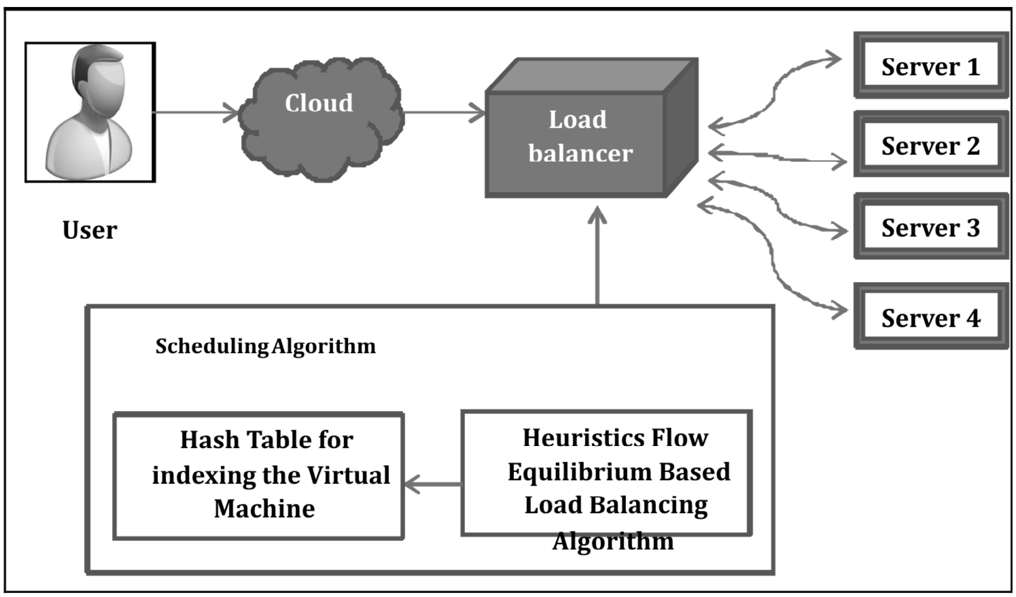

Figure 1: Proposed System Architecture

The above figure 1 explains that the working process of the proposed system. The cloud user request resources in the cloud environment. The related resources are fetched from the cloud service provider server and those services are balanced by the load balancer. The load balancer allocates, each process to the particular machine based on the scheduling algorithm. Then the step by step procedure of the Heuristics Flow Equilibrium Load Balancing algorithm is explained as follows.

Heuristics Flow Equilibrium Based Load Balancing Algorithm:

The proposed algorithm aims at optimizing the scheduling of incoming jobs by balancing the load among virtual machines throughout the execution of the tasks and thereby decreasing the response time of the users.

The algorithm uses the following heuristics:

- Number of virtual machines for each physical machine get varied based on its level of configuration.
- The incoming workflow is divided into sub tasks by considering the dependency relationship
- The sub tasks are then classified into simple, medium and complex based on their resource requirements
- Physical machines along with their virtual machines are identified for their lower, medium and higher configuration

- Each virtual machine is assumed to have resources such as memory, bandwidth capacity, speed and I/O access.

Each physical machine maintains an index that contains name of the virtual machine, number of tasks allocated to it, execution status of the virtual machines, resource status, utilization status and so on. The index table gets updated whenever a virtual machine is removed from physical machine or a new virtual machine is added to the physical machine. The execution status of every task is also updated in the table.

The incoming workflow is divided into subtasks and a classifier is used to classify the tasks based on its interdependency, complexity of the tasks and resource requirements. The tasks are divided based on the execution time and the minimum execution time tasks are grouped.The classified tasks are made into groups based on its level of complexity, dependency, etc and their loads are calculated. Then they are scheduled accordingly to the virtual machines group. A load balancer is used to perform the following operations:

(i) Calculate the capacity of each virtual machine and determine the cumulative total capacity

(ii) Update the indexes of each virtual machine when a new task is being assigned or completed tasks is removed from the VM.

(iii) Determine whether there is an exceed in the load of VMs when compared to their tolerable capacity and also check whether there is a load imbalance between virtual machine groups. Then the working flow of the load balancing process is shown as follows,

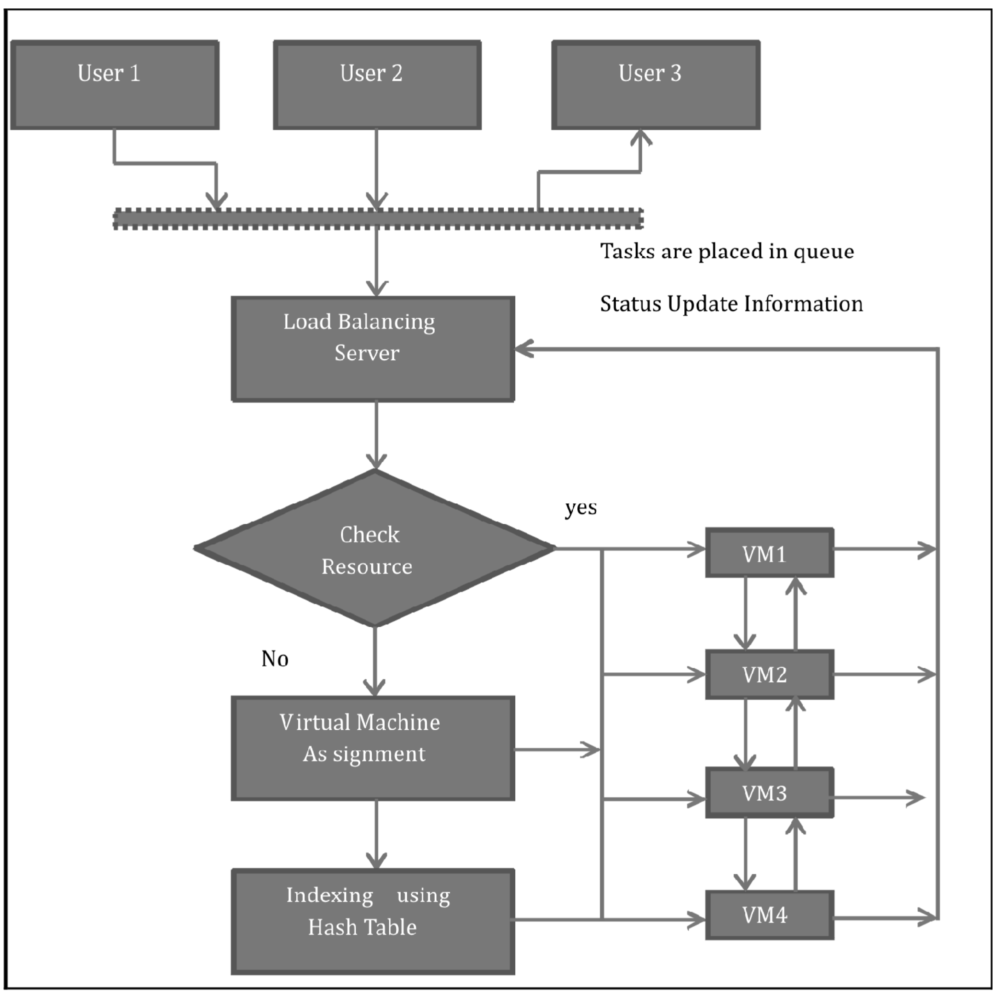

Figure 2: Load Balancing Workflow

The figure 2 describes the flow of virtual machine assignment based on hash table indexing and load balancing.Here the user tasks are placed in the queue and assigned to the virtual machine by using the hash indexing table. During the assignment, the resources availability of the virtual machine should be estimated.Based on the resource particulars, virtual machine has been identified and the resources are assigned to that virtual machine for reducing the response time. Then the scheduling and load balancing process is explained by using the following algorithm.

Algorithm for Heuristics Flow Equilibrium Based Load Balancing Algorithm

Input: Set of Incoming tasks T{T_1, T_2, T_3,…T_n } and set of Physical Machines having Virtual Machine as {P_1(VM_1,VM_2,VM_3,…VM_n} and P_2(VM_{n+1},VM_{n+2},…VM_m)}

Output: Every incoming task *i* optimally assigned to the virtual machines there by balancing the load dynamically.

Procedure:

Step 1: Classify the incoming tasks into n subtasks

Step 2: Group the sub tasks based on its dependency and complexity

Step 3: For every group of tasks, calculate the load,

do

{

$L_{VMi,} = \frac{N(T,t)}{VM_{bw_j}(t)}$ (Load balancing of each virtual machine)

Then calculate the resource utilization of the virtual machine

$R(T_i) = ET_i * \left[\frac{(n*R_k)}{m}\right]$ (Resource Utilization of each virtual machine)

}

Step 4: Allocate the task to virtual machine according to minimum load balanced and minimum resource utilization VM.

Mathematical Model to Calculate The Load Value and Resource Utilization

Let us consider an ordered set of m virtual machines as VM=(VM_1,VM_2,VM_3,…VM_n} and a set of n incoming tasks as T={T_1, T_2, T_3,…T_n}. Let us take Gmax as the maximum throughput obtained during the execution of tasks and ET_{ij} be the Execution Time for task T_i and Virtual Machine VM_j.

Execution Time of all tasks in VM_j can be calculated by the formula,

$$ET_j = \Sigma_{i=1}^{n} ET_{ij} \text{ (j lies between 1 to m)} \quad (1)$$

Throughput can be calculated by the following formula,

$$Gmax = \{\max_{i=1} CT_i, \max_{j=1} \Sigma_{i=1} ET_{ij}\} \quad (2)$$

where the first component gives the completion time of task i,

the second component gives the execution time of VMj in task i.

In addition the capacity, load, resource utilization, execution time and variance of the virtual machine has been calculated for determining the virtual machine allocation process which calculated as follows,

Capacity of VM_j

$$C_j = P_{nj} * P_{mips_j} + VM_{bwj} * VM_{i/o} \tag{3}$$

Where,

P_{nj} – Number of Processors in VM_j

P_{mips_j} – Million Instructions per second of all processors in VM_j

VM_{bwj} – Communication Bandwidth ability of VM_j

$VM_{i/o}$ -I/O Contention of VM_j

Overall Capacity of all active VMs

$$C = \Sigma_{i=1} C_i \tag{4}$$

where,

C_i – is the value of active VM_i calculated in *Eq. 3*

Load Calculation on a single VM

Total Length of tasks assigned to VM is given by,

$$L_{VMi,t} = \frac{N(T,t)}{VM_{bwj}(t)} \tag{5}$$

$N(T, t)$ – Number of tasks active at time ‘t’ on service queue of VM

$VM_{bwj}(t)$ - Communication bandwidth of VM_j at time ’t’

Load Calculation of all active VMs

$$L = \Sigma_{i=1}^{m} L_{VMi} \tag{6}$$

Execution Time of a Single VM

Now the Execution Time of a Single VM can be calculated as,

$$ET_i = \frac{L_{VMi}}{C_i} \tag{7}$$

Execution Time of all active VMs

Therefore, the Execution Time of all active VMs can be calculated as,

$$ET = \frac{L}{C} \tag{8}$$

Variance Calculation of Load

$$\sigma = \sqrt{\frac{1}{m} * \Sigma_{i=1}^{m} (ET_i - ET)^2} \tag{9}$$

$$VarianceV = \left(\frac{1}{m} * (ET_i - ET)^2 \right) \tag{10}$$

Overall Resource Utilization

$$R(T_i) = ET_i * \left[\frac{(n * R_k)}{m} \right] \tag{11}$$

$R(T_i)$ – Resources utilized for Task T_i

n- Number of Virtual Machines

R_k – Number of units of resources available in VM

m - Units of Resources that each server has

ET_i -Execution Time of Task i

Thus the proposed Heuristics Flow Equilibrium Based Load Balancing Algorithm schedule the job to various users with minimum response time and minimum resource utilization. Then the overall load of the particular virtual machine also reduced because it divides the whole task into sub tasks which lead to increase in the throughput of the load balancer. The performance of the proposed system is evaluated by using the experimental result and discussion.

5. PERFORMANCE ANALYSIS

This section discusses about the performance analysis of the proposed Heuristics Flow Equilibrium Based Load Balancing Algorithm using the execution time of the virtual machine, response time, throughput metrics. The experiments are carried out in Java based implementation and the virtual machines are chosen according to the data centers. Then the RAM, hard disks are selected and the resources are provided based on the task execution time. The table 1 represent that the overall performance analysis of the proposed system with several existing methods [21] such as Central Load Balancing Decision Module (CLBDM), Ant Colony Optimization Based Load Balancing Algorithm (ACO), Honeybee Foraging load balancing Algorithm (Honey Bee Foraging) and Join Idle Queue Load Balancing Algorithm (JIQ).

Table 1
Representation of Performance of the Load Balancing Algorithm in Cloud

Performance Metrics	*CLBDM*	*ACO*	*Honeybee Foraging*	*JIQ*	*HFEL*
Throughput	High	Low	Low	Moderated	High
Speed	Moderated	Low	Moderated	Low	High
Complexity	Moderated	High	Moderated	High	Low
Fault Tolerance	Low	High	Moderated	Low	High
Response Time	High	Moderated	Low	Moderated	Low
Performance	Moderated	Low	Moderated	Low	High

The above table 1 clearly shows that the proposed system provides the optimized results when compared to the other load balancing algorithms. The response time of the proposed load balancing algorithm is listed in the table 2.

Table 2
Response Time of the Different Load Balancing Techniques

Load Balancing Techniques	*Response Time (ms)*
CLBDM	5.36
ACO	4.16
Honey Bee	3.47
Foraging	2.01
JIQ	1.34
HFEL	0.33

The graph representation of the proposed system response time is shown in the figure 3.

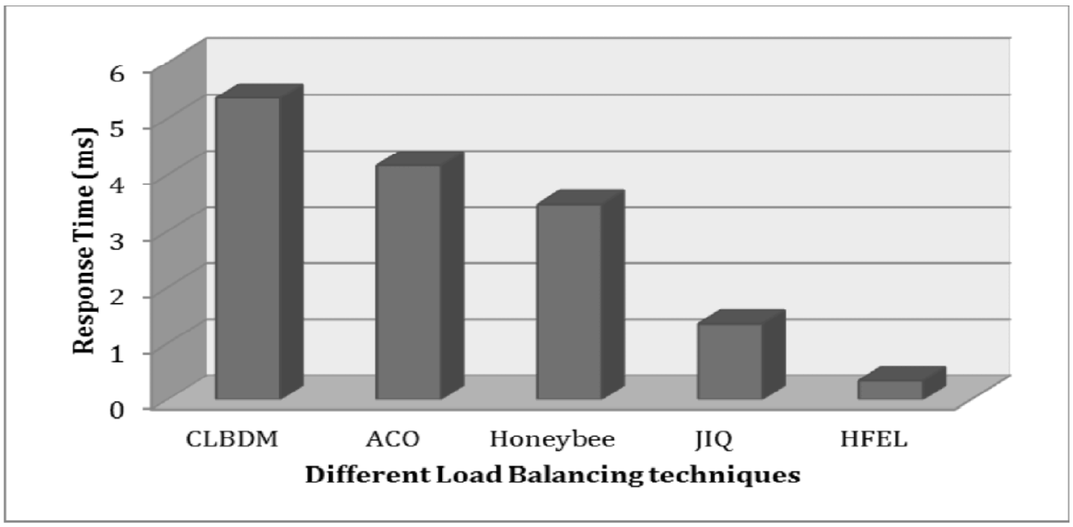

Figure 3: Response Time of the Different Load Balancing Techniques

The above figure 2 clearly shows that the proposed system consumes minimum time which means the algorithm produces the resources with minimum time when compared to other load balancing algorithm. Then the throughput of the proposed system is increased which is shown in the figure 4.

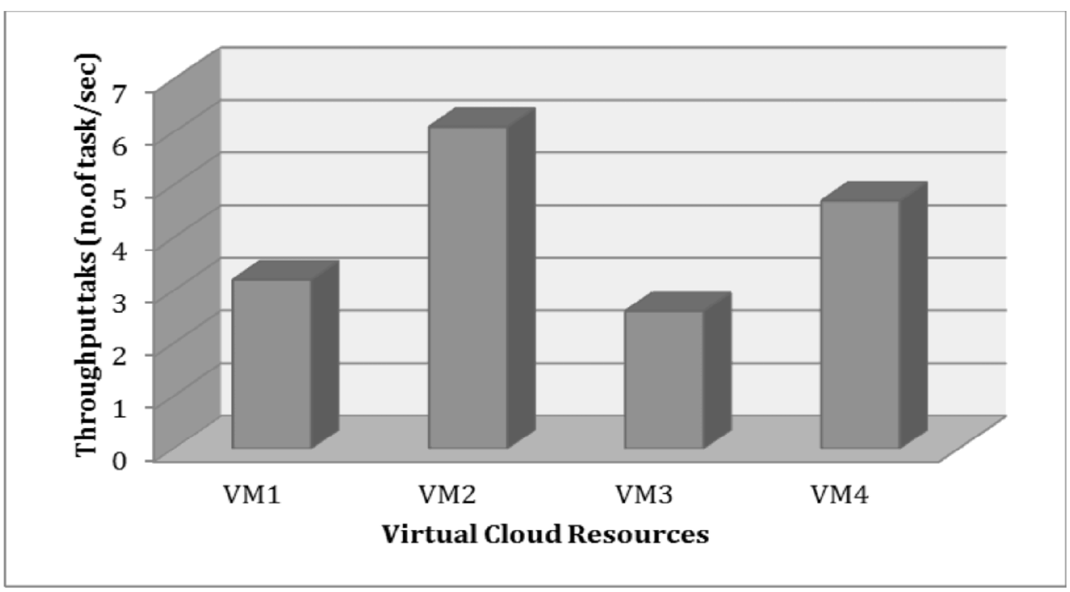

Figure 4: Throughput of the Virtual Cloud Resources

Thus the figure 4 explains that the throughput of the different virtual resource on various tasks which produces the highest throughput when compared to the different virtual machines. Based on the resources the scheduling utilization is explained in the figure 5.

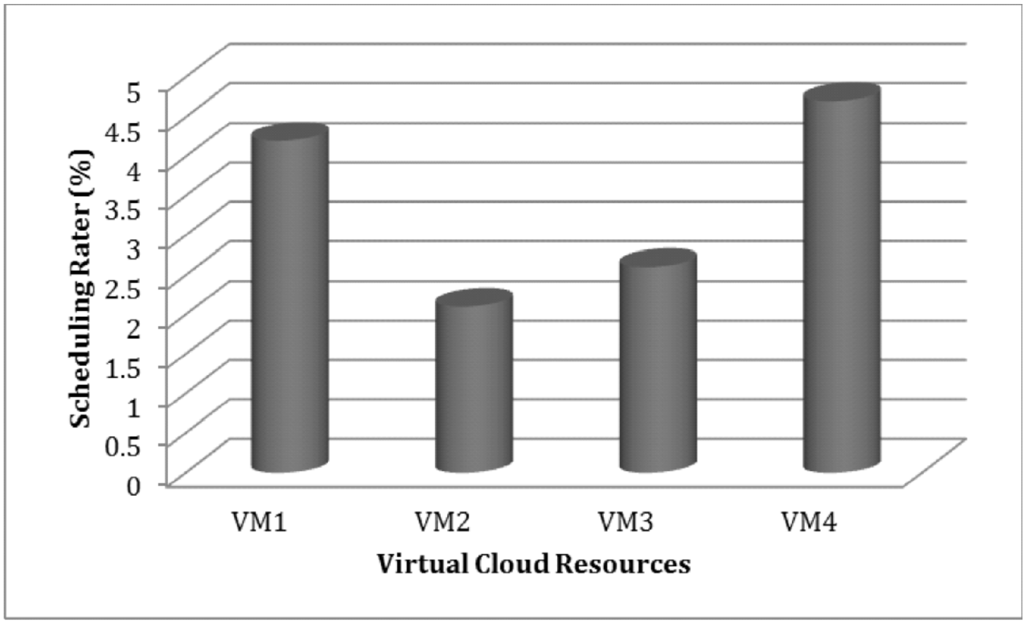

Figure 5: Scheduling Rate of the Virtual Cloud Resources

The figure 5 shows that the VM4 has high scheduling rate (97%) than compared to the other virtual machines which means that the tasks are effectively utilized by the VM4 for every resource. Based on the scheduling rate, response time, the task resource utilization rate is shown in following table 3.

Table 3
Resource Utilization Rate

Tasks	*Resource Utilization Rate*			
	VM1	*VM2*	*VM3*	*VM4*
T1	1.3	2.36	3.65	2.3
T2	2.6	1.35	4.39	3.14
T3	3.7	3.48	5.47	4.01
T4	4.01	3.47	4.36	5.46

Finally, the related graph representation of the resource utilization rate of different virtual machine using the proposed load balancing technique is shown in the figure 6

Thus the proposed system ensures the minimum response time based load balancing approach in the cloud environment.

6. CONCLUSION

Thus the paper discusses the proposed Heuristics Flow Equilibrium Based Load Balancing Algorithm for scheduling the various tasks to the different virtual machines in the cloud environment. The Heuristic method indexed each virtual machine based on the hash table which is used to schedule the process. During the scheduling process, each task is subdivided into different groups and the complexity, resource utilization of the virtual machine is estimated. Based on the complexity and resource utilization the task is allocated to

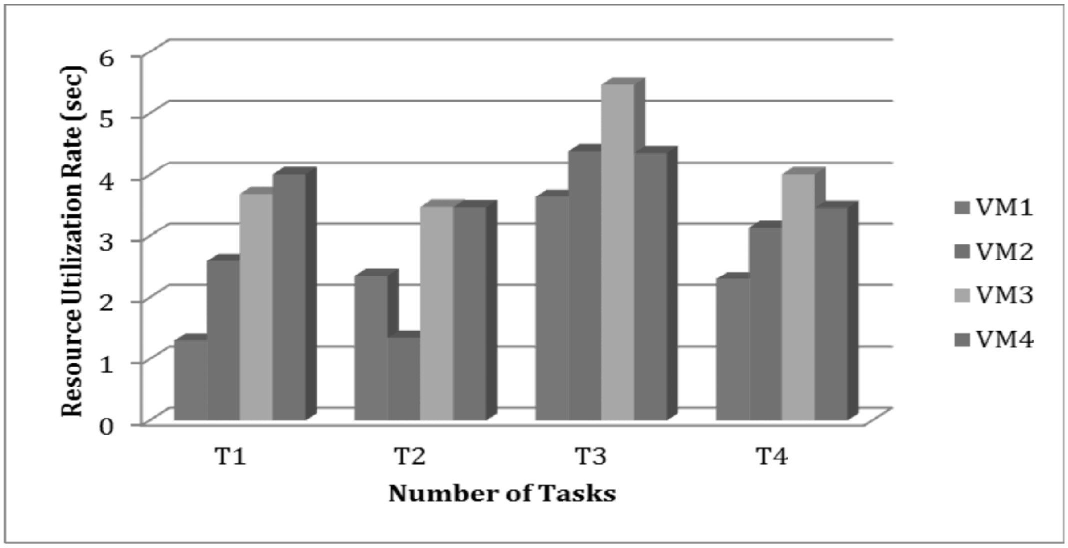

Figure 6: Resource Utilization Rate

the virtual machine with minimum response time. Thus the heuristic approach schedules the task and effectively utilizes the cloud resources and improves the response time of the users. Then the performance of the system is evaluated using the response time, average throughput and scheduling rate. Thus the proposed system consumes minimum response time 0.33 ms and the task scheduled rate is 97% when compared to the existing methods. This enhanced scheduled rate helps the tasks to utilize the virtual machines in an effective manner.

REFERENCES

[1] Mayanka Katyal, Atul Mishra, "A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment", Article can be accessed online at *http://www.publishingindia.com.*

[2] Abhijit A. Rajguru, S.S. Apte, "A Comparative Performance Analysis of Load Balancing Algorithms in Distributed System using Qualitative Parameters", *International Journal of Recent Technology and Engineering (IJRTE)* ISSN: 2277-3878, Volume 1, Issue 3, August 2012.

[3] Zhang Bo, Gao Ji ; Ai Jieqing, "Cloud Loading Balance algorithm" *International Conference on Information Science and Engineering in IEEE* 2010.

[4] Domanal, Reddy, "Optimal load balancing in cloud computing by efficient utilization of virtual machines" *International Conference on Communication Systems and Networks in IEEE* 2014.

[5] Domanal, Reddy, "Load Balancing in Cloud Computing using Modified Throttled Algorithm" IEEE International Conference on Cloud Computing in Emerging Markets, 2013.

[6] XuCh aoqun, ZhuangYi and ZhuWei," A Load Balancing Algorithm with Key Resource Relevance for Virtual Cluster" (2013) ,Vol 6, No. 5 *International Journal of Grid and Distributed Computing,* pp 1-16.

[7] Preeti Gulia and Deepika Nee Miku," Analysis and Review of Load Balancing in Grid Computing using Artificial Bee Colony" (2013), *International Journal of Computer Applications* (0975 – 8887), Volume 71– No. 20,

[8] Omer Khalid, Ivo Maljevic, Richard Anthony," Dynamic Scheduling of Virtual Machines Running HPC Workloads in Scientific Grids"(2009), IEEE.

[9] Stelios Sotiriadis, Nik Bessis, Pawel Gepner, Nicolas Markatos "Analysis of requirements for virtual machine migration in dynamic clouds".

[10] Victor Toporkov, Anna Toporkova, Alexey Tselishchev, Dmitry Yemelyanov, and Petr Potekhin,"Preference-Based Fair Resource Sharing and Scheduling Optimization in Grid VOs",(2014),Elseveir, ICCS 2014. 14th International Conference on Computational Science, Volume 29, 2014, PP 831–843.

[11] Kousik Dasgupta, Brototi Mandal, Paramartha Dutta, Jyotsna Kumar Mondal,Santanu Dam, "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing", Elseveir, International Conference on Computational Intelligence: Modeling Techniques and Applications,(CIMTA) (2013).

[12] Karthik Kumar, Jibang Liu, Yung-Hsiang Lu, Bharat Bhargava, "A Survey of Computation Offloading for Mobile Systems", © Springer Science+Business Media, LLC 2012, Mobile Netw Application, DOI 10.1007/s11036-012-0368-0.

[13] Ankit Anand, Mohit Dhingra, J. Lakshmi, S. K. Nandy ," Resource usage monitoring for KVM based virtual Machines".

[14] Alin Murara_su, Josef Weidendorfer, and Arndt Bode," Workload Balancing on Heterogeneous Systems: Case Study of Sparse Grid Interpolation".

[15] Brototi Mondal, Kousik Dasgupta, Paramartha Dutta, "Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach", *Procedia Technology* 4 (2012) 783–789.

[16] Branko Radojevic, Mario Žagar, "Analysis of Issues with Load Balancing Algorithms in Hosted (Cloud) Environments", available at., *https://bib.irb.hr/datoteka/536452.MIPRO_-_LB_v3.pdf*

[17] Ratan Mishra and Anant Jaiswal, "Ant colony Optimization: A Solution of Load balancing in Cloud", *International Journal of Web & Semantic Technology* (IJWesT) Vol.3, No.2, April 2012.

[18] Martin Randles, Enas Odat, David Lamb, Osama Abu- Rahmeh and A. Taleb-Bendiab, "A Comparative Experiment in Distributed Load Balancing", 2009 Second International Conference on Developments in eSystems Engineering.

[19] Yi Lua, Qiaomin Xiea, Gabriel Kliotb, Alan Gellerb, "Join-Idle-Queue: A Novel Load Balancing Algorithm for Dynamically Scalable Web Services, *http://research.microsoft.com/pubs/153348/idleq.pdf*.

[20] Omer Khalid, Ivo Maljevic, Richard Anthony ,"Deadline Aware Virtual Machine Scheduler for Grid and Cloud Computing" IEEE 24th International Conference on Advanced Information Networking and Applications, 2010.

[21] Dharmesh Kashyap, Jaydeep Viradiya, "A Survey Of Various Load Balancing Algorithms In Cloud Computing", *International Journal of Scientific & Technology Research*, Volume 3, Issue 11, November 2014.