ISSN:0975-9646

Improve the Efficiency of Load Balancing in Cloud Environment using DAG and Honey Bee Algorithm

Abhishek Kumar Tiwari,
M.Tech Scholar, CSE ,
OIST,Bhopal, India

Sreeja Nair,
Department of CSE,
OIST,Bhopal, India

Abstract -Load balancing is core part of public cloud computing. The process of load balancing increase the performance of cloud based services. Cloud based service provide hardware, software and platform as service. For the balancing of public cloud used two types of approach one is traditional approach and other is swarm based approach. The swarm based approach used particle of swarm optimization, ant colony optimization and glowworm swarm algorithm. for the balancing of load also used different policy such as busy and idle condition such technique are called cloud part ion based load balancing technique. In this paper modified the load balancing policy based on teacher based learning optimization. The teacher based learning optimization well knows meta-heuristic function used for the purpose of optimization and searching process. The modified load balancing policy simulated in cloudsim simulator. The cloudsim simulator developed in java technology and free available for research purpose. Our modified load balancing policy compare with two different techniques one is round robin and other is glowworm algorithm. our modified load balancing policy reduces load effect about 10-12%.

Keywords: - Honey Bee, Virtual Machine (VM), Cloud computing, LBMM.

I. INTRODUCTION

Cloud computing provides much more effective computing by centralized memory, processing, storage and bandwidth. It should make sure that the tasks are not loaded heavily on one VM and also ensure that some VMs do not remains idle and/or under loaded In cloud computing technology the data and applications are maintained using the internet and central remote servers. Over the last few years Cloud Computing has been gaining immense popularity where user can pay (as you use) for software, hardware [8].

To improve the response time of user's submitted applications so that there should be maximum utilization of available resources we use of load balancing algorithms. Load balancing methods aims to speed up the execution of applications by removing tasks from over loaded VMs and assigning them to under loaded VMs and execution of applications of resources whose workload varies at run time in an unpredictable manner.

Load balancing is the process of improving the performance of a parallel and distributed system through a redistribution of load among the processors or nodes As Load Balancing is one of the major issues related to cloud computing, the load may represent a CPU capacity, memory, network load etc. It is necessary to distribute the load equally among the nodes in a network. This results in agile and efficient performance of the system. Thereby it avoids heavily loading or under loading of nodes in a network.

Honey Bee Foraging Algorithm Achieves global load balancing through local server action Honey Bee Behaviour inspired Load Balancing [HBBLB] a technique which helps to achieve even load balancing across virtual machine to maximize throughput [1]. It considers the priority of task waiting in queue for execution in virtual machines. After that work load on VM calculated decides whether the system is overloaded, under loaded or balanced. And based on this VMs are grouped. New according to load on VM the task is scheduled on VMs. Task which is removed earlier. To find the correct low loaded VM for current task, tasks which are removed earlier from over loaded VM are helpful.

LOAD BALANCING

Load balancing is used to distributing a larger processing load to smaller processing nodes for enhancing the overall performance of system. In cloud computing environment load balancing is required distribute the dynamic local workload evenly between all the nodes. Load balancing helps in fair allocation of computing resource to achieve a high User satisfaction and proper Resource utilization .High resource utilization and Proper load balancing helps in minimizing resource consumption. It helps in implementing fail over, scalability, and avoiding bottlenecks.

Load balancing is a techniques that helped networks and resources by providing a Maximum throughput with minimum response time [6]. Load balancing is dividing the traffic between all servers, so data can be sent and received without any delay with load balancing. In cloud environment many algorithms are available that helps in proper traffic Loaded between all available servers .Most of them can be applied in the cloud environment with suitable verifications. In cloud computing environment load balancing algorithms can be divided into two main groups. first algorithm type is Batch mode heuristic scheduling

algorithms (BMHA) and second is online mode heuristic algorithms [4].

In BMHA Jobs are combined together when they are arriving in the system [3]. The BMHA scheduling algorithm will start after a fixed time period. The examples of BMHA based algorithms are: First Come First Served Scheduling algorithm (FCFS), Round Robin scheduling algorithm (RR), Min Min algorithm and Max Min algorithm. In On-line mode heuristic scheduling algorithm, all Jobs are scheduled when they are arriving in the system. The cloud environment is a heterogeneous system and in this speed of each processor varies quickly and easily. The online mode heuristic scheduling algorithms are more appropriate and better for a cloud environment.

Rest of this paper is organized as follows in Section II discusses about load balancing algorithm in cloud computing environment description, Section III discusses about the Problem Statement. Section IV Describes the proposed methodology. Section V discusses comparative result analysis. Finally, concluded in section VI.

METRICS FOR LOAD BALANCING

- Throughput: - It is used to calculate the all tasks whose execution has been completed. The performance of any system is improved if throughput is high.
- Fault Tolerance: -It means recovery from failure. The load balancing should be a good fault tolerant technique.
- Migration time: -It is the time to migrate the jobs or resources from one node to other nodes. It should be minimized in order to enhance the performance of the system.
- Response Time: - It is the amount of time that is taken by a particular load balancing algorithm to response a task in a system. This parameter should be minimized for better performance of a system.
- Scalability: - It is the ability of an algorithm to perform Load balancing for any finite number of nodes of a system. This metric should be improved for a good system.
- Overhead: Overhead is the estimation of extra cost involved while making an algorithm to execute efficiently. For an efficient algorithm, there should be very low overhead
- Resource Utilization: This is to ensure that the resources that make up the system are properly utilized in such a way that no tasks should remain idle or keep waiting for the resource to get executed.
- Carbon Emission (CE) - calculates the carbon emission of all the resources in the system. As energy consumption and carbon emission go hand in hand, the more the energy consumed, higher is the carbon footprint. So, for an energy-efficient load balancing solution, it should be reduced.

II LOAD BALANCING ALGORITHM

LOAD BALANCE MIN-MIN (LBMM)

Scheduling algorithm [14] and new optimized Load Balancing Max-Min-Max (LB3M) [15] had main objective to minimize execution time of each task, also avoid unnecessary replication of task on the node thereby minimizing overall completion time. Opportunistic Load balancing algorithm when combined with LBMM (OLB + LBMM) [14] keeps every node in working state to achieve load balance. Similar to LBMM, LB3M [15] also calculate average completion time for each task for all nodes. Then mark the task with maximum average completion time. After that it dispatches the task of marked node to the unassigned node with minimum completion task, thus balancing the workload evenly among all nodes.

HONEYBEE FORAGING LOAD BALANCING ALGORITHM

It is a nature inspired decentralized load balancing technique which helps to achieve load balancing across heterogeneous virtual machine of cloud computing environment through local server action and maximize the throughput [2]. The current workload of the VM is calculated then it decides the VM states whether it is over loaded ,under loaded or balanced .according to the current load of VM they are grouped. The priority of the task is taken into consideration after removed from the overload VM which are waiting for the VM .Then the task is schedule to the lightly loaded VM. The earlier removed task are helpful for the finding the lightly loaded VM. These tasks are known as scout bee in the next step. Honey Bee Behavior inspired Load Balancing technique reduces the response time of VM and also reduces the waiting time of task [1].

ACTIVE CLUSTERING

It is considered as a self-aggregation algorithm, works on the principle of grouping the similar nodes and work together on these available groups [10]. A set of processes is iteratively executed by each node on the network. Initially any node can become an initiator and selects another node from its neighbors to be the matchmaker node satisfying the criteria of being a different type than the former one. The matchmaker node then forms a connection between neighbors of it which are similar to the initiator. The matchmaker node, then removes the connection between itself and the initiator.

III PROBLEM STATEMENT

The process of load balancing and task scheduling impart a major role in success of cloud computing. In review process we found that various factor effect the performance of cloud computational in concerns of storage of network data and sharing of resource. The sharing of cloud resource generates a network overload, the network overload arise the problem of bandwidth and stack overflow. Some problem related to cloud computing is given below.

1. Increasing the time span for process[2]
2. Failure of resource allocation[12]
3. Traffic overhead of network[13]

4. Waste of resource[4]
5. Cost of monitoring of resource[15]
6. Process feedback system[16]

For the minimization of this entire problem, used heuristic function by various author in load balancing in cloud computing.

IV Proposed Methodology

In this section discuss the modified algorithm of honey bee for cloud load balancing. For the modification of honey bee algorithm used directed acyclic graph technique. The directed acyclic graph technique allocates the dedicated job load to group of virtual machine for the processing of job transfer. The transferred job allocated in sequence of total job for the processing of job scheduler according to their available virtual machine. here define some parameter for the processing of job to dedicated virtual machine.

- Let n is the no. of jobs (j1, j2, and j3… jn).
- Let m is the no. of virtual machine (v1,v2,……., vm)
- Compute the value of best patch according to the define constraints.
- For each resource obtain the information like bandwidth, computing capacity and current load from job scheduler.
- For each job obtain the job size and the time needed to complete to complete the job.
- Create job matrix for the process and apply best patch.
- $$Mintmize \sum_{x} L(x) = J(x), B(x) \dots\dots\dots\dots\dots\dots\dots (a)$$
- Here j is total job and B is forged bee
- $$B(x) = C(x) + L(x) \dots\dots\dots\dots\dots\dots\dots\dots\dots (b)$$
Here C is cost of job and L is minimized pool
- $$B(x) \leq \sum_{t=1}^{k} Ml \dots\dots\dots\dots\dots\dots\dots\dots\dots (c)$$
Total forged bee for DAG allocation process.
- $$F(x) \leq \sum_{t=1}^{k} DAG(t,k) \dots\dots\dots\dots\dots (d)$$
- The processing of optimal patch according to their allocation
- $$As(Li,Bj) = \begin{Bmatrix} Allocated(Ta, tready(Li, at)) if\ Li \neq At\ empty \\ 0\ otherwise \end{Bmatrix} \dots\dots\dots\dots\dots (e)$$
- Finally allocated job to virtual machine
- $$VMload = Max_{Li \in At(DAG(t,k))} + \sum Bt \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (f)$$
Job is processed for their dedicated machine.

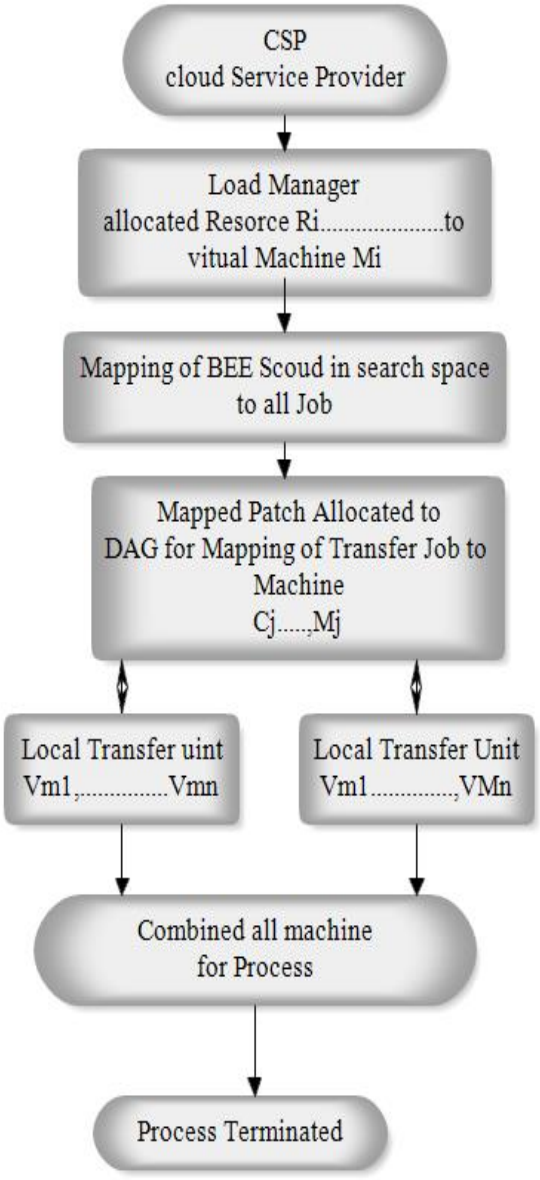

Figure 1: process block diagram of load balancing process based on BEE-DAG.

V Experimental Results Analysis

In this section we perform experimental process of cloud computing techniques with simulation tools. To interact with various services in the cloud and to maintain the resources in a balanced manner to fulfill the requirement of resources/infrastructure by those services, several techniques are required. To evaluate the performance of cloud computing techniques in cloud computing environment for the load balance and resource management, here we are using various numbers of techniques such as Round Robin, BEE and IBEE as a proposed method. For the further implementation and comparison for performance evaluation we used java programming languages with NetBeans IDE 8.0.1 tools for complete implementation/results process.

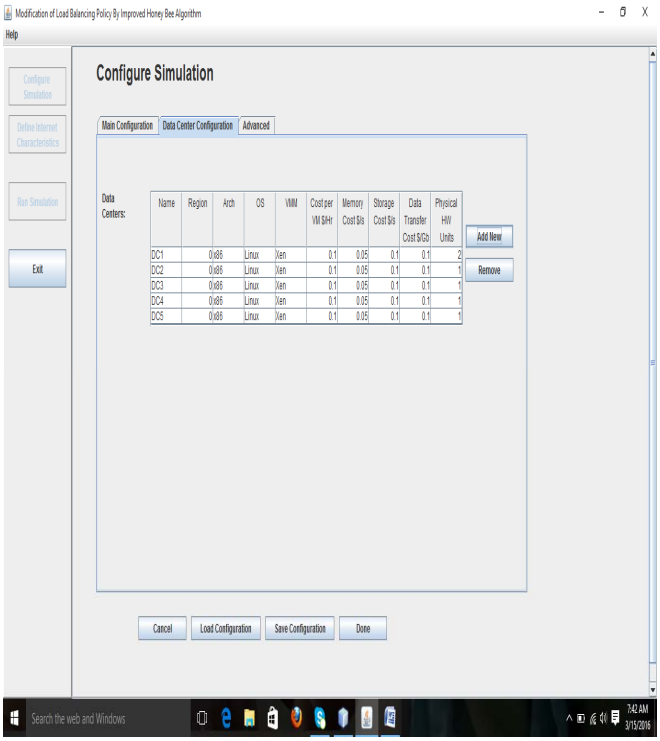

Figure 2: Shows that the Simulation window for cloud computing analyst with adds the data centre configuration value.

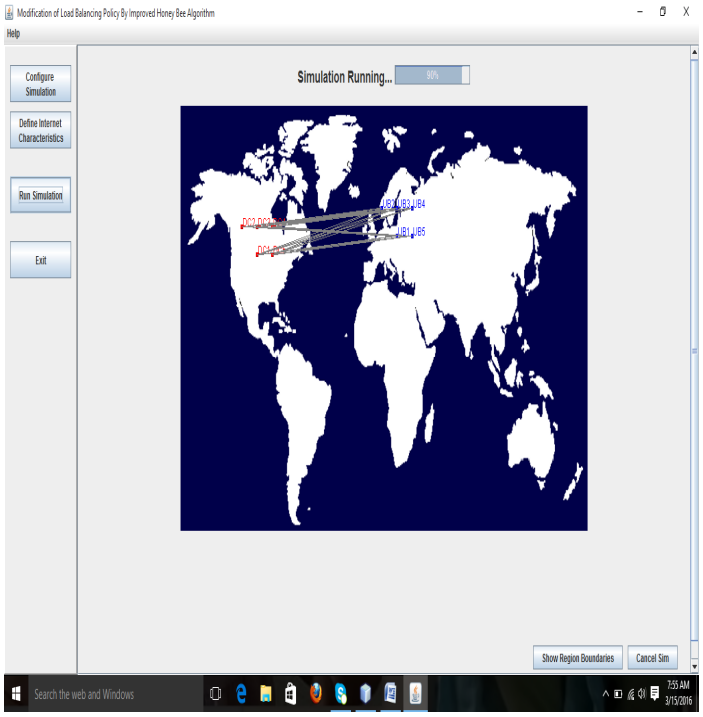

Figure 4: Shows that the Simulation window for cloud computing analyst for Honey Bee methods with the input value is 5 for the experimental work.

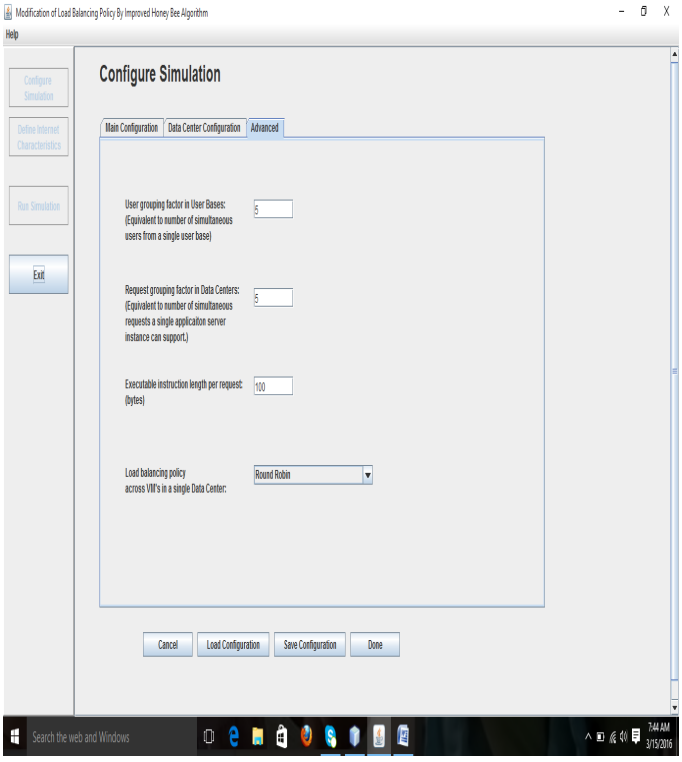

Figure 3: Shows that the Simulation window for cloud computing analyst for Round Robin methods with the input value is 5 for the experimental work.

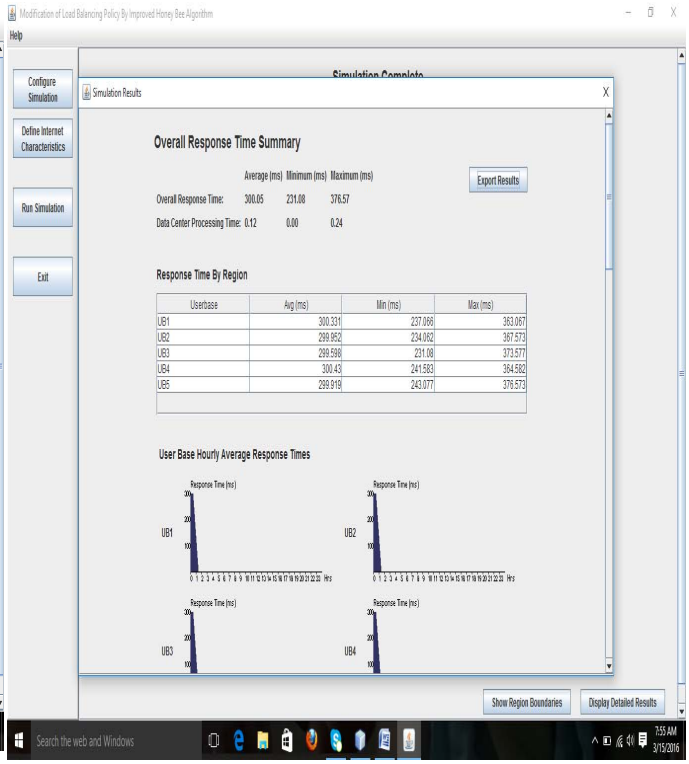

Figure 5: Shows that the Results window for cloud computing analyst for Honey Bee methods with the input value is 5 for the experimental work.

Table 1: Shows that a comparative performance evaluation using various methods for the input value is 5.

Number of Input	Method Name	Average Time	Minimum Time	Maximum Time
5	Round Robin	0.31	0.02	0.61
	Bee	0.22	0.01	0.43
	IBee	0.12	0.01	0.24

Table 2: Shows that a comparative performance evaluation using various methods for the input value is 20.

Number of Input	Method Name	Average Time	Minimum Time	Maximum Time
20	Round Robin	0.48	0.02	0.95
	Bee	0.34	0.01	0.66
	IBee	0.19	0.01	0.38

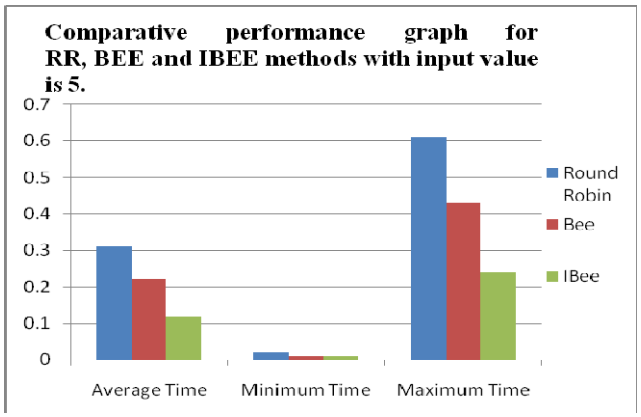

Figure 6: Shows that the comparative performance evaluation for the cloud computing load balancing mechanism for the Round Robin, BEE and IBEE methods, here the input value is 5.

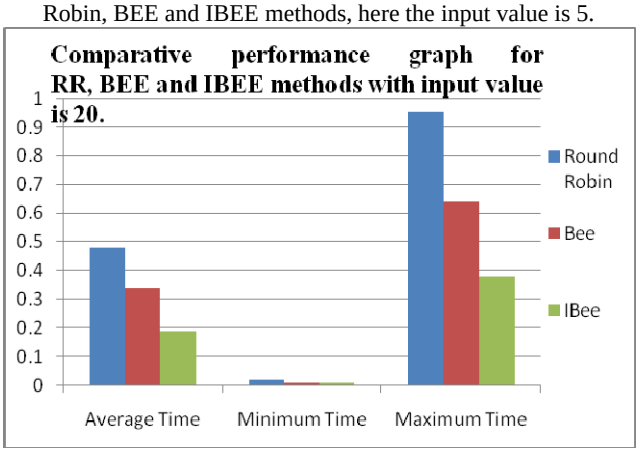

Figure 7: Shows that the comparative performance evaluation for the cloud computing load balancing mechanism for the Round Robin, BEE and IBEE methods, here the input value is 20.

VI Conclusion And Future Work

In this paper we used BEE algorithm and DAG allocation for load balancing policy in cloud environments. The IBEE optimization set the diverse property of virtual machine and request job. The define fitness constraints function partially allocated job for dedicate machine and the distribution of job according to the process job scheduler. For the evaluation of performance used cloud simulator software such is called cloud analyst. The cloud analysis software is bag of composition of cloud environment and load balancing policy.

In scenario of policy design two services one is BEE policy and other is IBEE based policy. The IBEE based policy reduces the load effect approx 10-12% in compression of BEE algorithm. The BEE and DAG based load balancing policy is very efficient for the proper allocation of job according to dedicated virtual machine. The partial allocation of job allocation policy faced problem of minimum time span. The minimum time span factor effect the efficiency factor of proposed policy.

References:-

[1] Harshit Gupta, Kalicharan Sahu "Honey Bee Behavior Based Load Balancing of Tasks in Cloud Computing" International Journal of Science and Research, 2014. Pp 842-846.

[2] Obaid Bin Hassan, A Sarfaraz Ahmad "Optimum Load Balancing of Cloudlets Using Honey Bee Behavior Load Balancing Algorithm" International Journal of Advance Research in Computer Science and Management Studies, 2015. Pp 334-338.

[3] Salim Bitam "Bees Life Algorithm for Job Scheduling in Cloud Computing" , ICCIT, 2012. Pp 186-191.

[4] Mohamed Firdhous, Osman Ghazali, Suhaidi Hassan, Nor Ziadah Harun, Azizi Abas "honey bee based trust management system for cloud computing" Proceedings of the 3rd International Conference on Computing and Informatics, ICOCI, 2011. Pp 126-131.

[5] Gaochao Xu, Junjie Pang, Xiaodong Fu "A Load Balancing Model Based on Cloud Partitioning for the Public Cloud" Tsinghua Science And Technology 2013 PP 34-39

[6] Siva Theja Maguluri, , R. Srikant "Scheduling Jobs With Unknown Duration in Clouds" IEEE 2014 PP 1938- 1951.

[7] Mauro Andreolini, Sara Casolari, Michele Colajanni, Michele Messor "Dynamic load management of virtual machines In a cloud architectures" PP 1-13.

[8] Rajiv Ranjan, Liang Zhao, Xiaomin Wu, Anna Liu "Peer-to-Peer Cloud Provisioning: Service Discovery and Load-Balancing" CSE UNSW PP 1-27. .

[9] Philipp Hoenisch, Stefan Schulte, Schahram Dustdar "Workflow Scheduling and Resource Allocation for Cloud-based Execution of Elastic Processes" 2013 IEEE.

[10] Yossi Azar, Naama Ben-Aroya ,Nikhil R. Devanur "Cloud Scheduling with Setup Cost" SPAA 2013 PP 23–25.

[11] Borja Sotomayor, Ruben S. Montero, Ignacio M. Llorente, Ian Foster "An Open Source Solution for Virtual Infrastructure Management in Private and Hybrid Clouds" IEEE Internet Computing, Special Issue On Cloud Computing 2009 PP 1-11.

[12]Aarti Singha, Dimple Junejab, Manisha Malhotra "Autonomous Agent Based Load Balancing Algorithm in Cloud Computing", International Conference on Advanced Computing Technologies and Applications , 2015 PP 832-841.

[13] Sasmita Parida, Suvendu Chandan Nayak "Study of Deadline Sensitive Resource Allocation Scheduling Policy in Cloud Computing" IJCSMC, 2014, PP 521 – 528.

[14] Suguna R, Divya Mohandass, Ranjani R "A Novel Approach For Dynamic Cloud Partitioning And Load Balancing In Cloud Computing Environment" 2014 JATIT PP 662-667.

[11] Amittai Aviram, Sen Hu, Bryan Ford Yale ,Ramakrishna Gummadi "Determinating Timing Channels in Compute Clouds" 2010 Pp 1-6.

[15] Siva Theja Maguluri , R. Srikant "Heavy Traffic Optimal Resource Allocation Algorithms for Cloud Computing Clusters" 2012.

[16] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, A. Vakali "Cloud computing: Distributed internet computing for IT and scientific research" Internet Computing, 2009. PP.10-13.

[17] P. Mell , T. Grance, "The NIST definition of cloud computing" 2012.

[18] N. G. Shivaratri, P. Krueger, M. Singhal, "Load distributing for locally distributed systems" 1992 PP. 33-44.