Multi-parameter optimization in cloud load balancing problem

[1] Monika, [2] Dr. Vivek Jaglan, [3]Jugnesh Kumar
[1] Department of C.S.E., Bhagwant University Ajmer Rajasthan
[2] Department of C.S.E., Amity University, Gurugram, Haryana
[3] Professor & Director, St. Andrews Institute of Technology and Management, Gurugram

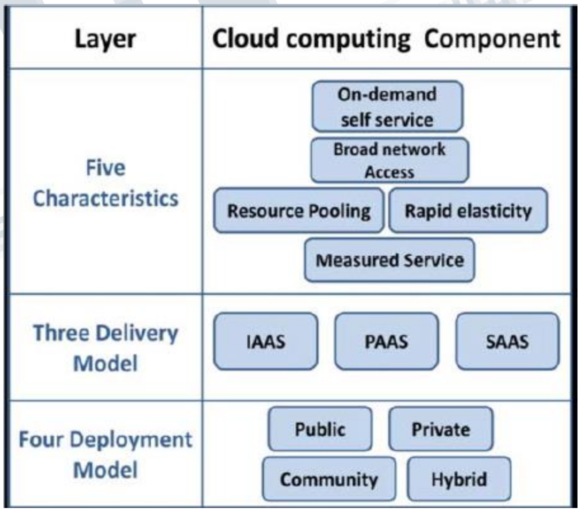
Abstract - **Cloud computing is exceptionally composed setup gives stage as an administration, framework as an administration and programming as administration. It causes customers to utilize required administrations and pay as indicated by the use of administration. The standard part of distributed computing is virtualization that arrangements with the development and administration of virtual machines effectively. Distributed computing condition gives numerous assets and administrations to imparting to their customers. Information stockpiling on distributed computing is expanding step by step which causes shortage of assets on cloud server farms. Additionally a few server farms are over-burden and some are under stacked. Along these lines stack adjusting on cloud server farms is required. With stack adjusting idea a few assignments of over-burden servers are exchanged to under stacked servers. For the most part stack adjusting calculations work powerfully. There are numerous dynamic load adjusting calculations exists for adjusting the work stack on cloud server farms. In this paper Hybrid approach is connected for stack adjusting utilizing Round Robin, Equally Spread Current Execution and (ESCE) Throttled calculations.**

Keywords — **Cloud Computing, Load Balancer, Round Robin, Throttle Method and ESEC**

1. INTRODUCTION

Cloud computing gives adaptable approach to hold information and records which includes virtualization, dispersed figuring, and web administrations. The principle point of cloud servers is to impart tremendous measure of assets to their customers with minimal effort. The customers can utilize the cloud assets by enrolling with particular server and send demands for the assets. The server after validation gives wanted administrations to the asking for customers. The distributed computing now daily is confronting a continuous test of load adjusting [1]. The fundamental purpose behind this test is the expansion in the clients interest for cloud administrations. So it is for all intents and purposes difficult to keep up the at least one free administration to satisfy the request. Furnishing every server with one request to satisfy will come about into movement on the server and at last the crash of the framework. It is utilized by Cloud specialist organization (CSP) in its own particular distributed computing stage to give a high effective answer for the client. Likewise, a bury CSP stack adjusting instrument is expected to build a minimal effort and endless asset pool for the purchaser [2].

ssA general model of Cloud Computing is appeared in figure 1 underneath.



Layer	Cloud computing Component
Five Characteristics	On-demand self service Broad network Access Resource Pooling Rapid elasticity Measured Service
Three Delivery Model	IAAS PAAS SAAS
Four Deployment Model	Public Private Community Hybrid

Figure 1: Model of Cloud Computing

Load balancing is helped to distribute the dynamic workload across multiple nodes to ensure that no single node is overloaded. It helps in proper utilization of resources. It also improves the performance of the system. Many existing algorithms provide load balancing and better resource utilization [3]. Load balancing is the process of finding overloaded nodes and then transferring the extra load to other nodes. In this paper Hybrid approach is applied for load balancing using Round

Robin, Equally Spread Current Execution and (ESCE) Throttled algorithms.

II. LOAD BALANCING IN CLOUD COMPUTING

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 is a techniques that helped networks and resources by providing a Maximum throughput with minimum response time. Load balancing is dividing the traffic between all servers, so data can be sent and received without any delay with load balancing Load balancing required to improve the performance of the system by minimize the overall completion time and avoid the situation where some resources are heavily loaded or others remains under loaded in the system [4].

Quantitative Metrics

- Throughput

It is the number of tasks completed per unit time. Its value should be high.

- Overhead

It is the extra cost associated with algorithm like cost for implementation, inter-process communication and data migration etc. It should be minimized.

- Migration Time

Time taken to transfer load from one node to another is known as migration time. It should be minimized for an effective algorithm.

- Response Time

It"s the time when a load balancing algorithm starts giving reaction. It should be minimized for a good algorithm.

- Resource Utilization

An efficient load balancing algorithm makes optimized utilization of resources [23].

- Performance

Performance means overall system usefulness. This parameter should be high.

- Complexity

Complexity of an algorithm means how hard it is to implement and understand. Complexity of an algorithm should be small.

- Computing power

It"s the ability of system to perform varieties of tasks efficiently. It"s also the measure of speed of system. For an algorithm this parameter should be high.

- Parallel Elements

It is the number of computing units working simultaneously. Larger value of parallel elements means more tasks can be handled at the same time.

- Power consumption

It is the measure of power consumed by system. This parameter should be minimized for an efficient algorithm.

- Carbon emission

This parameter is directly proportional to power consumption. More power consumption means more carbon emission.

Qualitative Metrics

- Fault Tolerance

It is ability of system to automatically overcome faulty situations and continue to operate with slight degradation in performance. This parameter should be high for an effective algorithm.

- Fairness

An algorithm should be fair enough to distribute load on all severs equally without making them overloaded and under-loaded.

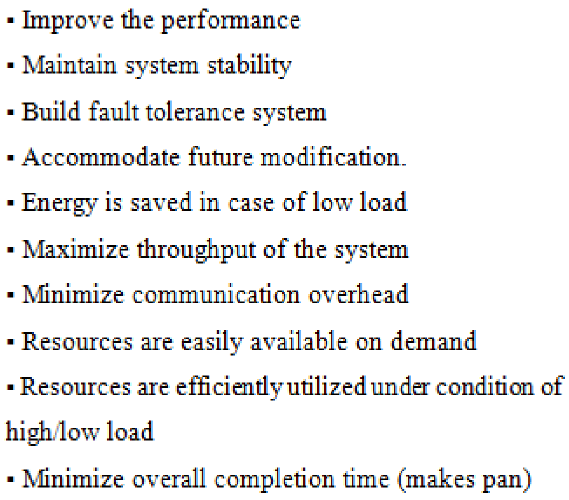
- Reliability

It"s ability of algorithm to perform with same pace and efficiency in long run. Value of this metric should be high.

- Security

Algorithm should processes data only for authenticated users within their authorized workspaces. Security should be high

The goals of load balancing are:



- Improve the performance
- Maintain system stability
- Build fault tolerance system
- Accommodate future modification.
- Energy is saved in case of low load
- Maximize throughput of the system
- Minimize communication overhead
- Resources are easily available on demand
- Resources are efficiently utilized under condition of high/low load
- Minimize overall completion time (makes pan)

Basically there are four policies of load balancing algorithms [16]:

- Load information

All the nodes must be aware about current load status.

- Transfer policy

Overloaded nodes must transfer their load to under loaded nodes.

- Location policy

It means how far customer is from server geographical location.

- Selection policy

It means which task needs to be moved to customer side [16].

Load balancing can be performed at hardware and software level. In [17] a hybrid load balancing approach named duet has been proposed which combines benefits of both hardware and software load balancers. Hardware load balancers offer low latency and high capacity but they are very expensive. On the other hand software load balancers are cost effective, offer high availability and reliability but have high latency and low capacity.
Main sides of load balancing algorithms are approximation and comparison of load, firmness and performance evaluation of nodes, internode traffic optimization, selection of suitable node for load transfer, communication among nodes, to maintain backup in case of failure, reduction in cost and low power consumption [18]. The task of load balancing is divided into two parts one is resource allocation and another task scheduling [9].

- Resource allocation

It is the process of provisioning of cloud resources to different tasks on basis of demand. This provisioning should be done in such a way that neither the resources nor the VM‟s (Virtual Machines) are underutilized.

- Task scheduling

It‟s the next step after resource allocation. Here it is decided whether the provisioned resources are for exclusive use or not (or they are shared). Task scheduling algorithms decide whether resources are available for full time or time-sharing basis [9].
The concept of load balancing and how load balancer works in cloud environment is shown in figure 2.This diagram consists of three phases, data center controller, load balancer and VM manager [19]. A user can make multiple requests at a time at cloud interface. Data center controller receives requests from users and sends them to load balancer. Load balancer performs resource allocation and task scheduling as mentioned above. It then runs appropriate load balancing algorithm to choose a VM suitable for that requested application and returns its id to VM manager. VM manager then allocates user request to the selected VM for processing.

Figure 2 below shows load balancing in cloud environment.

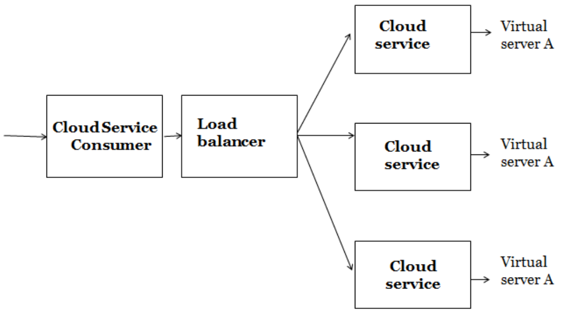


Fig. Load balancer in cloud computing

Figure 2: Load Balancer in cloud Environment

III. EXISTING LOAD BALANCING ALGORITHMS

To design an effective load balancing policy and to determine how to increase the cloud resource usage are the two main goals of a cloud service provider. The VM scheduling algorithms for load balancing helps in allotment of VMs efficiently on need. Basically, a VM load balancing algorithm decides which VM is to allocate when request is made by cloud consumer. Numerous VM load balancing algorithms that have been proposed are discussed here [6][10]:

1. Round Robin Load Balancing Algorithm

It is a very simple load balancing algorithm that places the newly coming request on the available virtual machines in a circular manner. The major advantage of this algorithm is its simplicity and easy implementation. The main drawbacks are that it requires the prior knowledge of user tasks and system resources & it do not make use of current state of the system

2. Throttled Load Balancing Algorithm

It is a dynamic approach. In this, user submits its request to the Data Center Controller (DCC). Data Center Controller asks the VM Load Balancer to determine the appropriate virtual machine that can handle that much workload easily. Throttled VM Load Balancer keeps a virtual machine list and their status (available/busy). If a suitable VM is found on memory space, cores or availability basis, then throttled VM Load Balancer accept the cloudlet request and allot the cloudlet request over that virtual machine. Otherwise, clients have to wait in the waiting queue until a suitable VM becomes available. Among all, it is best approach for load balancing, since it maintains the present state of all VMs in data center. But the major drawback is that it works properly only if all VMs in a data center have same

3. ESCE load Balancing Algorithm

ESCE stands for Equally Spread Current Execution. It is also called Active VM Load Balancing algorithm. This algorithm is based on spread spectrum technique. As the name indicates, it equally distributes the workload on each VM in data center. A job queue keeps all the cloudlet requests that need the VM for their execution. ESCE VM Load Balancer (VMLB) also maintains a list of virtual machines. The VM Load Balancer continuously checks the job queue and VM list. If a VM is found free, then cloudlet request will be allotted over that VM. At the same time, VMLB inspect the overloaded VMs. If any virtual machine is found overloaded, then VMLB move some load to an idle or an under loaded virtual machine, so as to reduce some load of overloaded VM. The main drawback is high computational overhead. hardware configuration.

IV. PROPOSED WORK

The goal of the proposed work is to design an efficient scheduling algorithm that uniformly distribute workload among the available virtual machines in a data center and at the same time, decrease the overall response time and data center processing time. The proposed approach is a combination of Round Robin, Throttled (TVLB) and ESCE algorithm. TVLB algorithm makes use of states of VMs. A virtual machine state may be either AVAILABLE or BUSY. AVAILABLE state indicates that the virtual machine is idle/free and ready for cloudlet allotment, where BUSY state indicates that the current virtual machine is busy in execution of previous cloudlets and is not available to handle any new cloudlet request. This current load state of a VM helps in taking decision whether to allocate cloudlets to virtual machines or not.

Active VM Load Balancing algorithms continuously monitor the job queue for new cloudlets and allot them to the bunch of idle/free VMs. It also maintains the list of cloudlets allocated to each virtual machine. This allocated cloudlet list helps in determining whether a VM is overloaded or under loaded at particular moment of time. On the basis of this information, VM load Balancer moves some load from overloaded VMs to the VM having minimum number of cloudlets, so as to maintain a high degree of balance among virtual machines. These features combined together, make the proposed scheduling algorithm more efficient & effective and help in fair distribution of the load.

Figure 3 below shows the proposed load balancing algorithm.

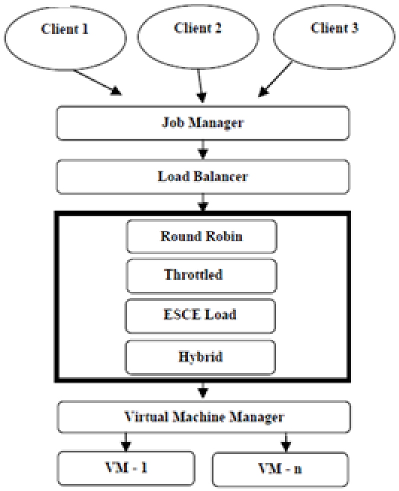


Figure 3: load balancing Algorithms

The algorithm for proposed work:

The steps for performing load balancing on cloud environment using hybrid approach of round robin, throttle and equally spaced algorithm are listed below:

Step 1. The input for the proposed algorithm is cloud data centers that provide services to cloudlet (clients' requests).

Step 2. Initially clients' requests are performed in round robin fashion i.e., as soon as client's request arises and server status is available then the server is allocated to the requesting client.

Step 3. Now after allocating server the throttle load balance technique checks the requested resources (Number of CPUs, Memory, Processing time etc.) & allocate the cloud data center server to the client that satisfying the client need.

Step 4. The process also checks the desired threshold value for each server. If the server utilization exceeds the specified threshold (>75%) then the server of another cloud data center will start for satisfying the request.

Step 5. When the client stops the task then the service allocated to the client is released & same can be reallocated to another client in the waiting.

Step 6. In this step our implementation performs equally spread current execution algorithm to equally distribute the load on various active cloud data center servers.

Step 7. We have use another threshold value that becomes the basic of balancing the load on servers. If load of any active server is less than specified threshold value (<25%) and there is another server with required space then load of first server is transferred to second server and first server is closed.
Hence the work provides efficient load balancing among the active servers and server with low utilization is closed after moving their load to another server with required space.

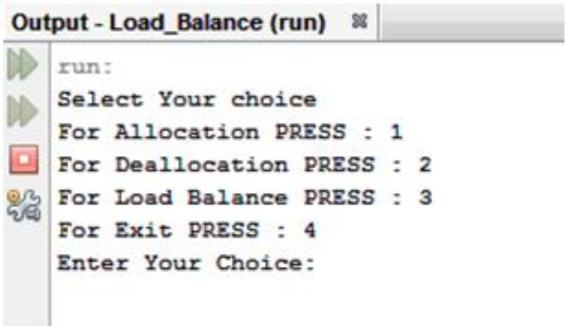
V. IMPLEMENTATION& RESULTS

Our implementation for performing load balancing on cloud environment using hybrid approach of round robin, throttle and equally spaced algorithm work as follows:

Initially some servers on cloud data centers startup & wait for user's requests. As user's requests arrive, they are performed in round robin fashion i.e., as soon as client's request arises and server status is available then the server is allocated to the requesting client. Now after allocating server the throttle load balance technique checks the requested resources (Number of CPUs, Memory, Processing time etc.) & allocate the cloud data center server to the client that satisfying the client need. The process also checks the desired threshold value for each server. If the server utilization exceeds the specified threshold (>75%) then the server of another cloud data center will start for satisfying the request.

When the client stops the task then the service allocated to the client is released & same can be reallocated to another client in the waiting. Now our implementation performs equally spread current execution algorithm to equally distribute the load on various active cloud data center servers.

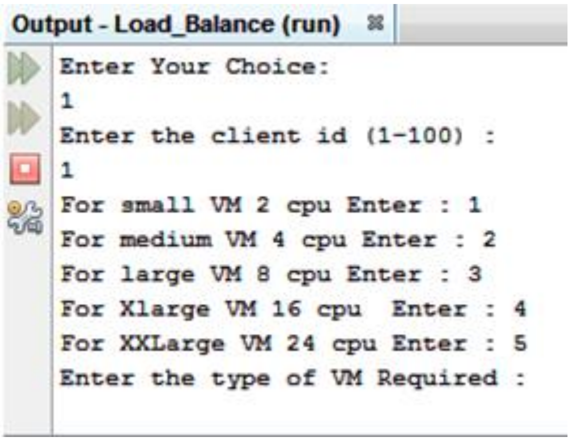
We have use another threshold value that becomes the basic of balancing the load on servers. If load of any active server is less than specified threshold value (<25%) and there is another server with required space then load of first server is transferred to second server and first server is closed. Hence the work provides efficient load balancing among the active servers and server with low utilization is closed after moving their load to another server with required space.Figure 4 shows the main menu of our implementation.



```
run:
Select Your choice
For Allocation PRESS : 1
For Deallocation PRESS : 2
For Load Balance PRESS : 3
For Exit PRESS : 4
Enter Your Choice:
```

Figure 4: Main menu of implementation

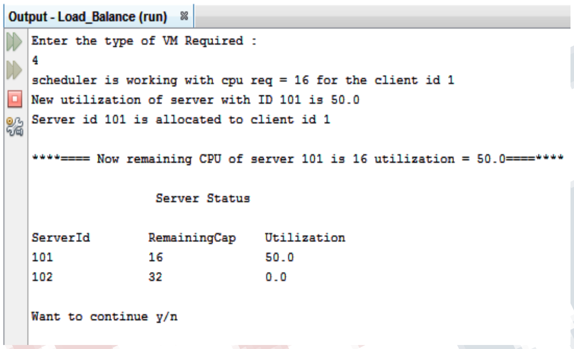
When user select option 1 i.e., allocation of resources then it ask for client id and numbers of CPUs required as shown in figure 5.



```
Enter Your Choice:
1
Enter the client id (1-100) :
1
For small VM 2 cpu Enter : 1
For medium VM 4 cpu Enter : 2
For large VM 8 cpu Enter : 3
For Xlarge VM 16 cpu  Enter : 4
For XXLarge VM 24 cpu Enter : 5
Enter the type of VM Required :
```

Figure 5: Allocation of resources for client id 1

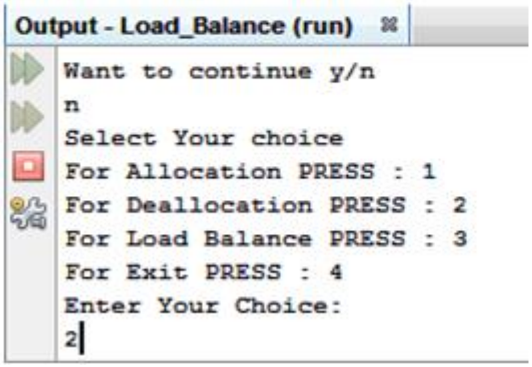
Figure 6 shows the effect after allocation of resources



```
Enter the type of VM Required :
4
scheduler is working with cpu req = 16 for the client id 1
New utilization of server with ID 101 is 50.0
Server id 101 is allocated to client id 1

****==== Now remaining CPU of server 101 is 16 utilization = 50.0====****

                    Server Status

ServerId          RemainingCap     Utilization
101               16               50.0
102               32               0.0

Want to continue y/n
```

Figure 6: Effect after resource allocation

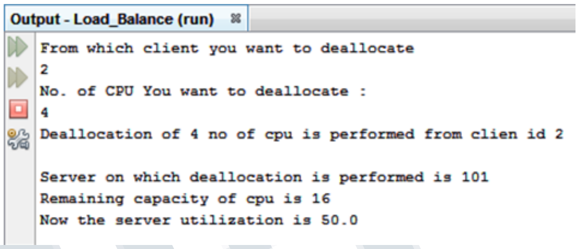
When we done with allocation then press n. Main menu options will display again on the screen as shown in figure 7



```
Want to continue y/n
n
Select Your choice
For Allocation PRESS : 1
For Deallocation PRESS : 2
For Load Balance PRESS : 3
For Exit PRESS : 4
Enter Your Choice:
2
```

Figure 7: Selecting de allocation choice

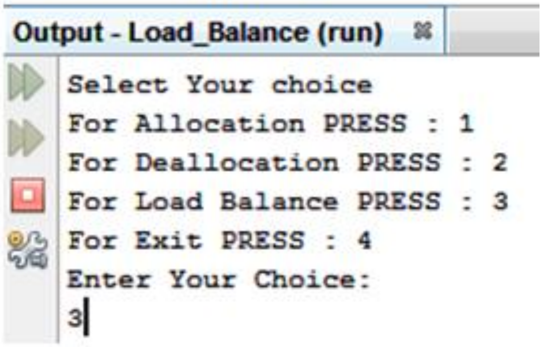
In main menu options now select the option 2 for deallocation of resources. It will ask for client id and numbers of CPUs to be deallocated. If client id is not correct or numbers CPUs to be deallocated are more than the allocation then appropriate error message is displayed. If everything is OK then deallocation is performed. The remaining capacity and utilization value are updated for the server on which deallocation are performed as shown in figure 8.



```
From which client you want to deallocate
2
No. of CPU You want to deallocate :
4
Deallocation of 4 no of cpu is performed from clien id 2

Server on which deallocation is performed is 101
Remaining capacity of cpu is 16
Now the server utilization is 50.0
```

Figure 8: Effect of deallocation

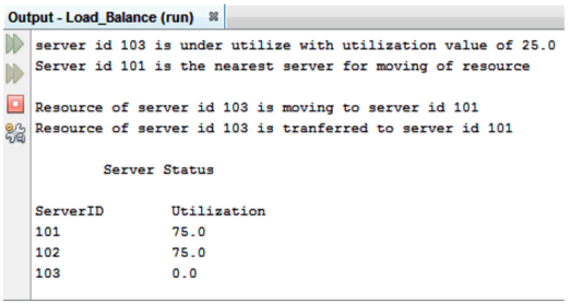
Now press the option 3 for performing the load balance as shown in figure 9 below.



```
Select Your choice
For Allocation PRESS : 1
For Deallocation PRESS : 2
For Load Balance PRESS : 3
For Exit PRESS : 4
Enter Your Choice:
3
```

Figure 9: Selecting the option 3 for load balancing

Figure 10 below shows the effect of load balancing.



```
server id 103 is under utilize with utilization value of 25.0
Server id 101 is the nearest server for moving of resource

Resource of server id 103 is moving to server id 101
Resource of server id 103 is tranferred to server id 101

          Server Status

ServerID          Utilization
101               75.0
102               75.0
103               0.0
```

Figure 10: Effect of load balance

In this work, we propose the hybrid approach of three load balancing algorithms to overcome the drawback of existing methods. We first assign the load on the server using round robin fashion then we use throttle concept to find the suitable machine for current tasks and finally we use equally spread technique to equally distribute the load on various virtual machines to balance the load [11][12].

REFERENCES

[1]. Vishwas Bagwaiya, Sandeep k. Raghuwanshi, "Hybrid Approach Using Throttled And ESCE Load Balancing Algorithms In Cloud Computing".

[2]. Mamta Khanchi, Sanjay Tyagi, "AN EFFICIENT ALGORITHM FOR LOAD BALANCING IN CLOUD COMPUTING",© International Journal of Engineering Sciences & Research Technology.

[3]. Bhavisha Patel, Shreyas Patel, "Various Load Balancing Algorithms in cloud Computing", IJARIIE-ISSN(O)-2395-4396. Vol-1 Issue-2 2015.

[4].Mithun Dsouza, Mohammed Rizwan, Ramnath Gaonkarand, S. Sathyanarayana, "SCHEDULING AND LOAD BALANCING TECHNIQUES IN CLOUD COMPUTING: A SURVEY", International Journal of Latest Trends in Engineering and Technology Special Issue SACAIM 2016, pp. 309-316 e-ISSN:2278-621X.

[5]. Bhasker Prasad Rimal, Enumi Choi, "A Taxonomy and survey of cloud computing systems", 2009 Fifth International Joint Conference on INC, IMS and IDC.

[6]. Rajwinder Kaur and Pawan Luthra, "Load Balancing in Cloud Computing".

[7] Amandeep Kaur Sidhu, Supriya Kinger, "Analysis of Load Balancing Techniques in Cloud Computing", International Journal of Computers &Technology Volume 4 No. 2, March-April, 2013, ISSN 2277-3061.

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

[9]. Fang, Y., Wang, F., and Ge, J., "A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing", Lecture Notes in Computer Science, Jg. 2010(6318): pp. 271-277, (2010).

[10]. Abhinav Hans, Sheetal Kalra, "Comparative Study of Different Cloud Computing Load Balancing Techniques", 2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom).

[11].V Ravi Teja Kanakala,V. Krishna Reddy and K. Karthik, "Performance Analysis of Load Balancing Techniques in Cloud Computing Environment" 978-1-4799-6085-9/15/$31.00©2015 IEEE.

[12]. Subhadra Bose Shaw, "A Survey on Scheduling and Load Balancing Techniques in Cloud Computing Environment", 2014 5th International Conference on Computer and Communication Technology (ICCCT).