Volume 5, Issue 9, September 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Optimizing the Server through Load Balancing Techniques

S. Divya Meena[*]
ME-CSE
Kingston Engineering College,
Vellore 632059, Tamil Nadu, India

K. Chinetha
Assistant Professor
Kingston Engineering College,
Vellore 632059, Tamil Nadu, India

Abstract-- Cloud computing is an aggregation of cloud and computing. Cloud refers to the pool of heterogeneous resources and huge infrastructure which includes applications delivered to end user, hardware and software. Computing focuses on efficient use of these heterogeneous resources, providing high availability at minimum cost. Load balancing is a methodology that distributes the workload among multiple systems in order to achieve optimal resource utilization, maximum throughput at minimum response time and in overall to avoid overloading or under-loading of systems. Load balancing is achieved by applying certain scheduling algorithm to the process. This paper presents several load balancing methods, to achieve server utilization. In this paper we discuss the problem of balancing the load and assigning them to servers.

Keywords- Load balancing, server utilization, Cloud simulation, Server optimization, master-slave processes

I. INTRODUCTION

Cloud computing is a new approach that reduces IT complexity by leveraging the efficient pooling of on-demand, self-managed virtual infrastructure and consumed as a service. It is Internet based computing, whereby shared resources, software and information are provided to computers and other devices on-demand, like a public utility. In cloud computing, the requests arrive randomly and get allocated to servers randomly, thereby the utilization of CPU varies and so most of the servers are either idle most of the time or is heavily loaded or is under-loaded. Hence, load balancing is one of the major challenges in cloud computing. Load balancing is a methodology that distributes the workload among multiple systems in order to achieve optimal resource utilization, maximum throughput at minimum response time and in overall to avoid overloading or under-loading of systems. Load balancing requires keeping track of certain types of information such as Number of requests waiting to be processed, the rate at which these requests arrives at the server and CPU processing rate. This information has to be exchanged among neighbouring process in order to improve the overall performance. Without load balancing, while browsing websites, users will face delays, timeouts and low response time. Load balancing can alleviate this problem by applying redundancy in servers. We have tried to solve this problem by introducing several algorithms. The objective of the paper is not only to establish several load balancing algorithms but also provide the execution analysis of these algorithms using the simulation tool Cloud Analyst. Section (1) provides the basic idea behind Server Utilization. The techniques that could be applied to server such as Server consolidation are discussed in section (2). Load balancing concepts and its algorithms are explained in section (3). Section (4) presents the cloud simulation technique. The experimental results of executing the algorithm are presented in section (5). Section (6) gives the conclusion and future work.

II. SERVER CONSOLIDATION

Server consolidation is the process of consolidating or aggregating the servers to make several servers into one efficient server. This technique was developed as a means to overcome the server sprawl problem. It is a situation where several servers remain unused, consuming more space and becoming an overhead to the datacenter. Server consolidation technique makes efficient use of computer servers and reduces the number of servers required for an organization. This kind of consolidation is achieved through virtualization techniques. Virtualization enables several applications to run on a single server. Server consolidation increases the efficiency of servers, increases the utilization of servers, and reduces the cost by reducing the number of servers required and thereby reducing the maintenance cost. This reduces the overall CAPEX cost. It also reduces the power consumption of datacenter, as the number of servers required in server consolidation is very less.

III. LOAD BALANCING

Load balancing is the process of balancing the load by distributing the load among various systems through network links. It avoids overloading on particular system, by dividing the traffic/request equally for each system. To achieve this we would require certain algorithms that would allow for automatic load balancing service, in case of dynamic environment.

The two major needs for load balancing are;

1. To improve the utilization of resources (server/ system)

2. To improve performance by maximizing the throughput, minimizing the overall response time and minimizing the total waiting time.
3. To build a fault-tolerant system.

The algorithm we design should be cost-effective, scalable and flexible. Regardless of the equal number of jobs allotted to each server, the algorithm should implement prioritization, in order to provide better service. The real challenge lies in designing such an efficient algorithm. Load balancers in Cloud can manage online traffic. In such load balancer, the traffic is distributed automatically among resources. The following figure shows the load distribution in typical datacenter;

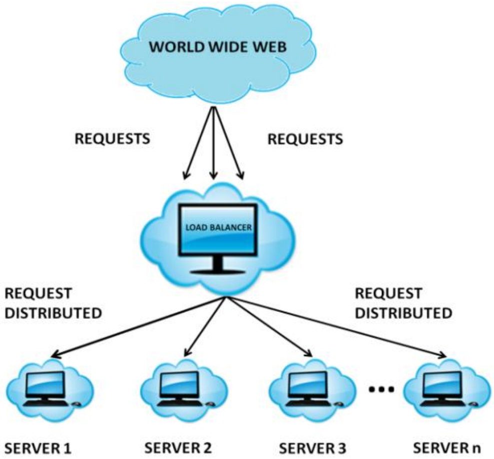


Figure 3.1: Load Balancing

A. LOAD BALANCER

MASTER-SLAVE PROCESS

Master node: It is responsible for handling the incoming request

Scheduler: It accepts the incoming request (may be batch-processing) and schedules the given task equally among the slaves.

Data Index: It maintains the index of all the data in the slaves and when a process requests for a particular piece of data, we can refer to the data index and find in which slave node does the data lies.

Task Processor: It accepts the task given by the scheduler and performs the necessary operation.

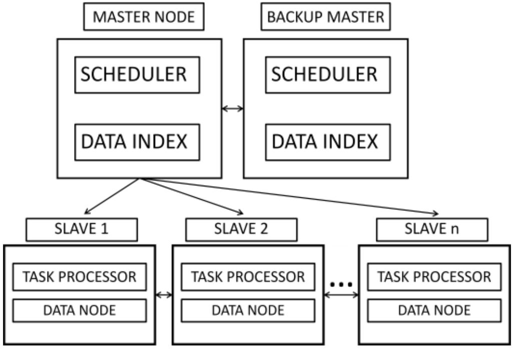


Figure 3.2: Master Slave Process

Data Node: The processed data is stored as information in the data node. Apart from this, each node maintains a set of data that may be accessed by the processor.

Backup Master: In case of a master node failing, it would become a single point of failure. To avoid such a bottleneck, we maintain a backup master node, which gets updated periodically by the main master node. The main master node on recovering takes back its position and gets the necessary information from the backup master node.

Backup Slaves: It is far most sufficient if utmost 3 backups are maintained for each slave nodes (works in most of the situation). If a slave node fails, its backup node1 can take in charge. If in case all backup nodes of a slave node also fails, before the slave node recovers, neighboring slave node can take up the process of failed slave node. The slave node on recovering takes back its position and gets the necessary information from the neighbor slave node or its own backup nodes, if available.

B. LOAD BALANCING ALGORITHM

Cloud has numerous resources which are not used in most cases. But there are cases where more of these resources are required. This situation is what is called ad Over-provisioning and Under-provisioning of resources respectively. Both the cases represent a bad state of Data Center. Managing these resources requires a proper plan and proper layout. The overall idea behind these concepts is Load Balancing, which is about balancing the load equally among resources or executing algorithm that works optimistically. In order for an algorithm to work efficiently, it must consider all the situations. So we classify our algorithm into three broad categories that will all the situations. Following are the three broad categories of load balancing;

1. Based on the location (Centralized or distributed)
2. Based on the load type (Static or Dynamic)
3. Token based vs. Non-token based method

For all these three classifications, we will use our basic Master-Slave diagram. The algorithm will differ in the way Master behaves.

1) BASED ON THE LOCATION:

The load balancing algorithm based on location is of two types namely; Centralized and Decentralized.

i. CENTRALIZED APPROACH:

In this approach, a central Master node is responsible for scheduling and allocating the incoming request to all slave nodes. The master node may employ any of the algorithms discussed below. The master node will store the entire knowledge base of the cloud by storing the Index of the data in slave, for easier access when a processor requests for the data. This way, the time required is reduced, for, not all the nodes are searched for a particular piece of data. But this increases the chances of a Master node becoming an overhead and because of this; the failure rate of master node is high. In case of master node failure, we will use our Backup master node. In the mean time, the failed master node can recover, though it is not easy and cheap.

ii. DECENTRALIZED/ DISTRIBUTED APPROACH:

This method has no central control of any nodes. All nodes including slaves work by themselves. No node is responsible for scheduling or allocating the resources. So the process is randomly scheduled to the nodes. In this method, every node monitors the network to make the load balancing decision. Every node maintains the local knowledge base and this information are broadcasted to every other node in the network. This way the load is balanced among the nodes. In case, a node is under-loaded, it sends a message to all other nodes stating that it is ready to share the workload of any over-loaded node. But if a node is over-loaded, again it sends a request message to all other nodes asking for the neighboring nodes to share its load. This method of sharing the load of other nodes is said to be load-balancing. The load balancing request is processed based on FIFO order. The failure of any particular node is identified by the absence of the broadcast message for a long time. In that case, the neighboring nodes will share the failed nodes workload. Hence, the distributed approach is said to be fault tolerant and balanced and there is no master node that is overloaded in this case.

2) BASED ON THE LOAD TYPE

Based on load type, the load balancing algorithm is classified into two types namely: Static load and Dynamic load

i. STATIC LOAD:

When the number of resources and its capacity is fixed, then it is called as Static. Static is not scalable i.e., it cannot be changed at the run time. In this method, not only the resources are fixed, the processing power and memory capacity is also fixed. In case of centralized approach, static method employs equal number of request, which will be scheduled to each node by the master. In case of decentralized approach, we can employ FCFS algorithm, where the first request will be assigned to the first node and the second request to the second node and so on. The nodes can have a field for indicating if it has received any task to be processed or not. If several nodes' fields indicate that they were not assigned the job, then the request will be assigned to the first node among them. If all the nodes were assigned a job at least once, then all their respective fields will be same. In such a case, the algorithm, in order to find the next node to be assigned with a task, will have to rely on the timestamp maintained by the nodes. The timestamp indicates the time at which the last task was assigned to it. The node with longest timestamp will be assigned the task. Though this method is simple to implement, it will not be suited for most of the real-time situations, as real-time process are mostly dynamic in nature.

ii. DYNAMIC LOAD:

When the number of resources and its capacity is not fixed, then it is called as Dynamic. Dynamic is scalable i.e., it can be changed at the run time. So the dynamic approach is adaptable though little difficult to implement. In static method, tasks are allotted to nodes based on the memory capacity and processing constraint. In dynamic method, there is no such constraint. So we will place time as a constraint, to avoid a particular node from being overloaded all the time. This algorithm will work similar to Round robin. Here we will assign an equal burst time for each node and the incoming request will be sent to the first node for the given burst time. After the burst time of first node expires, the requests will be forwarded to the next node. Order of the node is maintained in the same way as static method. So the load will be balanced among the nodes. Dynamic scheduling is a better choice to static scheduling, as the real-time cloud environment is dynamic in nature.

3) TOKEN BASED ALGORITHM

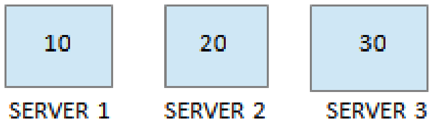
Following is a sample code for token based load balancing algorithm;

```
Token_Based_Load_Balancing ()
{
initialize the entire node in the Slave_node_List;
calculate the capacity of each slave node in the Slave_node_List;
initialize token_field of entire slaves with no null;
if(capacity_of_a_slave_node< capacity_of_all_other_slave_node)
{
```

```
assign the token to the slave node
}
while (new request are received by the Master node)
do
{
master node queue the requests;
master node removes the first request from the beginning of the queue;
if (token_field of first slave is set && node allocation status == AVAILABLE)
{
the request is allocated to the slave node;
 }
else
{
allocate a different node to the request using Round Robin Algorithm;
update the entry of the token and the node in the Slave_node_List
}}}
```

4) ALGORITHM PROBLEM

Let us consider 3 servers and its weight is given in numerical.



The weights indicate the capacity of server. Higher weight indicates that it can handle more traffic. We divide the incoming request by the weight of the server and sort it from smallest to largest. For example, let us consider that each server serves 10 requests, and then the order is given by;

C ---> 10/30=0.33
B ---> 10/20=0.50
A ---> 10/10=1.00

From the above calculation it is clear that the server C is least loaded. So, the next few requests (say 5) will go to Server C. The 6th request will go to either C or B. And the 7th request will go to the server that didn't handle the 6th request.

The job of the load balancer is indeed complicated, because it will have to keep track of the request that are being currently serviced by the servers and decrement the count when the request is completed. Also it has to keep track of the request and where they go.

IV. SIMULATION IN CLOUD

Simulation imitates the real-time environment. We can use simulation to test the algorithm at hand. Thus it is used to test the efficiency of the actual system before it is constructed as a final model. In Cloud environment, the resources are dynamically shared according to client’s request. At times when it is difficult to measure the performance of the application in real time, we can go for simulations that is very much useful for users to get practical response in spite of having real environment.

CloudAnalyst

CloudAnalyst is a Cloud Simulator tool that facilitates in modeling the workload characteristics of a data center. It considers the geographic distribution of users and data centers. It is completely based on GUI. It is used to model and analyze the real world problems.

The main features of Cloud Analyst are;

1. User friendly GUI simulation
2. Can be extended with Java Connectivity
3. High degree of configurability and flexibility
4. Ability to perform various experiments with replication

A. DECENTRALISED/ DISTRIBUTED APPROACH

Distributed approach works by distributing the workload among multiple servers and there is no central server in this method. By far, this is the best method. Some of the algorithms that work by this method are;

1. Round Robin
2. Equally Spread Current Execution Algorithm
3. Throttled Load Balancing Algorithm

a) Round Robin:

Round Robin works by scheduling the incoming request sequentially to the servers. This is a random sampling based algorithm. This works best if all servers have same resource capacity. If the capacity is not same in all servers, then the overall performance will degrade. This algorithm selects the load from heavily loaded servers and transfers it to lightly loaded servers.

b) Equally Spread Current Execution Algorithm

ESCE algorithm works by spreading the workload randomly among the servers. The algorithm checks the size of the incoming load and assigns this load to any of the lightly loaded server. It is based on spread spectrum technique that spreads the load among multiple servers.

c) Throttled Load Balancing Algorithm

Throttled algorithm works by requesting the load balancer to check for the least loaded server or a server that is easily accessible. It then transfers the request to that server. As the name implies, this algorithm is mostly about controlling the decision of choosing the server for next request. This algorithm is based on Virtual machines.

V. PERFORMANCE ASSESSMENT

For the three algorithms – round robin, ESCE, Throttled; we have used CloudAnalyst tool to evaluate the performance of these algorithms.

Components of CloudAnalyst:

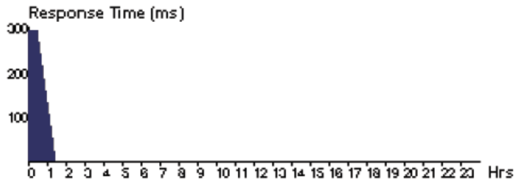
1. ***User Base***: It refers to single user. But a data center can't be built for a single user, so the number of user should be high for a better utilization of resources and for an efficient simulation. User Base generates the traffic representing the users in user base.
2. ***Datacenter***: It is responsible for managing the data activities, creating servers and destroying servers. It accepts the incoming request/ traffic from user base and routes it to various servers via internet.

A. RESPONSE TIME BY REGION

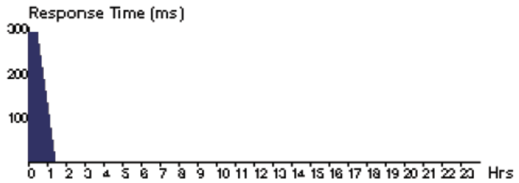
Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.26	0.10	0.64
DC2	0.40	0.04	0.67
DC3	0.32	0.10	0.69
DC4	0.42	0.13	0.69
DC5	0.30	0.02	0.66

B. USERBASE HOURLY RESPONSE TIME

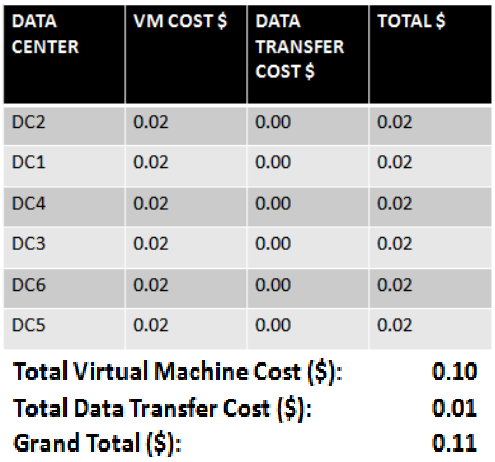
UB1



UB2

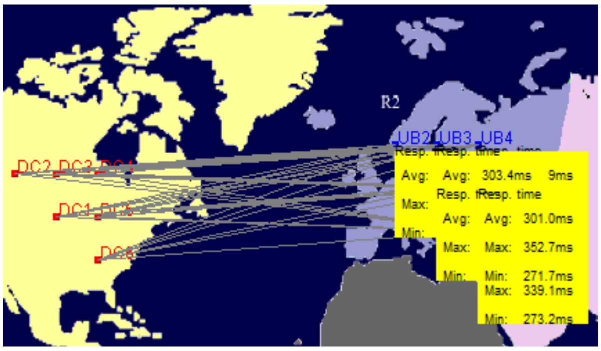


C. OVERALL COST



DATA CENTER	VM COST $	DATA TRANSFER COST $	TOTAL $
DC2	0.02	0.00	0.02
DC1	0.02	0.00	0.02
DC4	0.02	0.00	0.02
DC3	0.02	0.00	0.02
DC6	0.02	0.00	0.02
DC5	0.02	0.00	0.02

Total Virtual Machine Cost ($):	**0.10**
Total Data Transfer Cost ($):	**0.01**
Grand Total ($):	**0.11**



VI. CONCLUSION

One of the major issues in cloud Computing is Load Balancing. The efficient utilization of resources and response time is the major challenge faced in data centers. Lack of algorithm that implements some techniques to improve the utilization of resources and to reduce the response time will only increase the operational cost and give customer dissatisfaction. In this paper, we have discussed several algorithms and have focused on improving the resource allocation technique and thereby reducing the response time. We have considered several environments under which the three algorithms namely Round Robin, Equally Spread Current Execution Algorithm and Throttled Load Balancing Algorithm run. Based on the Compile time or the Run time simulation, the algorithm is classified into static or dynamic environment respectively. While load balancing technique in static environment is more stable and easy to predict, dynamic load balancing promises to provide better performance. The algorithm was implemented using CloudAnalyst. The simulation results shows that the cost and time has reduced up to 50%-60% and this shows the effectiveness of our algorithm. The algorithms used in load balancing can be used in research. In the future, we may implement the same algorithm using neural networks in Artificial Intelligence. While this paper focuses on manual load allocation, in future with neural networks we can assign automatic load balancing in data centers.

REFERENCE

[1] Mayanka Katyal, Atul Mishra, *A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment*, *December 2013* International Journal of Distributed and Cloud Computing, Volume 1 Issue 2.

[2] Fazel Mohammadi , Dr. Shahram Jamali , and Masoud Bekravi "Survey on Job Scheduling algorithms in Cloud Computing"International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) Volume 3, Issue 2, March – April 2014.

[3] Dharmesh Kashyap, Jaydeep Viradiya(2014), *A Survey Of Various Load Balancing Algorithms In Cloud Computing*, INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 3, ISSUE 11, NOVEMBER 2014 ISSN 2277-8616 115 IJSTR©2014 www.ijstr.org

[4] Upendra Bhoi, Purvi N. Ramanuj, ―Enhanced Max-min Task Scheduling Algorithm in Cloud Computing‖ International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 2, Issue 4, April 2013

[5] Karanpreet Kaur, Ashima Narang, Kuldeep Kaur, "Load Balancing Techniques of Cloud Computing", International Journal of Mathematics and Computer Research, April 2013

[6] Dhinesh B. L.D , P. V. Krishna, ―Honey bee behavior inspired load balancing of tasks in cloud computing environments‖, in proc. Applied Soft Computing, volume 13, Issue 5, May 2013, Pages 2292-2303.

[7] Tushar Desai, Jignesh Prajapati ,‖ A Survey Of Various Load Balancing Techniques And Challenges In Cloud Computing‖ International Journal Of Scientific & Technology Research , Volume 2, Issue 11, November 2013

[8] Baris Yuce , Michael S. Packianather ,Ernesto Mastrocinque , Duc Truong Pham and Alfredo Lambiase "Honey Bees Inspired Optimization Method: The Bees Algorithm" insects 1 July 2013; Published: 6 November 2013.

[9] Shilpa V Pius, Shilpa T S "Survey on Load Balancing in Cloud Computing" International Conference on Computing, Communication and Energy Systems (ICCCES-2014).

[10] Kousik Dasguptaa, Brototi Mandalb, Paramartha Duttac, Jyotsna Kumar Mondald, Santanu Dame, A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing" International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA) vol 10,2013.

[11] N. S. Raghava and Deepti Singh "Comparative Study on Load Balancing Techniques in Cloud Computing" OPEN JOURNAL OF MOBILE COMPUTING AND CLOUD COMPUTING Volume 1, Number 1, August 2014.

[12] Doddini Probhuling L**.,** *LOAD BALANCING ALGORITHMS IN CLOUD COMPUTIN*, International Journal of Advanced Computer and Mathematical Sciences ISSN 2230-9624. Vol4, Issue3, 2013, pp229-233.,http://bipublication.com

[13] Nusrat Pasha, Dr. Amit Agarwal Dr. Ravi Rastogi, *Round Robin Approach for VM Load Balancing Algorithm in Cloud Computing Environment*, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 5, May 2014 ISSN: 2277 128X

[14] Amandeep Kaur sidhu1 and Supriya Kinger2, "Analysis of Load Balancing Techniques in Cloud Computing", International Journal of Computers & Technology, volume 4, No. 2, March- April 2013, pg 737- 741.

[15] Pooja Samal1 and Pranati Mishra2, "Analysis of Variants in Round Robin Algorithms for Load Balancing in Cloud Computing", (IJCSIT) International Journals of Computer Science and Information Technologies, Volume 4 (3), 2013, pg. no. 416- 419.

[16] B. Santosh Kumar1 and Dr. Latha Parthiban2, "An Implementation of Load Balancing Policy for Virtual Machines Associated with a Data Centre", International Journal of Computer Science & Engineering Technology (IJCSET), volume 5 no. 03, March 2014, pp. 253- 261.

[17] Sonika Matele1, Dr, K James2 and Navneet Singh3, "A Study of Load Balancing Issue Among Multifarious Issues of Cloud Computing Environment", International Journals of Emerging Technolog Computational and Applied Science (IJETCAS), volume 13- 142, 2013, pg. 236- 241

[18] Dr. Rakesh Rathi1, Vaishali Sharma2 and Sumit Kumar Bole3, "Round Robin Data Center Selection in Single Region for Service Proximity Service Broker in Cloud Analyst" , International Journal of Computer & Technology, Volume 4 no. 2, March- April 2013, pg. no. 254- 260.

[19] Kunal Mahurkar1, Shraddha Katore2 and Suraj Bhaisade3, Pratikawale4, "Reducing Cost of Provisioning in Cloud Computing", International Journal of Advance in Computer Science and Cloud Computing, Volume- 1, Issue- 2, nov.- 2013, pg. 6- 8.

[20] Dr Hemant S. Mahalle1, Prof Parag R. Kaver2 and Dr. Vinay Chavan3, "Load Balancing on Cloud Data Centres", Internatinal Journal of Advanced Reserch in Computer Science and Software Engineering, volume 3, issue 1, January 2013, pp. 1- 4.

[21] Subasish Mohapatra1, Subhadarshini2 and K. Smruti Rekha3, "Analysis of Different Varients in Round Robin Algorithms for Load Balancing in Cloud Computing", International Journal of Computer Application, Volume 69- no. 22, may 2013, pp. 17-21.

[22] Ajay Gulati1 and Ranjeev K. Chopra2, "Dynamic Round Robin for Load Balancing in a Cloud Computing", International Journal of Computer Science and Mobile Computing, volume 2, issue 6, June 2013, pg 274- 278.

[23] Pooja Samal, Pranati Mishra, ‖Analysis of variants in Round Robin Algorithms for load balancing in Cloud Computing‖ (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (3) , 2013, 416-419.

[24] International Conference on Cloud and Service Computing 978-1-4577-1637-9/11/$26.00 ©2011 IEEE. Shenzhen, China: ZTE Corporation.

[25] Venubabu Kunamneni, "Dynamic Load Balancing for the cloud", International Journal of Computer Science and Electrical Engineering, 2012.

[26] Nidhi Jain Kansal, Inderveer Chana, "Cloud Load Balancing Techniques: A Step Towards Green Computing", International Journal of Computer Science, January 2012

[27] Che-Lun Hung1, Hsiao-hsi Wang2 and Yu-Chen Hu2, ―Efficient Load Balancing Algorithm for Cloud Computing Network‖. IEEE Vol. 9, pp: 70-78, 2012

[28] Prof Meenakshi Sharma1 and Pankaj Sharma2, "Performance Evaluation of Adaptive Virtual Machine Load Balancing Algorithm", International Journal of Advanced Computer Science and Applications, volume 3, no. 2, 2012, pp. 86-88.

[29] Computing and Applications 978-0-7695-4943-9/12 $26.00 © 2012 IEEE DOI 10.1109/NCCA.2012.29.

[30] Al Nuaimi, K., Mohamed, N., Al Nuaimi, M. & Al-Jaroodi, J. (2012). *A survey of load balancing in cloud computing: challenges and algorithms*. 2012 IEEE Second Symposium on Network Cloud.