Chaotic Biogeographic Based Optimization Load Balancing in Cloud Computing

Swati Rana
NCCE, Israna, Panipat, India.

Sukhvir Singh
NCCE, Israna, Panipat, India.

Abstract – Cloud computing is the enactment of using a network of remote servers hosted on the internet to accumulate, manage and process data rather than a local server or a personal computer. Cloud computing is known as a popular and important term in the IT society these days. As the main goal of cloud computing we can mention the better use of distributed resources and applying them to attain a higher throughput, performance and solving large scale computing problems. In this thesis we focus on load balancing algorithm, chaotic BBO. As we observe in previous work that they assign jobs to data centers in some way like either in increasing or in decreasing order. That' why there cost and load on data centers increased. Now in our proposed work we emphasis on load balancing of cloud computing environment using chaotic biogeographic based optimization in which we randomly assign the jobs to data centers to decrease the cost, time or load on data centers.

Index Terms – load balancer, cloud computing, biogeographic based optimization, Chaotic biogeographic based optimization.

1. INTRODUCTION

Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., computer networks, servers, storage, applications and services), which can be rapidly provisioned and released with minimal management effort.

Figure 1: Cloud Computing

Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in either privately owned, or third-party data centers that may be located far from the user–ranging in distance from across a city to across the world.

Cloud computing relies on sharing of resources to achieve coherence and economy of scale, similar to a utility (like the electricity grid) over an electricity network. Advocates claim that cloud computing allows companies to avoid up-front infrastructure costs (e.g., purchasing servers). As well, it enables organizations to focus on their core businesses instead of spending time and money on computer infrastructure. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables information technology (IT) teams to more rapidly adjust resources to meet fluctuating and unpredictable business demand. Cloud providers typically use a "pay as you go" model. This will lead to

Unexpectedly high charges if administrators do not adapt to the cloud pricing model.

In 2013, it was reported that cloud computing had become a highly demanded service or utility due to the advantages of high computing power, cheap cost of services, high performance, scalability, accessibility as well as availability. Some cloud vendors are experiencing growth rates of 50% per year, but being still in a stage of infancy, it has pitfalls that need to be addressed to make cloud computing services more reliable and user friendly.

2. PROBLEM STATEMENT

As we studied or noticed in our base paper that there were high cost as well as load on our data centers or virtual machines. The main idea behind their work was that they assign the jobs to data centers in some order like either in increasing or decreasing order. This is the main disadvantage of their work because if we calculate the cost according to them, then we get high cost and also we get high load on our data centers or we can say our virtual machines. So, these were the main problem that they sort their jobs in some order which assign to data centers.

3. PROPOSED WORK

In this thesis, we proposed to optimize our arrangement through chaotic biogeographic based optimization. Through chaotic biogeographic based optimization we get less cost for our task. From below diagram if we calculate the cost then we get less cost, less load as compare to previous research. We are assigning the jobs to CPU such that we get the less cost. This work is done through optimization.

3.1 ARCHITECTURE OF LOAD BALANCER

Cloud computing is the practice of using a network of remote servers hosted on the internet to store, manage and process data rather than a local server or a personal computer. Cloud computing is known as a popular and important term in the IT society these days. As the main goal of cloud computing we can mention the better use of distributed resources and applying them to achieve a higher throughput, performance and solving large scale computing problems.

To achieve these kinds of goals, improving the general performance of system, maintain stability, availability and some other features for a cloud computing data centre, we need a mechanism which is called load balancing. Load balancing is the process of distributing the general system work load among all nodes of the distributed system to improve both resource utilization and job response time while avoiding a situation where some nodes are overloaded while others are under loaded or idle. In order to avoid failure, load balancing is often used by controlling the input traffic and stop sending the workload to resources which become overloaded and non-responsive.

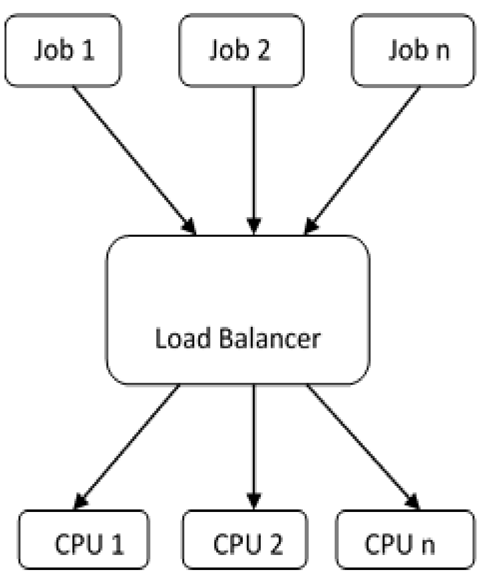

Figure 3.1: Load Balancer

3.2 ASSIGNING JOBS TO CPU

In our proposed work, we implemented Chaotic Biogeographic Based Optimization algorithm in which we optimize the jobs which are located to CPU. We optimize our arrangement to get less cost and less load as compared to previous research.

In the following figure 3.2(a) we assign the jobs to CPU. Firstly we arrange the jobs in some order like in this we arrange the jobs in increasing order to keep in mind that lesser jobs completed in less time. Therefore, lesser jobs get highest priority. Now if we calculate the total cost from the below table 1 then we get 244.

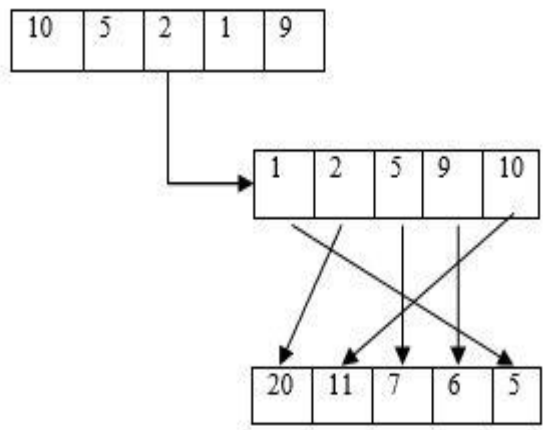

Figure 3.2(a): Assigning jobs to CPU

Table 1: Total cost

Jobs	CPU	Cost
1	5	5
2	20	40
5	7	35
9	6	54
10	11	110

If we calculate the total cost then we get 244.

In the below figure we assign the jobs to CPU chaotically means randomly. We set the number of iterations to get less cost and less load on CPU or virtual machines. From below table we calculate the total cost then we get 181. This is less as compared to table 1.

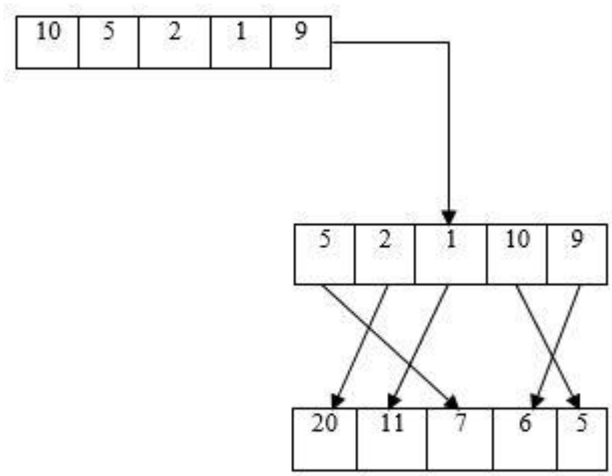

Figure 3.2(b): Assigning jobs to CPU

Table 2: Total cost

Jobs	CPU	Cost
5	7	35
2	20	40
1	11	22
10	5	50
9	6	54

If we calculate the total cost then we get 181.

3.3 ALGORITHM STEPS

The CBBO algorithm can be described in the following way.

Step 1) Initialize the BBO parameters like habitat modification probability Pmod, mutation probability, maximum mutation rate r_{max}, max immigration rate /, max emigration rate E, lower bound and upper bound for immigration probability per gene, step size for numerical integration dt, number of habitat N, number of suitability index variables m, elitism parameter “p” which indicates the number of best habitats to be retained in the habitat matrix as it is, from one generation to the next without performing migration operations on them, etc. Set maximum number of iteration. Generate the SI Vs of the given problem within their feasible region using random number. A complete solution consisting of suitability index variabless is known as one habitat H. There are several numbers of habitats to search the optimum result.

Suppose we are minimizing a function f(x)$\sum_{i=1}^{m} x_1^2 and 0 < xi < z$.Initialize several numbers of habitats depending upon the habitat size within feasible region. Each habitat represents a potential solution to the given problem. So total habitat in matrix form is written in the following forms:

$$\begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & \cdots & x_{1m} \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & \cdots & x_{2m} \\ x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & \cdots & x_{3m} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{N1} & x_{N2} & x_{N3} & x_{N4} & x_{N5} & \cdots & x_{Nm} \end{pmatrix}$$

Step 3) Calculate the HS1 value for each habitat of the population set for given emigration rate /z, immigration rate A. For the function f(x) = $\sum_{i=1}^{m} x_i^2$, habitat suitability index of all *N suitability index variables* sets is calculated as per the following way:

Habitat suitability index$_{1=}$ $[x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2 + x_{15}^2 + \dots\dots\dots . x_{1m}^2]$

Habitat suitability index$_2$=$[x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2 + x_{15}^2 + \dots\dots\dots . x_{1m}^2]$

Habitat suitability index$_3$=$[x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2 + x_{15}^2 + \dots\dots\dots . x_{1m}^2]$

……………………………………………………

Habitat suitability indexN=$[x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2 + x_{15}^2 + \dots\dots\dots . x_{1m}^2]$

Calculate the number of valid species out of all habitats using their habitat suitability index values. Those habitats, whose fitness values, i.e., habitat suitability index values, are finite, are considered as valid species S.

Step 4) Based on the optimum habitat suitability index value, elite habitats are identified.

Step 5) probabilistically immigration rate and emigration rate are used to modify each non-elite habitat using migration operation. The probability that a habitat Hi is modified is proportional to its immigration rate A; and the probability that the source of the modification comes from a habitat Hj is proportional to the emigration rate /ij. Habitat modification using migration operation can be described as follows.

Select a habitat H_t with probability proportional to ,

If *Hi* is selected

For *j* = 1 to *N*

Select another habitat Hj with probability proportional to //., If Hj is selected

Randomly select an *suitability index variables* from habitat Hj

Replace a random *suitability index variables* in Hi with that selected *suitability index variables* of Hj

End

End

End

From this algorithm, we note that elitism can be implemented by setting $\lambda = 0$ for the best p habitats. After each habitat is modified, its feasibility as a problem solution should be verified. If it does not represent a feasible solution, then the above procedure is ignored and the same procedure is performed again in order to map it to the set of feasible solutions. After modification of each non-elite habitat using migration operation, each habitat suitability index is recomputed.

Step 6) for each habitat, the species count probability is updated using (17). Mutation operation is performed on each non-elite habitat as discussed in Section IV-B and habitat suitability index value of each habitat is computed again. Mutation operation can be described as follows.

For $i = 1$ to N

For $j = 1$ to m

Use μi and μi to compute the probability P_t using (17)

Select a *suitability index variables* $H_i(j)$ with probability proportional toPi

If $Hi(j)$ is selected

Replace with a randomly generated *suitability index variables*within its feasible region

End

End

End

As with habitat modification, elitism can be implemented by setting the probability of mutation selection Pi to zero for the best p habitats. After each habitat is modified, its feasibility as a problem solution should be verified. If it does not represent a feasible solution, then the above step is ignored and the above-mentioned method is applied again in order to map it to the set of feasible solutions.

Step 7) Go to step 3) for the next iteration. This loop can be terminated after a predefined number of iterations have been found.

3.4 Flowchart

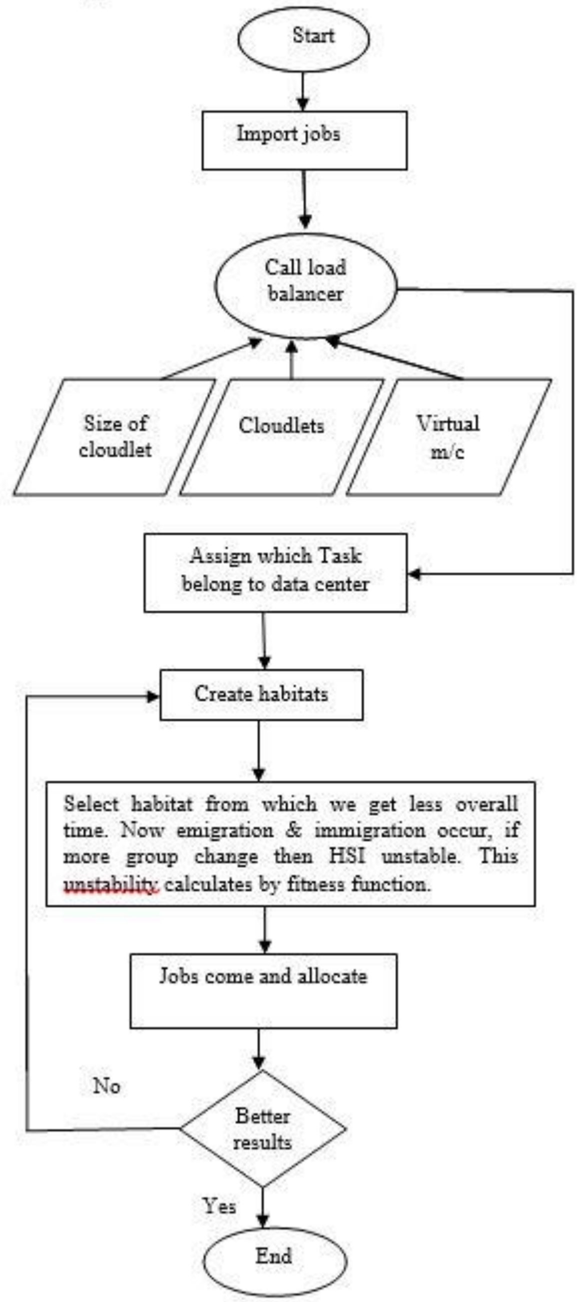

Figure 3.4: Flowchart

4. IMPLEMENTATION AND RESULTS

4.1 IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve change over and evaluation of changeover methods. We did our implementation using MATLAB.

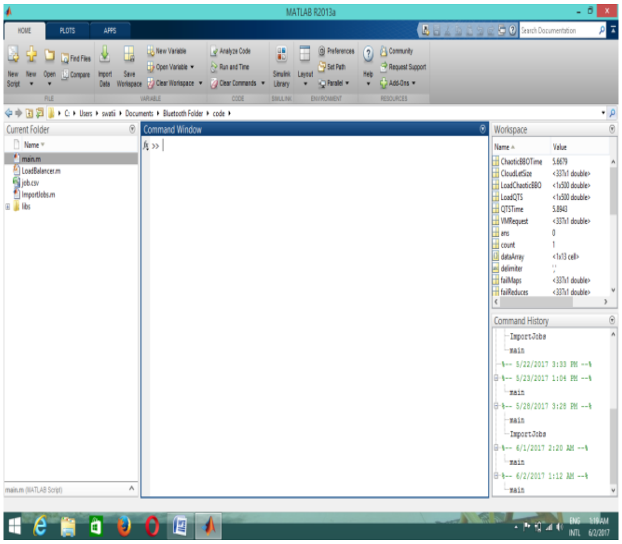

Figure 4.1: Main screen

4.2 RESULTS

4.2.1 Chaotic map for BBO Optimization

This graph shows our chaotic Biogeographic based optimization. This is our proposed algorithm graph.

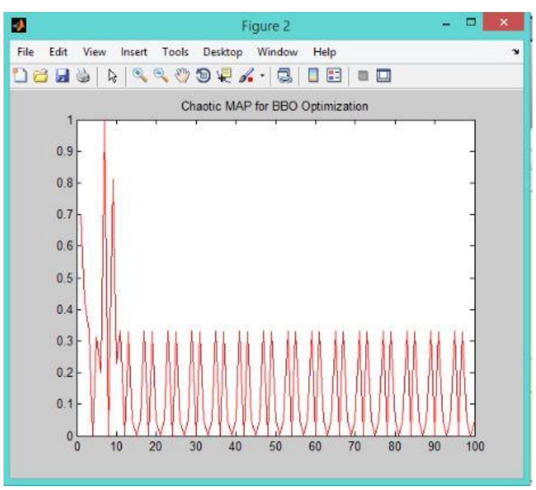

Figure 4.2.1: Chaotic map for BBO Optimization

4.2.2 Comparison of cost

This screenshot show the comparison of chaotic biogeographic based optimization cost to the base paper. This clearly shows that we get less cost for our proposed algorithm as compared to our base paper [7].

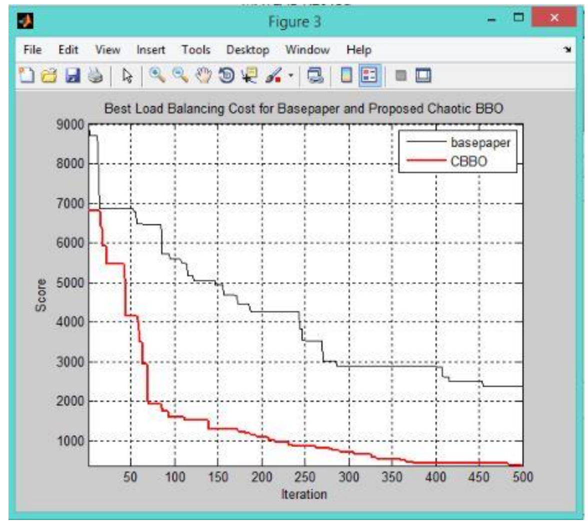

Figure 4.2.2: Best load balancing cost for base paper and proposed CBBO

4.2.3 Loads on server

This screenshot shows the load on servers for jobs. In this we show that we take two servers one for CBBO and another for base paper. This clearly shows that we get fewer loads on our server and more load on another server which is for base paper [7].

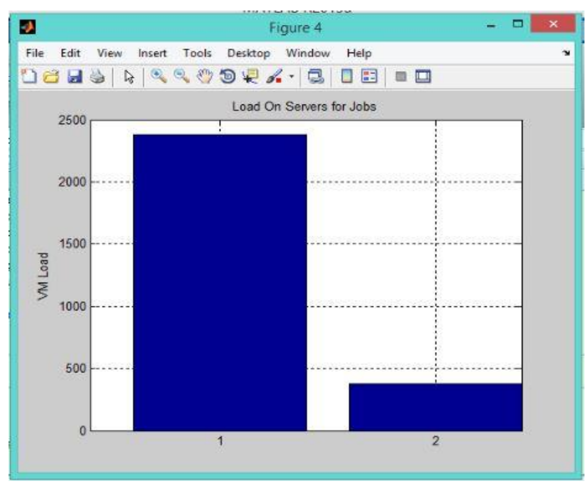

Figure 4.2.3: Load on server for jobs

5. CONCLUSION

In this thesis, we observe the problem of load balancer in cloud computing. To ensure the load balancing in cloud we proposed an effective algorithm through which we randomly assign the jobs to data centers through load balancer. In this we take number of iterations by which we select the best job allocation, and then through that allocation we get less load and less cost. As we proposed algorithm which is named as chaotic BBO (Biogeographic Based Optimization) it give better results, we used BBO algorithm as a reference, it is concluded that for balancing the load the proposed algorithm overcome the previous problem successfully.

6. FUTURE SCOPE

In future we can work for optimization of energy of cloud data centers along side with parallelization of Batch jobs using MAP Reduce Framework. we get better energy efficiency. By Efficient Utilization of Task loads and when we evenly distribute the task energy and power is minimized. Also minimize the energy consumption by minimizing the number of servers used.

REFERENCES

[1] Ali M Alakeel, "A Guide to Dynamic Load Balancing in Distributed Computer Systems," IJCSNS International Journal of Computer Science and Network Security, vol.10 No.6, pp. 153-160 June 2010.

[2] O. M. Elzeki, M. Z. Reshad, M. A. Elsoud, " Improved Max-Min Algorithm in Cloud Computing," International Journal of Computer Applications, vol 50 – No.12, pp. 22-27, July 2012.

[3] Srinivas Sethi, Anupama Sahu Suvendu Kumar Jena,"Efficient load Balancing in Cloud Computing using Fuzzy Logic," IOSR Journal of Engineering (IOSRJEN), vol 2, pp. 65-71, July 2012.

[4] Tushar Desai, Jignesh Prajapati,"A Survey Of Various Load Balancing Techniques And Challenges In Cloud Computing," International journal of scientific and technology research, vol 2, pp. 158-161, November 2013.

[5] Amandeep, Vandana Yadav2, Faz Mohammad," Different Strategies for Load Balancing in Cloud Computing Environment: a critical Study," International Journal of Scientific Research Engineering & Technology (IJSRET), vol. 3, pp. 85-90, April 2014.

[6] Ren Gao, Juebo Wu," Dynamic Load Balancing Strategy for Cloud Computing with Ant Colony Optimization," Future Internet, pp. 465-483, July 2015.

[7] Sanjay Chakraborty, Nilotpal Choudhury," A Study of a New Dynamic Load Balancing Approach in Cloud Environment," World Journal of Computer Application and Technology, pp. 31-37, April 2016.

[8] Kousik Dasgupta, Brototi Mandlab, Paramartha Duttac, Jyotsna Kumar Mondald ," A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing," International Conference on Computational Intelligence, pp. 340-347,October 2013.

[9] Aayushi Sharma, Umang Singh," Study on Load Balancing Techniques in Ant colony Optimization for Cloud Computing," International Journal of Computer Applications, pp. 5-10, April 2016.

[10] Thomas Yeboah, Odabi," Hybrid Bee Ant Colony Algorithm for Effective Load Balancing And Job Scheduling In Cloud Computing," West African Journal of Industrial And Academic Research, vol. 13, pp. 54-59, April 2015.

[11] Manisha Verma, Neelam Bhardwaj, Arun Kumar Yadav," Real Time Efficient Scheduling Algorithm for Load Balancing in Fog Computing Environment, " I.J. Information Technology and Computer Science, pp. 1-10, April 2016.

[12] Anjali, Jitender Grover, Manpreet Singh, Charanjeet Singh, Hemant Sethi," A New Approach for Dynamic Load Balancing in Cloud Computing" IOSR Journal of Computer Engineering, pp. 30-36, August 2015.

[13] Gulshan soni, Mala Kalra," A Novel Approach for Load Balancing in Cloud Data Center," InternationalAdvance Computing Conference (IACC), pp. 807-812, July 2014.

[14] Suresh M, Santhosh Kumar, "A Load Balancing Model In Public Cloud Using ANFIS AND GSO, " International Conference on Intelligent Computing Applications, pp. 85-89, Feb. 2014.

[15] Samarsinh Prakash Jadhav, Priya R. Deshpande, "Load Balancing in Cloud Computing," International Journal of Science and Research (IJSR), Volume 3 Issue 6, pp. 2282-2285, June 2014.

[16] Sukhvir Kaur, Supriya Kinger," Review on Load Balancing Techniques in Cloud Computing Environment," International Journal of Science and Research (IJSR), Volume 3 Issue 6, pp. 2249-2504, June 2014.

[17] Vartta Siyal, Dr. Naveen Choudhary, Dr. Dharm Singh, "Efficient Load Distribution of VM in Cloud Computing," International Journal of Scientific and Engineering Research, Volume 4, Issue 12, pp. 2105-2108, December-2013

[18] Jaspreet Kaur, "Comparison of load balancing algorithms in a Cloud," International Journal of Engineering Research and Applications, Vol. 2, Issue 3, pp.1169-1173, May-Jun 2012.

[19] Rong wang, Bin rao, "Research on the Cloud Computing Load Balance Degree of Priority Scheduling Algorithm based on Convex Optimization Theory," International Conference on Informatization in Education, Management and Business, pp. 156-160, September 2015.

[20] Suriya Begum, Dr. Prashanth CSR, "Review of Load Balancing in Cloud Computing," IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, pp. 343-352, January 2013.

[21] Rajeev Kumar, Tanya Prashar, "Performance Analysis of Load Balancing Algorithms in Cloud Computing," International Journal of Computer Applications, Volume 120 – No.7, pp. 19-27, June 2015.

[22] Liu Kun, Xu Gaochao, Chen Jingxia2, "Research on Cloud Computing Load Balancing Based on Virtual Machine Migration," The Open Cybernetics & Systemics Journal, pp. 1334-1340, July 2015.

[23] D. Saranya, L. Sankara Maheswari, "Load Balancing Algorithms in Cloud Computing: A Review," International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, pp. 1107-1111, July 2015.

[24] Vinay Darji, Jayna Shah, Rutvik Mehta, "Survey paper on various load balancing algorithms in cloud computing," International Journal of Scientific & Engineering Research, Volume 5, Issue 5, pp 583-587, May-2014.