An Implementation of Load Balancing Policy for Virtual Machines Associated With a Data Center

B.SANTHOSH KUMAR
Assistant Professor, Department Of Computer Science, G.Pulla Reddy Engineering College.
Kurnool-518007, Andhra Pradesh.India.
Tel: 91-9963038639
E-mail: santhoshkumar.bala@gmail.com

DR.LATHA PARTHIBAN
Department of Computer Science, Community College, Pondicherry University.
Tel: 91-9884378241
E-mail: lathaparthiban@yahoo.com

Abstract

The main aspect of any cloud application is to provide simultaneous access to number of users by reducing the cost of maintenance. This is mainly achieved by the process of implementing virtualization effectively. Virtualization is considered to be the core of cloud computing and it mainly deals with the process of creating and maintaining the virtual machines efficiently. This paper focuses on implementing an algorithm to distribute the load among virtual machines created in a data center. The algorithm mainly relies upon Least Frequently Used mechanism. Each time when a task has to be submitted to a data center for processing choice is made upon the virtual machine which has been assigned tasks least number of times. This process is mainly employed to keep the virtual machines busy in processing for most of the time. The simulation tool Cloud Analyst is used to develop and test the algorithm. An analysis also has been made on the proposed algorithm with the ones that are already in practice with the simulation tool.

Keywords: Virtualization, Least Frequently Used, Cloud Analyst, Cloud Computing.

I. INTRODUCTION

Cloud Computing can be considered as a model for developing, maintaining and accessing applications by paying for the resources which are only used for certain time. To serve huge number of requests from different users located at different parts of the globe on a pay-per-usage model the process of virtualization has been followed in cloud environment. There are service providers who are responsible for maintaining the application on cloud environment. The most predominant providers are Google, Amazon, Microsoft and many others. In order to serve the huge traffic these service providers maintain data centers all over the world where the data is stored in bulk and requests are processed. Depending on the number of requests to be processed by a data center virtual machines are created where different operating systems and configurations can be run on a single central processing unit. It gives the illusion to the users that numbers of CPUs are existing for processing the requests. This whole process can be considered as virtualization and is monitored by software known as hypervisor. The hypervisor is mainly responsible in creating and maintaining the virtual machines. The requests from all the users are considered as individual tasks and each of the virtual machines is assigned a task and effort is made to keep them busy for longer duration. This is considered to be load balancing policy which makes an effort to maximize the throughput of virtual machines.

In order to simulate the whole process on personal computers a tool by name Cloud Analyst has been developed by the CLOUDS laboratory in university of Melbourne. The main intention behind developing this tool is to simulate the traffic generated by most used applications such as face book, Gmail and analyze the response times at each data center. The simulation process is divided into regions denoted by R**i** where **'i'** indicates the region number and number of requests generated in each region can be considered as a User Base denoted by UB**i** where **'i'** indicates the user base number. The data centers are denoted by DC**i** where **'i'** indicates the corresponding data center number. The following figure gives a better understanding of how the regions are divided for simulation and the table1 summarizes the regions. [1] Elaborates each and every detail of the tool.

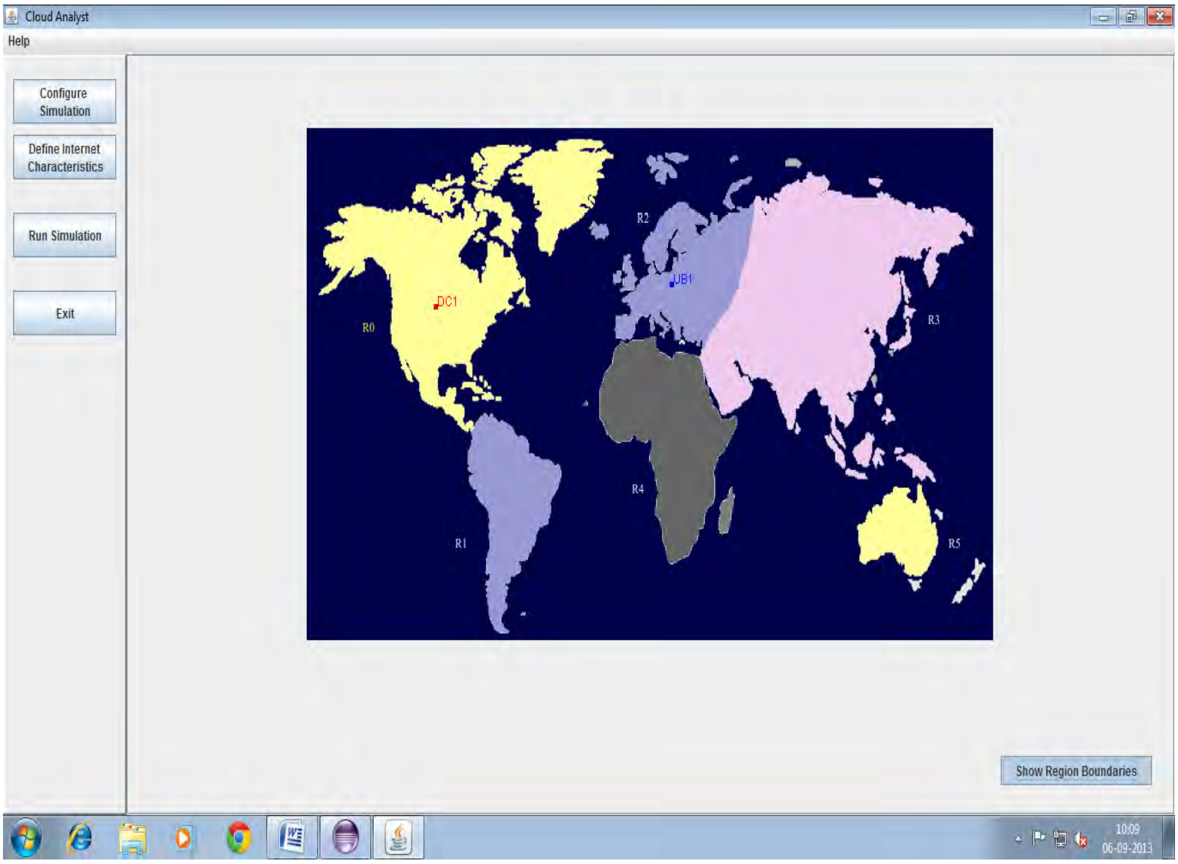


Figure 1: Screen shot illustrating the region boundaries as well as the home screen.

Region	**Cloud Analyst Region Id**
North America	Ro
South America	R1
Europe	R2
Asia	R3
Africa	R4
Oceania	R5

Table 1: Table representing the region Ids and their corresponding names

Further the paper is organized as follows. Section II gives an overview of the related work that has been done before. Section III gives a comprehensive understanding of the proposed algorithm along with the steps to be followed in the simulation process. Section IV draws a performance analysis of the proposed work with the already existing load balancing policies in the tool. Section V briefs the conclusion and future enhancements that can be done based on the current work.

II. RELATED WORK

A considerable amount of work has been done in framing the load balancing policies in a cloud environment. [2] Gives an idea on load balancing the virtual machines so that they can be efficiently used. [3] Makes a comparative study of different algorithms based on distributed load balancing in a cloud environment. [4] Emphasizes on a load balancing policy based on game theory and partitioning the cloud. Beginning from the era of grid computing numerous tools have been developed to demonstrate the load balancing algorithms. In order to make studies related to grid computing [5] gives an understanding of a tool known as GangSim. SimGrid simulation framework introduced in [6] supported the scheduling policies in distributed applications. Simulation of dynamic grid replication strategies have been demonstrated in [7] using a tool OptorSim. [8] Focuses on toolkit for modeling and simulation of grid computing known as GridSim.

The rapid advancement in technology led to the cost maximization for developing and maintaining the application. In order to reduce the cost and improve the maintenance of applications pay-per-usage concept has been introduced which eventually led to the development of cloud computing environment. This led to the development of many tools which simulated the cloud applications. The most predominant one among them was the CloudSim [9] which was developed for both modeling and simulation of cloud computing environment. Based on this tool a lot of research has been performed in developing many tools useful for cloud simulation. Cloud Analyst [10] can be considered as the most effective tool based on CloudSim as this gave a visual idea of simulating the huge traffic associated with some of the rich online applications such as face book. A cloudlet which is considered to be a mini cloud is created and submitted to different data centers of the service providers across the globe. The simulator allows the user to specify the number of virtual machines and their corresponding images which should be created in each data centre. Depending on the requests from each user base the traffic is diverted to different data centers .The data centers create the virtual machine instances and submit the requests according to load balancing policies such as round robin, throttled requests. The main components or classes used to develop cloud analyst can be considered as Data Centre Broker, VmScheduler, Cloudlet and many more.

III. PROPOSED WORK

Algorithm **vmLeastFrequentlyUsed**

Input: Task ‘T’ to be allotted to a virtual machine for processing.

1. Initialize vmCount ← number of virtual machines created in a data center.
2. Create a Hash Map vmAllocation< vmid , count>
   Where vmid refers to id of virtual machine created. Generally it will be from 0 to n.
   count refers to number of times that vmid is assigned a task.
3. Initialize count to zero for all vmid.
4. If task ‘T’ is in waiting state and has to be assigned to any virtual machine DO

```
For i ← 0 to vmCount-1
    For j ← i+1 to vmCount-1
        If( vmAllocation. Get (i). count <= vmAllocation. Get (j).count)
            Assign ‘T’ to virtual machine with vmid ‘i’
            vmAllocation.put (i, count+1)
        End
    End
End
```

5. If task ‘T’ is completed make the count as zero for that particular vmid.
6. Repeat steps 4, 5 until all the tasks are allotted to the virtual machines created in a data center.

In addition to the load balancing policies existing in cloud analyst such as round robin, throttled requests a new algorithm which is based on the least frequently used mechanism is proposed as shown above. The algorithm depicted above is added as a part of cloud analyst tool in a file called vmLeastFrequentlyUsed which extends the abstract class vmLoadBalancer. In order to simulate the algorithm, 6 user bases have been added in 6 regions and 4 data centers have been created. In order to add the data centers Data Center configuration tab is selected in the Configure Simulation screen and four of them have been added. Figure 2 gives an understanding of how the process is done.

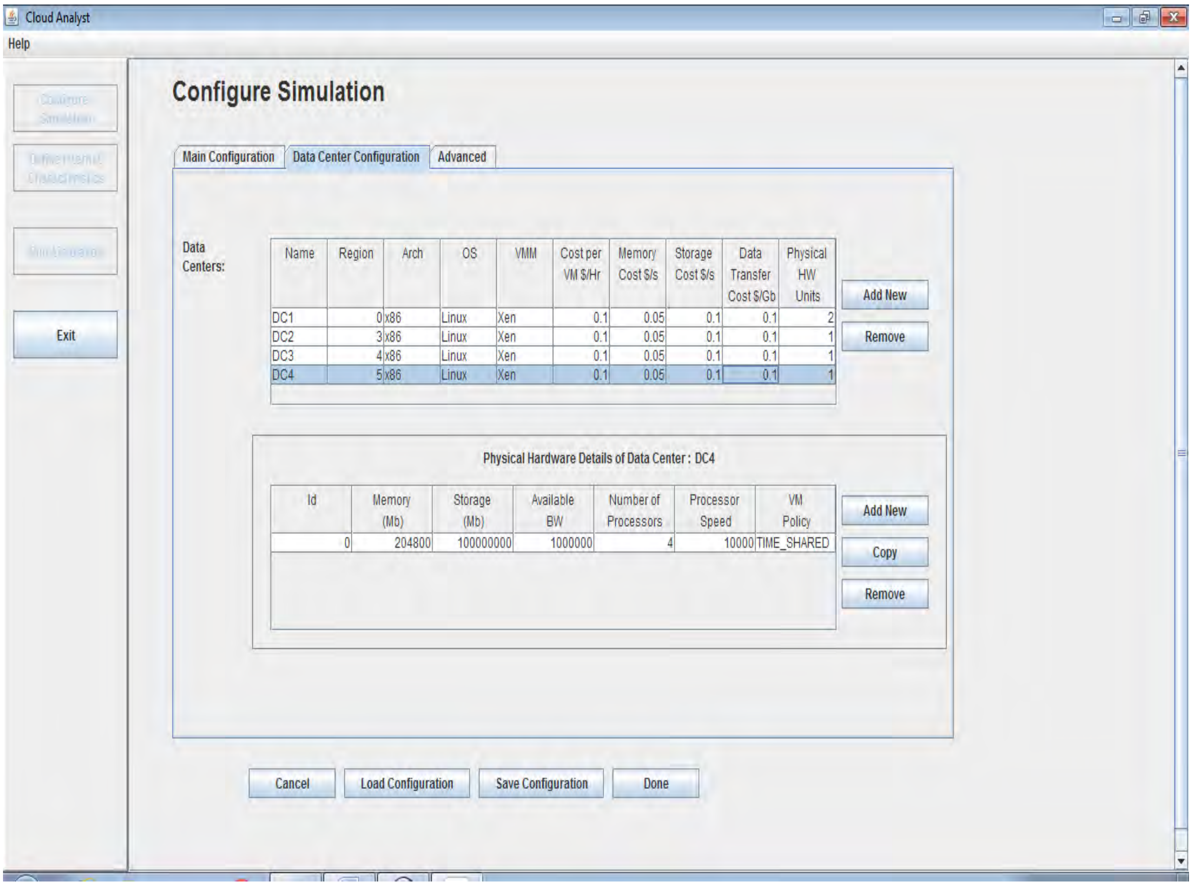


Figure 2: Screen shot indicating the creation of data centers.

After the data centers are added the tab main configuration is selected on configure simulation screen to add 6 user bases .Further the screen is also designed to add the number of virtual machines in each data center and the image size of each virtual machine and the memory ,bandwidth occupied by each of them can be specified. Figure 3 gives a snapshot of how the whole process is done.

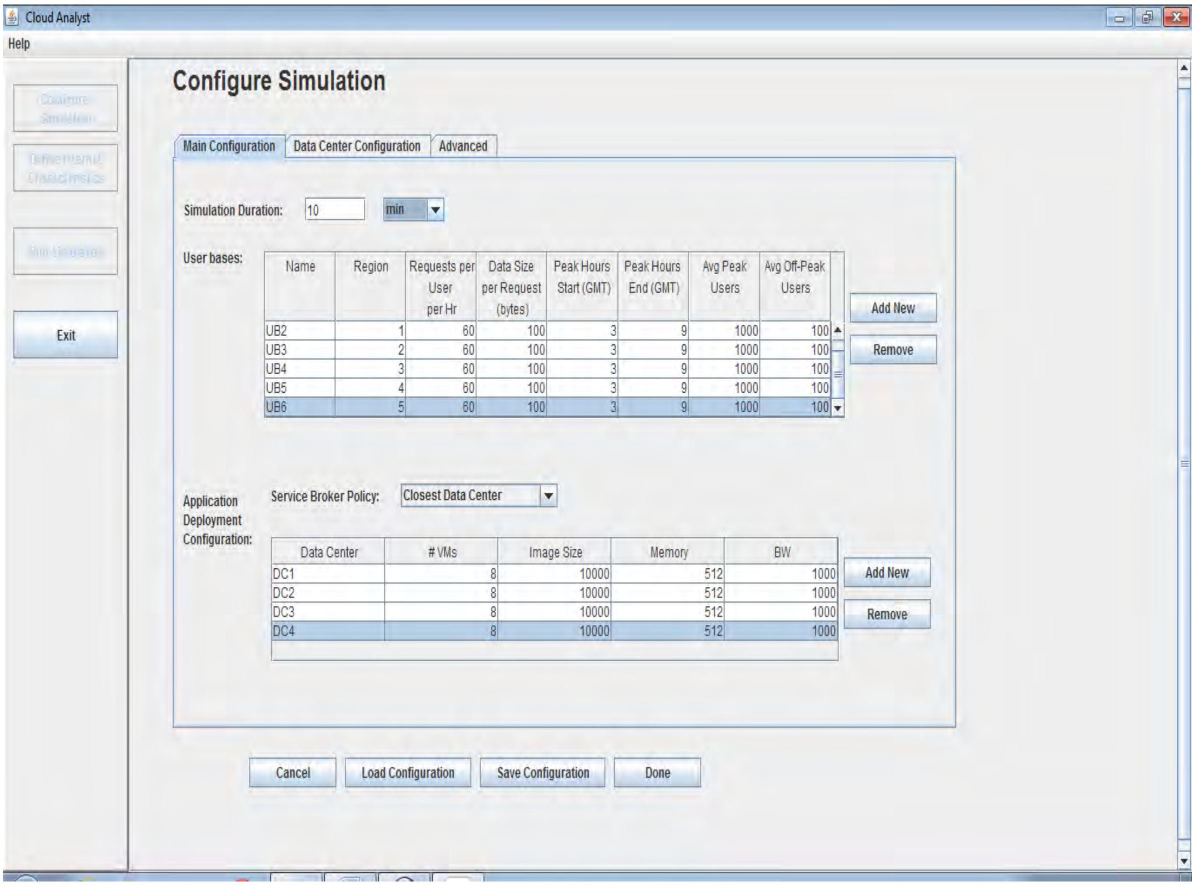


Figure 3: Screen indicating the addition of user bases and data center characteristics.

The user bases UB1,UB2,UB3,UB4,UB5,UB6 are added respectively with each of them in different regions and the requests per user per hour is specified as 60. There is a provision to select the broker policy by which the data center is selected for processing. This simulation provides three different broker policies namely closest data center, optimize response time and Reconfigure dynamically with load. In the present scenario closest data center policy is selected which results in request processing at the data center which is close to the corresponding user base. The number of virtual machines to be created at each data center is specified as 8 and their image size, memory and bandwidth are specified accordingly.

After the whole process of creating the data centers and user bases is finished the advanced tab on configure simulation screen is selected. It gives an option to add the load balancing policy for virtual machines created in a single data center. The algorithms which already exist with the simulation tool for load balancing are Round Robin, Throttled and Equally spread load. Round Robin policy allocates the tasks to virtual machines in a sequential manner. Throttled based policy allots the tasks to each virtual machine until its full capacity and then shifts to another virtual machine. Equally spread load algorithm tries to even out the number of active tasks assigned to each virtual machine. In addition to these algorithms a new load balancing policy option is added to the drop down list and when selected the proposed algorithm vmLeastFrequentlyUsed is executed. Figure 4 gives the scenario where user is given the option to select least frequently used load balancing policy. The code for the proposed algorithm is added to a file by name vmLeastFrequentlyUsed which extends the abstract class vmLoadBalancer.

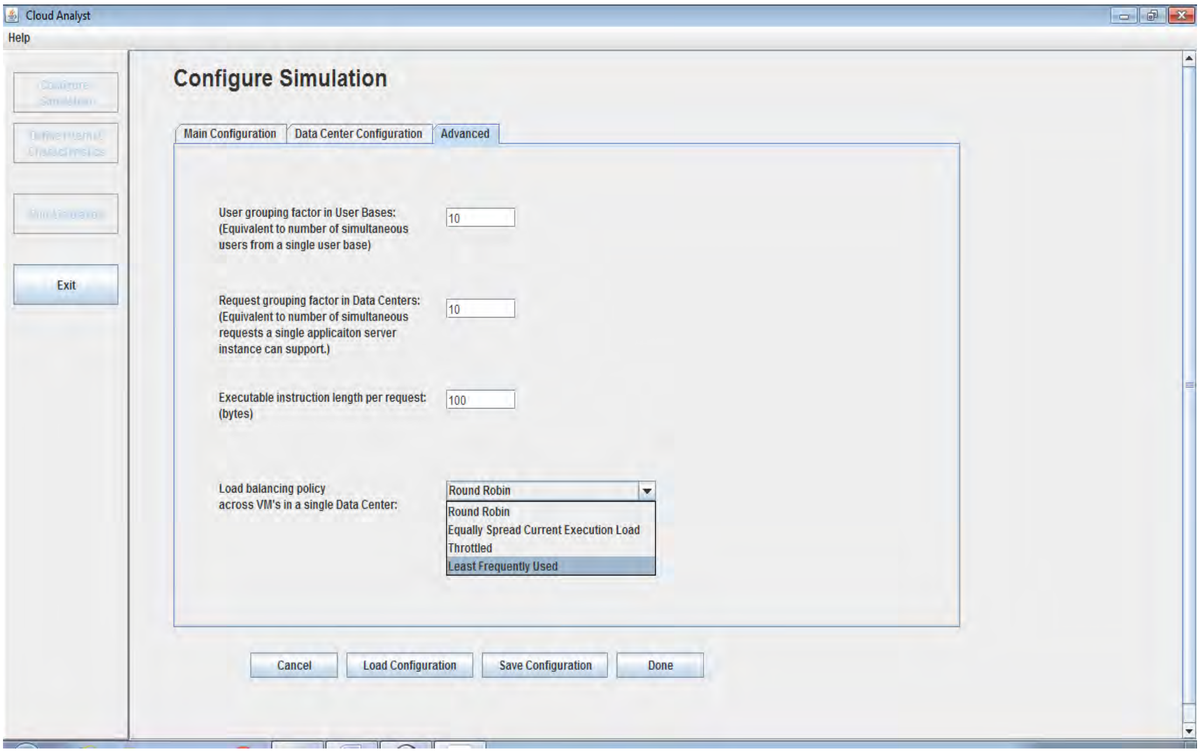


Figure 4: A new option Least Frequently Used is added to the Load balancing policy.

As shown in the above figure the option Least Frequently Used which is newly added is selected. After everything is done the option Run Simulation is selected in the left pane which executes the code including the proposed algorithm. The results are displayed both in graphical format as well as in table format. Figure 5 gives the simulation complete screen which shows the results graphically. Table 2 and Table 3 displays the results which indicates the Response time by region and Data center request servicing times.

Userbase	Avg(ms)	Min(ms)	Max(ms)
UB1	49.97	37.61	60.36
UB2	200.23	172.04	234.11
UB3	300.87	252.12	364.61
UB4	49.93	39.21	57.71
UB5	50.57	42.49	57.98
UB6	50.22	42.65	59.96

Table 2: Average response time by Region where User bases are created.

Data Center	Avg(ms)	Min(ms)	Max(ms)
DC1	0.28	0.02	0.85
DC2	0.55	0.11	0.96
DC3	0.60	0.04	0.97
DC4	0.55	0.15	0.95

Table 3: Results indicating the request servicing times at each data center.

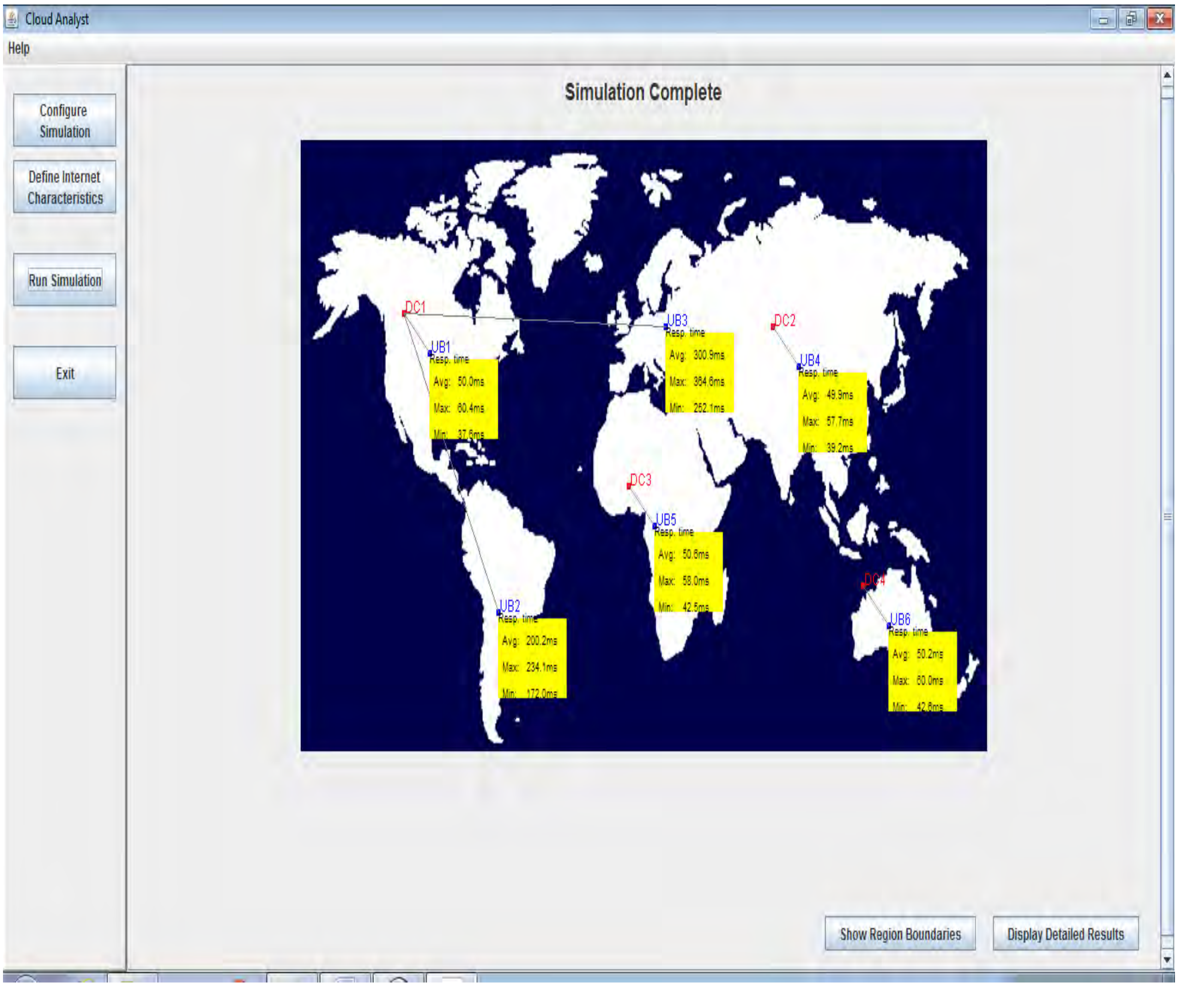
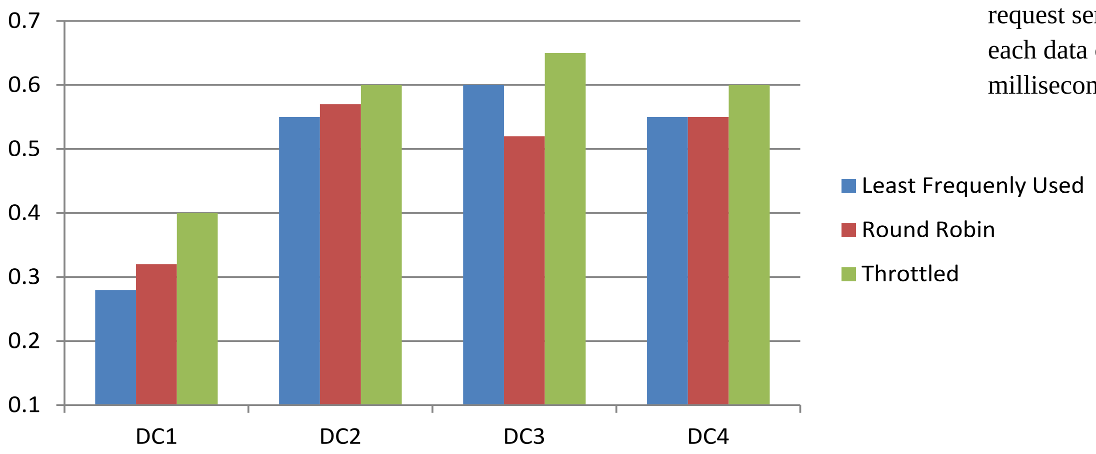


Figure 5: Graphical Representation of the response times at each user base created in 6 regions.

IV. PERFORMANCE ANALYSIS

The performance of the proposed algorithm is compared with the algorithms which already exist in the simulation tool and an analysis is drawn as show below in figure 6.All the data regarding data center configurations as well as user bases are taken same for all the algorithms and results are obtained. This can be achieved easily by selecting the Load Configuration option in the simulation tool where already entered data is just loaded again for other simulations. The X-axis indicates the Data centers and Y-axis indicates the average request servicing time at each data center in milliseconds. A load balancing is considered to be efficient when the processing of requests at each data center takes less time. A study shows that proposed algorithm least frequently used has a minimum average request servicing time when compared to already existing policies like round robin and throttled. But when data center DC3 is considered round robin performs well when compared to least frequently used policy. So on an average proposed algorithm would yield better results but sometimes it might be less effective when compared to other algorithms.



X-axis indicates Data Centers.
Y-axis indicates average request servicing times at each data center in milliseconds.

Figure 6: Comparison of load balancing policies performed at each data center.

V. CONCLUSION AND FUTURE ENHANCEMENTS

If the virtualization process is implemented effectively it results in a best cloud computing environment. The process of virtualization is guided by load balancing of virtual machines which mainly concentrates on improving the throughput of each virtual machine. In order to improve the load balancing policies within a data center a new algorithm based on least frequently used is proposed and results show that on an average it performs well when compared to other existing algorithms. The proposed algorithm can be combined with other features such as least recently used and priority based load balancing so that the results might be still more effective and it might result in better performance than other algorithms all the time.

REFERENCES

[1] http://www.cloudbus.org/students/MEDC_Project_Report_Bhathiya_318282.pdf
[2] Jianhua Gu,Guofei Sun,Tianhai Zhao."A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment “Parallel Architectures, Algorithms and Programming (PAAP), 2010 Third International Symposium on cloud. 18-20 Dec 2010.
[3] Randles,M.,Lamb,D.,Taleb-Bendiab,A."A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing" proceedings of Advanced Information Networking and Applications Workshops (WAINA), 2010.
[4] Gaochao Xu, Junjie Pang, and Xiaodong Fu."A Load Balancing Model Based on Cloud Partitioning for the Public Cloud”. proceedings of Advanced Information Networking and Applications Workshops (WAINA), 2010.
[5] C. L. Dumitrescu and I. Foster. GangSim: a simulator for grid scheduling studies. Proceedings of the IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2005), Cardiff, UK, 2005.
[6] A. Legrand, L. Marchal, and H. Casanova. Scheduling distributed applications: the SimGrid simulation framework. Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2003.
[7] W. Bell, D. Cameron, L. Capozza, P. Millar, K. Stockinger, F. Zini. Simulation of dynamic Grid replication strategies in OptorSim. Proceedings of the 3rd International Workshop on Grid Computing (Grid 2002), Baltimore, U.S.A. IEEE CS Press: Los Alamitos, CA, U.S.A., November 18, 2002
[8] R. Buyya and M. Murshed. GridSim: A Toolkit for the Modelling and Simulation of Distributed Resource Management and Scheduling for Grid Computing. Concurrency and Computation: Practice and Experience, 14(13-15), Wiley Press, Nov.-Dec., 2002.
[9] Rodrigo N. Calheiros,Rajiv Ranjan, Anton Beloglazov,César A. F. De Rose and Rajkumar Buyya "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms." Proceedings of Software: Practice and Experience (SPE), Volume 41, Number 1, Pages: 23-50, ISSN: 0038-0644, Wiley Press, New York, USA, January, 2011.
[10] Bhathiya Wickremasinghe ,Roderigo N. Calherios “Cloud Analyst: A Cloud-Sim-Based Visual Modeler For Analyzing Cloud Computing Environments And Applications”. Proc Of IEEE International Conference On Advance Information Networking And Applications, 2010

Authors Profile:

B.Santhosh Kumar did his B.Tech from Rajeev Gandhi Memorial College Of Engineering and Technology and M.S from Western Kentucky University, USA. He is currently working as an assistant professor in CSE department in G.Pulla Reddy Engineering College, Kurnool. His areas of interests include Cloud Computing, Cryptography and Web Technologies. He is currently pursuing his Ph.D. from Pondicherry University. He has published a paper in 1 International Journal.

Dr.Latha Parthiban has obtained her B.E in Electronics and Communication Engineering from University of Madras, M.E. in Computer Science and Engineering from Anna University, Chennai and has obtained her PhD from Pondicherry University. Her experience spans over 19 years in various Engineering colleges and her research interest includes Data Mining, Image Processing and Cloud Computing. She has published papers in 36 international journals and presented papers in 43 international and national conferences. She has also published a book in the area of computer aided diagnosis.