International Journal of Latest Research in Engineering and Technology (IJLRET)
ISSN: 2454-5031
www.ijlret.com || Volume 02 - Issue 12 || December 2016 || PP. 69-75

A new efficient Virtual Machine load balancing Algorithm for a cloud computing environment

Miss. Rajeshwari Nema
MTECH Student Department of Computer Science and Engineering,
BTIRT, Sagar

Mrs. Sahana T. Edwin
Prof and Head Department of Computer Science and Engineering,
BTIRT, Sagar

Abstract: The performances of computational resources are depending on several concepts, one of which is load balancing concept. The load balancing concept is totally dependent on the how much amount of work allotted to the node for a specific time period of time. This is the time where node has to manage and work according to the given priority. There are several load balancing algorithms are available for the improvement and optimization of cloud performances. The nature of the load balancing algorithm can be static or dynamic, although some algorithms are simple but under the some factor they work more effectively. Cloud computing is a service oriented architecture which is provided via internet. The objective of the service provider is to provide maximum available resources and this can be achieved by implementing load balancing algorithm which helps in gaining load balancing objective. In this paper we proposed an algorithm which is experimented on a Cloud Analyst environment.

I. Introduction:

Cloud Computing can be considered as a platform for development, maintenance and accessing applications by user by paying for the resources which are only used for certain time. To serve for huge number of requests from different types of users located at different parts of the world on a pay-per-usage bases, the process of virtualization has been followed in cloud computing environment. There are so many service providers who are responsible for maintaining the application on cloud environment. The most predominant cloud service providers are Google, Amazon, Microsoft and many others. In order to serve the huge traffic to world 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, no. of virtual machines are created where on a single CPU different operating systems and their configurations can be run. It gives the illusion to the users that, there are numbers of CPUs are involved for processing the requests. This whole process can be considered as virtualization and it is monitored by the software called as hypervisor. The hypervisor is mainly responsible in creating and maintaining the virtual machines. The requests from all the users are considered as individual request and each of the virtual machines is assigned a request and effort is made to keep them busy for longer time. 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 work on personal computers a tool, Cloud Analyst has been come into existence by the CLOUDS laboratory in the University of Melbourne. The main intention for developing this tool is to simulate the traffic generated by most visited 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. Elaborates each and every detail of the tool.

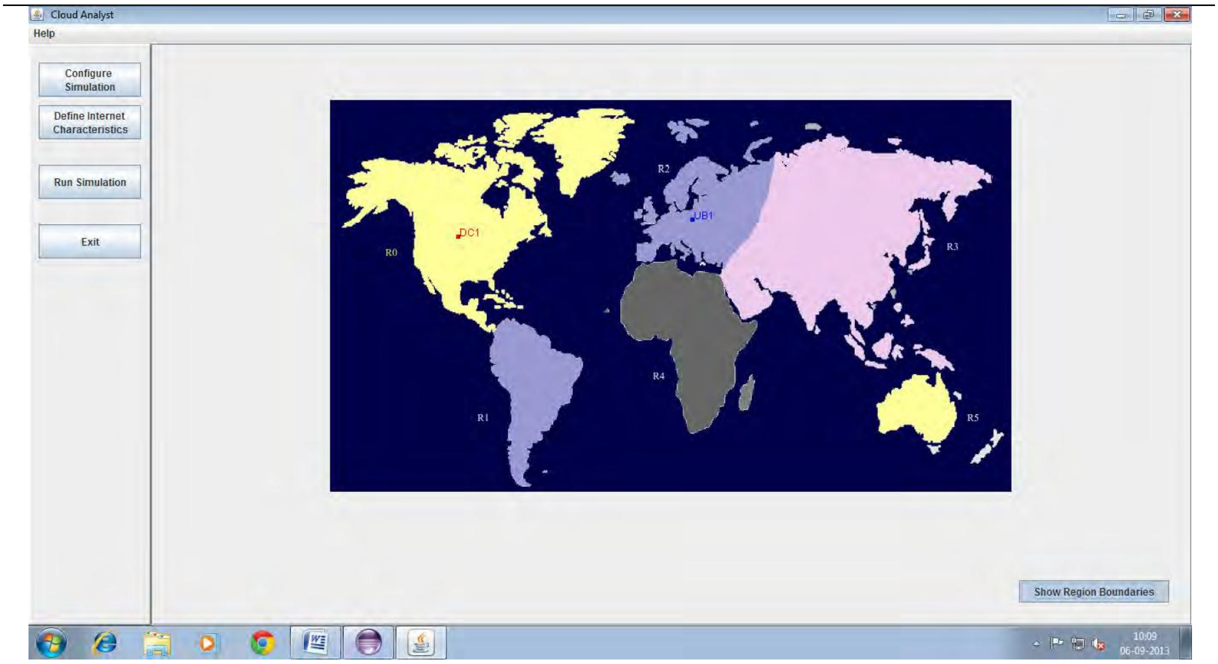


Figure 1: Screen shot illustrating the region boundaries as well as the home screen.

Region	**Cloud Analyst Region Id**
North America	R0
South America	R1
Europe	R2
Asia	R3
Africa	R4

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

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

II. Related Work :

A considerable related amount of work has been framing in the load balancing policies in a cloud environment. It Gives an idea on load balancing policy virtual machines so that they can be efficiently used on data center It show a comparative study of different load balancing algorithms based on distributed load balancing in a cloud environment. Emphasizes on a load balancing policy that is based on game theory and cloud partitioning. Beginning from the era of grid computing numerous tools have been developed to work on the cloud load balancing algorithms. In order to do work related to grid computing gives an understanding of a existing tool known called Gang Sims, Sims Grid simulation framework introduced in supported the scheduling policies in distributed cloud applications. Simulation of dynamic grid replication policies have been demonstrated in using a tool OptorSim. Focuses on toolkit for modeling and simulation of grid computing known as Grasim.

The rapid advancement in technology lead to the maximizing cost for developing and maintaining the cloud application. In order to diminish the cost and improve the maintenance of applications as-per-pay usage based concept has been introduced which ultimately lead to the development of cloud computing environment. This lead to the development of many cloud tools which simulate the cloud applications. The most leading one among them is the CloudSim [9] which was developed for both modeling and simulation of cloud computing application. Based on this tool a lot of research has been done and useful for developing many tools useful for cloud simulation.

Cloud Analyst can be considered as the most recent and effective tool which is based on Cloud Sim.This gave a visual idea of simulating the massive traffic associated with some of the rich online applications such as face book and Gmail. A cloudlet which is considered to be a mini cloud is developed and submitted to different data centers of the cloud service providers across the world. This simulator allows the user to specify the number of virtual machines which is created in each data centre. Depending on the requests from each user base the traffic is divided into different data centers. The data centers creates the virtual machine instances and submit the requests according to load balancing policies such as round robin, throttle requests. The main components or classes used to develop cloud analyst can be considered as Data Centre Broker, VmScheduler, Cloudlet and many more.

III. Load Balancing:

Load Balancing is a Concept to distribute the load among the multiple computers or a computer clusters through internet to achieve optimal resource utilization by maximizing throughput and minimizing overall response time. Load Balancing is used for avoiding too much load on the resources and dividing the traffic between the servers and data centers . Data can be sent and received without delay. Load Balancing is used for minimizing the total waiting time of the resources. In cloud computing, load balancing are uses for balancing the load on virtual machine and cloud centers.

IV. Existing Scheduling Algorithm in Cloud Computing:

Virtual machine enables the abstraction of an Operating System and software running on it from hardware. The interior hardware services is interrelated to the Clouds which is modeled in the Cloudsim simulator by a Data centre for handling service requests. These request are application within VM’s which need to be allocated to share of processing power on Data centre’s , components. Data centres object manages the data centre management activities such as VM creation and destruction and does the routing of user request received from user base to the VM’s. The data centre controller uses a VmLoadBalancer to determine which virtual machine should be assigned to the next request of VM for processing. There are three types of VmLoadBalancer that is Round Robin, Throttled and active monitoring load balancing algorithms.

A. Round Robin Load Balancer- It is one of the simplest scheduling algorithm that work on the principle of time slices, it is small amount of time. Here the time is divided into multiple slices and each node is assigned for a particular time slice or time interval i.e. it utilizes the principle of time scheduling. Each node is given a quantum and its operation. The resources of the service provider are provided to the requesting client on the basis of time slice.

B. Throttled Load Balancer (TLB)- This algorithm are work on the concept of that number of cloudlets are allocated to a single Virtual Machine at any given time. If there are more request groups are present as compare to the number of available Virtual Machine at data centre, allocates the incoming request in a queue until the next VM becomes available.

C. Active Monitoring Load Balancer (AMLB)- The Active Monitoring Load Balancer algorithm maintains request on each VM’s and maintain the number of request currently allocated to which Virtual Machine, when a request is allocate to a new VM, first identified is selected by AMLB and returns the Virtual Machine id to the data centres controller. The data centres controller send the request to the Virtual Machine identified by its id. The data centre controller notifies to AMLB to new allocated request and cloudlets is sent to it.

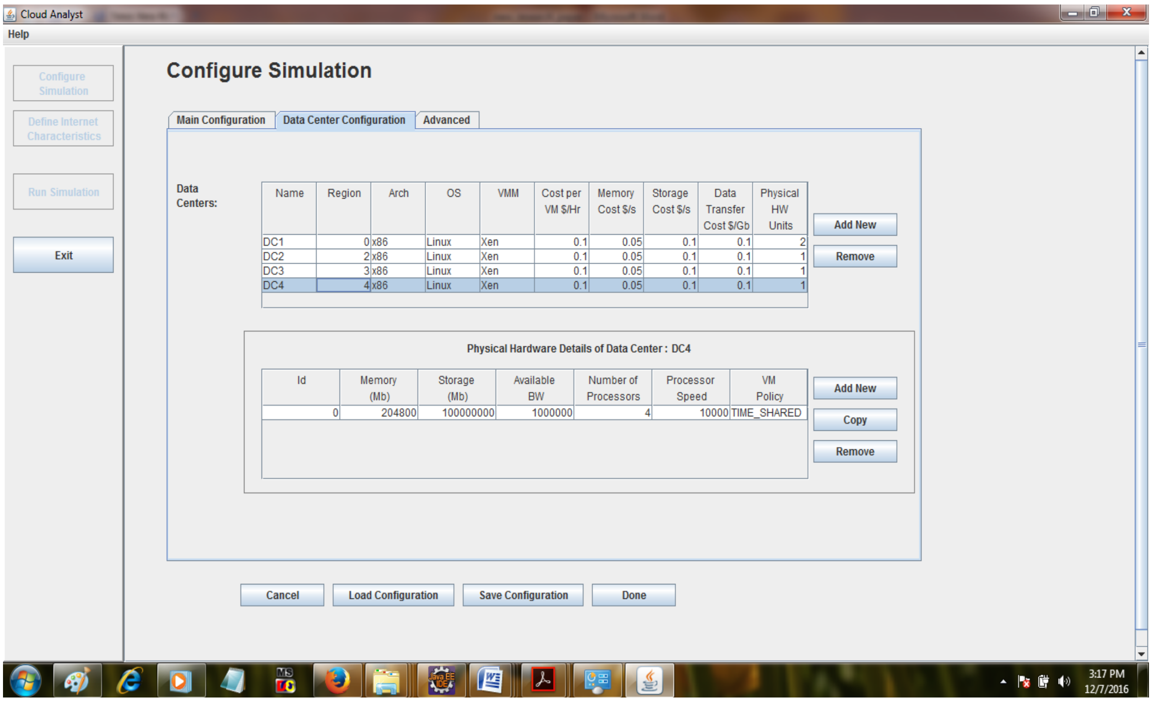
V. The Proposed Algorithm- Enhance Round Robin VM Load Balancing:

- The proposed algorithm is an improvement over the Round Robin Load Balancing algorithm. The Round Robin algorithm does not save the state of previous Virtual machine allocation to a request from a given user base while the same state is saved in ERR VM load balancer. The Enhance Round Robin VM Load balancer maintain two data structure which is discussed below. Hash Map- it stores the entry for the last VM allocated to a request from a given requested user base. VM State List- stores the allocation busy or available status of each VM.

VI. Algorithem:

```
Enhance Round_Robin_Load_Balancing ()
 {
 Initialize all the VM allocation status to as AVAILABLE in the data center VM state list;
Initialize hash map with zero entries;
While(new request are received by the Data Centre Controller)
Do
{
    Data Center Controller queue the requests;
    Data Centre Controller removes a request from the beginning of the queue;
    If(hash map contain any entry of a VM corresponding to the current requesting user base && VM allocation status == AVAILABLE)
    {
        The VM is reallocated to the user base request;
    }
    Else
    {
        Allocate a VM to the user base request using Round Robin Algorithm;
        Update the entry into the user base and the hash map and the VM state list into the VM;
    }
}
}
```

In addition to the load balancing policies existing in cloud analyst such as round robin, we proposed a new algorithm which is based on the existing round robin mechanism is proposed as shown above. The algorithm depicted above is added as a part of cloud analyst tool in a file called vmEnhanceRoundRobin 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.



Configure Simulation

Data Centers:

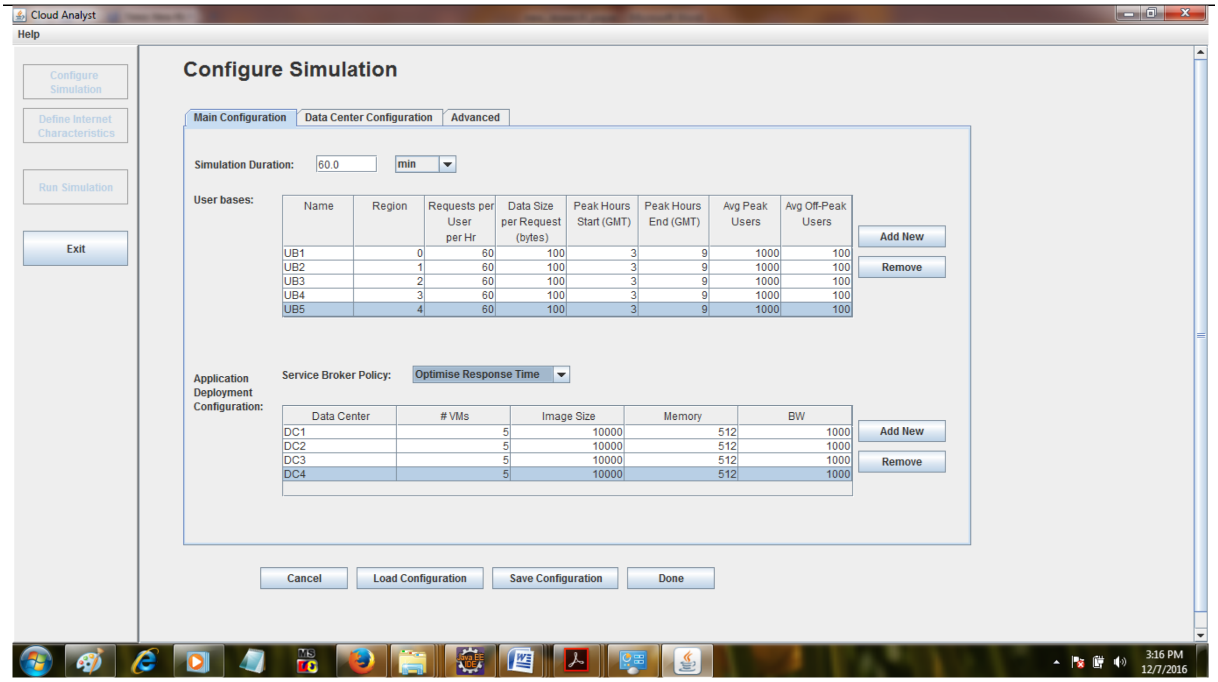
Name	Region	Arch	OS	VMM	Cost per VM $/Hr	Memory Cost $/s	Storage Cost $/s	Data Transfer Cost $/Gb	Physical HW Units
DC1	0	x86	Linux	Xen	0.1	0.05	0.1	0.1	2
DC2	2	x86	Linux	Xen	0.1	0.05	0.1	0.1	1
DC3	3	x86	Linux	Xen	0.1	0.05	0.1	0.1	1
DC4	4	x86	Linux	Xen	0.1	0.05	0.1	0.1	1

Physical Hardware Details of Data Center : DC4

Id	Memory (Mb)	Storage (Mb)	Available BW	Number of Processors	Processor Speed	VM Policy
0	204800	100000000	1000000	4	10000	TIME_SHARED

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

After Selected the main configuration go to the Data Center Configuration and add the data center based on the user base. Further the another screen is designed to add the number of virtual machines in each data center and the memory , available bandwidth allocated to each virtual machine can be specified.
Figure 3 gives a snapshot of how the whole work is done.



Configure Simulation

Simulation Duration: 60.0 min

User bases:

Name	Region	Requests per User per Hr	Data Size per Request (bytes)	Peak Hours Start (GMT)	Peak Hours End (GMT)	Avg Peak Users	Avg Off-Peak Users
UB1	0	60	100	3	9	1000	100
UB2	1	60	100	3	9	1000	100
UB3	2	60	100	3	9	1000	100
UB4	3	60	100	3	9	1000	100
UB5	4	60	100	3	9	1000	100

Application Deployment Configuration:

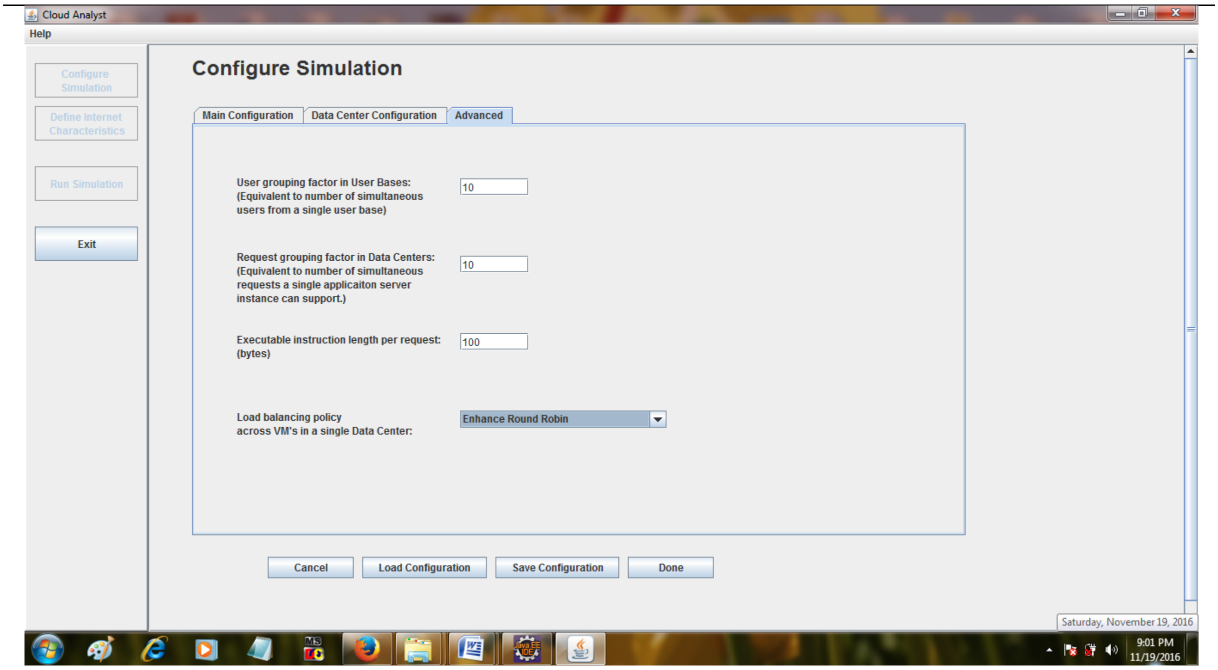
Service Broker Policy: Optimise Response Time

Data Center	# VMs	Image Size	Memory	BW
DC1	5	10000	512	1000
DC2	5	10000	512	1000
DC3	5	10000	512	1000
DC4	5	10000	512	1000

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

The user bases UB1, UB2, UB3, UB4, UB5 are added respectively with each of them in different regions and the requests is specified per user per hour as 60. There are the broker policy by which the data center is selected for processing. This simulation tool provides three different broker policies closest data center, optimize response time and Reconfigure dynamically with load. In the present scenario optimum response time 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 5 and their memory and bandwidth are specified accordingly.

After creating the data centers and user bases is select the advanced tab on configure simulation screen. It provide an option to select the load balancing policy for virtual machines created in a single data center. The algorithms which are already available 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 for a time slice. Throttled based policy allocates the tasks to each virtual machine until its full capacity and than shifts task 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 vmEnhanceRoundRobin is executed. Figure 4 gives the scenario where user is given the option to select Enhance Round Robin load balancing policy. The code for the proposed algorithm is added to a file by name vmEnhanceRoundRobin which extends the abstract class vmLoadBalancer.



Configure Simulation

User grouping factor in User Bases: (Equivalent to number of simultaneous users from a single user base) 10

Request grouping factor in Data Centers: (Equivalent to number of simultaneous requests a single applicaiton server instance can support.) 10

Executable instruction length per request: (bytes) 100

Load balancing policy across VM's in a single Data Center: Enhance Round Robin

Figure 4: A new option Enhance Round Robin is added to the Load balancing policy.

As shown in the above figure the option Enhance Round Robin Used which is newly added is selected. After everything is done click on option Run Simulation is selected in the left pane which executes the code including with the proposed algorithm. The results are come 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.

Data Center	**Avg (ms)**	**Min (ms)**	**Max (ms)**
DC1	0.28	0.02	0.86
DC2	0.40	0.02	0.88
DC3	0.47	0.02	0.88
DC4	0.48	0.03	0.90

Table 2: Average Request Servicing Times by New Enhance Round Robin Policy

Data Center	**Avg (ms)**	**Min (ms)**	**Max (ms)**
DC1	0.35	0.02	0.89
DC2	0.43	0.02	0.88
DC3	0.51	0.02	0.88
DC4	0.41	0.02	0.88

Table 3: Average Request Servicing Times by Existing Round Robin Policy

VII. Performance Analysis:

The performance of the proposed algorithm is compared with the Existing Round Robin 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 Enhance Round Robin used has a minimum average request servicing time when compared to already existing policies like round robin. But when data center DC3 is considered round robin performs well when compared to Enhance Round Robin used policy. So on an average proposed algorithm would yield better results but sometimes it might be less effective when compared to other algorithms.

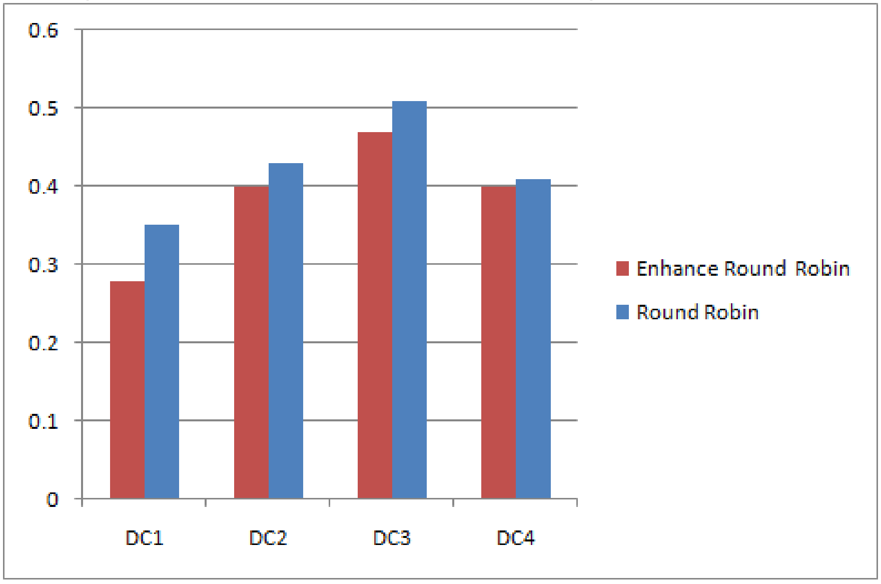


Fig. 5 Comparison of Both Load Balancing Policy performed at each data center

VIII. Conclusion and enhancement :

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 Enhance Round Robin is proposed and results shown that on an average Request Service Time is well when compared to other existing algorithms. The proposed algorithm can produce better result in term of Request Service Time, means it fulfill the request as early, that's why it is better as compare to existing round Robin Algorithm.

IX. 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.