ISSN:0975-9646

A Comparative Study on Load Balancing Algorithms with Different Service Broker Policies in Cloud Computing

Sonia Lamba , Dharmendra Kumar

United College of Engineering and Research,Allahabad, U.P, India.

Abstract- Cloud computing is emerging as a new paradigm of large scale distributed computing. It enables a wide range of users to access scalable, virtualized hardware, distributed and/or software infrastructure over the Internet. One of the challenging scheduling problems in Cloud datacenters is to take the allocation and migration of reconfigurable virtual machines into consideration as well as the integrated features of hosting physical machines. In order to select the virtual nodes for executing the task, Load Balancing is used. It is the process of distributing workload across multiple computers, or other resources over the network links to achieve optimal resource utilization, minimum data processing time, minimum average response time, and to avoid overload. Load Balancing ensures that all the processors in the system as well as in the network do approximately the equal amount of work at any instant of time. To simulate large scale applications Cloud-Analyst, the Cloud-Sim based tool is used. The simulator uses different Service Broker Algorithms, Load Balancing Algorithms etc. with taking some parameters under consideration as per requirement. The objective of this paper is to analyze the performance of existing load balancing algorithms with different service broker policies.

Keywords: Cloud Computing, Load Balancing, Service Broker, Cloud Analyst.

INTRODUCTION

Cloud Computing, a framework for enabling convenient, and on-demand network access to a shared pool of computing resources [1], is emerging as a new paradigm of large scale distributed computing [2]. It has moved computing and data away from desktop and portable PC's into large datacenters. It has the capability to harness the power of Internet and Wide Area Network (WAN) to use resources that are available remotely, thereby providing cost effective solution to most of the real time requirements [3][6]. It has widely been adopted by the industry, though there are many existing issues like Load Balancing, Virtual Machine Migration, Server Consolidation, Energy Management etc that are fully addressed [4][5]. Central to these issues is the issue of Load Balancing. Load Balancing is a methodology to distribute the workload across multiple computers, or other resources over the network links to achieve optimal resource utilization, minimum data processing time, minimum average response time and to avoid overload. Load Balancing ensures that all the processors in the system as well as in the network do approximately the equal amount of work at any instant of time.

Cloud computing thus involving distributed technologies to satisfy a variety of applications and user needs. Sharing resources, software, information via internet are the main functions of cloud computing with an objective to reduced capital and operational cost, better performance in terms of response time and data processing time, maintain the system stability and to accommodate future modification in the system. So there are various technical challenges that need to be addressed like Virtual machine migration, Server consolidation, fault tolerance and high availability and scalability but central issue is the load balancing. It avoids a situation where some of the nodes are heavily loaded while other nodes are idle or doing little work. It also ensures that all the processors in the system or every node in the network does approximately the equal amount of work at any instant of time [7][8]. It helps in preventing bottlenecks of the system which may occur due to load imbalance. When one or more components at any service fail, load balancing facilitates continuation of the service by implementing fair over i.e. it helps in provisioning and de-provisioning of instances of applications without fail. It also ensures that every computing resource is distributed efficiently and fairly.

Load Balancing serves two important needs, primarily to promote availability of cloud resources and secondarily to promote performance. In order to balance the requests of the resources it is important to recognize a few major goals of Load Balancing algorithms:

a.) ***Cost Effective***- Primary aim is to achieve an overall improvement in system performance at a reasonable cost.

b.) ***Scalability and Flexibility***- The distributed system in which the algorithm is implemented may change in size or topology. So the algorithm must be scalable and flexible enough to allow such changes to be handled easily.

c.) ***Priority***- Prioritization of the resources or jobs need to be done on before hand through the algorithm itself for better services to the important or high prioritized jobs in spite of equal service provision for all the jobs regardless of their provision.

NEED OF LOAD BALANCING IN CLOUD COMPUTING

Load Balancing in clouds is a mechanism that distributes the excess dynamic local workload evenly across all the nodes. It is used to achieve a high user satisfaction and resource utilization ratio [9], making sure that no single node is overwhelmed, hence improving the overall

performance of the system. Proper load balancing can help in utilizing the available resources optimally, thereby minimizing the resource consumption. It also helps in implementing fail-over, enabling scalability, avoiding bottlenecks and over-provisioning, reducing response time etc.

Apart from above-mentioned factors, load balancing is also required to achieve Green Computing in clouds which can be done with the help of the following two factors:

Reducing Energy Consumption - Load balancing helps in avoiding overheating by balancing the workload across all the nodes of a cloud, hence reducing the amount of energy consumed.

Reducing Carbon Emission – Energy Consumption and carbon emission go hand in hand. The more the energy consumed, higher is the carbon footprint. As the energy consumption is reduced with the help of load balancing, so is the carbon emission helping in achieving Green Computing.

EXISTING LOAD BALANCING ALGORITHMS FOR CLOUD COMPUTING

The three existing algorithm to distribute the workload across multiple nodes over the network links to achieve optimal resource utilization, minimum data processing time, minimum average response time, and to avoid overload are:

Round Robin Algorithm (RR):

It is the simplest algorithm that uses the concept of time quantum or slices. Here, time is divided into multiple slices and each node is given a particular time quantum and within this time quantum the node will perform its operations. In this algorithm, the DataCenterController assign the request to a list of VM's on a rotating basis. The first request is allocated to a VM picked randomly from the group and then DataCenterController assigns the subsequent requests in a circular order. Once the virtual machine is assigned the request, the VM is moved to the end of the list.

In this RRLB, there is a better allocation concept known as *Weighted Round Robin Allocation* in which one can assign a weight to each VM so that if one VM is capable of handling twice as much load as the other, the powerful server gets a weight of 2. In such cases, DataCenterController will assign the two requests to the powerful VM for each request assigned to a weaker one. Round Robin Algorithm selects the load on random basis, and therefore leads to a situation where some nodes are heavily loaded and some are lightly loaded. Though, the algorithm is very simple but there is an additional load on the scheduler to decide the size of quantum [5]. It has longer average waiting time, higher context switches, higher turnaround time and low throughput.

Equally Spread Current Execution Algorithm (ESCE):

1. In this algorithm, the Load Balancer maintains an index table of VM's and the number of requests currently allocated to the VM's. At start all VM's have 0 allocations.
2. When a request to allocate a new VM from the DataCenterController arrives, it parses the index table and identifies the least loaded VM. If there are more than one, the first identified is selected.
3. The Load Balancer returns the VM ID to the DataCenterController.
4. The DataCenterController sends the request to the VM identified by that ID.
5. The DataCenterController notifies the Load Balancer of the new allocation.
6. The Load Balancer updates the allocation table incrementing the allocation count for that VM.
7. When the VM finishes processing the request and DataCenetrController receives the response cloudlet, it notifies the Load Balancer of the VM de-allocation.
8. The Load Balancer updates the allocation table by decrementing the allocation count for the VM one by one.

In Equally Spread Current Execution Algorithm, a communication exist between the load Balancer and the DataCenterController for updating the index table leading to an overhead. Further, this overhead causes delay in providing response to the arrived requests.

Throttled Load Balancing Algorithm (TLB):

1. In this algorithm, the Load Balancer maintains an index table of VM's as well as their states (Available/Busy).
2. When a request to allocate a new VM from the DataCenterController arrives, it parses the index table from top until the first available VM is found.
3. If VM is found, the Load Balancer returns the VM ID to the DataCenterController.
4. The DataCenterController send the request to the VM identified by that ID.
5. The DataCenterController notifies the Load Balancer of the new allocation.
6. The Load Balancer updates the allocation table by incrementing accordingly.
7. When the VM finishes processing the request and the DataCenterController receives the response cloudlet, it notifies the Load Balancer of the VM de-allocation.
8. The Load Balancer de-allocate the same VM whose Id is already communicated.

The purpose of algorithm is to find the expected Response Time of each Virtual Machine because Virtual Machines are of heterogeneous capacity with regard to its processing performance, the expected response time can be found with the help of the following formulas:

Response Time = $Fin_t – Arr_t + TDelay$ (1)

Where, Arr_t is the arrival time of user request and Fin_t is the finish time of user request and the transmission delay can be determined by using the following formulas:

TDelay = Tlatency + Ttransfer (2)

Where, TDelay is the transmission delay, Tlatency is the network latency and Ttransfer is the time taken to transfer the size of data of a single request (D) from source location to destination location.

Ttranfer = D/Bwperuser (3)

Bwperuser = Bwtotal/Nr (4)

Where, Bwtotal is the total available bandwidth and Nr is the number of user requests currently in transmission. The Internet Characteristics also keeps track of the number of user requests in-flight between two regions for the value of Nr.

METRICS FOR LOAD BALANCING IN CLOUD

The existing load balancing techniques in clouds, consider various parameters like performance, response time, scalability, throughput, resource utilization, fault tolerance, migration time and associated overhead. But, for an energy efficient load balancing, metrics like energy consumption and carbon emission should also be considered.

Overhead Associated- It determines the amount of overhead involved while implementing a load balancing algorithm. It is composed of overhead due to movement of tasks, inter processor and inter process communication. This should be minimized so that a load balancing technique can work efficiently.

Throughput- It is used to calculate the no. of tasks whose execution has been completed. It should be high to improve the performance of the system.

Performance- It is used to check the efficiency of the system. It has to be improved at a reasonable cost e.g. reduce response time while keeping acceptable delays.

Resource Utilization- It is used to check the utilization of resources. It should be optimized for an efficient load balancing.

Scalability- It is the ability of an algorithm to perform load balancing for a system with any finite number of nodes. This metric should be improved.

Response Time- It is the amount of time taken to respond by a particular load balancing algorithm in a distributed system. This parameter should be minimized.

Fault Tolerance- It is the ability of an algorithm to perform uniform load balancing in spite of arbitrary node or link failure. The load balancing should be a good fault-tolerant technique.

Migration Time- It is the time to migrate jobs or resources from one node to other. It should be minimized in order to enhance the performance of the system.

Energy Consumption- It determines the energy consumption of all the resources in the system. Load balancing helps in avoiding overheating by balancing the workload across all the nodes of a cloud, hence reducing energy consumption.

Carbon Emission- It calculates the carbon emission of all the resources in the system. As energy consumption and carbon emission go hand in hand, the more the energy consumed, higher is the carbon footprint. So, for an energy-efficient load balancing solution, it should be reduced.

CLOUD ANALYST

Cloud Analyst is developed on the top of Cloud-Sim and the Cloud-Sim [13][14] is developed on the top of the Grid-Sim. Some new extensions are introduced in Cloud Analyst:

Application Users

There is the requirement of autonomous entities to act as traffic generators and behavior needs to be configurable.

Internet

It is introduced to model the realistically data transmission across Internet with network delays and bandwidth restrictions.

Simulation defined by time period

In Cloud-Sim, the process takes place based on the pre-defined events. Here, in Cloud Analyst, there is a need to generate events until the set time-period expires.

Service Brokers

DataCenterBroker in Cloud-Sim performs VM management in multiple data centers and routing traffic to appropriate data centers. These two main responsibilities were segregated and assigned to DataCenterController and CloudAppServiceBroker in Cloud Analyst.

GUI and ability to save simulations and results

The user can configure the simulation with high level of details using the GUI. It makes easy to do the simulation experiments and to do it in repeatable manner. Using the GUI introduced here, we can also save the simulation configuration as well as the results in the form of PDF files for future use.

Cloud Analyst is implemented including these features:

REGION

In the Cloud Analyst, 6 Regions are there based on the 6 main continents in the world. To have the relisting simplicity for the large scaled testing in Cloud Analyst [11].

USER BASE

A User Base models a group of users that is considered as a single unit in the simulation and its main responsibility is to generate traffic for the simulation. A single User Base may represent thousands of users but is configured as a single unit and the traffic generated in simultaneous bursts representative of the size of the User Base. The modeler may choose to use a User Base to represent a single user, but ideally a User Base should be used to represent a larger number of users for the efficiency of simulation [11].

VM LOAD BALANCER

VM Load Balancer is useful to determine which VM should be assigned the requests (Cloudlet) for processing. Three policies are included currently in the Cloud Analyst [11].

a.) Round Robin Load Balancer
b.) Active Monitoring Load Balancer
c.) Throttled Load Balancer

INTERNET CLOUDLET

It is a grouping of user requests. The number of requests grouped into a single Internet Cloudlet. This Internet Cloudlet is configurable in Cloud Analyst. The Internet Cloudlet is having information such as the size of a request execution command, size of input and output files, the originator and target application id used for routing by the Internet and the number of requests.

CLOUD APPLICATION SERVICE BROKER

A Service Broker decides which data center should provide the service to the requests coming from each User Base. And thus, Service Broker controls the traffic routing between User Bases and Data Centers.

Currently, Cloud Analyst uses three types of Service Brokers each implementing a different routing policy:

Service Proximity Based Routing

Here, the shortest path to the data center from the user base, depended on the network latency is selected and according to that, the service broker routes the traffic to the closest data center with the consideration of transmission latency.

Performance Optimized Routing

In this routing policy, service broker actively monitors the performance of all data centers, and based on that, directs traffic to the data center with best response time.

Dynamically Reconfiguring Routing

This router has one more responsibility of scaling the application deployment depended on the current load it faces. This policy increases and decreases the number of virtual machines allocated in the data centers. This will be done taking under consideration the current processing times and best processing time ever achieved.

SIMULATION AND RESULTS ANALYSIS

The Cloud Analyst is a GUI based tool which is developed on Cloud-Sim architecture. Cloud-Sim [7][9] is a toolkit used for modeling, experimentation and simulation. The deployment of large scale applications is quite economical and easy by using clouds. The cloud also generates the new issues for developers. The various users access the Internet applications around the world, and because, popularity of applications may vary along the world, so experience in the use of application can also vary.

In order to analyze various load balancing policies configuration of the various components of the cloud analyst tool need to be set. We have set the parameters for the user base configuration, data center configuration, and advanced configuration as shown in figure 1, figure 2 and figure 3 respectively. The output screen of Cloud Analyst is shown in figure 4. The location of user bases has been defined in six different regions of the world. We have taken four datacenters to handle the request of these users. One datacenter is located in region 0, second in region 1, and third in region 2 and fourth in region 3. On DC1, DC2, DC3 and DC4 number of Virtual Machines allocated are 50. The duration of simulation is 60 hrs.

Cloud Analyst enables the modeler to execute the simulation repeatedly with the modifications to the parameters quickly and easily. The graphical output of the simulation results can be analyzed more easily and efficiently. After performing the simulation the result computed by Cloud Analyst for Round Robin Load Balancing algorithm with Closest Data Center Service Broker policy is shown figure 5, 6 and 7. We have used the above defined configuration for each load balancing policy one by one with three different service broker policies and depending on that the result calculated for the metrics like response time, request processing time and cost in fulfilling the request has been shown in Tables.

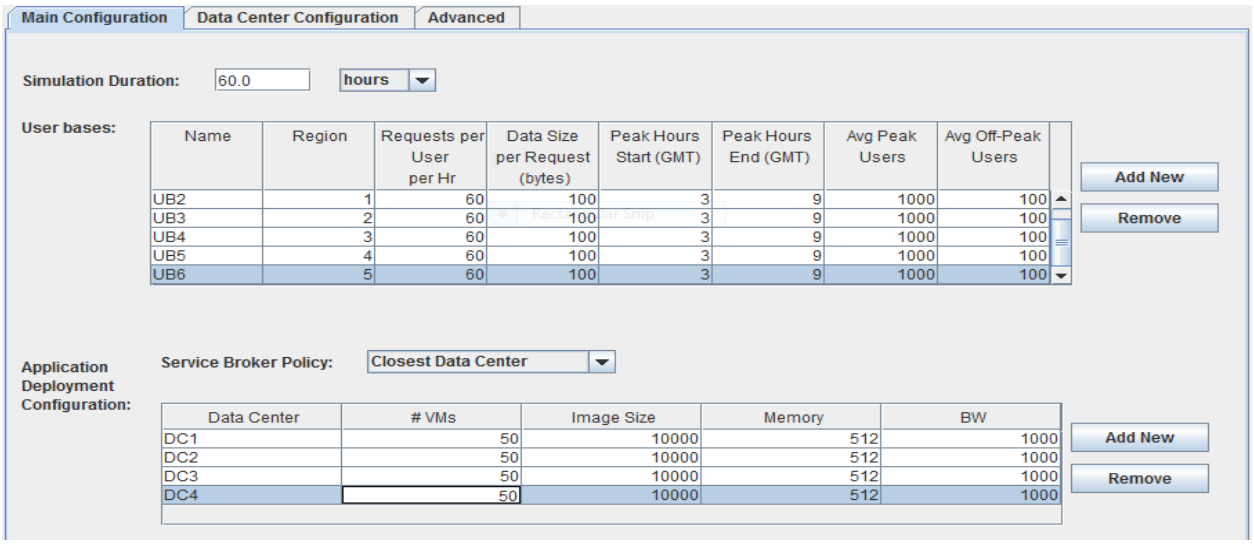

Figure 1. Main configuration Screen in Simulator

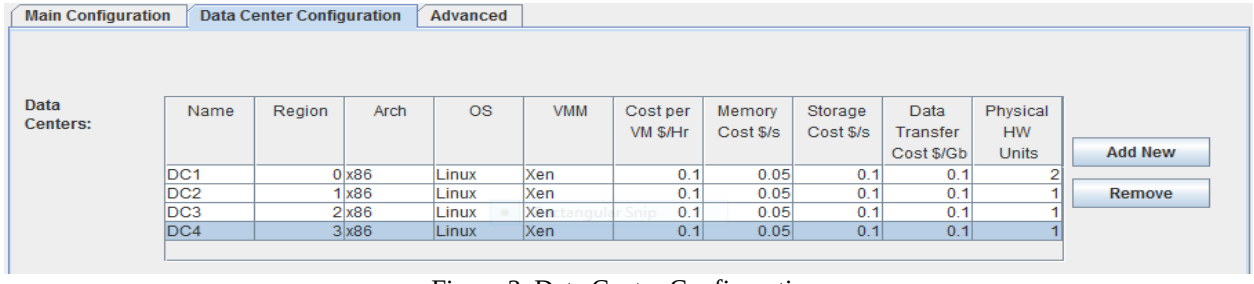

Figure 2. Data Center Configuration

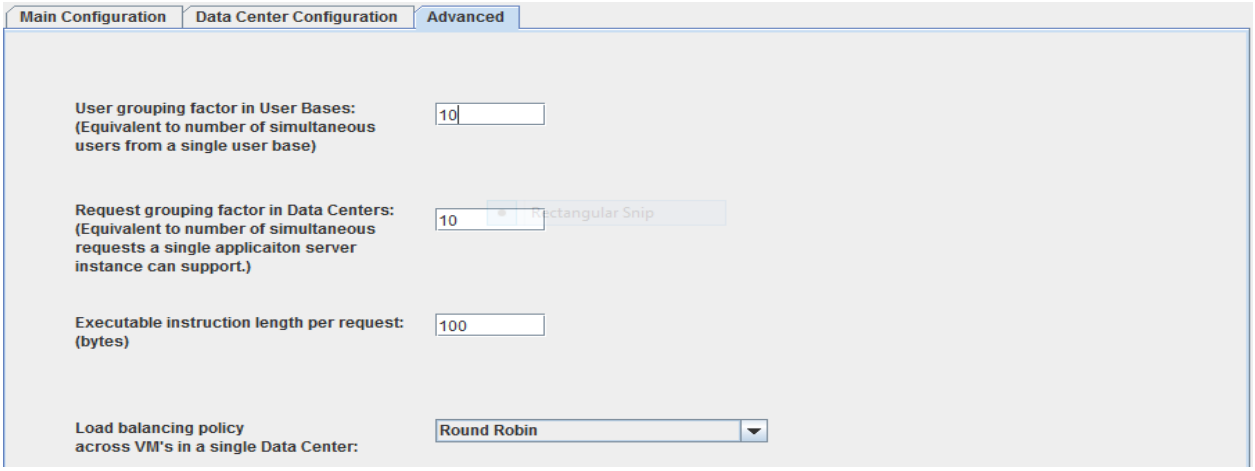

Figure 3. Advanced Configuration

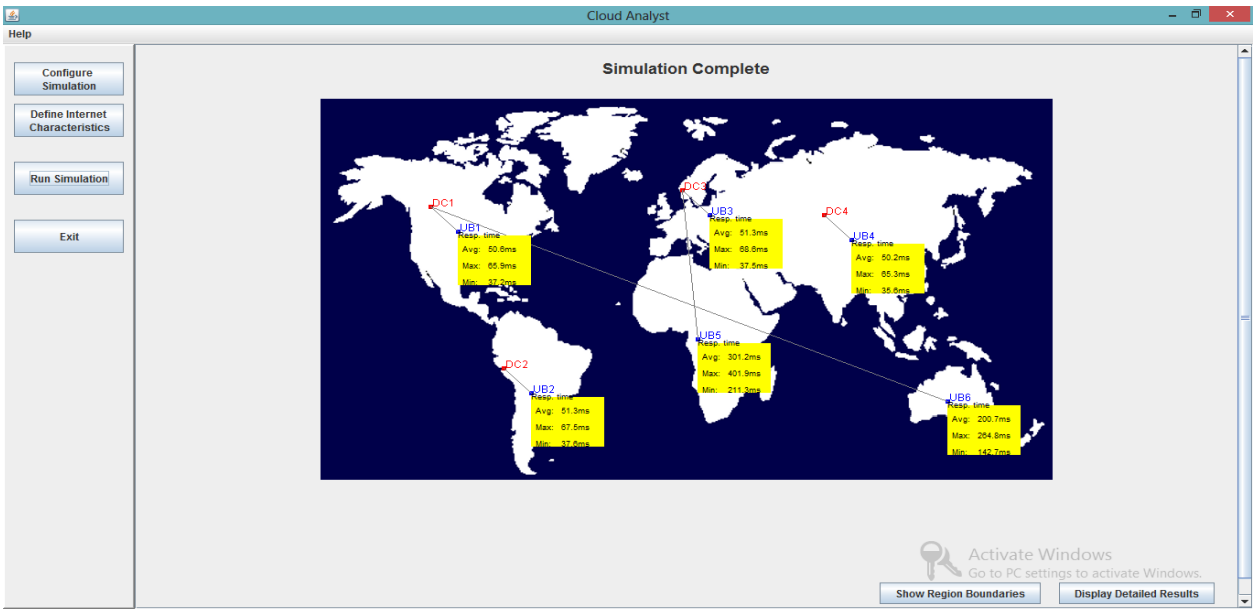

Figure 4. Output Screen of Cloud Analyst

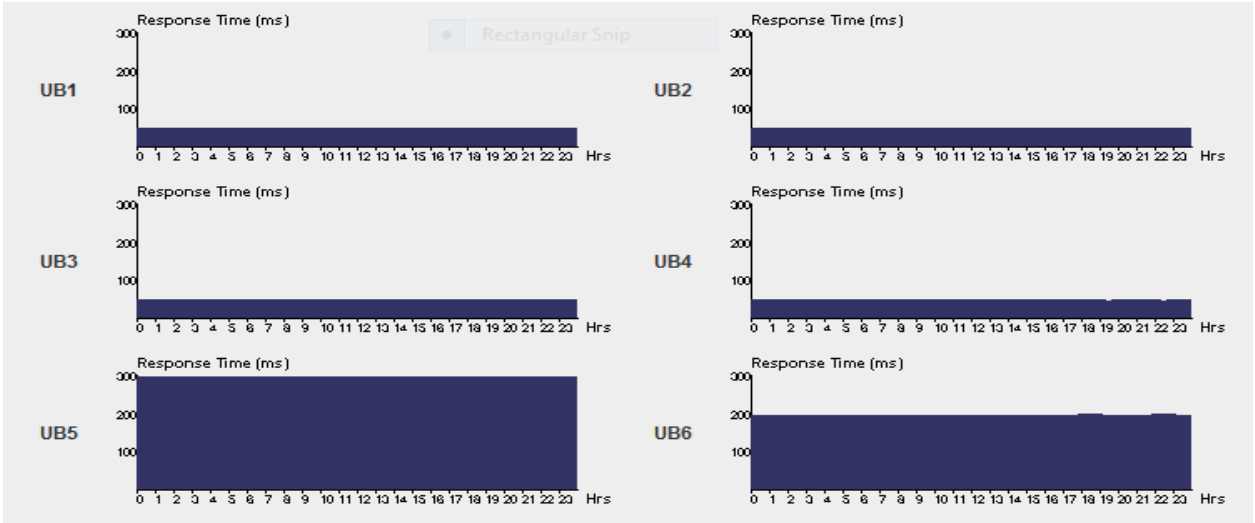

Figure 5. User Hourly Response Time

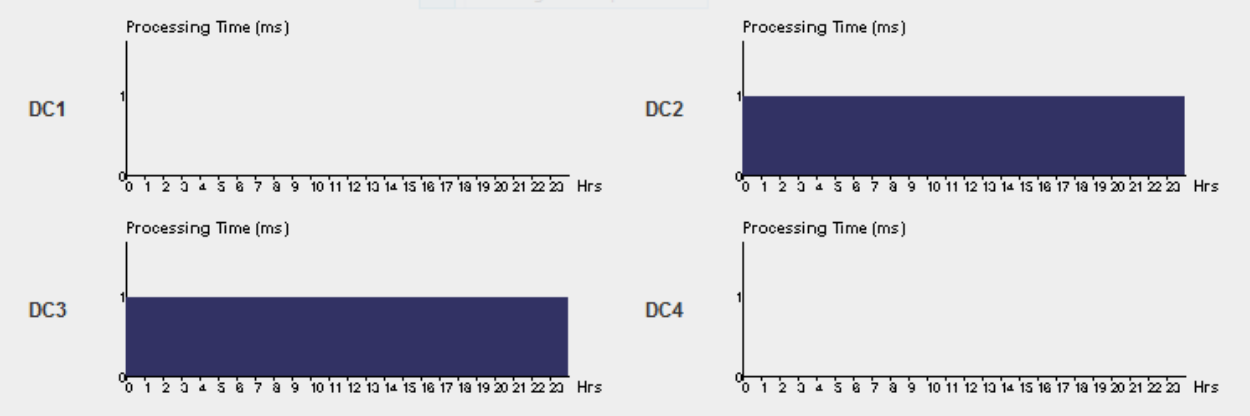

Figure 6. Data Processing Times

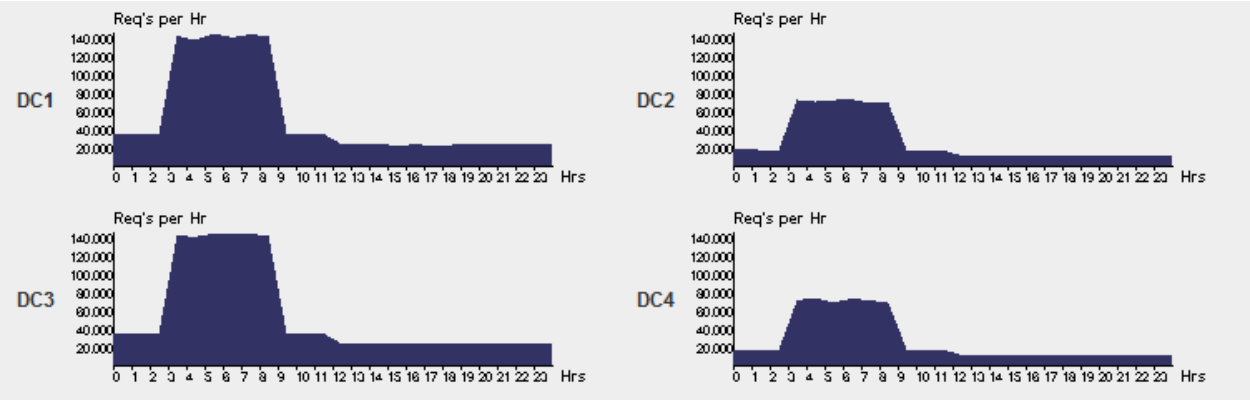

Figure 7. Data Center Loading Times

Table 1. Overall Response Time Summary

Load Balancing Algorithms With Different Service Broker Policies	Overall Response Time		
	Avg(ms)	Min(ms)	Max(ms)
Round Robin With Closest Data Center	117.71	35.63	401.88
Round Robin With Optimize Response Time	117.73	36.01	555.63
Round Robin With Reconfigure Dynamically	163.54	37.78	67245.01
Equally Spread Current Execution With Closest Data Center	117.71	35.63	401.88
Equally Spread Current Execution With Optimize Response Time	117.73	36.01	540.56
Equally Spread Current Execution With Reconfigure Dynamically	134.75	37.78	73847.01
Throttled load Balancing With Closest Data Center	117.71	35.63	401.88
Throttled Load Balancing With Optimize Response Time	117.73	36.01	555.63
Throttled Load balancing With Reconfigure Dynamically	134.70	37.78	73847.01

Table 2. Data Center Processing Times

Load Balancing Algorithms With Different Service Broker Policies	Data Center Processing Time		
	Avg(ms)	Min(ms)	Max(ms)
Round Robin With Closest Data Center	1.15	0.02	2.56
Round Robin With Optimize Response Time	1.15	0.02	2.35
Round Robin With Reconfigure Dynamically	47.00	0.03	67193.51
Equally Spread Current Execution With Closest Data Center	1.15	0.02	2.17
Equally Spread Current Execution With Optimize Response Time	1.15	0.02	2.17
Equally Spread Current Execution With Reconfigure Dynamically	18.19	0.03	73794.50
Throttled load Balancing With Closest Data Center	1.15	0.02	2.17
Throttled Load Balancing With Optimize Response Time	1.15	0.02	2.17
Throttled Load balancing With Reconfigure Dynamically	18.15	0.06	73794.50

Table 3. Overall Processing Cost Summary

Load Balancing Algorithms With Different Service Broker Policies	Processing Cost		
	Total VM Cost	Total DC Cost	Grand Total
Round Robin With Closest Data Center	930.03	43.01	973.04
Round Robin With Optimize Response Time	930.03	43.01	973.04
Round Robin With Reconfigure Dynamically	2408.11	43.01	2451.12
Equally Spread Current Execution With Closest Data Center	930.03	43.01	973.04
Equally Spread Current Execution With Optimize Response Time	930.03	43.01	973.04
Equally Spread Current Execution With Reconfigure Dynamically	2408.27	43.01	2451.28
Throttled load Balancing With Closest Data Center	930.03	43.01	973.04
Throttled Load Balancing With Optimize Response Time	930.03	43.01	973.04
Throttled Load balancing With Reconfigure Dynamically	2408.11	43.01	2451.12

From Table 1, 2 & 3, it is inferred that Throttled Load Balancing Algorithm provides best response time, data center processing time with small processing cost as compared to Round Robin and Equally Spread Current Execution Algorithm. Among Different Service Broker Policies Closest Data Center is the best as it forwards the request to the closest data center and thus results in lesser response time.

CONCLUSION

Cloud Computing has widely been adopted by industry, though there are many existing issues like load balancing, Migration of Virtual Machines, Server Unification etc. which have not been fully addressed. On the contrary load balancing is the most central issue in the system i.e., to distribute load balancing in an efficient manner. It also ensures that every computing resource is distributed efficiently and fairly. Existing load balancing techniques/algorithms that have been studied mainly focus on reducing overhead, reducing the migration time and improving performance etc. The response time is a challenge of every engineer to develop the product that can increase the throughput in the cloud based sector. The several strategies lack efficient scheduling and load balancing resource allocation techniques leading to increased operational cost.

REFERENCES

[1] G. Pallis, “Cloud Computing: The New Frontier of Internet Computing”, IEEE Journal of Internet Computing, Vol. 14 No. 5, September/October 2010, pages 70-73.

[2] Armbrust M., Fox A., GriffithR., Joseph A. D., Katz R., Konwinski A., Lee G., Patterson D., Rabkin A., Stocia I. and Zaharia M. (2009) Above the Clouds: A Berkeley View of Cloud Computing, EECS Department, University of California, 1-23.

[3] A. Bhadani, and S. Chaudhary, “Performance evaluation of web servers using central load balancing policy over virtual machines on cloud”, Proceedings of the Third Annual ACM Bangalore Conference(COMPUTE), January 2010.

[4] B. P. Rima, E. Choi, and I. Lumb, “A Taxanomy and Survey of Cloud Computing Systems”, Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, Seoul, Korea, August 2009, pages 44-51.

[5] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud Computing: state of the art and research challenges", Journal of Internet Services and Applications, Vol. 1, No. 1, April 2010, pages 7-18.

[6] R. P. Mahowald, Worldwide Software as Service 2010-2014 Forecast: Software will never be same, In, IDC, 2010.

[7] Mishra, Ratan, Jaiswal, Anant,”Ant Colony Optimization: A Solution Of Load Balancing In Cloud”, April 2012, International Journal Of Web & Semantic Technology; Apr 2012, Vol. 3 Issue 2, P33.

[8] Eddy Caron, Luis Rodero-Merino”Auto Scaling, Load Balancing and Monitoring in Commercial and Open Source Clouds”Research Report, January 2012.

[9] Z. Zhang and X. Zhang, "A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation", Proceedings of 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), Wuhan, China, May 2010, pages 240-243.

[10] Saroj Hiranwal, Dr. K.C. Roy, "Adaptive Round Robin Scheduling Using Shortest Burst Approach Based On Smart Time Slice", International Journal Of Computer Science And Communication July-December 2011, Vol. 2, No. 2,Pp. 319-323.

[11] Bhathiya Wickremasinghe, "Cloud Analyst: A Cloud-Sim-Based Tool for Modeling And Analysis Of Large Scale Cloud Computing Environments MEDC Project", Report 2010.

[12] 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.

[13] R. Buyya, R. Ranjan, and R. N. Calheiros, “Modelling And Simulation Of Scalable Cloud Computing Environments And The CloudSim ToolKit: Challenges And Opportunities,” Proc. Of The 7th High Performance Computing and Simulation Conference (HPCS 09), IEEE Computer Society, June 2009.

[14] www.cloudbus.org/cloudsim.