International Journal of Electrical and Computer Engineering (IJECE)
Vol. 6, No. 3, June 2016, pp. 963 ~ 973
ISSN: 2088-8708, DOI: 10.11591/ijece.v6i3.7943

Load Balancing Techniques for Efficient Traffic Management in Cloud Environment

Talasila Sasidhar, Vani Havisha, Sai Koushik, Mani Deep, V Krishna Reddy
Department of Computer Science and Engineering, K L University, India

Article Info

Article history:

Received Apr 29, 2015
Revised Jan 19, 2016
Accepted Feb 2, 2016

Keyword:

Cloud data centers
Load balancing
Traffic management
Virtualization

ABSTRACT

Cloud computing is an internet based computing. This computing paradigm has enhanced the use of network where the capability of one node can be utilized by other node. Cloud service provides access on demand to distributive resources such as database, servers, software, infrastructure etc. in pay as you go basis. Load balancing is one of the vexing issues in distributed environment. Resources of service provider need to balance the load of client request. Load balancing is adapted in order to increase the resource consumption in Data centers that leads to enhance the overall performance of system achieving client satisfaction.

Corresponding Author:

Talasila Sasidhar,
Departement of Computer Science and Engineering,
K L University,
Vaddeswaram 522502, Guntur District, Andhra Pradesh, India.

1. INTRODUCTION

Traffic engineering in cloud data centers has become a major challenge particularly when the legacy protocols are employed in data centers .Data centers offer limited and un-scalable traffic management. Although the use of VLANs is a way to provide scalable traffic management. Generally broadcast domains are created by routers. But with the Virtualization of LAN's, a switch creates the broadcast domain. One needs VLAN when there are more than 200 devices on LAN, and there exists a lot of broadcast traffic on LAN, or when enabling a single switch into multiple virtual switches. With virtualization of a LAN, a device can be connected to one switch, another device can be connected to a different switch, and those devices can still be on the same broadcast domain.

Devices on different VLAN's communicate with a router which is used to route between the subnets. Configuring VLAN's can vary even between different models of switches. VLAN's offer higher performance for medium and large LAN's as on account they limit the broadcasts. As the amount of traffic and the number of devices raise so does the number of broadcast packets. VLAN's may even be considered for providing security because a user essentially puts one group of devices, in one VLAN, on their own network. A trunk port is a special port that runs ISL to carry traffic from more than one VLAN.

But VLAN's encloses few disadvantages as it is difficult to manage VLAN rather than managing only LAN, Traffic between VLAN's must go through router .i.e. one shall need a router, then should setup the routing protocol and trunk, there is a high risk of virus attacks because if one system of a VLAN is infected by virus then it may infect all the systems of that VLAN, Administrator needs to add additional layer of security, It allows to implement the logical grouping of devices by function instead of location. Existing paper introduced a novel decomposition approach to solve the VLAN mapping problem in cloud data centers

through column generation, which is an effective technique that is proven to reach optimality by exploring only a small subset of the search space.

Different load balancing algorithms have been proposed in order to manage the resources of service provider efficiently and effectively. This project presents the performance analysis of an efficient method for load balancing to solve some of the key features like overload rejection, process migration, and fault tolerance in cloud.

2. RELATED WORK

2.1. Load Balancing

Load balancing is one of the crucial issues of cloud computing which divides the workload dynamically among the processors by improving the performance of the system .The total processing time a machine requires to execute all the tasks assigned to it is termed as Workload. Load balancing is done so that every virtual machine in the cloud system does the same amount of work throughout resulting in increasing the throughput and minimizing response time. Balancing the load of virtual machines uniformly means that no machine is either idle or partially loaded but machines are loaded equally.

2.2. Benefits

By distributing the workload among the processors results in utilizing the available resources optimally by reducing the response time, enhancing the overall performance by achieving maximum client satisfaction. Also helps in implementing fail over, Enabling Scalability, there by voiding bottlenecks and over provisioning. Load balancing is needed for achieving green computing in clouds as only limited energy is consumed and less amounts of carbon is emitted.

Finally the goal of load balancing is to improve the performance substantially. With the help of load balancing a backup plan is maintained even when a system fails partially. Load balancing helps in continuing the service by provisioning and de-provisioning the instances of applications without fail. It maintains system stability. Load balancing accommodates future modification in the system.

2.3. Categories of load balancing algorithms

Broadly, Load balancing algorithms are categorized into three sets: Symmetric, Sender Initiated and Receiver Initiated. Symmetric load balancing is a combination of receiver initiated and sender. Based on the current state of the system load balancing is split into two categories a) Static Algorithm, b) Dynamic Algorithm

a) Static Algorithm – In this algorithm each server is assigned a weight and accordingly the highest weighted server receives more connections. In this situation all weights are equivalent and servers receive a balanced traffic [1].
b) Dynamic Algorithm – Allocates the accurate weights on servers by searching in the entire network and a lightest weighted server is preferred to balance the traffic

The main difference is, although based on a simple rule where more loads are conjured up on servers and resulting in imbalanced traffic, where as in dynamic load balancing is predicted on a query that can be made frequently on the servers, but sometimes existed traffic will prevent queries to be answered and correspondingly more added overhead. The following is the interaction among the components of a dynamic load balancing algorithm.

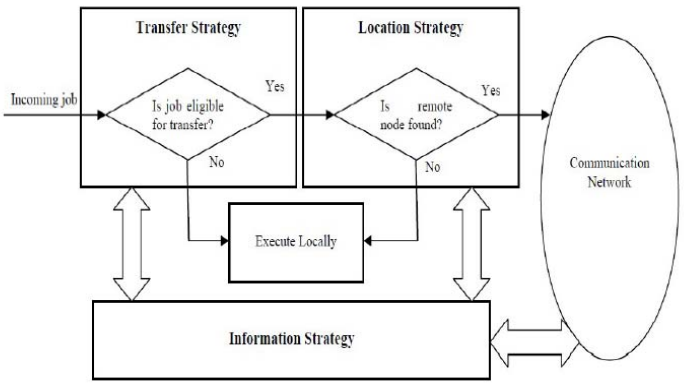

Figure 1. Interaction among the components of a dynamic load balancing algorithm

2.4. Load balancing Algorithms

To achieve the maximum load by distributing the workload among the multiple network links we employee other algorithms to distribute the load and also check the performance and cost.

2.4.1. Round Robin Algorithm

Round Robin is one of the existing load balancing techniques that distributes multiple network links to achieve maximum throughput [2] and minimum response time to avoid overloading. Here scheduling time quantum plays an important role.

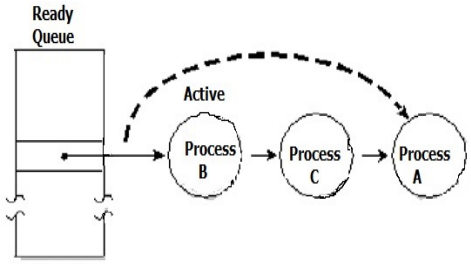

Figure 2. Round Robin Algorithm

Round Robin uses the time quantum concept where the time is divided into multiple segments and each node is given a particular time interval and a node has to perform its actions within this allocated time interval only. The resources provided to client on the based on time quantum. If the time quantum is large round robin algorithm is same as the FCFS. If the time quantum extremely too small then Round Robin scheduling is called processor.

Here selection of load on context switches and sharing of algorithm is random and this leads to situation where some nodes are heavily loaded and some are lightly loaded. Though the algorithm is very simple the additional load on the scheduler decides the size of quantum whereby it has longer average waiting time, high number of context switches, higher turnaround time [2] and low through put.

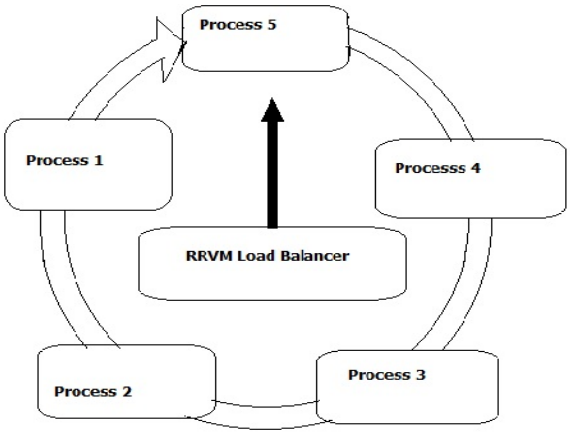

Figure 3. Execution of processes within time quantum in circular queue

Step- by – Step:

1. Round RobinVM Load Balancer (RRVM Load Balancer) maintains an index of VM's and state of VM's (busy/available).Initially, all VM's have zero allocation.
   a. The datacentercontroller receives the cloud request/cloudlets.
   b. It stores arrival /burst time of the user requests.
   c. The requests allocated to VM's are based on the states known from VM queue
   d. The RRVM Load Balancer allocates the time quantum for user request execution.

2. a. The RRVM Load Balancer calculates turn-around time for each process.
   b. It also calculates the response time and average waiting time of user requests.
   c. It decides the scheduling order
3. After the execution of cloudlets, the VM's are de-allocated by the RRVM Load Balancer
4. The datacentercontroller checks for the new/pending /waiting requests in queue

Continue from step-2.

2.4.2. Throttled Load Balancing Algorithm (TLB)

The total execution time in this algorithm is estimated in three stages. In the first stage virtual machines formed are ideally waiting for the scheduler to schedule the jobs in the queue, once jobs are allocated, the virtual machines in the cloud starts processing, which is the second stage, and finally in the third stage the cleanup or the destruction of the virtual machines occurs.

The proposed algorithm will improve the performance by providing the resources on demand, resulting in increased number of job executions and thus reducing the rejection in the number of jobs submitted. The throughput of the computing model can be estimated as the total number of jobs executed within a time span without considering the virtual machine formation time and destruction time.

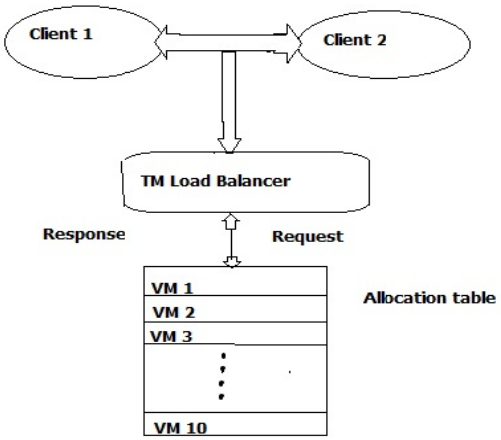

Figure 4. Throttled scheduling process

The proposed algorithm will improve the performance by providing the resources on demand, resulting in increased number of job executions and thus reducing the rejection in the number of jobs submitted.

Step- by – Step:

1. ThrottledVMLoadBalancer (TM Load Balancer) maintains an index table of VMs and the state of the VM (BUSY/AVAILABLE).
2. When VM is started it is said to be available., a DataCenterController receives a new request
3. DataCenter Controller inquires the new TMLoadBalancer for next location
4. TMLoadBalancer parses the allocation table from top and until the first available VM is found or parsed completely
5. If found:
   a. The TM Load Balancer returns the VM id to the DataCenterController.
   b. The DataCenterController sends the request to the VM identified by that id
   c. DataCenterController notifies the TM Load Balancer of the new allocation
   d. TM Load Balancer updates the allocation table accordingly
6. If not found:
   a. The TM Load Balancer returns -1
   b. The DataCenterController queues the request
   c. When the VM finishes processing the request, and the DataCenterController receives the response cloudlet, it notifies the TM Load Balancer for de-allocation.
   d. The DataCenterController checks for the left over waiting requests in the queue. If there exists any, it continues from step 3

Continue from step 2

2.4.3. Equally Spread Current Execution Algorithm (ESCE)

Here load balancer makes an effort to allocate equal load to all the virtual machines connected with the data centre. Load balancer maintains an index table of VM's along with number of requests currently assigned for each Virtual Machine (VM). When a request is originated from the data centre to allocate the new VM, then Load Balancer scans the entire index table for least loaded VM. If more than one VM is found then load balancer selects the first identified VM for handling the client/node's request, and also returns the VM id to the data centre controller.

The data centre identifies VM by id and communicates the request to it. The data centre revises the index table by increasing the allocation count of identified VM. When VM executes the assigned task, a request is communicated to data centre which is further notified by the load balancer that again revises the index tables by decreasing the identified VM's allocation count by one even though there remains an additional computation overhead for scanning the queue again and again.

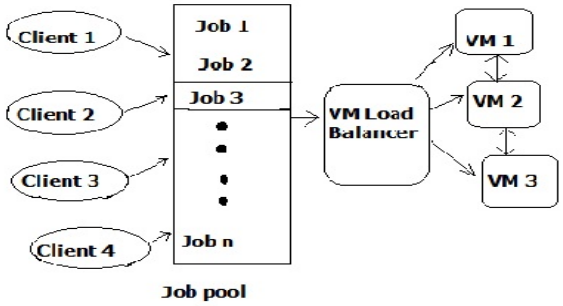

Figure 5. ESCE Process

Step- by – Step:

1. Find the next available VM.
2. Check for all current allocation count, if it is less than max length of VM, allocate the VM
3. If available VM is not allocated create a new one, Count the active load on each VM
4. Return the id of those VM which is having least load.
5. The VMLoadBalancer will allocate the request to one of the VM
6. If a VM is overloaded then the VMLoadBalancer will distribute some of its work to the VM having least work so that every VM is equally loaded.
7. The datacentercontroller receives the response to the request sent and then allocate the waiting requests from the job pool/queue to the available VM & so on
8. Continue from step-2

2.5. Deploying Algorithms of load balancing

2.5.1. Cloud analyst – simulation tool

Cloud analyst is actually a toolkit for simulation of cloud scenarios to support evaluation of social network tools according to geographic distribution of users and data centers [3]. Cloud analyst features are shown in Table 1. In this simulation tool communities of users and data centers supporting the social networks are characterized and based on their location, parameters such as user experience using the social network application and load on the data center are logged/obtained. Cloud Analyst is able to display the output in graphical form [4].

Table 1. Cloud Analyst Features

Parameters	Cloud Analyst
Communication on Network	Limited
Graphical Reports	Capable to display
Availability	Open Source
Platform	SimJava
Simulation time	Seconds
Language/Script	Java
Physical Models	None
Energy Models	None
Power Saver Modes	None

All components in Cloud Analyst communicate through the process of message passing. The lowermost layer is responsible for managing the communication between various components. The second layer has all the sub layers in it that have the main cloud components [5].

Cloud Analyst [6] is a GUI based tool which was developed on CloudSim [7] architecture [8]. CloudSim is a toolkit which permits a user to perform modeling, simulation. The cloud analyst tool as shown in Figure 6 removes all the complexities by developing GUI so that focus can be done on simulation rather than on programming. A user has access to perform simulations repeatedly with slight change in parameters very easily and quickly. The cloud analyst allows users to set the location of data centers for generating the application. In this various configuration parameters can be set such as number of users, number of request generated per user per hour, number of virtual machines, number of processors, amount of storage, network bandwidth and other necessary parameters. Taking the parameters into account the tool computes the simulation result and result is displayed in graphical form.

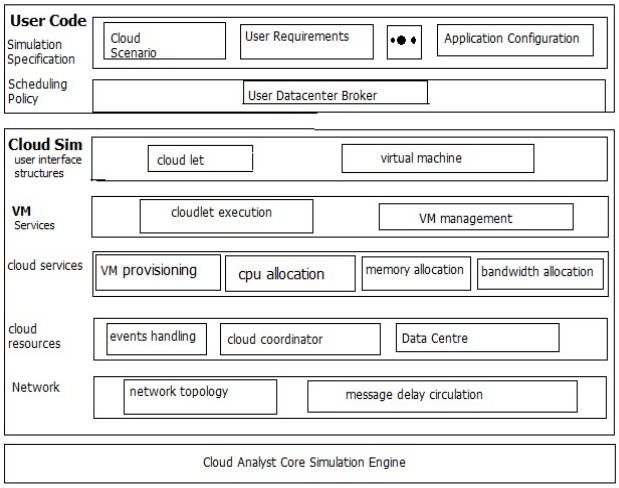

Figure 6. The Cloud Analyst Architecture

The outcome comprises response time, processing time, cost etc. By performing various simulation operations the cloud provider can focus on the most ideal approach for allocating the resources, choosing the data center, optimizing cost based on request. The various activities performed in cloud analyst tool are summarized as Figure 7.

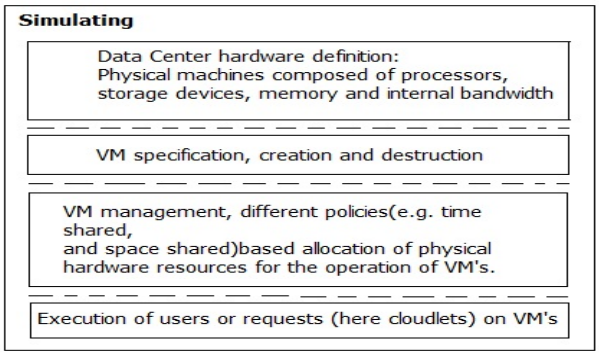

Figure 7. Tasks of Cloud analyst

The main components of cloud analyst tool are:

Simulation: By considering the various parameters this tool executes the simulation and outcomes the required results.

User Base: Here user base is modeled to represent the users who deploy the application.

Data Center Controller: Plays crucial role in controlling the various data center activities.

GUI Package: A graphical interface is displayed for various user interfaces to configure the diverse simulation parameters in an efficient way. The GUI of cloud analyst is shown in figure below

Internet Characteristics: Various internet characteristics are modeled for simulation, which incorporates the measure of latency and bandwidth and current performance level of the data centers for assigning between the regions.

Vm Load Balancer: Responsible for allocating the load on various data centers based on the request generated by users. One of the policy has to be selected from round robin algorithm, equally spread current execution load, and throttled.

Cloud App Service Broker: Handles the traffic routing between user bases and data centers by modelling service broker. The service broker can use one of the routing policies from the given three policies with option of choosing either closest data center or optimize response time and reconfiguring dynamically with load. The closest data center routes the traffic from the source user base to the closest data center in terms of network latency. Reconfiguring dynamically with load routing policy works on the principle load of whenever the performance of particular data center degrades below a given threshold value then the load of that data center is equally distributed among other data centers.

Simulation Configuration: Configuring various component of the cloud analyst tool need to be done various component of the cloud analyst tool for analyzing load balancing policies. In Figure 10, Figure 11 and Figure 12 parameters for the user base configuration, application deployment and data center configuration are shown. From the figure we can infere that the location of user bases has been defined in six different regions of the world. In the table below we have two data centers in use to handle the request of the client's/users.

The given in Table 2 represents the user base configuration and application deployment configuration. Simulation 180 mins.

Table 2. User Base Configuration

Name	Region	Requests per user	Data size per request	Peak hours start (GMT)	Peak hours end (GMT)	Avg.peak users	Avg.Peak-off users
UB1	0	3	1000	13	15	400000	40000
UB2	1	12	1000	15	17	100000	10000
UB3	2	8	1000	20	22	300000	30000
UB4	3	9	1000	1	3	150000	15000
UB5	4	7	1000	21	23	50000	100

Service Broker Policy : Closest Data Center

Application Deployment:

Table 3. Application Deployment Configuration

Data Center	#VM's	Image Size	Memory	BW
DC1	20	100	1024	10
DC2	20	100	1024	1000

Data Center Configuration:
Data Centers:

Table 4. Data Center Configuration

Name	Region	Arch	OS	VMN	Cost $/Hr	Memory Cost	Storage Cost $/Hr	Data Transfer Cost per $/Gb	Physical HW Units
DC1	0	x36	Linux	Xen	01	035	01	01	2
DC2	2	x86	Linux	Xen	01	035	01	01	1

3. RESULTS AND ANALYSIS

After simulating, the result computed by cloud analyst is as shown in the following figures. Configuration for each load balancing policy 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 Figures 9, 10, 11.

3.1. Response Time

Response time for each user base and overall response time is calculated by the cloud analyst for each loading policy and results are tabulated in the Table 5, 6 and 7 respectively. We can infer from the figure that overall response time of Round Robin policy and ESCE policy is almost same while that of Throttled policy is very much low as compared to other two policies.

Table 5. Overall Response Time of Round Robin Algorithm

Overall Response Time Using Robin Policy	Average (ms)	Minimum (ms)	Maximum (ms)
Overall Response time	754.81	67.97	1589.10
Data Center Processing time	472.77	0.40	1064.89

Response time by region

User Base	Average (ms)	Minimum (ms)	Maximum (ms)
UB1	172.608	67.974	244.91
UB2	229.71	177.144	340.717
UB3	243.605	162.125	340.563
UB4	1,173.705	303.74	1589.994
UB5	317.408	278.357	356.425
UB6	212.468	169.318	327.106

Table 6. Overall Response Time of ESCE Algorithm

Overall Response Time Using ESCE	Average (ms)	Minimum (ms)	Maximum (ms)
Overall Response time	757.45	67.97	1580.08
Data Center Processing time	475.50	0.40	1053.09

Response time by region

User Base	Average (ms)	Minimum (ms)	Maximum (ms)
UB1	172.897	65.767	244.378
UB2	229.507	177.144	356.597
UB3	243.925	162.125	340.241
UB4	1,173.218	303.34	1580.882
UB5	318.247	278.357	375.242
UB6	212.526	169.318	327.052

Table 7. Overall Response Time of Throttled Algorithm

Overall Response Time Using Throttled	Average (ms)	Minimum (ms)	Maximum (ms)
Overall Response time	511.33	63.30	1456.36
Data Center Processing time	246.06	0.40	935.23

Response time by region

User Base	Average (ms)	Minimum (ms)	Maximum (ms)
UB1	117.943	63.3	194.085
UB2	225.713	177.144	328.153
UB3	160.77	72.256	303.224
UB4	781.09	299.547	1456.362
UB5	318.209	278.357	374.415
UB6	210.389	169.318	336.314

3.2. Data Center Request Servicing Time

Data Center Request Servicing Time for each data center calculated by the cloud analyst for each loading policy has been shown in the Table 8 respectively. This has tabulated that servicing time of Round Robin policy and ESCE algorithm is almost same while that of Throttled policy is very much low as compared to other two policies.

Table 8. Overall Data Center Request Serving Time of Algorithms

Data Center Request Servicing Time			
For Round Robin Algorithm			
Data Center	Average (ms)	Minimum (ms)	Maximum (ms)
DC 1	68.673	1.911	173.345
DC 2	646.238	0.404	1064.888
For ESCE Algorithm			
Data Center	Average (ms)	Minimum (ms)	Maximum (ms)
DC1	68.876	1.911	171.756
DC2	649.911	0.404	1053.088
For Throttled Algorithm			
Data Center	Average (ms)	Minimum (ms)	Maximum (ms)
DC 1	37.348	1.911	120.454
DC 2	334.222	0.404	935.23

3.2.1. Load Balancing Challenges – Cloud Computing

In cloud computing, load balancing is required to distribute the dynamic local workload evenly across all the nodes. It assists in high user satisfaction and resource utilization ratio by guaranteeing a proficient, reasonable distribution of each processing resource. Appropriate load balancing supports in thinning resource utilization, actualizing fail-over, enabling scalability and elasticity, keeping away from bottlenecks etc.. [9],[10]. Despite the fact that cloud computing is on pace. Research in cloud computing is still in its initial stages, and some experimental difficulties stay unsolved by established researchers, especially load adjusting difficulties [11].

Elasticity is key feature in cloud where resources can be allocated or released automatically. And a user can we use or release the resources of the cloud, by keeping the same performance as traditional systems by making use of best possible resources.

3.3. Virtual Machines Migration

With virtualization, an entire machine can be seen as a file or set of files, to unload a heavily Loaded machine, it is possible to shift a virtual machine between physical machines. The main objective is to distribute the load in a datacenter or set of datacenters. Then dynamic distribution of load by moving the virtual machine by users is unanswerable as this keeps away from the bottlenecks in Cloud computing framework.

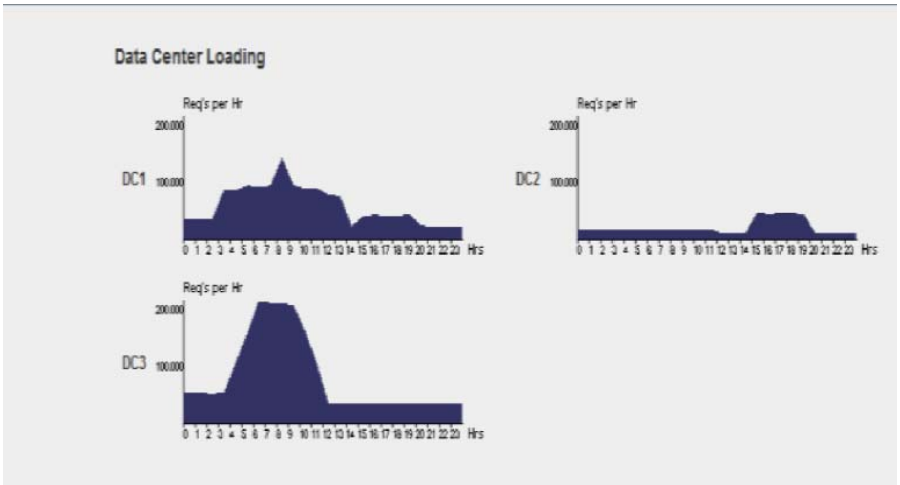

Figure 8. Data Center Loading Time

Figure 9. User Hourly Response Time

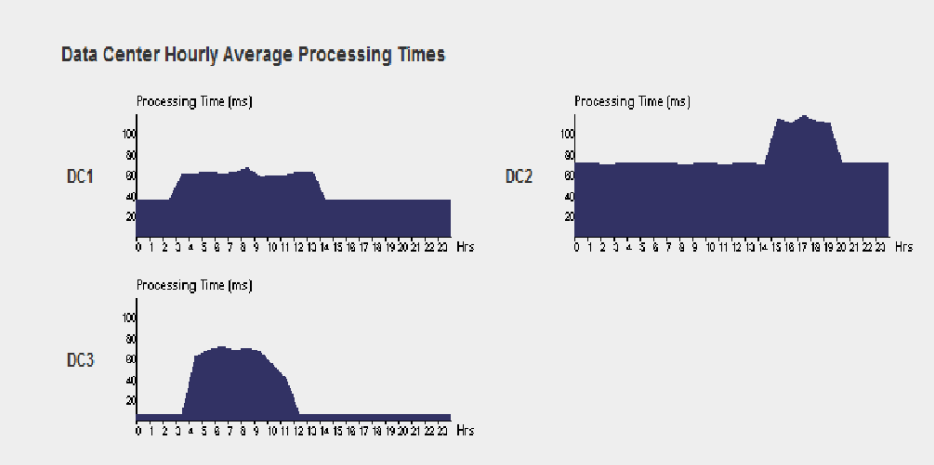

Figure 10. Data Center Hourly Processing Times

3.4. Energy Management

Economy of sale is a beneficiary factor that supports cloud. Energy saving is a crucial note that allows a set of global resources supported by condensed providers. If so then how a user can use a part of Datacenter while maintaining standard performance remains unsolvable.

3.5. Emergence of Small Data Centers for Cloud Computing

Small datacenters are beneficial as they are less expensive .Small providers deliver the cloud computing services leading to geo-diversity computing. Yet at the same time Load balancing will become a problem on a global scale to ensure an adequate response time with an optimal circulation of a resource.

3.6. Stored Data Management

From the past, data stored across the network has an exponential raise even for organizations by outsourcing their data storage or for individuals, the management of data storage has become a major challenge for cloud computing. Then the distribution of the data for optimum storage in cloud for a quick access is the present day's challenge.

4. CONCLUSION

Load Balancing distributes the dynamic local workload evenly across all the nodes in the clouds. Load Balancing strives to achieve a high user satisfaction and resource utilization ratio by avoiding situation where left over nodes are either heavily balanced or idle. There by overall performance and resource utility of the system increases. With proper balancing, resource utility ratio is maintained minimum which will further reduce energy consumption.

In this paper Existing load balancing techniques that have been discussed which focus on reducing associated overhead, service response time and improving performance etc. but none of the techniques has considered the energy consumption and carbon emission factors. Yet at the same time there are numerous existing issues which have not been fully addressed like Load Balancing, Virtual Machine Migration, Server Consolidation, and Energy Management. Key to these issues is the issue of load balancing, that is obliged to distribute the excess dynamic local workload evenly to all the nodes in the Cloud to attain to a high client fulfillment and resource utilization ratio.

REFERENCES

[1] R. X. T. and X. F. Z., "A Load Balancing Strategy Based on Combination of Static and Dynamic," in *Database Technology and Applications (DBTA), 2010 .2nd International Workshop (2010)*, pp.1-4.

[2] S. Hiranwal, K. C. Roy, "Adaptive Round Robin Scheduling Using Shortest Burst Approach Based On Smart Time Slice," *International Journal Of Computer Science and Communication*, vol/issue: 2(2), pp. 319-323, 2011.

[3] M. Rahul and J. Prince, "Study and Comparison of CloudSim Simulators in the Cloud Computing," *The SIJ Transactions on Computer Science Engineering & its Applications (CSEA)*, vol/issue: 1(4), pp. 111-115, 2013.

[4] B. Wickremasinghe, "CloudAnalyst: A CloudSim based Tool for Modeling and Analysis of Large Scale Cloud Computing Environments," *MEDC Project Report*, 2009.
[5] K. Gaganjot and K. Pawan, "Study of Comparison of Various Cloud Computing Simulators," *2nd National Conference in Intelligent Computing & Communication*.
[6] "Cloud Analyst: A Cloud-Sim-based Tool for Modeling and Analysis of Large Scale Cloud Computing Environments," *MEDC Project Report Bhathiya Wickremasinghe*.
[7] R. Buyya, *et al*., "Modeling 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.
[8] http://www.cloudbus.org/cloudsim
[9] A Book by O' Reilly on "Cloud Security and Privacy".
[10] S. V. Gorge, *et al.*, "Multi-Tier Distributed Load Balancing," *CS598RHC Literature Survey*.
[11] M. Khiyaita, *et al.*, "Load Balancing Cloud Computing: State of Art," 9778-1- 4673-1053-6/12/$31.00, 2012.