Volume 6, Issue 5, May 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Load Balancing in Cloud Computing Using Modified Optimize Response Time

Vidhi Tailong, Vivek Dimri (Assistant Professor)
Department of Computer Science and Engineering, Sharda University, Greater Noida,
Uttar Pradesh, India

Abstract: ***Cloud computing comes as a new approach for organizing, and applications accessing over the large network. Cloud computing brings many advantages like flexibility, cost, and resource availabity for service user. These facilities further demand for cloud services. Cloud computing has many issues like power management, security and load balancing. We are addressing a main issue of cloud computing that is load balancing, this paper is describing about the load balancing issue in cloud computing, what is the impact of load unbalancing in the network, need of load balancing, also providing an approach to for enhancing the load distribution process . Basically load balancing is the process of shifting work load among various processors so that whole system become reliable and collision free and work can be distribute fair in between all the server . It not only increases the efficiency of virtual machines, also improves the performance of the system.***

Keywords: CLBDM, LBMM, COMP, DDFTP

I. INTRODUCTION

Today everyone wants to use smart phone so the demand of internet is also increasing day by day. Host request for the resources and internet made available those resources to host. Where these resources are resided? What is the mechanism inside this? These are the basic questions comes in mind. The answer of this question is network, all the resources are resides inside different servers and these servers are residing inside a network. Cloud computing is a new internet concept becoming popular to provide different type of resources/services to the host/user (Niroshinie and Seng) we can also say that cloud computing is the internet based computing in which the services like storage, server and application are provided to organizations computer using internet.

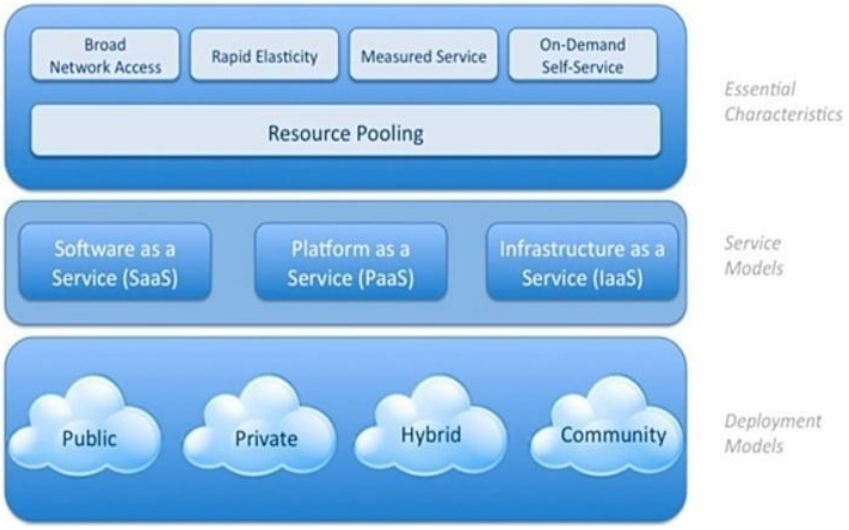


Fig. 1 Cloud computing architecture

Note. [The data in figure are from
A Survey of Various Load Balancing Techniques
(Tushar Desai, JigneshPrajapati)

Three main services provided by the cloud are IaaS, SaaS, and PaaS(Supriya and sukhvir, 2012). But there are many issues in cloud computing environment like security, power consumption, and load balancing. Load balancing is the main issue in the cloud environment (Dr. Chander and Minakshi, 2015). Load balancing is the concept in which the network workload is distribute between other nodes in the network. On the other words when sudden work load comes in the network of cloud computing the load balancer distribute that load among the other computers in the network. To enhance the global throughput of these cloud environments, workloads should be evenly distributed among the available

resources. Through load balancing we can achieve optimal resources utilization, minimum response time, maximum throughput and avoid overload. Cloud computing has some benefits like scalability, virtualization, Mobility, Low infrastructure cost, Increased storage.

II. CLOUD COMPUTING CHARACTERISTICS

On demand services Cloud computing provides on demand services without any involvement of human interaction examples are: - Gmail, Microsoft, Google and various application are the services provided by cloud computing.

Broad access- Cloud computing provides excess of network through any platform like laptops, mobile phone, desktop etc.

Resource pooling- It provide pooling of resources means collection of resources when any client request for a particular resources it will be available for the client.

Rapid elasticity- Cloud provides services every time everywhere quickly and also elastically.

Load Balancing- The process of shifting work load among various servers is known as load balancing.

It not only increases the efficiency of virtual machines, improves the performance of the system but also reduces energy consumption and ensures, maximize throughput, maximum resource utilization and reduce response time, thereby reducing the number of job rejection. Load balancing has increasingly become important as the number of users is increasing on the cloud. Multiple requests from clients reach the load balancer which distributes each of them across multiple computers or network devices. There are mainly two different type of load balancing dynamic and static load balancing.

2.1 Advantages of load balancing

- To expand the performance significantly.
- To have a backup plan in case the system fails even partially.
- To keep the framework stable.
- To give future improvement in the framework
- To distribute the work load fairly.

2.2 Need of load balancing

- To improve the performance substantially.
- To have a backup plan in case the system fails even partially.
- To maintain the system stability.
- To accommodate future modification in the system.

2.3 Load balancing algorithms:-

Round robin-- In round robin, a number of requests are assigned by datacenter to a list of VMs on a rotating basis (Amjad Mahmood and Irfan Rashid, 2011). The first request is assigned to a VM- selected randomly from the list of VMs and then the Data Center controller assigns the particular requests in a circular order. Once the VM is assigned the request, the VM id is moved to the end of the list. In this manner, Round Robin Load Balancer works.

Throttled load balancing-- In Throttled load balancing, a record of the state of each virtual machine (busy/ideal) is maintained. If a request arrived regarding the allocation of virtual machine, ID of ideal virtual machine is send by TLB to the data center controller and data center controller allocates the ideal virtual machine.

Least-connection-- The least-connection scheduling algorithm (Amjad Mahmood and Irfan Rashid,2011). Directs network connections to the server with the least number of established connections. This is one of the dynamic scheduling algorithms; because it needs to count live connections for each server dynamically. For a virtual server that is managing a collection of servers with similar performance, least-connection scheduling is good to smooth distribution when the load of requests vary a lot (Amjad Mahmood and Irfan Rashid,2011).

Active Clustering -- Active Clustering works on the principle of grouping similar nodes together and working on these groups. The performance of the system is enhanced with high resources thereby in-creasing the throughput by using these resources effectively. It is degraded with an increase in system diversity (Anandharajan, Dr. M.A. Bhagyaveni, 2011).

Min-Min Algorithm-- It begins with a set of all unassigned tasks. First of all, minimum completion time for all tasks is found. Then among these minimum times the minimum value is selected which is the minimum time among all the tasks on any resources. Then according to that minimum time, the task is scheduled on the corresponding machine. Then the execution time for all other tasks is updated on that machine by adding the execution time of the assigned task to the execution times of other tasks on that machine and assigned task is removed from the list of the tasks that are to be assigned to the machines. Then again the same procedure is followed until all the tasks are assigned on the resources. But this approach has a major drawback that it can lead to starvation.

III. DESCRIPTIVE STUDY

In this segment we talk about existing load balancing systems in cloud computing. Here we characterize load balancing algorithm in two principle sorts that are Static load balancing and Dynamic load balancing. In 2009, B Sotomayor et al [4] presented a static surely understood load balancing procedure called Round Robin, in which all procedures are separated in the midst of every single accessible processor. The assignment request of procedures is kept up locally which is autonomous of the distribution from the remote processor. In this strategy, the solicitation is sent to

the hub having minimum number of associations, and as a result of this sooner or later of time, some hub might be intensely loaded and other stay unmoving [4]. This issue is explained by CLBDM. In 2010, S C. Wang et al. [5] displayed a dynamic load balancing algorithm called load balancing Min-Min (LBMM) strategy which depends on three level systems. This procedure utilizes Opportunistic Load Balancing algorithm which keep every hub occupied in the cloud without considering execution time of hub. Due to this it causes bottle neck in framework. This issue is fathomed by LBMM three layer engineering. To begin with layer solicitation administrator which is in charge of getting undertaking and doling out it to one administration supervisor to second level. On getting the solicitation administration director isolate it into subtasks. After that administration administrator will dole out subtask to administration hub to execute errand. In 2011, B. Radojevic et al [6] presented a static load balancing algorithm called CLBDM (Central Load Balancing Decision Model). CLBDM is an upgrade of the Round Robin method. This depends on session exchanging at application layer. In round robin, solicitation is sent to the hub having minimum number of associations. RR is improved and in CLBDM, the count of the association time between the customer and the hub is done and if the association time goes over the limit then issue is raised. In the event that an issue is emerges, then the association between the customer and the hub is ended and the Task is sent to the further hub utilizing Round Robin law. In 2011, L. Colb et al [7] presented the Map Reduced based Entity Resolution load balancing strategy which depends on extensive datasets. In this procedure, two principle assignments are done: Map undertaking and Reduce errand which the creator has depicted. For mapping errand, the PART strategy is executed where the solicitation substance is apportioned into parts. And afterward COMP strategy is utilized to think about the parts lastly comparable elements are assembled by GROUP technique and by utilizing Reduce assignment. Map undertaking peruses the elements in parallel and processes them, so that overloading of the assignment is lessened. In 2011, J Hu et al. [8] presented a static booking technique of load balancing on virtual machine resource. This method considers the chronicled information furthermore the present condition of framework. Here, focal scheduler and resource screen is utilized. The booking controller checks the accessibility of resources to perform an undertaking and appoints the same. Resource accessibility subtle elements are gathered by resource screen. In 2011, J Al-Jaroodi et al. [9] proposed a dynamic load balancing method named DDFTP (Duel Direction Downloading Algorithm from FTP server). This can likewise be actualized for load balancing in cloud computing. In DDFTP, document of size m is separated into m/2 segment and every hub begins preparing the assignment. For instance if one server begins from 0 to incremental request than other will begin from m to adverse request autonomously from each other. As on downloading two continuous hinders the undertaking is considered as completed and relegated next errand to server. In light of decrease in system correspondence in the middle of customer and hub organize overhead is lessened. In 2012, K. Nishant et al [10] presented a static load balancing procedure called Ant Colony Optimization. In this procedure, an ant begins the development as the solicitation is started. This method utilizes the Ants conduct to gather data of cloud hub to allot assignment to the specific hub. In this strategy, once the solicitation is started, the ant and the pheromone begins the forward development in the pathway from the "head" hub. The ant moves in forward bearing from an overloaded hub searching for next hub to check whether it is an overloaded hub or not. Presently if ant find under loaded hub still it move in forward heading in the way. Furthermore, on the off chance that it finds the overloaded hub then it begins the retrogressive development to the last under loaded hub it discovered beforehand. In the algorithm [8] if ant found the objective hub, ant will submit suicide with the goal that it will forestall pointless in reverse development. In 2012, T. Yu Wu et al. [11] presented a dynamic load balancing strategy called Index Name Server to minimize the information duplication and excess in framework. This method takes a shot at mix of de duplication and access point streamlining. To compute ideal choice point some parameter are characterized: hash code of information square to be downloaded, position of server having target piece of information, move quality and most extreme transfer speed. Another estimation parameter to discover climate association can deal with extra hub or is at occupied level B(a), B(b) or B(c). B(a) signify association is extremely occupied to handle new association , B(b) signifies association is not occupied and B(c) means association is restricted and extra study expected to know more about association. In 2013, D. Babu et al [13] proposed a Honey Bee Behavior enlivened Load Balancing procedure which accomplishes even load balancing crosswise over virtual machine to amplify throughput. It considers the need of assignment sitting tight in line for execution in virtual machines. After that work load on VM figured chooses climate the framework is overloaded, under loaded or adjusted. Furthermore, in view of this VMs are assembled. New as indicated by load on VM the errand is booked on VMs. Assignment which is uprooted before. To locate the right low loaded VM for current undertaking, errands which are expelled before from over loaded VM are useful.

IV. METHODOLOGY

This research will use CloudAnalyst as a framework in the simulator environment. Implementation has been started with installation of simulation package CloudAnalyst on Windows 8.0. Thereafter Java version 7 is installed and class path along with other necessary execution setup requirement is fulfilled. The minimum requirement of this experiment is VM (Virtual machine) memory of 1GB, VM bandwidth of 1000 and local operating system used as a host. In this simulation setup, equally spreads algorithms have been executed with enhanced optimize time and compare with old.

4.1 Simulation Scenario

The simulated scenario corresponds to the peer to peer architecture since the masters in the peer to peer architecture correspond to datacenters and the slaves correspond to Virtual Machines (VMs). We further have user bases (UBs) from where the user requests are generated in the form of Cloudlets.The analysis is done based on the following parameters: load distribution on datacenters, number of tasks executed, and average processing time for execution of tasks.

4.2 Performance metrics

1. Load per node- the load on a slave node is calculated in terms of requests received by the node during the total simulation time. Ideally, the load on all nodes should be equal. In our model the load on all the virtual machines is almost equal.
2. Percentage of task executed- this parameter calculated the number of tasks processed vs. the number of task received by a slave.
   Percentage of tasks = Tasks executed/total tasks received*100
3. Average processing time for execution of tasks- the processing is the time taken to transfer and execute the task along with the time taken for the transfer of the result back to the user.

4.3 Algorithm Used

Round robin algorithm

Round-robin load balancing is one of the simplest methods for distributing client requests across a group of servers. It is one of the simplest scheduling techniques that utilize the principle of time slices. Here the time is divided into multiple slices and each node is given a particular time slice or time interval i.e. it utilizes the principle of time scheduling.

Step 1:- Round robin Vm load balancer maintain an index of vms and state of the vms (busy/available).At start all vm's have zero allocation.

Step 2:- (a) The data center controller receives the user request/cloudlets.
(b) It stores the arrival time and burst time of the user requests.
(c) The request is allocated to vms on the basis of their states known from the vm queue.
(d) The round robin vm load balancer will allocate the time quantum for user request execute.

Step 3:- (a) The round robin vm load balancer will calculate the turn- around time of each process.
(b) It also calculates the response time and average waiting time of user requests.
(c) It decides the scheduling order.

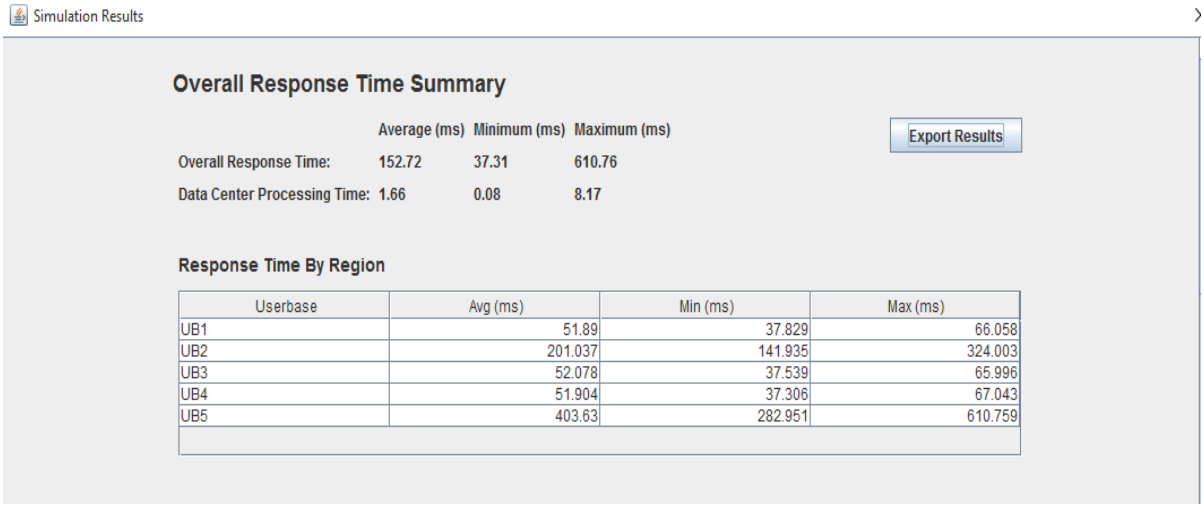
Step 4:- After the execution of cloudlets, the vms are de-allocated by the round robin vm load balancer.
Step 5:- The data center controller checks for new/pending/waiting requests in queue.
Step 6:- Continue from step2.

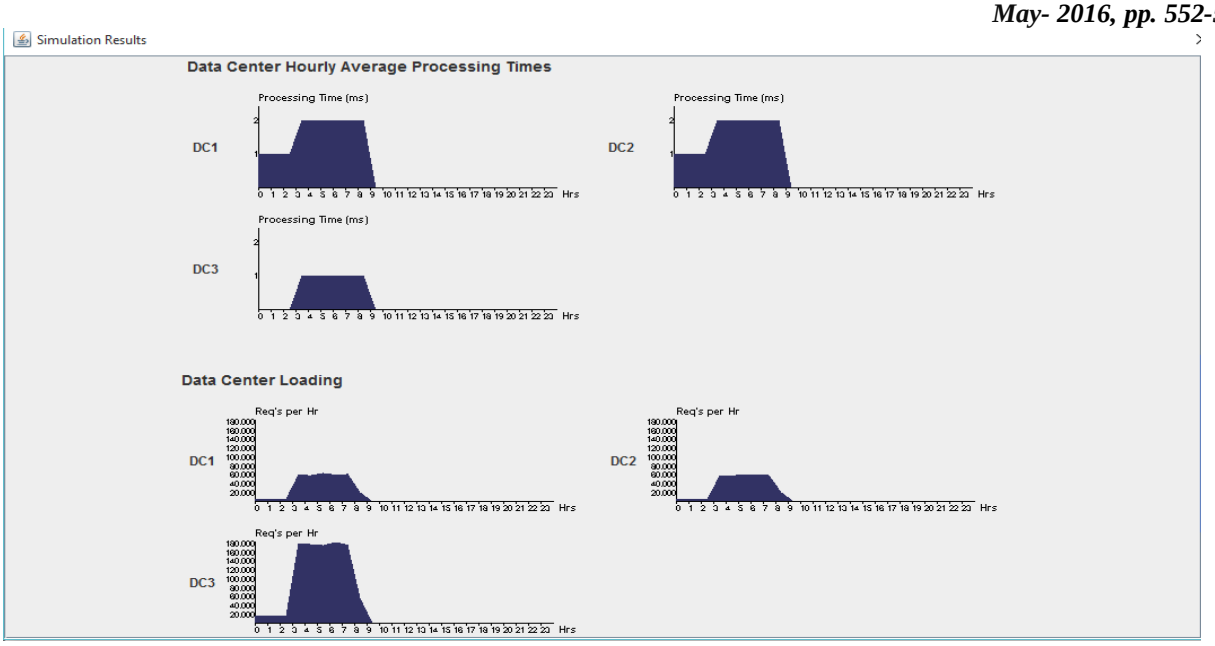
V. EXPERIMENTAL RESULT

We did an experiment in service broker policy of cloud analyst, the experiment includes sorting and after sorting mapping function will run to map the user bases with the data center. Service broker policy is the policy by which an algorithm decides to distribute load among the data center. We used optimize response time service broker policy in which data center choose according to their response time. We apply a sorting in the optimize response time service broker policy and then find out the results and compare with the result which is without sorting .we are using round robin algorithm for distribution of load.

According to new response time with sorting and mapping

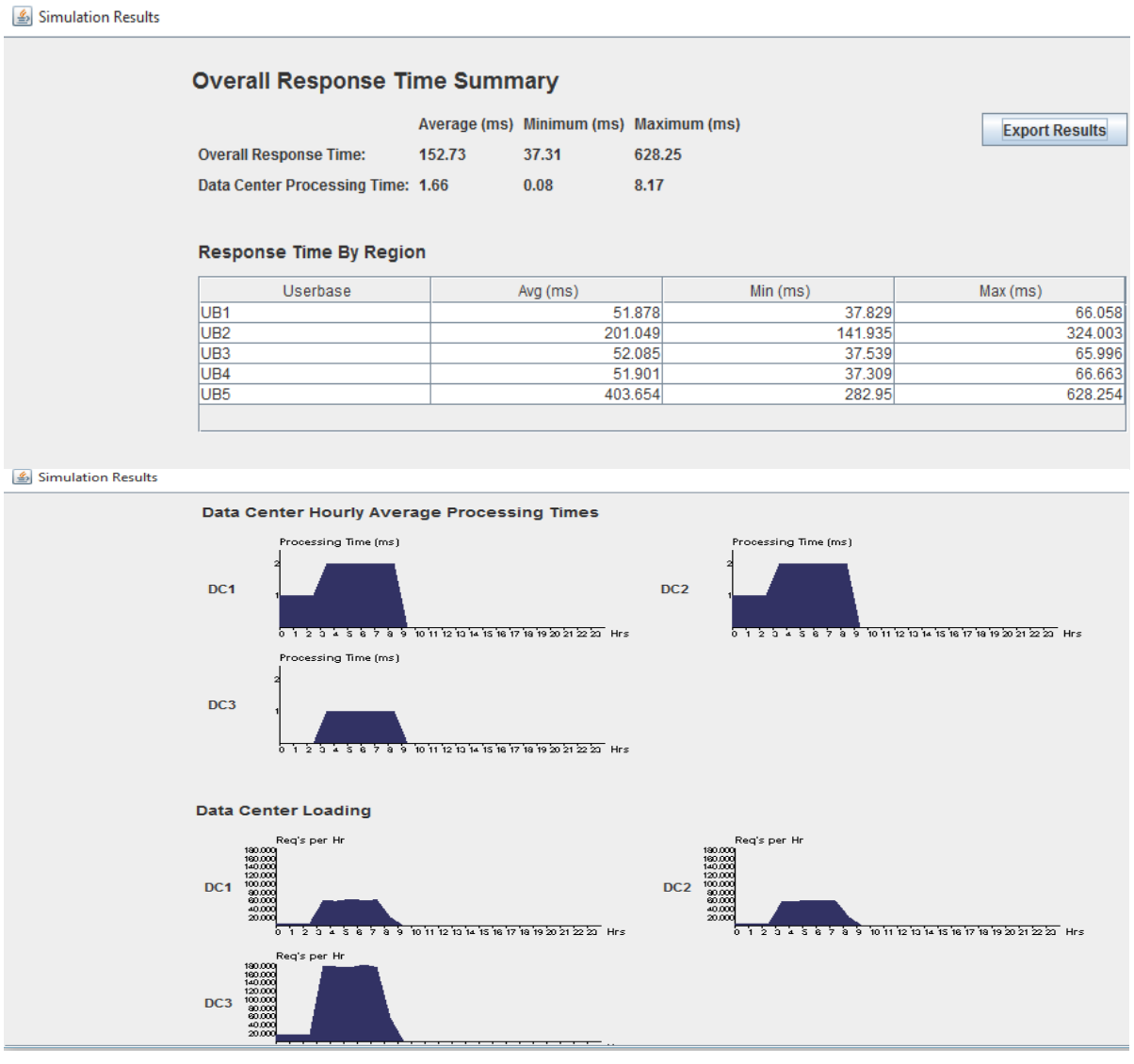
Overall response time	Avg	Min	Max
	152.72	37.31	610.76
Data Center processing time	1.66	0.08	8.17





According to old response time without sorting and mapping –

Overall response time	Avg	Min	Max
	152.73	37.31	628.25
Data Center processing time	1.66	0.08	8.17



VI. PERFORMANCE MATRIX

- Load per node – the load on a slave node is calculated in terms of requests received by the node during the total simulated time. Ideally, the load on all nodes should be equal and in case of optimize response time the load per virtual machines is almost equal as compared to our service broker policy.
- Percentage of tasks executed – This parameter calculates the number of tasks processed vs. the number of tasks received by a slave.
  Percentage of tasks = Tasks Executed/Total Tasks
  Received * 100.
  - Request received is 335366 and request processed is 317864 Percentage of task executed in optimize response time without sorting and mapping using round robin = 94.78%
  - Request received is 351006 and request processed is 338005 Percentage of task executed with enhanced optimize response time using round robin = 96.29%
- Average processing time for execution of tasks – The processing time is the time taken to transfer and execute the task along with the time taken for the transfer of the results back to the user base. For optimize response time without sorting and mapping is 1.66 and for enhanced response time average processing time is also 1.66.

VII. CONCLUSION AND FUTURE ENHANCEMENT

We modified service broker policy(optimize response time) of cloud analyst in that we incorporate sorting and mapping so that policy have all the data center list according to the response time in ascending order so that it can further map the user bases according to the list of data centers. We have used round robin algorithm for load distribution among data centers, modified response time service broker policy shows better results.

In round robin algorithm using modified optimize response time service broker policy the maximum response time for request is 610.76 and in case of old optimize response time service broker policy maximum response time is time is 628.25. Percentage of task executed in optimize response time without sorting and mapping using round robin = 94.78% and Percentage of task executed with enhanced optimize response time using round robin = 96.29% which is good.

REFERENCE

[1] Singh, A., D. Juneja, et al. (2015). "Autonomous Agent Based Load Balancing Algorithm in Cloud Computing." Procedia Computer Science 45: 832-841.

[2] Kherani, F. and J. Vania (2014).Load Balancing in cloud computing. International Journal of Engineering Development and Research, IJEDR.

[3] Mondal, B., K. Dasgupta, et al. (2012). "Load balancing in cloud computing using stochastic hill climbing-a soft computing approach." Procedia Technology 4: 783-789.

[4] Dorigo, M., M. Birattari, et al. (2006). "Ant colony optimization." Computational Intelligence Magazine, IEEE 1(4): 28-39.

[5] N. S. Raghava and Deepti Singh "Comparative Study on Load Balancing Techniques in Cloud Computing" OPEN JOURNAL OF MOBILE COMPUTING AND CLOUD COMPUTING Volume 1, Number 1, August 2014.

[6] Dhurandher, S. K., M. S. Obaidat, et al. (2014). A cluster-based load balancing algorithm in cloud computing. Communications (ICC), 2014 IEEE International Conference on, IEEE.

[7] Kaur, Sukhvir, and Supriya Kinger. "Review on Load Balancing Techniques in Cloud Computing Environment." IJSR, ISSN: 2319-7064.

[8] A. M. Alakeel, A guide to dynamic load balancing in distributed computer systems, in: IJCSNS International Journal of Computer Science and Network Security,VOL.10 No.6., 2010, pp. 153–160.

[9] Karger, David R., and Matthias Ruhl. "Simple efficient load balancing algorithms for peer-to-peer systems." In Proceedings of the sixteenth annual ACM symposium on Parallelism in algorithms and architectures, pp. 36-43. ACM, 2004.

[10] Xu, Zhiyong, and Laxmi Bhuyan. "Effective load balancing in p2p systems." In Cluster Computing and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on, vol. 1, pp. 81-88. IEEE, 2006.

[11] Karger, David, and Matthias Ruhl. "New algorithms for load balancing in peer-to-peer systems." (2003).

[12] Qiao, Ying, and Gregor V. Bochmann. "Load balancing in peer-to-peer systems using a diffusive approach." Computing 94, no. 8-10 (2012): 649-678.

[13] Hsiao, H.-C., H.-Y.Chung, et al. (2013). "Load rebalancing for distributed file systems in clouds." Parallel and Distributed Systems, IEEE Transactions on24(5): 951-962.

[14] Karger, D. R. and M. Ruhl (2004). Simple efficient load balancing algorithms for peer-to-peer systems. Proceedings of the sixteenth annual ACM symposium on Parallelism in algorithms and architectures, ACM..

