LOAD BALANCING OF VIRTUAL MACHINES USING SERVICE BROKER ALGORITHM

Sameena Naaz[1], Iffat Rehman Ansari[2], Amir Khan[3]

[1,3]Department of Computer Science and Engineering, Jamia Hamdard New Delhi, India

[2] Corresponding Author , University Women's Polytechnic, Aligarh Muslim University, Aligarh, U.P., India

ABSTRACT

Cloud computing is a term which is associated with of virtualization, distributed computing, networking, software and web services. A cloud is made up of several elements such as clients, datacenter and distributed servers. Some of the desired features of cloud environment includes fault tolerance, high availability, scalability, flexibility and minimum overhead for users. It should also have features such as reduced cost of ownership, on demand services etc.

In cloud computing the computing resources that is the service nodes are deployed in the network is a manner which eases the execution of complicated tasks that require large-scale computation. The selection of nodes for executing a task in cloud computing must be considered properly so that the available resources can be used efficiently. In this work a new load balancing algorithm for virtual machine has been developed using the service broker approach and it has been simulated using Cloud Analyst. This algorithm has been compared with the Round Robin algorithm and the throttled load balancing algorithm and it has been found that it performs better than both the algorithms in terms of overall response time and data center processing time.

Keywords: *Virtual Machine, Load Balancing, Cloud Analyst, Round Robin Algorithm, Throttled Algorithm.*

I. INTRODUCTION

Cloud computing is an alternative to all the computer hardware and software that we are using sitting on our desktop, or someplace inside our company's network. It is provided to us as a service by another company and accessed over the Internet, commonly in a completely logical way. The offered Cloud Service Models are classified as Infrastructure as a service (IaaS), Platform as a service (PaaS) and Software as a service (SaaS). Gmail, Google Docs, Microsoft Windows Azure, Google App Engine and Amazon Elastic Compute Cloud (EC2) are examples of cloud service. The offered Cloud Deployment models are classified as public or private or combined of both [1]. The importance of these services is highlighted in a recent report from Berkeley as: "Cloud computing, the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service" [2].

ISBN : 978-93-86171-07-8

Clouds [3] aim to power the next generation data centers by exposing them as a network of virtual services (hardware, database, user-interface, application logic) so that users are able to access and deploy applications from anywhere in the world on demand at competitive costs depending on users QoS (Quality of Service) requirements [4]. Developers with innovative ideas for new Internet services are no longer required to make large capital outlays in the hardware and software infrastructures to deploy their services or human expense to operate it [2]. It offers significant benefit to IT companies by freeing them from the low level task of setting up basic hardware and software infrastructures and thus enabling more focus on innovation and creation of business values. We can refer to Cloud computing as a bunch of distributed servers known as masters, providing demanded services and resources to different clients. Physical resources can be broken into a number of logical slices called Virtual Machines (VMs).

There are several challenges in Cloud Computing that need to be resolved before exploiting the features of this technology. Some challenges include security issues [5], legal and compliant issues [6], load balancing [5], reliability [5], ownership [5], performance and QOS [6], interoperability issues [6], data management issues [6], multi-platform support [5]. Load balancing is a methodology to distribute workload across multiple computers, or other resources over the network links to achieve optimal resource utilization, maximize throughput, minimum response time, and avoid overload [5].

The load balancing in these Virtual Machines is performed to determine which Virtual Machine is assigned to the next cloudlet [7]. This paper introduces a new VM load balancing algorithm and compares it's performance with the already existing algorithms Round robin and throttled algorithm. The paper primarily talks about implementation of Efficient Service Broker algorithm in which the effective selection of data center for forthcoming request is done, based on their processing capability. This research depicts that how the effective service broker algorithm leads to reduction of load on data centers and minimization in response time felt by users.

II. EXISTING VM LOAD BALANCER

Virtual machine facilitates the abstraction of an OS and Application running on it from the hardware. The interior hardware framework services interrelated to the Clouds is formed in the simulator by a Datacenter element for handling service requests. The particular requests are application elements sandboxed within VM's components. Datacenter object supervise the data center management activities such as VM creation and destruction. It routes the user requests from the User Bases via the Internet to the VMs. The Data Center Controller, uses a VM Load Balancer to regulate which VM should elect the next request for processing. Most common VM Load Balancing algorithms are Round Robin and Throttled algorithms.

2.1. Round Robin

It is one of the simplest scheduling techniques which utilizes the standard of time slices. Here the time is divided into multiple slices, each node is given a particular time slice or time interval i.e. it applies the principle of time scheduling. Each node is provided a quantum and in this quantum the node would perform its operations. The resources of the service provider have been provided to the requesting client; on the basis of this time slice [8].

RISEM - 16
ISBN : 978-93-86171-07-8

The algorithm is very simple however there is an additional load on the scheduler to elect the size of quantum.

2.2 Throttled

In this algorithm the client first requests the load balancer to find a suitable Virtual Machine to perform the required operation. The process first starts by maintaining a list of all the VMs each row is individually indexed to speed up the lookup process. If a match is found on the basis of size and availability of the machine, then the load balancer accepts the request of the client and allocates that VM to the client. If, however there is no VM available that matches the criteria then the load balancer returns -1 and the request is queued.

III. PROPOSED VM LOAD BALANCING ALGORITHM

In this paper a new algorithm for load balancing has been proposed by making some modifications in the existing algorithms. The new algorithm has been simulated on Cloud Analyst and the results have been compared with two existing algorithms: round robin and throttled.

The Proposed Load balancing algorithm has been divided into three parts. The first phase is the initialization phase. In the first phase, the expected response time of each VM is to be found. In second Phase find the efficient VM, in Last Phase return the ID of efficient VM.

- This algorithm finds the expected response time of each Virtual machine.
- When a request comes to allocate a new VM from the Data Center Controller, the algorithm find the most effective VM (effective VM having least loaded and minimum expected response time) for allocation.
- It stores the ID of the efficient VM to the Datacenter Controller.
- Datacenter Controller declare the new allocation
- The algorithm updates the allocation table and expands the allocation count for that VM.
- When the VM completes the processing, the request, the Data Center Controller receives the response. The data center controller notifies the algorithm, for the VM de-allocation.

The proposed algorithm finds the expected Response Time of each Virtual Machine at the Datacenter controller because virtual machine can be of heterogeneous platform, the expected response time can be found with the help of the following formulas.

ResponseTime = Fint – Arrt (1)

Where, Arrt is the arrival time of user request and Fint is the finish time of user request after servicing the request at datacenter the result will be transmitted at the requested UserBase. So the transmission delay can be determined using the following formulas

TDelay = Tlatency + ResponseTime (2)

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

RISEM - 16
ISBN : 978-93-86171-07-8

Destination to source and Response Time is the time taken to service the request at the datacenter.

IV. SIMULATION

The proposed algorithm is implemented through simulation package CloudSim[9]. The application is deployed in one data center having 50 virtual machines (with 1024Mb of memory in all VM running on physical processors capable of speeds of 100 MIPS) and parameter Values are as under.

Table 1: Parameter Value

Parameter	**Value**
Data Center OS	Linux
VM Memory	1024MB
Datacenter Architecture	X86
Service Broker Policy	Optimize response Time
VM Bandwidth	1000

V. RESULTS AND DISCUSSIONS

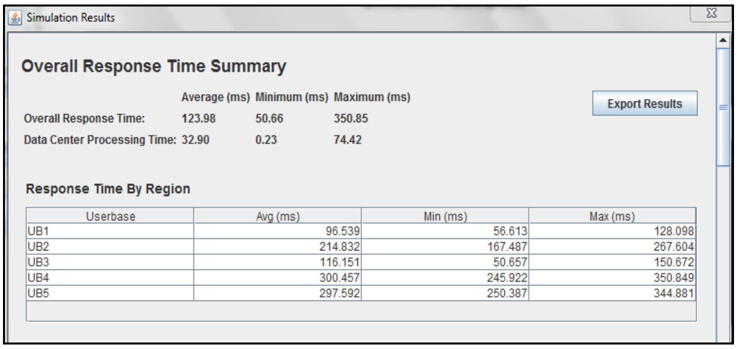
The simulation has been done on the CloudSim toolkit and the results have been compared for the three algorithms discussed above. We have applied the above defined configuration for each load balancing policy one by one and depending on that the conclusions have been drawn based on the the metrics overall response time and data center processing time. Overall response time computed by the CloudSim for each loading policy has been shown below.

Round Robin Algorithm: Figure 1 shows the data center request processing time and Figure 2 shows the overall response time of round robin algorithm when implemented with CloudSim.

Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	20.499	0.24	39.879
DC2	56.296	0.233	74.417

Figure 1: Data Center request servicing time for round robin algorithm



Simulation Results

Overall Response Time Summary

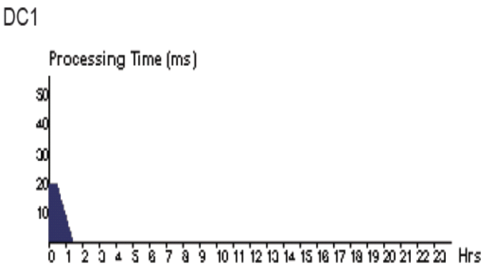
	Average (ms)	Minimum (ms)	Maximum (ms)
Overall Response Time:	123.98	50.66	350.85
Data Center Processing Time:	32.90	0.23	74.42

Export Results

Response Time By Region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	96.539	56.613	128.098
UB2	214.832	167.487	267.604
UB3	116.151	50.657	150.672
UB4	300.457	245.922	350.849
UB5	297.592	250.387	344.881

Figure 2: Overall response time for round robin algorithm



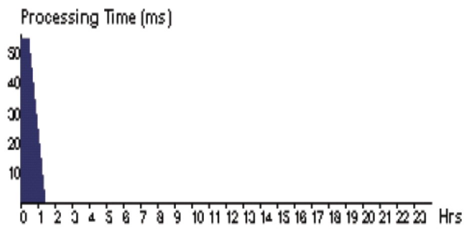
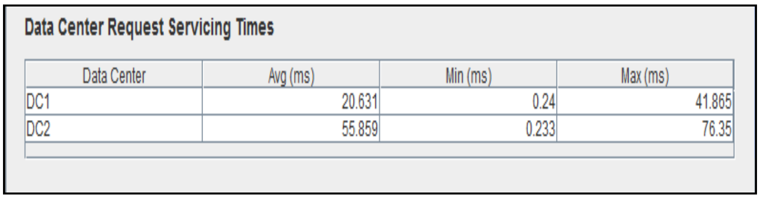


Figure 3(a): Datacenter1 processing time

Figure 3(b): Datacenter 2 processing time

The processing time of datacenter 1 and 2 is shown in Figure 3(a) and (b) respectively.

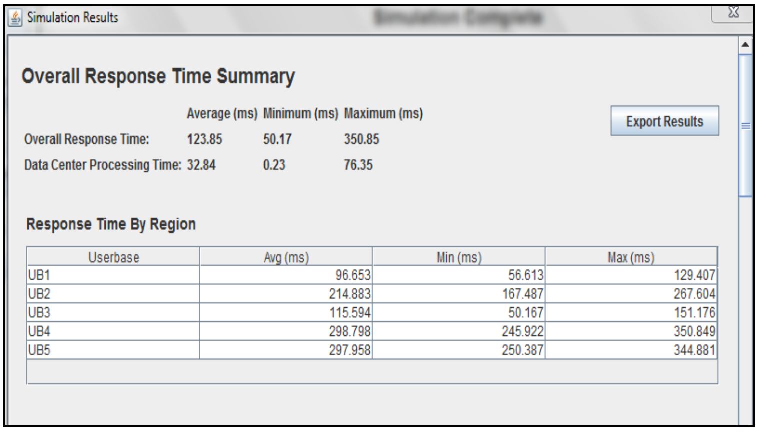
Throttled Algorithm: Figure 4 shows the data center request processing time and Figure 5 shows the overall response time of throttled algorithm when implemented with CloudSim.



Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	20.631	0.24	41.865
DC2	55.859	0.233	76.35

Figure 4: Data Center request servicing time for throttled algorithm



Simulation Results

Overall Response Time Summary

	Average (ms)	Minimum (ms)	Maximum (ms)
Overall Response Time:	123.85	50.17	350.85
Data Center Processing Time:	32.84	0.23	76.35

Export Results

Response Time By Region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	96.653	56.613	129.407
UB2	214.883	167.487	267.604
UB3	115.594	50.167	151.176
UB4	298.798	245.922	350.849
UB5	297.958	250.387	344.881

Figure 5: Overall response time for throttled algorithm

RISEM - 16
ISBN : 978-93-86171-07-8

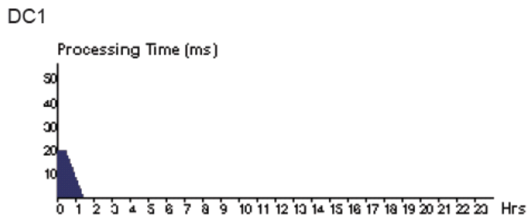


Figure 6(a): Datacenter 1 processing time

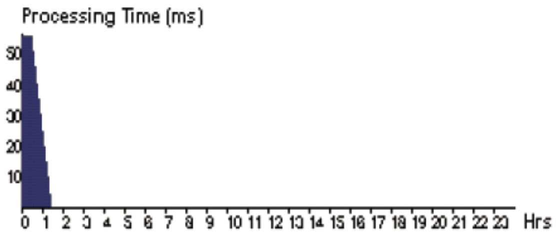


Figure 6(b): Datacenter 2 processing time

The processing time of datacenter 1 and 2 is shown in Figure 6(a) and (b) respectively.

VM Load Balancing Algorithm: Figure 7 shows the data center request processing time and Figure 8 shows the overall response time of throttled algorithm when implemented with CloudSim.

Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	10.575	0.24	31.086
DC2	28.659	0.233	65.893

Figure 7: Data Center request servicing time for VM load balancing algorithm

Simulation Results

Overall Response Time Summary

	Average (ms)	Minimum (ms)	Maximum (ms)
Overall Response Time:	109.54	50.49	350.85
Data Center Processing Time:	16.84	0.23	65.89

Export Results

Response Time By Region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	87.267	56.613	122.714
UB2	214.881	167.487	267.604
UB3	87.181	50.49	138.733
UB4	300.346	245.922	350.849
UB5	298.687	250.387	344.881

Figure 8: Overall response time for VM load balancing algorithm

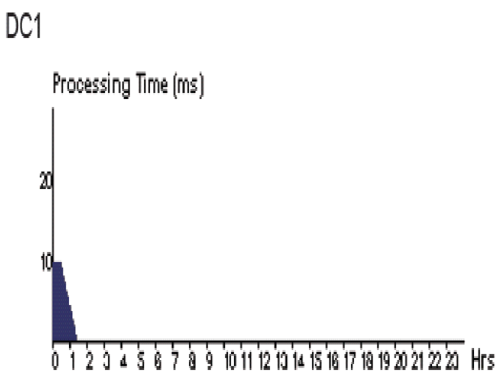


Figure 9(a): Datacenter1 processing time

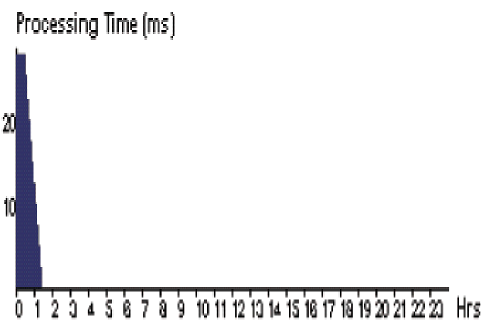


Figure 9(b): Datacenter 2 processing time

The processing time of datacenter 1 and 2 is shown in Figure 9(a) and (b) respectively.

From the Figures above it can be seen that average processing time at datacenter and response time of the query generated by the user base in Round robin scheduling is quiet high compared to the Throttled policy and VM load balancing policy

Similarly, in case of throttled load balancing policy time taken to process query at datacenter and the overall response time of user base is quiet better than Round robin scheduling.

The proposed VM Load balancing has best performance compared to other all loads balancing policy.

VI. CONCLUSION

In this paper a new VM load balancing algorithm was proposed and then implemented in CloudSim cloud computing environment using java language. Proposed algorithm finds the expected response time of each resource (VM) plus it sends the ID of virtual machine having minimum response time to the data center controller in consequence of allocation to the new request. According to this work we conclude that if we select an efficient virtual machine then it effects the complete performance of the cloud Environment and also decreases the average response time.

REFERENCES

[1] Christopher S Yoo. Cloud computing: Architectural and policy implications. Review of Industrial Organization, 38(4):405–421, 2011.

[2] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia. Above the Clouds: A Berkeley View of Cloud computing. Technical Report No. UCB/EECS-2009-28, University of California at Berkley, USA, Feb. 10, 2009.

[3] A. Weiss. Computing in the clouds. NetWorker, 11(4):16–25, Dec. 2007.

[4] R. Buyya, C. S. Y eo, and S. V enugopal. Market- oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications, 2008.

[5] Suriya Begum, Dr. Prashanth C.S.R. "Review of Load Balancing in Cloud Computing." International Journal of Computer Science Issues-IJCSI (January 2013) 343-352.

[6] Raza Abbas Haidri, C. P. Katti, P. C. Saxena. "A Load Balancing Strategy for Cloud Computing Environment." International Conference on Signal Propagation and Computer Technology (ICSPCT)IEEE (July 2014) 636-641.

[7] Bhathiya Wickremasinghe and Rajkumar Buyya. Cloud- analyst: A cloudsim-based tool for modelling and analysis of large scale cloud computing environments. MEDC Project Report, 2009.

[8] Sandeep Sharma, Sarabjit Singh, Meenakshi Sharma "Performance Analysis of Load Balancing Algorithms", World Academy of Science, Engineering and Technology, 38, 2008 pp. 269- 272.

[9] Rodrigo N. Calheiros, Rajiv Ranjan, César A. F. De Rose, Rajkumar Buyya, "Modeling and Simulation of Scalable Cloud Computing Environment and CloudSim Toolkit: Challenges and opportunities", International Conference on high performance Computing and simulation, 21-24 June 2009, pp. 1-11.