ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse83942020.pdf
https://doi.org/10.30534/ijatcse/2020/83942020

A New Load Balancing Algorithm for Better Resource Utilization in Cloud Computing Environment

Sanjiv Kumar Grewal[1], Dr.Neeraj Mangla[2]
[1]Research Scholar, CSE, MMDU, Mullana,Ambala, India, sanjeev.grewal1990@gmail.com
[2]Associate Professor, CSE, MMDU, Mullana,,Ambala, India, neerajmangla@mmumullana.org

ABSTRACT
Cloud computing is a developing concept that has been used in every sector either government or private sector. Powerful data centres are used on clouds to handles the large number of requests. Dynamic pool of requests and virtualization has been provided by the cloud. To handle large number of requests, a proper load balancing technique is required that can equally share the loads among data centres. In this paper, a novel load balancing algorithm is proposed that will help in efficient utilization of resources. Set the priority of each request and Virtual Machines (VM) based on some predefined parameters. Experimental result shows that proposed technique is more efficient and less time consuming.

Key words: Cloud computing, Load balancing, Cloud analyst, Cloud-sim, Resource utilization.

1. INTRODUCTION

Cloud computing is the innovation that offers types of assistance to its customers "on-request". Cloud computing has the ability to support the users on request. The cloud is a common system, which is utilized by the users to store information and run programming. Cloud Service provider (CSP) manages the elements on the cloud. The cloud suppliers offer "Administrations" to the customer however not the artefact. So, the customer can just utilize administrations gave however they don't claim the assets.

Cloud has the computing framework as well as software manage the activities. Customer and Cloud are attached through web. Customer creates requests and sends it by means of web to the Service provider. Cloud is liable for executing the request of the client. The whole setup for the request execution is the obligation of cloud service provider or cloud administration. [1]

Progressively, any number of requests can be processed by the cloud and related software deal with the execution of all the assignments. So, related algorithm of load balancing should be used so that proper utilization of resources can be done [2].

Appearance of every incoming request is entirely unpredictable. So, number of resources can be scaled up or down on the cloud. In this way, customer of the cloud isn't made a big deal about the load. Figure 1 shows the architecture of cloud computing environment [3].

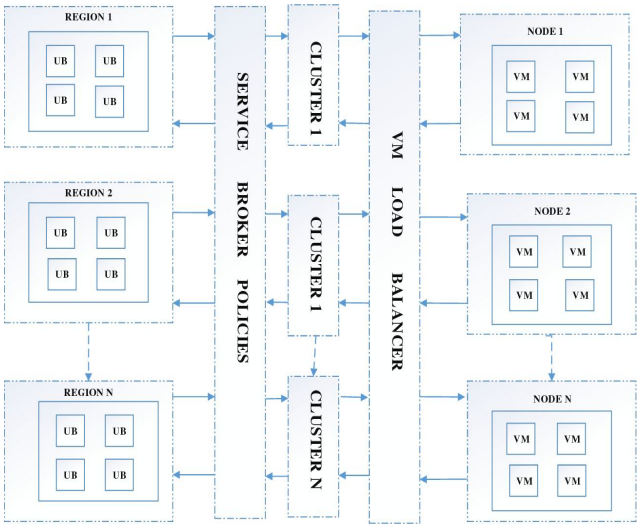

Figure 1: Cloud Figuring Architecture

Load balancing in a framework or system is the way toward circulating task over the processing parts of the framework with the end goal that all assets/resources of the framework are effectively used. The presentation of the system diminishes without load balancing technique on the cloud [4]. Load balancing is typically implemented at IaaS level. There are two points of view for usage of load balancer in distributed cloud. First is the specialized point of view, which deals with execution time, responsiveness and accessibility after usage of load balancer. Second viewpoint is the Business point of view, which deals with expense of extra equipment required for load balancer, execution of the load balancer etc [5].

Essentially, two methodologies used in load balancing algorithms which are as follows: Static and dynamic methodologies. Static methodology thinks about the past condition of the system. Round robin load balancing approach is one genuine model for this type which separates the resources to the incoming tasks on basis of time. These kinds of approaches are not so much popular generally because it requires the earlier information [6]. On the other hand,

dynamic methodology thinks about the present status of the system. Artificial Ant Colony Search is an example of dynamic load balancing algorithm [7].

This paper includes a new load balancing algorithm that will help in efficient utilization of resources. A request from user base arrives at datacenter. First thing is to calculate the size of request, execution time and waiting time of each request (cloudlet). Set the priorities of each request based on their execution time and waiting time and schedule them accordingly. All virtual machines (VMs) are assigned with a weight based on some parameters (Ram, bandwidth, no of processors and processing speed). Sort all VMs in descending order based on their weight. A VM with more weight has the highest priority and assign the request to highest priority VM. If highest priority VM is overloaded then assign the request to next highest weighted VM and continuing this process until all request has been processed.

Organization of rest of the papers is as follows: Section 2 includes the related work that has been completed. Section 3 discusses the methodology that will help in designing the proposed load balancing algorithm. Section 4 elaborates simulation results and discussion. Last section includes conclusion and future work.

2. RELATED WORK

B Preetjiet. al. [8] proposed a framework and made it ready for the green computing by distributing the virtual machine dependent on the load. Presentation of the calculation is investigated utilizing Cloud-Sim test system .The experimental result guarantees that all VM have equal work on server and prone to spoofing activities as compared to other techniques.

G.S. Singh & T. Vivek [9] proposed a hybrid algorithm (HACOBEE), which is the ideal answer for load balancing among hubs in a unique cloud condition that utilizes blend of 2 methodologies for example Artificial Bee Colony (ABC) and Ant Colony Optimization (ACO) approach for workloads dispersion among hubs of a cloud. In Ant Colony Optimization (ACO), ants constantly update a pheromone table while pushing ahead and in reverse from home to food source and the other way around. Best arrangement compares to most extreme pheromone store on the way. In Artificial Bee Colony (ABC), honey bees help in looking through food sources and best source is discovered computing the wellness esteem. In Mixture approach both ACO and ABC give a consolidated ideal answer for load distribution dependent on the pheromone esteems and honey bee fitness values. Hybrid methodology smoothens the framework working at ordinary just as during high workloads hours. Both ACO and HACOBEE algorithms have been utilized on comparative informational indexes where HACOBEE demonstrated better response times.

S. Roy et. al. [10] introduced a Fuzzy Based Least Response Time (FLRT) dynamic load balancing algorithm, which is successful for hard contribution from various frameworks. Consequently, FLRT is a new worldview which can choose closest neighbour edge server from client's present area where response time and load of the edge server is most minimal.

M. Vanithaa & P. Marikkannu [11] proposed a load balancing techniquewhich adjusts the load dependent on the need of the virtual machines and condition of the virtual machines utilized in the cloud.

P. Yadav & Prof.A. Vyas [12] proposed a cloud partitioning approach that stretched out for successful load balancing and resource optimization. Other than the perfect refresh period which decides changes in the condition of segments of data centre for advanced resource use. Cloud is thesimulation framework used to construct a model application for showing the confirmation of idea. Outcome of the framework demonstrated that the proposed arrangement improves execution, throughput and use of cloud resources.

A.N. Singh & S. Prakash [13] proposed a methodology in which following weight factors eg. Physical memory, transfer speed, quantity of processor, and speed of processor etc. are used for assigning the virtual machine. Subsequent to ascertaining the heaviness of each VM, and select those VM that is high in weight and accessible for task execution. Outcomes of work checked through Cloud Analyst that is Cloud Sim based test system and outcome shows proposed system is better than existing ones.

K. Ravindranathet. al. [14] proposed a novel algorithm which utilizes the resources proficiently by partitioning the tasks among them (utilizes exhibit for the productivity of assets) in a superior way. This algorithm adjusts the tasks as well as causes them to execute in a superior manner by allocating to the resources put together up with respect to the execution time of the task.

S. Afzal& G. Kavitha [15] proposed a proficient load balancing algorithm for effective usage of resources in distributed cloud computing condition and to compare the performance of proposed algorithms with other algorithms. The proposed algorithm consider size of cloudlet, expected completion time of assignments by virtual machine and runtime properties virtual machines to guide's the incoming requeststo virtual machine inunbiased and efficient manner.

A. Pradhanet. al. [16] proposed a hybrid meta-heuristic algorithm with nature inspired Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) approaches. Advanced outcomes achieved in terms of execution when proposed approach assessed on Cloud Analyst Simulation tool.

M. Yadav & S. Gupta [17] proposed aalgorithm that wipes out the downsides of executing a basic cooperative design in distributed computing by presenting an idea of assigning various time slices to individual processes relying upon their needs.

3. METHODOLOGY

3.1 Method for Setting Priority of each Request

In a cloud computing condition, different clients are submitting work demand with possible requirements that is same resource are requesting by multiple clients. For instance in a high execution computational condition which mostly manage scientific simulations for example, monsoon prediction, weather prediction, cyclone simulation, rainfalls emulation which requires enormous amount of registering resources, for example, processors, storage, servers and so forth [18]. Many clients are requesting these computational resources to run their model which is utilized for logical expectations. So at this point it will be issue for cloud administrator to conclude how to allocate the accessible resources among the requested clients.

Initial step of proposed algorithm is to find out the waiting time of all the tasks. Waiting time is determined dependent on the length of an incoming request. The length of cloudlet of each task is mapped in hash table.

Waiting Time = $\sum_{i}^{n} l(i) * t(i)$

Where
n= Number of CloudLet
l= Length of CloudLet
t= Execution time of VM
i = Whole number beginning from 0

The waiting time of each virtual machine is determined by including the result of the length of CloudLet by the execution time .This condition used to compute the waiting time of the virtual machine when it gets the request of length l. The execution time is determined on the basis of CloudLet size. To ascertain the execution time of the pay task is determined by partitioning size of CloudLet by the virtual machine power.

Execution Time = s/VMp
Where
S =CloudLet Size
VMp= Virtual Machines Force

s is estimated in MI (Million Guidelines) and Virtual Machine Limit is estimated on MIPS (million guidance for every seconds).

At the point when a request shows up at datacenter, figure the size of request, expected execution time and waiting time of each request(cloudlet). Set the needs of each request dependent on their execution time and waiting time and schedule them in like manner.

3.2 Method for Assigning Weight to Virtual Machines (VMs)

Compute the weight W of each virtual machine utilizing the following formula:

$W= \alpha_1 [t_1+t_2+....] + \alpha_2 [r_1+r_2+....] + \alpha_3 [np_1+np_2+....] + \alpha_4 [ps_1+ps_2+....]$

where α_1, α_2, α_3, and α_4 are the predefined weights of comparing system boundary, t_1,t_2 are transfer speed r_1, r_2 are RAM , np_1, np_2 are the used to represent processors quantity, and ps_1, ps_2 represents processor speed of the relating framework. Choosing the weight is a very typical task. In this methodology, weights are considered in the accompanying way: $\alpha_1 = 0.1$, $\alpha_2 = 0.2$, $\alpha_3 = 0.4$ and $\alpha_4 = 0.3$ so that their summation got one.

3.3 Proposed Algorithm

Figure 2 shows the pictorial representation of the various steps followed in the proposed algorithm and these steps are as follows:

step 1. Request from user arrives at the data center.
a) Get the size or length of the request (cloudlet).
b) Calculate the execution time and waiting time of each request.
step 2. Set the priority of each request depending on the parameters (execution time and waiting time).
step 3. Assign the weight of each VM based on equation 1.
step 4. Sort all VMs in descending order based on their weight.
step 5. Assign the highest priority request to VM that have more weight.
step 6. If (VM== Overloaded) then
Go to step 5.
Else
Process the request
step 7. If (any request to process) then
Go to step 2.
Else
Stop processing.

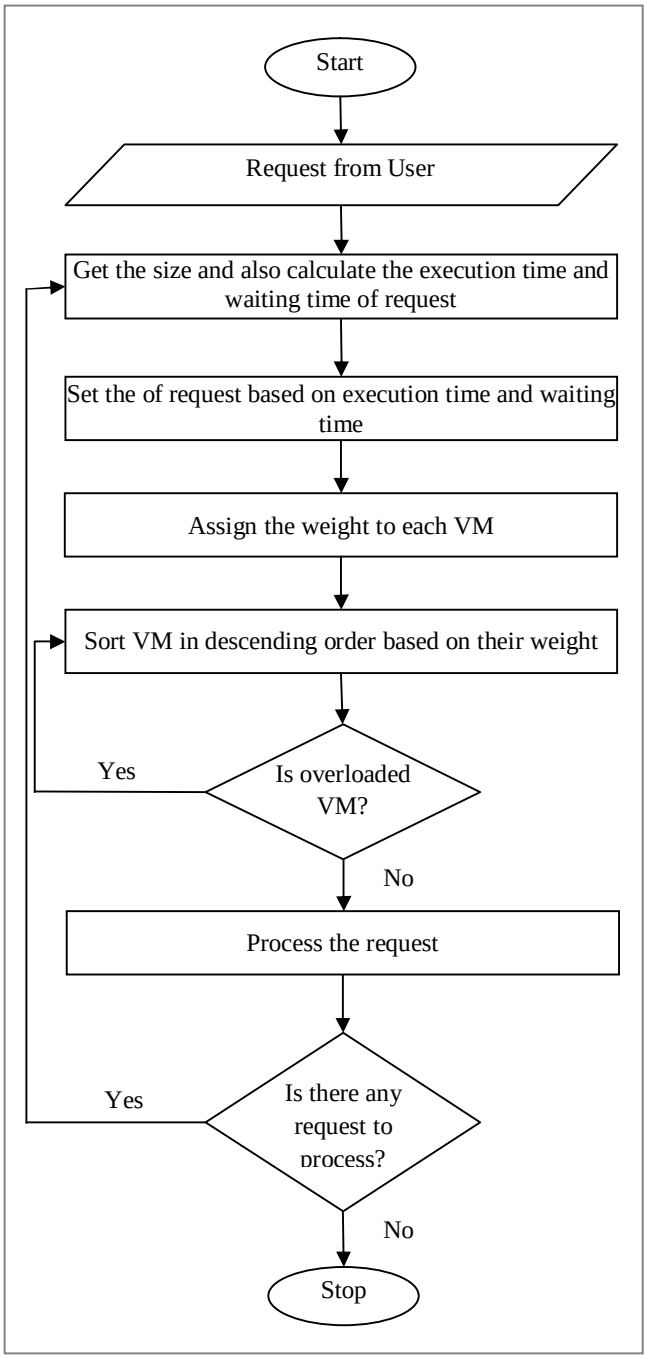

Figure 2:.Flow Chart of Proposed Algorithm

4. RESULT AND DISCUSSION

Cloud Analyst tool is used for the simulation and result analysis. This is a GUI based tool and developed on Cloud-Sim architecture. Table 1 shows Parameter and its set values for duration of 60 minutes.

Table 1: Parameter Values of Cloud Analyst Tool

PARAMETERS	VALUE
Number of Datacenters	3
User Bases	5
Service Broker Policy	Optimise Response Time
Physical H/w units (Physicals hosts) in each data center	4
Total No. of VMs	DC1-10,DC2-10, DC3 - 10
Processor Speed	2000 - 10000
VM Memory (RAM)	512 - 1024
VM Policy	Time Shared

Figure 3 and Figure 4 shows the parameters for the user base configuration, application deployment configuration, and data center configuration respectively. Five different region of the world defined for the location of user bases. Three data centres can handle the requests coming from these user bases. One data center (DC1) is located in region 0, second (DC2) & third (DC3) in region 2 &4 respectively. Numbers of VM on each DC are ten. Configuration of various component of Cloud analyst tool needs to be done to analyse various load balancing techniques. Figure 5 & 6 shows the parameter set for advanced configuration and out screen showing different regions respectively.

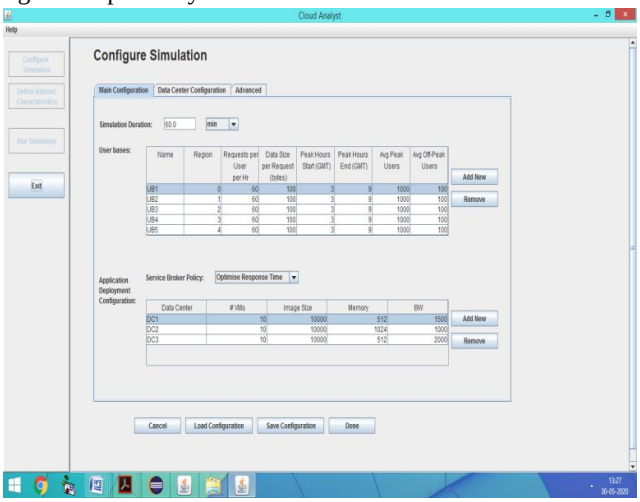

Figure 3: User Bases and Application Deployment Configuration of Simulator

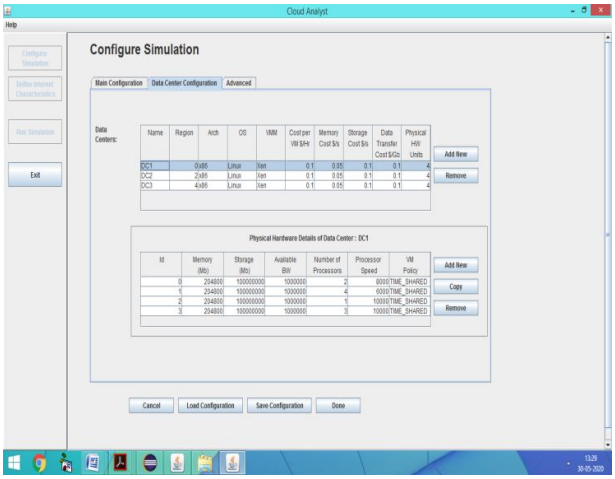

Figure 4: Data Center Configuration of Simulator

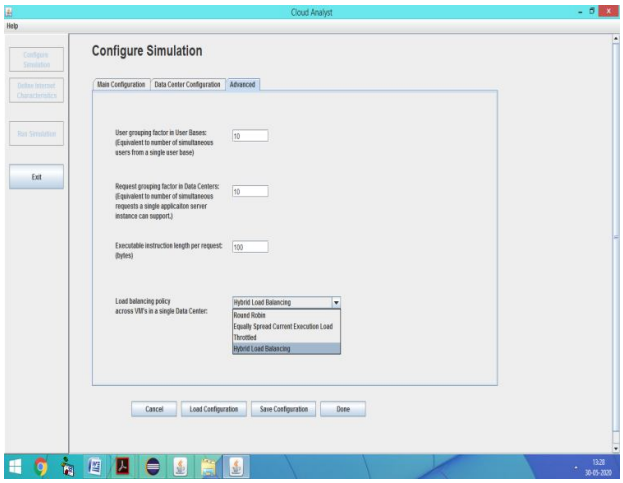

Figure 5: Advance Configuration

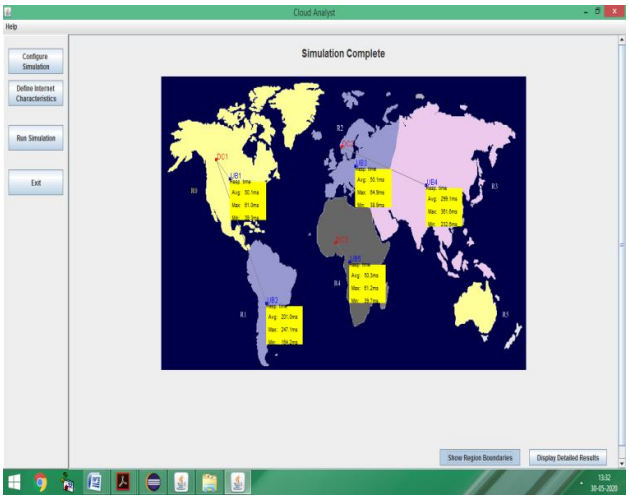

Figure 6: Output Screen of Simulator

Cloud analyst tool is used to calculate overall Response Time (RT) and Data Center Processing Time (DCPT) for Round Robin (RR), Throttled, Equally Spread Current Execution (ESCE) and proposed load balancing algorithm. Figure 7, 8, 9 and 10 shows the same respectively.

Overall Response Time Summary

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	130.32	38.85	361.61
Data Center processing time:	0.48	0.01	1.27

Figure 7: RT & DCPT Summary of RR Algorithm

Overall Response Time Summary

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	130.03	38.85	361.61
Data Center processing time:	0.45	0.01	1.27

Figure 8: RT & DCPT Summary of Throttled Algorithm

Overall Response Time Summary

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	130.02	38.85	361.61
Data Center processing time:	0.45	0.01	1.27

Figure 9: RT & DCPT Summary of ESCE Algorithm

Overall Response Time Summary

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	129.94	38.85	361.61
Data Center processing time:	0.44	0.01	1.27

Figure 10: RT & DCPT of Proposed Algorithm

Below table 2 shows the comparative analysis of proposed algorithm with the existing algorithms and the graphical representation of analysis are shown in figures 11 & 12. Proposed algorithm has less response and data processing time. Hence it is more efficient than the existing algorithm.

Table 2: Comparative Analysis of various Algorithms

Algorithms	**Avg Response Time (ms)**	**Avg Data Processing Time (ms)**
Round Robin	130.32	0.48
Throttled	130.03	0.45
ESCE	130.02	0.45
Proposed	129.94	0.44

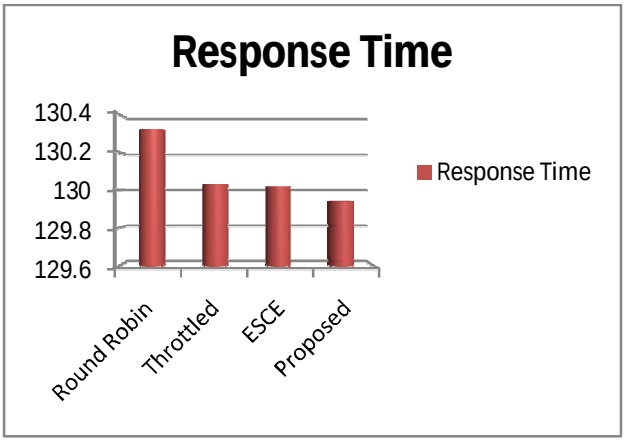

Figure 11:.RT Graph of Various Load Balancing Algorithms

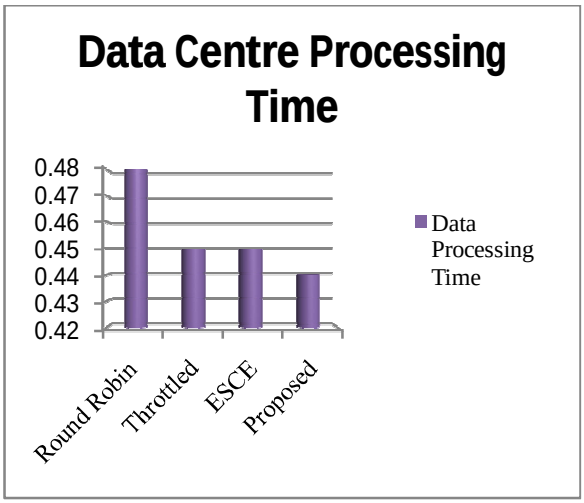

Figure 12:.DCPT Graph of Various Load Balancing Algorithms

5. CONCLUSION

In this paper, a novel load balancing algorithm is proposed that will help in efficient utilization of resources. Set the priorities of each request and VM based on some parameters (Ram, bandwidth, no of processors and processing speed). Sort all VMs in descending order based on their weight. A VM with more weight has the highest priority and assign the request to highest priority VM. If highest priority VM is overloaded then assign the request to next highest weighted VM and continuing this process until all request has been processed. Experimental results of the proposed work shows its efficiency. Future work will be focused on the development of new scheduling technique to process the request efficiently.

REFERENCES

[1] Rabi Prasad Padhy, ManasRanjanPatra,"Evolution of Cloud Computing and Enabling Technologies", International Journal of Cloud Computing and Services Science, Vol.1, No.4, pp 182-198, October 2012. https://doi.org/10.11591/closer.v1i4.1216

[2] S. Karimunnisa and Dr.V. S. Kompalli, "Cloud Computing: Review on Recent Research Progress and Issues," ternational Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), Vol. 8, No. 2, pp. 216-223, 2019.

[3] Shu-Ching Wang, Kuo-Qin Yan, Wen-Pin Liao, Shun-Sheng Wang, "Towards a Load Balancing in a Three-level Cloud Computing Network", International Conference on Computer Science and Information Technology [ICCSIT], Vol. 1, pp 108-113, 9-11 July 2010.

[4] Amritpal Singh, Rajeev kumarbedi, Sunil kumargupta, "Design and implementation of an efficient scheduling algorithm for cloud balancing in Cloud Computing", International Journal of Emerging Trends and Technology in computer Science [IJETTCS], Vol. 3, No 1, Jan-Feb 2014.

[5] A Khiyaita, M Zbakh, H El Bakkali, D El Kettani, "Load balancing cloud computing: state of art," 2nd National Days of Network Security and Systems [JNS2], 20-21 April 2012, Morocco, pp 106-109, DOI: 10.1109/JNS2.2012.6249253.

[6] Ruixia Tong, Xiongfeng Zhu, "A Load Balancing Strategy Based on the Combination of Static and Dynamic", 2nd International Workshop in Database Technology and Applications (DBTA), 27- 28 Nov 2010, Wuhan, pp 1-4, DOI: 10.1109/DBTA.2010.5658951.

[7] Jitendra Bhatia, Tirth Patel, HarshalTrivedi, VishrutMajmudar, "HTV Dynamic Load Balancing Algorithm for Virtual Machine Instances in Cloud", International Symposium on Cloud and Services Computing, 17-18 Dec 2012, Mangalore, pp 15-20, DOI: 10.1109/ISCOS.2012.25.

[8] B Preethi, C. Kamalanathan, Dr. S.M Ramesh, S. Shanmathi and P SathiyaBama, "Optimization Of Resources in Cloud Computing Using Effective Load Balancing Algorithms," International Advanced Research Journal in Science, Engineering and Technology, vol. 1, no. 1, pp. 20-22, Dec. 2014.

[9] G. S. Singh and T. Vivek, "Implementation of A Hybrid Load Balancing Algorithm for Cloud Computing," 2nd International Conference on Sciences, Technology and Management, DU, New Delhi, pp. 173-182, 2015.

[10] S. Roy, R. Bose and D. Sarddar, "Fuzzy Based Dynamic Load Balancing Scheme for Efficient Edge Server Selection in Cloud-Oriented Content Delivery Network using Voronoi Diagram,"IEEE International Advance Computing Conference (IACC), pp. 828-833, 2015.

[11] M. Vanithaa and P. Marikkannu, "Effective resource utilization in cloud environment through a dynamic well-organized load balancing algorithm for virtual machines," Computer and Electrical Enginineering, pp. 1-10, 2016.

[12] P. Yadav and Prof.A. Vyas,"Design and Implementation of Hybrid Algorithm for Load Balancing in Cloud Computing Environment," International Journal of Innovative Research in Computer and Communication Engineering, vol. 5, no. 6, pp. 11138-11143, 2017.

[13] A.N. Singh and S. Prakash, "WAMLB: Weighted Active Monitoring Load Balancing in Cloud Computing," Big Data Analytics, Advances in Intelligent Systems and Computing 654, Singapore, pp. 677-685, 2018.

[14] K. Ravindranath, J.D. Gadhia and K.S. Reddy, "A New Scheme For Maximum Utilization Of Resources In Load Balancing Algorithm's in Cloud Computing Architecture," International Journal of Recent Technology and Engineering (IJRTE), vol. 7, no. 6S, pp.567-569, 2019.

[15] S. Afzal and G. Kavitha, "Load balancing in cloud computing – A hierarchical taxonomical classification," Journal of Cloud Computing: Advances, Systems and Applications, pp. 1-24, 2019.

[16] A. Pradhan, S.K. Bisoy and P.K. Mallick,"Load Balancing in Cloud Computing: A Survey," Proceedings of IEPCCT, Singapore, pp. 99-112, 2020.

[17] M. Yadav and S. Gupta, "Hybrid meta-heuristic VM load balancing optimization approach," Journal of Information and Optimization Sciences, Vol. 41, No. 2, pp. 577–586, 2020.

[18] B. Mukhopadhyay, Dr. R. Bose and Dr. S. Roy, "A Novel Approach to Load Balancing and Cloud Computing Security using SSL in IaaS Environment," International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), vol. 9, No. 2, pp. 2363-2370, 2020.
https://doi.org/10.30534/ijatcse/2020/221922020