Creating Elasticity with Enhanced Weighted Optimization Load Balancing Algorithm in Cloud Computing

Neha Tyagi[1], Ajay Rana[2], Vineet Kansal[3]

[1]Ph.D scholar, Amity University, Noida Campus & Assistant Professor , IET, Lucknow, India
[3]G. L Bajaj Institute of Technology & Management, Greater Noida, India
[2]Amity University, Noida, India
[1]nehacs1988@gmail.com, [2]ajay_rana@amity.edu, [3]vineetkansal@yahoo.com

Abstract: This paper embraces an arrangement of rules for load balancing to improvise the general execution along with viability in assorted environment of cloud computing. The paper advocates a combination set of rules in light of priority and batch rules, taking points of interest of weighted optimization algorithm and similarly spread current execution algorithms.

These algorithm reflects the records of currentresource and the capability issue of Central Processing Unit to acquire the targets. Cloud Analyst simulator has been used to evaluate and compare the fused algorithm with different algorithms. The results demonstrated progress in migration time, reaction time and scalability, availability & reliability of resources with energy-consumption and throughput via considering the modern-day aid records and the Central Processing Unit potential factor in comparison with different algorithms.

Keywords: Virtualization, Virtual Machines, Distributed Cloud Computing, Load

I. INTRODUCTION

In Current Scenario, Cloud computing has taken shape of a new modern storing model which has been developed due to advancements in the internet. Distributed cloud computing is considered to be an advancement of disseminated frameworks, which in a heterogeneous situation offers a quick and on-request wide scope of administrations [1]. Theseconditions of heterogeneous nature implies to diverse equipment attributes which includes the equipment like storage, CPU, and memory among others [2]. Any entrepreneur may begin orgrow the framework and bringing down working and upkeeping costs without puting extra resources. Now the businesses has moved to a far extent from in the past storing its information on desktops to the Cloud servers situated very far distances from the place of business retrieving them large amounts of information [3]. It has the ability to utilize the energy of the internet with the local maintained assets that can be accessible remotely from anywhere in the world as desired by the businesses, accordingly giving the savvy answer for the vast majority of the genuine requirements [3-7].

II. TECHNIQUE OF LOAD BALANCING

Under theTechnique of Load balancing, there is a reassignment of total load. It is reassigned to the individual nodes of the collective cloud arrangement. This technique help in getting better of both dimensions namely the utilization of resources and time response requests. It tries to avoid sending signals to nodes which are overloaded and tries to make use of underutilized nodes. As a result due to load balancing all the nodes receives equally distributed work [7]**. A**chieving the optimal utilization of the resources like overall control of all the systems, least response time, and sidestep the overload condition can be summed up as the primary objective of Load Balancing.[8][20]. The heterogeneous environment condition is thought to be a major concern[9-11] [24-26]because such environment is a mixture of resources, so it behaves like a mixed cloud system and has diverse elements and diverse response times for any process to be executed[10-12] [25, 27]. Concrete load balancing system is the basis for the load balancing in distributed cloud computing. Load balancing varies from conventional load-balancing to concurrent computing. But the activity of the load balancing process as well as the structural design use to be entirely different in distributed cloud computing. It provides ample of new opportunities by being relatively cheaper. Figure 1 shows load balancing in cloud computing [29].

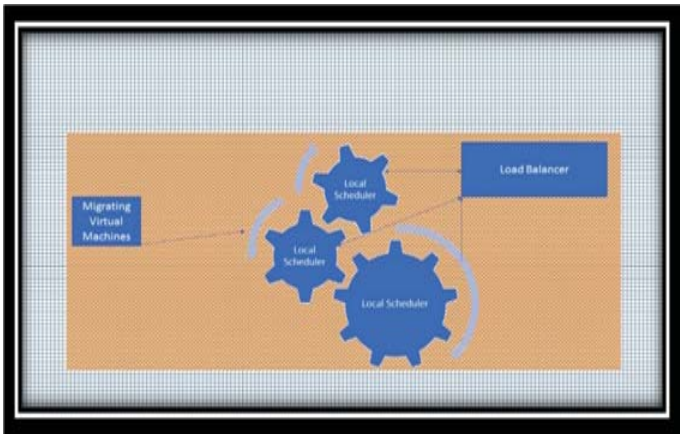

Fig. 1. Load Balancing Model

978-1-5386-9346-9/19/$31.00 ©2019 IEEE

2.1 Existing load adjusting calculations

This segment displays a few of the mainstream load adjusting calculations which need aid utilized within the cloud registering nature's domain. Previously, our work, we are setting off with aggravate investigations for percentage about these calculations what's more analyze them for our worth of effort.

1. Round robin algorithm: It will be viewed as likewise the greater part fundamental and the minimum perplexing planning algorithm [30,31], it utilizes the idea of period quantum what's more every processor detracts a period quantum, the procedures are separated. Between everyone processors concerning illustration seen done figure 2. Every transform is allocated to the processor for a Round structure request. On the methodology doesn't finish for a provided for time, it will make set in those ends. From claiming sitting tight queue, the detriment of this calculation is toward whatever side of the point about time a portion hubs might make. Intensely stacked also how to stay unmoving pulley[8] [20].

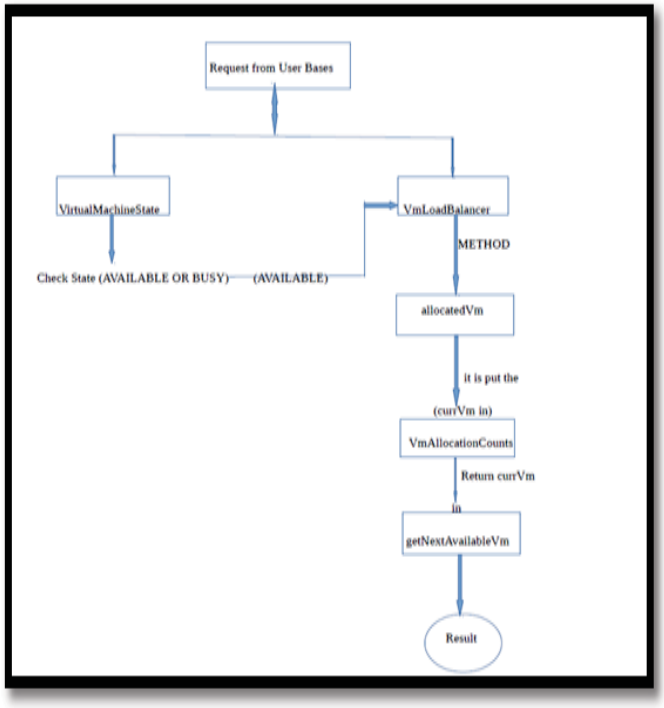

Fig. 2. Round Robin Flowchart

2. Equally Spread Current Execution: Equally Spread Current Execution calculation has been indicated in figure 3. It conveys the load haphazardly eventually the extent what's more exchange the load to that virtual machine which will be delicate. Stacked alternately handle that job not difficult and take securely occasion when what's more provide for expand throughput. It will be spread range technobabble on which those load balancers spread those loads of the particular occupation under control under various. Virtual machines [16].

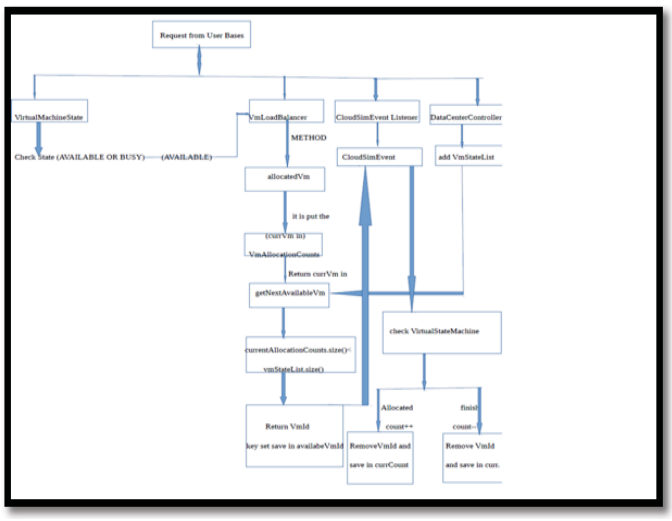

Fig. 3. Equally Spread Current Execution Flowchart

III. PROPOSED ALGORITHM

Those present load party planning calculations for heterogeneous of a processors energy clinched alongside cloud registering nature's domain will be not exceedingly productive [33]. The fundamental objective from claiming this exploration is on attain effective execution in heterogeneous of a processors energy for cloud registering. For this part, we will display the recommended a mixture of weighted-optimization and enhanced weighted optimization algorithm that takes favorable circumstances of both mentioned algorithms. In this Scrutinize we recommended a mixture algorithm that takes favorable circumstances from claiming both irregulars What's more calculations. The irregular calculation which haphazardly selects a virtual machine with the methodology the accepted tasks, cannot necessity perplexing calculation on settle on a choice at it doesn't select the best virtual machine. On the great holders kept all calculation selects those best virtual machines will handle the accepted task, yet the choice methodology needs a percentage complex calculation to figure out the best virtual machine. Those steps that accompanied well fulfill this worth of effort introduced in figure 4.

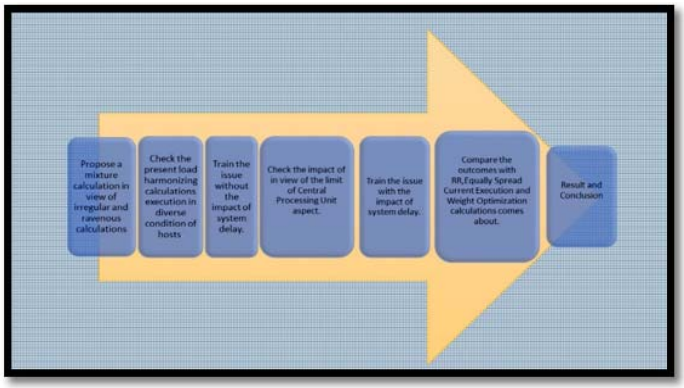

Fig. 4. Process of checking virtualization

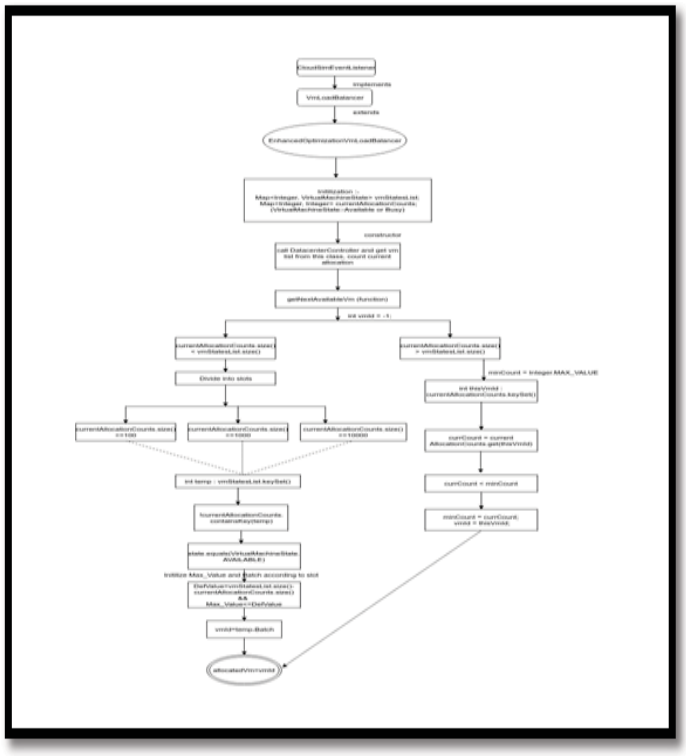

Fig. 5. Enhanced Weighted Optimization Algorithm

Initial we plan those suggested a mixture calculation dependent upon irregular also existing calculations. Those plan methodology incorporates advancement of the model, detail and planning the algorithm, checking the accuracy of Algorithm, What's more examination about the algorithm. Then we actualize all the recommended calculation utilizing Cloud analyst test system. Following that, we test the suggested algorithm utilizing cloud investigator test system. After that, we tried the suggested calculation in a heterogeneous from claiming processors energy without acknowledging system delay. After that, we tried those suggested algorithms previously, heterogeneous for processors force for acknowledging organize delay. At long last, we compared those outcomes of the recommended algorithm for current calculations effects. That algorithm adopts those aspects of randomization furthermore materialistic should make a proficient load adjusting what's more blankets their hindrances. The algorithm acknowledges the current asset data and the central processing unit limit variable should accomplish the destinations. Above mentioned figure 5 indicates the unique understanding for calculation.

3.1 Description of Proposed Algorithm

In the opening point, virtual machines are distributed over hosts as showed by the host capabilities. The biggest quantity of virtual machine is established and no more efficient host contingent upon the hosts central processing unit limit. For instance, in the event we have 5 virtual machines and 3 hosts, where the prime host has one machine and its speed is equal to 10,000, the 2nd host has two machines and the speed of each machine is equal to 10,000, and the 3rd host has three machines and the speed of each machine is equal to 100,000. In this way, the limit of the leading host is one* ten thousand is equal to 10,000, the 2nd host is two* ten thousand is equal to 20,000 and the 3rd host is three* ten thousand is equal to 30,000. So as showed by hosts abilities, the primary host will choose one virtual machine, the 2nd host will choose two virtual machines, and the 3rd host which becomes the biggest limit will choose three virtual machines. In the Next phase, the calculation used another list table to record the present burdens for each virtual machine. Also, which used to check the present burdens for the virtual machine at every cycle, the calculation read the estimate of virtual machine stack from the log table; when the server farm gets a demand from the clients, it sends the demand to the half-breed stock balancer. The half-breed calculation will choose N virtual machines, and eventually, it will pick the current stack for each chose the virtual machine [17-18]. At that point, it will pick a virtual machine that has minimum virtual machine current burdens too, reinforce the virtual machine id to data focus. The data focus will dole out the heap to the chosen virtual machine and refresh the estimate those virtual machines in the list table of current overloads. At long last when the virtual machine wraps up the demand, it will enlighten the server farm to revive its present load esteem.

3.2 Implementation of Proposed Algorithm

Cloud Analyst test system has been used in the present experiment. The experiment interpret test system specifications, for detail, consumer's configuration, data focus design, virtual systems pattern. In the beginning, trials are executed using the qualified method without deliberating the impact of system delay.Then heterogeneous condition of hosts is calculated, as each machine has a unique number of central processing system and speed. Then, the impact of the competence of central processing system point is investigated. At last, we tried the effect of the impact of system wait on the half-breed calculation while considering the capacity of central processing unit factor under the heterogeneous condition. We actualize some of the current load adjusting forecasts, for precedent, Round Robin, Equally Spread Current Execution and Weighted Optimization calculations. At that extent, we performed the crossbreed calculation namely Enhanced Weighted Optimization algorithm. The code of the crossover calculation is an informative buttress.

3.3 Evaluation

There are Diverse dimensions applied to estimate operations. In our stuff in we employed seven dimensions with control the execution similarly as observed:

1. **Migration Time:** Overall migration time is the time from when the process of migration leads from the origin machine until when the target VM obtains supervision and

the origin can be given up. In live migration, these two times
vary.uptime+=entry.getTime()+lastTimeSwitchedOn;dow ntime+=entry.getTime()

2. **Response Time:** The expired time between the finish of an analysis or request on a computer procedure and come out a reply; for illustration, the range of the time between a signal of the demise of an analysis and the publishing of the early quality of the reply at a userterminal. There is still the approach of *observed*responsetime, which is the time a user senses as open input and the end of the response. It is possible for distinguished responsetime to be extremely fast. However, this is not the regular criticism.*Response Time = Finisht - Arrivalt + Transmissiondelay; Arrivalt* means arrival time of user request and *Finisht* means 'finishing time of users request' and *Transmissiondelay* means the 'transmission delay'. However, *Tdelay* can be estimated as; Totaldelay = *Totallatency + Totaltransfer.*Here, ***Totallatency***is the network latency and ***Totaltransfer*** is the time taken in transferring the amount of data from a single request sent by source location to a destination. ***Tlatency*** is taken from the latency matrix held in the Internet characteristics.

3. **Throughput:**
throughPut1=(double)process/(double)timeDiff;
throughPut=throughPut1/100;

4. **Scalability:** scalability = (totAll / (succRes*100));

5. **Availability:** The availability means percentage of time required by a client to access the service. It is given by: (total service time) − (total time for which service was not available)/total service time

6. **Reliability:** (succRes / (totAll*100)); succRes +=entry. getAllocatedMips(); totReq +=entry.getRequestedMips();totAll=totReq + succRes;

7. **Energy-Consumption:**
EnergyconsumptionCostRs.:%.4f,energy*energyCostdatac enter.getPower()

IV. EXPERIMENTS AND RESULTS

The act of the Enhanced weighted optimization algorithm has been determined based on the proceeds of simulation performed in the Cloud Analyst. The classes of the Cloud Analyst simulator have been overridden the written algorithm. In the subsequent interpretations, the Migration Time, Response Time, Throughput, Scalability, Availability, Reliability and Energy-Consumption are investigated in the Round Robin, Equally Spread Current Execution, Weighted Optimization and Enhanced Weighted Optimization algorithms under the combination of heterogeneous and homogeneous activity lengths with heterogeneous resource conditions. Configuration details are given in Figure 8

Fig. 6. Cloud Analyst Regions

V. DISCUSSION & CONCLUSION

We have checked each of the four calculations on in an unexpected way unique .Sim setup documents. In every one of the cases, the proposed approach giving the best outcome if there should be an occurrence of various virtual machines 23, so relocation time 19.06, reaction time 224.26, throughput 100%, versatility 71.38, accessibility 51.28, dependability 89.14, Energy-Consumption 2.55. It shows the best near outcomes with other three Round Robin, Equally Spread Current Execution, Weighted Optimization and Enhanced Weighted Optimization approaches.

REFERENCES

[1] Zhang Z., McCalley J.D., Vishwanathan V. and Honavar V., June. Multiagent system solutions for distributed computing, communications, and data integration needs in the power industry. *Power Eng. Society Gen. Meeting, IEEE* 45-49, (2004).

[2] Kim JK, Shivle S, Siegel HJ, Maciejewski AA, Braun TD, Schneider M, Tideman S, Chitta R, Dilmaghani RB, Joshi R, Kaul A., Dynamically mapping tasks with priorities and multiple deadlines in a heterogeneous environment. *J Parallel Distri. Comp.***67**(2): 154-69, (2007).

[3] Sharma T. and Banga V.K., Efficient and Enhanced Algorithm in Cloud Computing. *Int. J Soft. Comp. Eng.* **3**(1)**:** 385-287, (2013).

[4] Zhang Q., Cheng L. and Boutaba R., Cloud computing: state-of-the-art and research challenges. *J Internet Ser. Appl*. **1:** 7-18, (2010).

[5] Pathak KK, Yadav PS, Tiwari R, Gupta TK. A modified approach for load balancing in cloud computing using extended honey bee algorithm. *Int. J. Research Rev. Eng. Sci. Tech.***1**(3)**:** 2-9 (2012).

[6] Tiwari M., Gautam K., and Katare K., Analysis of Public Cloud Load Balancing using Partitioning Method and Game Theory. *Int. J. Adv. Res. Comp. Sci. Soft. Eng.***4**(2)**:** 807-812 (2014).

[7] Mishra R, Jaiswal A., Ant colony Optimization: A Solution of Load Balancing in Cloud. *Int. J Web Semantic Tech*. **3**(2): 33 (2012).

[8] So K. Cloud computing security issues and challenges. *Int. J. Comp. Net.***3**(5): 247-255 (2011).

[9] Yao, J., & He, J. H. (2012, April). Load balancing strategy of cloud computing based on artificial bee algorithm. In *8th Int. Conf. Comp. Tech. Inf. Manage. IEEE,*1: 185-189 2012.

[10] Shameem P.M. and Shaji R.S. A Methodological Survey on Load Balancing Techniques in Cloud Computing. J. *Eng. Tech.* **4**(5): 3801-3812 (2013).

[11] Behal V. and Kumar A. Cloud Computing: Performance Analysis of Load Balancing Algorithms In Cloud Heterogeneous Environment. *The Next Gen. Infor. Tech. Summit (Confluence). Noida: IEEE,* 200 - 205 (2014).

[12] Kaushik V.K., Sharma H.K. and Gopalani D. Load Balancing in Cloud Computing Using High Level Fragmentation of Dataset. In *Int. Conf. Cloud, Big Data and Trus,t* 118-126 (2013).

[13] Pathak K.K., Yadav P.S., Tiwari R. and Gupta T., A Modified Approach for Load Balancing in Cloud Computing Using Extended Honey Bee Algorithm. *Int. J. Res. Rev. Eng. Sci. Tech.***1**(3): 12-19 (2012).

[14] Mehta R., Yask P. and Harshal T., Architecture for Distributing Load Dynamically in Cloud Using Server Performance Analysis Under Bursty Workloads. **1**(9): (2012).

[15] Dave S. and Maheta P., Utilizing Round Robin Concept for Load Balancing Algorithm at Virtual Machine Level in Cloud Environment. *Int. J. Comp. Appl.***49**(4): (2014).

[16] Mohialdeen I.A., Comparative Study of Scheduling Algorithms. *J. Comp. Sci.***2**(9): 252-263 (2013).

[17] Cormen TH, Leiserson CE, Rivest RL, Stein C, Introduction to algorithms third edition. *MIT Press,***23**: 631-8 (2009).

[18] Singh A, Bedi R, Gupta S. Design and implementation of an Efficient Scheduling algorithm for load balancing in Cloud Computing. *Int. J. Emerging Trends & Tech. Comp. Sci.***3**(1):- (2014).