

## Weighted Optimization Load Balancing Algorithm in Virtualization

<sup>1</sup>Neha Tyagi, <sup>2</sup>Ajay Rana and <sup>3</sup>Vineet Kansal

<sup>1</sup>G.L. Bajaj Institute of Technology and Management, Amity University,  
Noida Campus, Greater Noida, India

<sup>2</sup>Amity University, Noida, India

<sup>3</sup>Institute of Engineering and Technology, Lucknow, India  
nehacs1988@gmail.com

**Abstract:** Today, there is trend for distributed cloud computing to be followed as new worldwide pattern of processing. It is an advanced style of utilizing the power of internet to offer assets remotely. However, distributed cloud computing has many difficulties, for example, poor asset usage which has a deep effect on the execution of distributed computing. Such issues emerged because of the tremendous quantity of data. Therefore, the requirement intended for productive and capable distributed cloud computing load harmonizing calculations is a standout amongst the most imperative issues around to enhance the execution of cloud processing. Many investigators proposed numerous load balancing and process programming algorithms in the calculation of cloud but there still remains incompetency within the hardware functioning and lack of load balancing. Thus, this study endorses a set of rules for load balancing to enhance the overall performance and effectiveness in diverse cloud computing environment. We advocate a fusion set of rules based on priority and batch rules, taking advantages of weighted optimization algorithm and equally spread current execution algorithms. The algorithm reflects the current resource records and the capability issue of central processing unit to acquire the targets. The fused algorithm has been evaluated and are compared with different algorithms using Cloud Analyst Simulator. The results demonstrated progress in migration time, reaction time and scalability, availability and 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.

**Key words:** Virtualization, virtual machines, distributed cloud computing, load harmonizing, cloud simulator-cloud analyst, energy-consumption

---

### 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 (Zhang *et al.*, 2004). These heterogeneous condition implies to diverse equipment attributes including CPU, memory, storage and other equipment (Kim *et al.*, 2007). The entrepreneur can begin and also, expand without put extra resources into the framework and bringing down working and upkeeping costs. 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 (Sharma and Banga, 2013). 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 (Sharma and Banga, 2013). The National Institute of Standards and Technologies characterize a cloud processing as “distributed cloud computing is a model for empowering universal, advantageous, on-request allow access to a common pool of stored data (e.g., systems, servers, capacity, applications and administrations) that can be quickly retrieved and discharged with insignificant administration exertion or specialist organization interaction (Zhang *et al.*, 2010). Load synchronization is regarded as one of the difficulties needed to be solved in distributed cloud computing as it is the central point in

the better execution of the distributed cloud computing. The present load adjusting calculations in cloud configuring condition are not exceptionally productive (Pathak *et al.*, 2012). Load adjustment in distributed cloud computing conditions was an extremely wonderful job till today as the expectation of client who asks for entries on the server were most certainly not conceivable. Each virtual machine has its distinctive qualities, so, it turns into an exceptionally complex machine and thus, adjusts the heap among hubs (Tiware *et al.*, 2014). In present days, we can discover many research works that were done on adjustment of stacks in the processing of cloud, for example, Round robin, "equally spread current execution and weighted optimization load balancing calculation". There are different efforts done utilizing randomization, for example, "Subterranean insect state calculation" (Mishra and Jaiswal, 2012). In this examination, we propose a half breed calculation that takes focal points of both arbitrary and insatiable calculation. The trials were done by using cloud investigator to check the execution of the planned calculation in various workstation's control. The trials considered the impact of fixing the limit of central processing unit with the half and half calculation in the diverse condition of hosts and concentrated the impact of system delay on the hybrid calculation. The outcomes appeared upgraded by and huge latency and on getting ready time by considering the present quality information. Therefore, the machine i.e., central processing unit limit issue contrasted and totally different calculations and this means that the execution has created steps.

#### **Cloud computing and its characteristics and models:**

Distributed computing is characterized as a specialized idea where framework clients keep the data on remote servers which can't be seen by the other companies and steal information with the hacking applications. In addition, such data of businesses is stored on cloud servers in such a way at different places that it is impossible to retrieve data for an outsider, who doesn't know the passwords of these data (Almutairi *et al.*, 2011). Vast organizations, for example, Google, Amazon also, Microsoft give all the more intense, solid and cost-proficient cloud stages, some illustrations of developing cloud registering are Microsoft Azure cloud, Amazon EC2 cloud, Google App Engine are using cloud servers for their use (Zhang *et al.*, 2010; Kulkarni *et al.*, 2012; Buyya *et al.*, 2009). The cloud is a virtual place to keep electronic assets that looks after itself (Kaleeswari and Juliet, 2014). It is expanding continuously with various processing advances, for example, elite figuring,

appropriated frameworks, virtualization, stockpiling organizing, security, administration and computerization, Quality of Service, Service-Level-Agreement and Service-Situated-Architecture (Jararweh *et al.*, 2012). Distributed cloud computing gives a few components that make it appealing to IT industry, for instance (Zhang *et al.*, 2010; Sareen, 2013). Distributed cloud computing can be communicated by such transportation models to get back the information as and when required.

**Infrastructure as a service:** This prototype of cloud deals with users by providing hardware as a service with the assistance of internet like focal handling unit, amassing additionally etc. There are various providers like Amazon Elastic Cloud Computer, Rackspace and so on, (Almutairi *et al.*, 2011; Sareen, 2013).

**Platform as a service:** Distributed computing architecture gives a shape to an organization required for building application where customer uses instruments and libraries of cloud provider saving cost of hardware as cloud provider gives the basic hardware framework for customer use in a limited way. There are various providers like Google App Engine, Windows Azure (Almutairi *et al.*, 2011; Sareen, 2013).

**Software as a service:** Concentrate on giving various programming encouraged on the cloud and generally, acquainted with as on-request programming where in this sort of organization, the client should pay for utilization of programming. Typically buyer access to the item through the web, likewise, the customer uses the programming needn't waste time with any combination with another framework (Almutairi *et al.*, 2011; Sareen, 2013). There is various providers like Salesforce.com, Google Apps and so forth has showed up in Fig. 1.

**Open cloud:** A cloud service in which cloud service provider offers their benefits as a organization to the general public. Open cloud offers a couple of key preferences to customers including saving of capital on maintaining system and storing their everything including programs on their servers. But the drawback of open cloud is that it requires fine-grained control over data, framework and security settings which disrupts their practicality in various business situations (Zhang *et al.*, 2010; Mollah *et al.*, 2012).

**Private cloud:** Also, this model of cloud is proposed for specific use by a solitary affiliation. A private cloud system is built and regulated by an external cloud

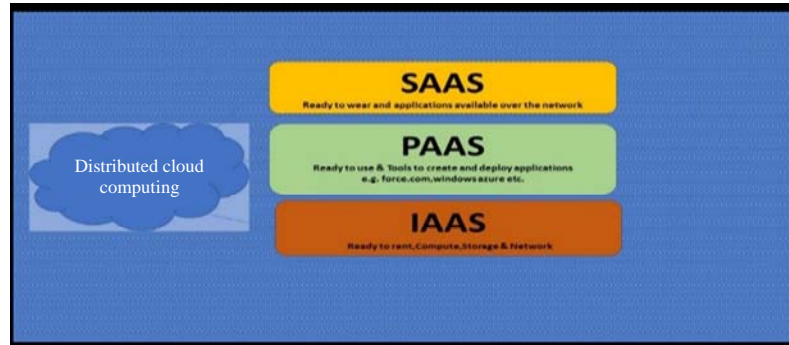


Fig. 1: Types of cloud service providers

provider for a company for their personal use. A private cloud is beneficial for a company as it gains significant level of control over execution of operations, better quality and more security. In any situation, they are frequently censured for resembling standard prohibitive servers and their points of interest are kept secret (Zhang *et al.*, 2010; Sajid and Raza, 2013).

**Hybrid cloud:** A hybrid cloud is a mixture of open and private cloud models in which the execution of each sides approach is limited. In a hybrid cloud, some bit of the organization structure runs in private cloud system while the remaining part of it runs in the open cloud system (Zhang *et al.*, 2010; Sajid and Raza, 2013).

Virtualization isolates assets and administrations from the fundamental physical conveyance condition (Sareen, 2013). Virtualization is regarded as a center of distributed cloud computing advancements and a standout amongst the most vital advancements which has powered this today's world (García-Valls *et al.*, 2014; Kumar and Aramudhan, 2013). Virtualization conceals a registering phase's substantial qualities from clients (García-Valls *et al.*, 2014; Kumar and Aramudhan, 2013). It permits reflection and segregation of lower level of functioning and hidden equipment. This empowers convenience of more elevated amount capacities and sharing or potentially the total of the substantial assets (Vouk, 2008). Virtualization signifies "something which isn't genuine", however, gives every one in the offices of a real existing look of any hardware system (Tiwari *et al.*, 2014). It is the programming usage of a personal computing machine which will execute distinctive projects as if it is a real machine but actually a remote machine is doing all the jobs (Rajoriya, 2014). Virtualization of the project has three benefits that formulate its relation with distributed computing which is (Sareen, 2013): segregating, separation and binding the data i.e., encapsulate the data.

## MATERIALS AND METHODS

### Distributed cloud computing issues and challenges:

There issues arose from distributed computing and needs to be solved as following.

**Aegis measures:** Obviously the issue of security is assumed to be the most critical part in cloud processing. Issue of security, for instance, information misfortune, phishing, protection and different dangers at the corporate level or singular level while utilizing the remotely situated processing assets in distributed computing has presented new security challenges for the world. So, to solve these issues, we need to device new methods to completely erase the effects of unlimited risks in the distributed computing servers (Kuyoro *et al.*, 2011).

**Application performance:** This is another issue to be solved in distributed cloud computing (Kuyoro *et al.*, 2011). Poor execution can be the result of the absence of assets, for instance, circular space, limited data transfer capacity, the speed of CPU is lowered, memory, hardware and so forth. The information serious applications are additionally being tested to deliver latest hardware assets. Poor execution can bring about the end of administration expenses, loss of clients and decrease incomes (Sajid and Raza, 2013). Execution can be found on various techniques, apparatuses and recreations for cloud situations, for example, fluffy frameworks and a device like Cloud Analyst (Khanghahi and Ravanmehr, 2013). There is a progression of factor that influences the execution, for example, security, recovery and fault resilience, service level understandings, bandwidth, storage limit, physical memory, disk limit, processor power, availability, number of clients and workload, usability, scalability and location, server farms and their separation from a client's area what's more, there is a progression of criteria for assessing the execution, for example (Khanghahi and

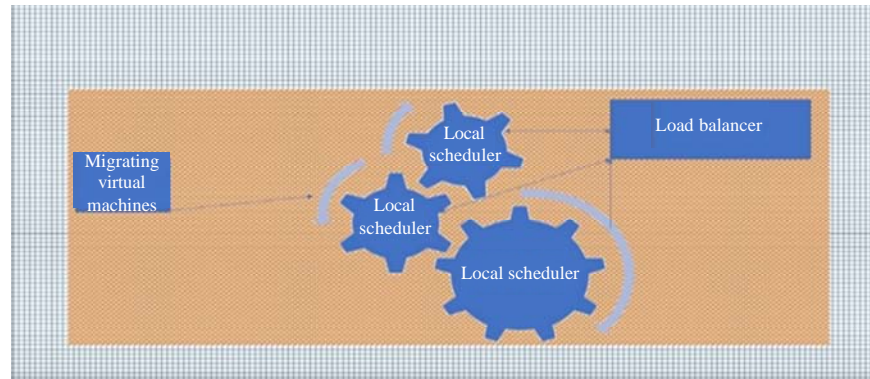


Fig. 2: Load balancing model

Ravanmehr, 2013; Deepika and Kumar, 2014): namely, average reply time per unit time, average holding up time per unit time, workload to be fixed every second (milli per second) or a unit of time, throughput required per second, the average time of handling specifically execution per second. Percentage of central processing unit use, the number of processes executed per unit time, the number of processes per unit time buffer, the number of rejected processes per unit time.

**Technique of load balancing:** Technique of load balancing reassigns the total load to the individual nodes of the collective cloud arrangement to get 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. Load balancing ensures that all nodes in the system are given the equal amount of work at any time (Sharma and Banga, 2013; Mohapatra *et al.*, 2013). The main objective of load balancing is to achieve the optimal utilization of the resources like over all control of all the systems, least response time and sidestep the overload condition (Kuyoro *et al.*, 2011). The heterogeneous environment condition is thought to be a major concern (Yao and He, 2012; Shameem and Shaji, 2013; Behal and Kumar, 2014) 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 (Shameem and Shaji, 2013; Kaushik *et al.*, 2013). While load balancing in distributed cloud computing is based on concrete load balancing system and it varies from conventional load-balancing to concurrent computing. But in distributed cloud computing the structural design and operation of the load balancing process is totally different and is designed, according to, the use of service provider servers to perform the load balancing which gives new

opportunities and economies of scale (Sethi *et al.*, 2012). Figure 2 shows load balancing in cloud computing (Pathak *et al.*, 2012).

Load balancer capable with assigns the virtual machine of the client asks for. After that, the information focus sends those signal of the clients after transforming by server-farms. The load adjusting may be extremely critical done cloud registering Earth. Those major objectives for the load adjusting calculations are: attain an in general change in framework execution toward a sensible expense (Kuyoro *et al.*, 2011). Should have a reinforcement arrangement in the event that the framework neglects indeed going incompletely (Rajoriya, 2014). To suit future adjustment in the system: the disseminated framework might change, for example, applying new toponymy what's more focused. In this system, a load adjusting algorithm must a chance to be versatile what's more adaptable should handle these transforms. Load adjusting calculation might be separated into two classes similarly as: static and progressive (Mohapatra *et al.*, 2013; Mehta *et al.*, 2012).

**Load adjusting static algorithms:** Static calculations partition that movement equivalently between servers and the load adjusting method. The general disservice of constantly on static schemes is that those last determination of a host to transform allotment will be settled on at the methodology will be made and can't make transformed throughout methodology execution to aggravate transforms in the framework load. Round robin calculations are a static load adjusting calculation a result those fill in load circulations the middle of processors is equivalent (Deepika and Kumar, 2014).

**Load adjusting dynamic algorithms:** For changing calculations choices with respect to load adjusting need aid in view of current state of the framework no former

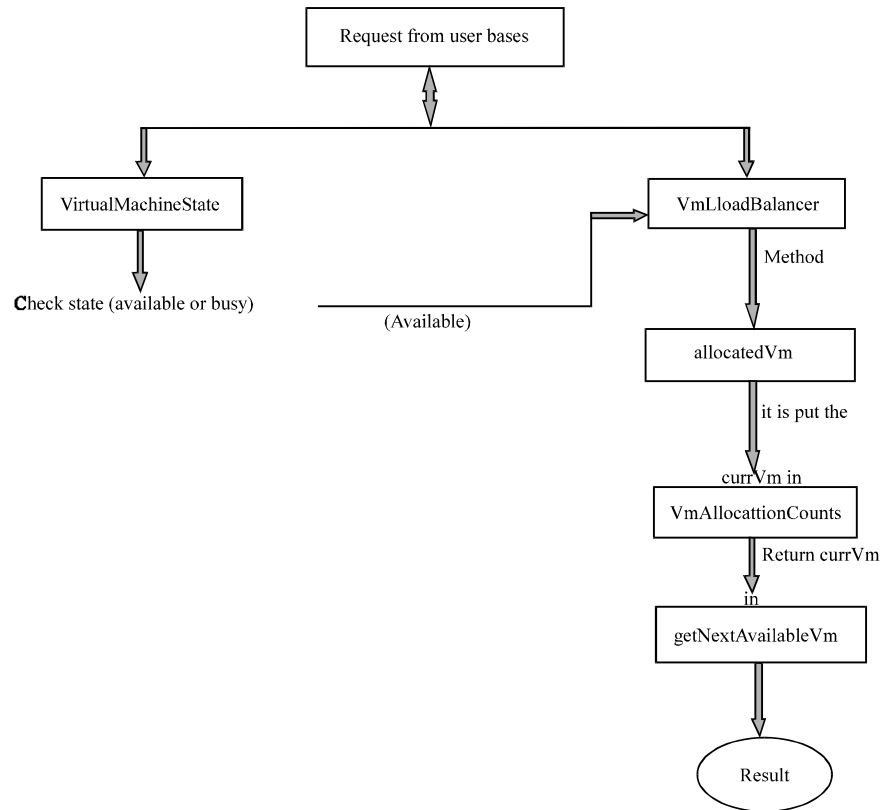


Fig. 3: Round robin flowchart

information will be required to load adjusting. Thereabouts it is superior to static approach. Element load adjusting camwood be done in two ways (Mishra and Jaiswal, 2012; Deepika and Kumar, 2014). Available in the framework and the assignment from claiming load adjusting is imparted around them. A benefit, from claiming this will be that regardless of particular case or more hubs in the framework fail, it will not cause those aggregate loads adjusting procedure to halt, it rather might influence the framework execution with a few degrees.

**Non-distributed dynamic load balancing:** The non-distributed, contains particular case hub answerable for load adjusting of the entire framework. The opposite hubs associate just for the national hub (Mehta *et al.*, 2012). In this exploration those suggested an algorithm will be a sort from claiming non-distributed element load adjusting.

**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 research, we

are setting off with aggravate investigations for percentage about these calculations what's more analyze them for our worth of effort.

**Round robin algorithm:** It will be viewed as likewise the greater part fundamental and the minimum perplexing planning algorithm (Dave and Maheta, 2014), 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 Fig. 3. 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 (Kuyoro *et al.*, 2011; Mohapatra *et al.*, 2013) Fig. 3.

**Equally spread current execution:** Equally spread current execution calculation as indicated done Fig. 4. 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

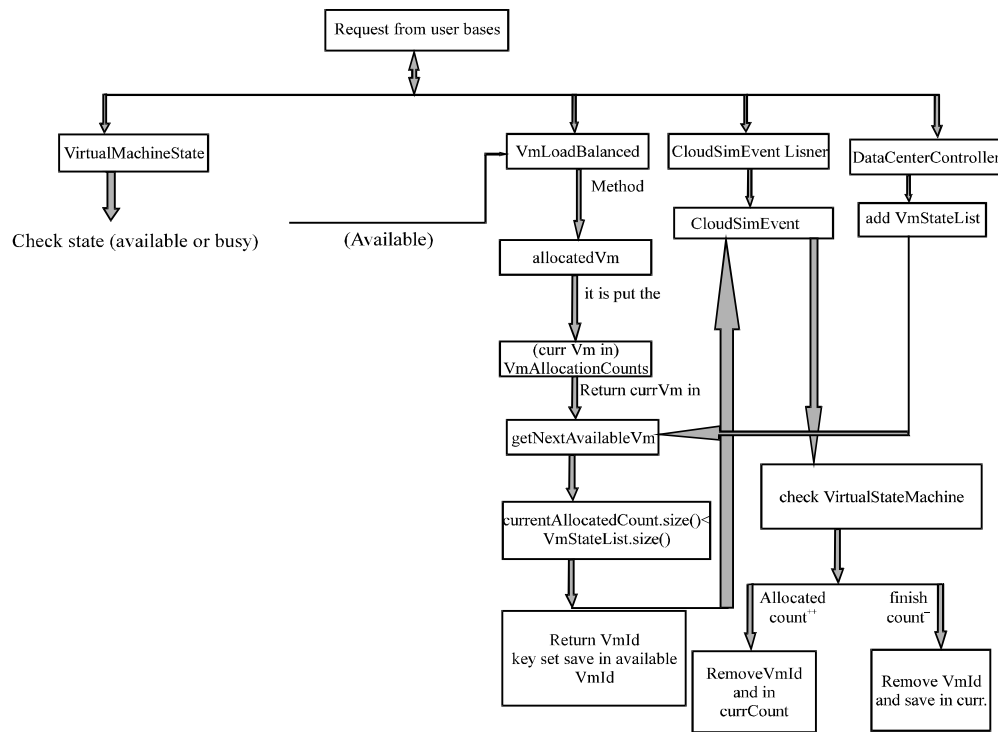


Fig. 4: Equally spread current execution flowchart

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, (Mohapatra *et al.* (2013) and Mohialdeen (2013) (Fig. 4).

In this investigation, we will suggest a hybridization of weighted-optimization and enhanced weighted optimization algorithm that takes focal point of both irregulars also, each calculation needs a portion points of interest and a percentage restriction. For instance, an irregular algorithm which haphazardly selects a virtual machine with the procedure the gained tasks don't the necessity. Mind boggling calculation will settle on a choice yet all it doesn't select those best VM. On the great holders kept all weighted optimization algorithm selects the best virtual machine on the handle the gained task, yet, the optimal procedure. Necessities exactly intricate calculation will discover the best virtual machine as shown in Fig. 5.

**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 (Mohialdeen, 2013). 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 Fig. 6.

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

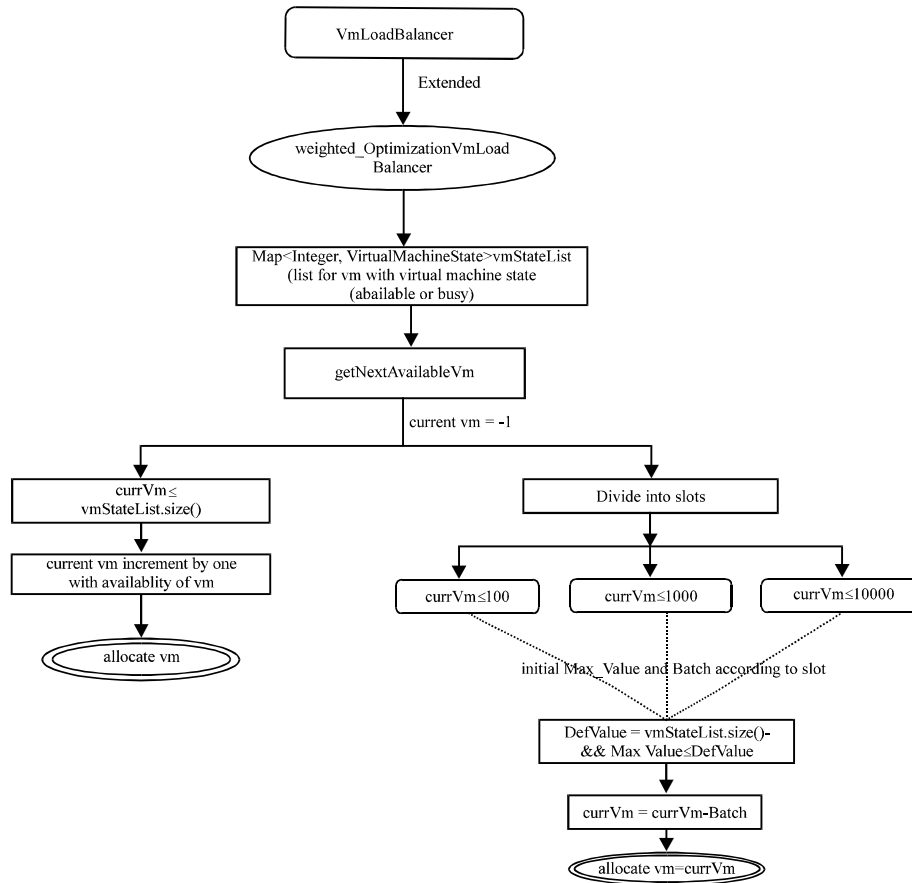


Fig. 5: Weighted optimization flowchart

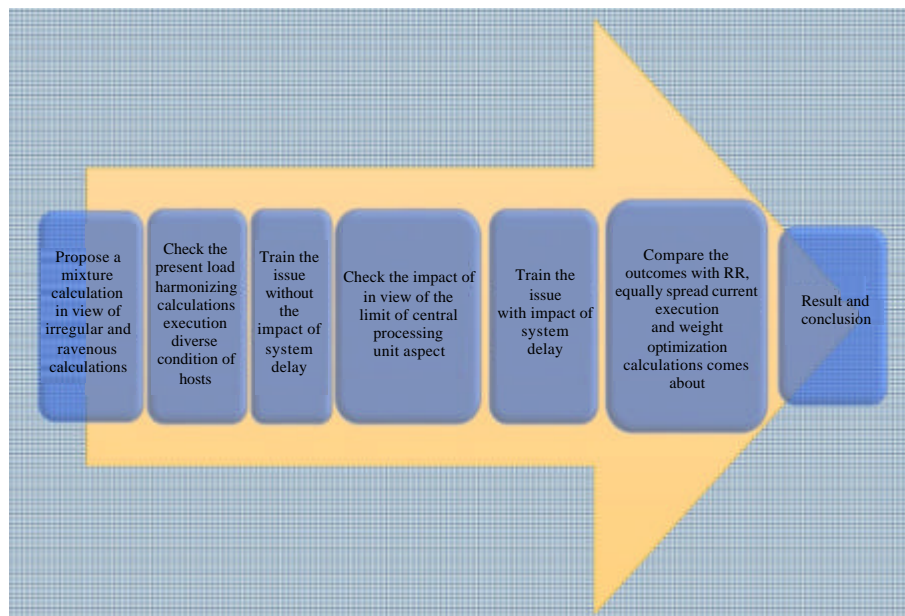


Fig. 6: Process of checking virtualization

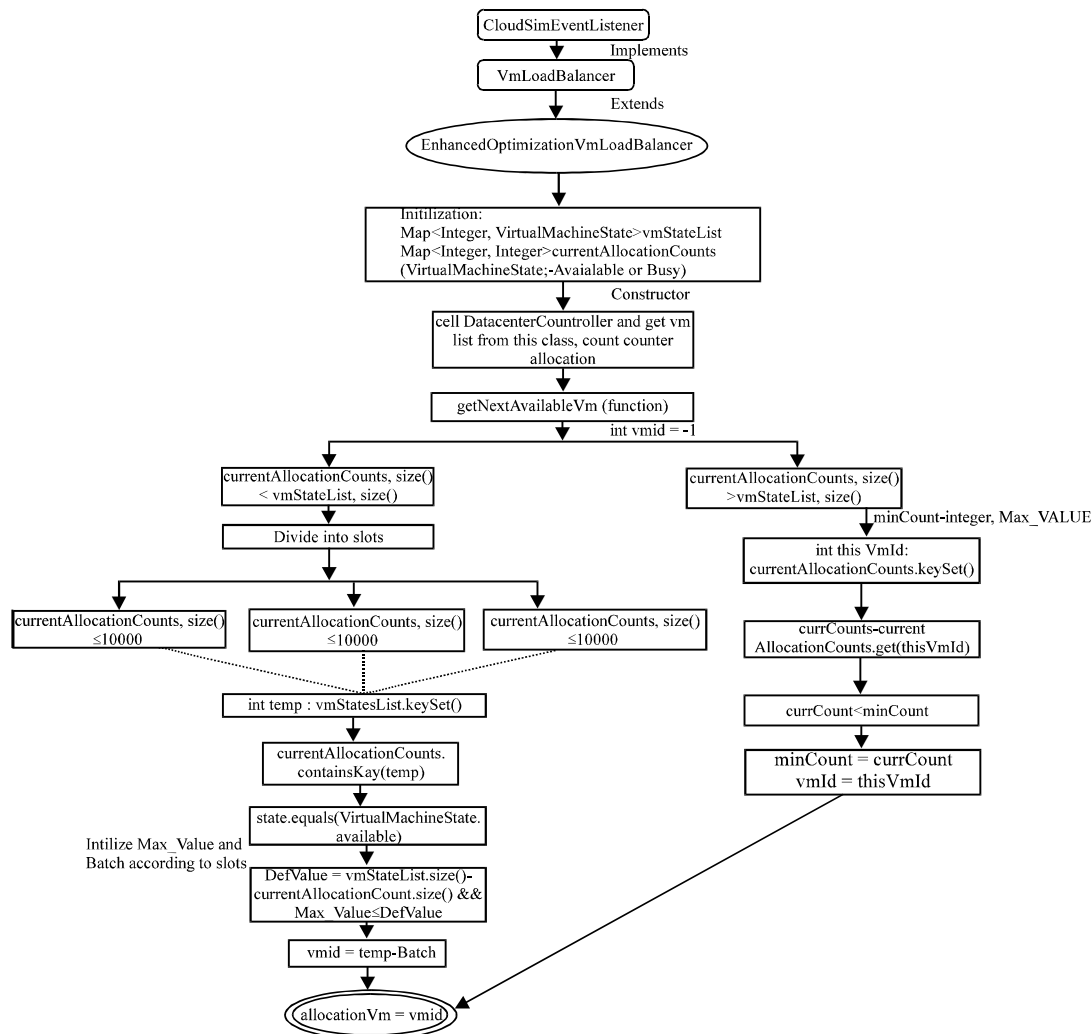


Fig. 7: Enhanced weighted optimization algorithm

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 Fig. 7 indicates the unique understanding for calculation.

**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 (Singh *et al.*, 2014). 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.

**Implementation of proposed algorithm:** The experimentation is accomplished using the Cloud Analyst test system. We interpret the test system specifications, for detail, consumers configuration, data focus design, virtual systems pattern and we looked at a few contours. The trials executed using the qualified method in the initial steps we mulled over the issue without the impact of system delay; we tried the calculation in the heterogeneous condition of hosts and each machine has a unique number of central processing system and speed and after that, we tried the impact of investigating the competence of central processing system point. At last, we tried the effect of the impact of system wait on the half-breed calculation with considering the capacity of central processing unit factor and in the heterogeneous condition of presents. 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 perform the crossbreed calculation namely enhanced weighted optimization algorithm. The code of the crossover calculation is an informative buttress.

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

**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 twotimesvary.uptime+ = entry.getTime()+lastTimeSwitchedOn;downtime+ = entry.getTime()

**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 user terminal. There is still the approach of observed response time which is the time a user senses as open input and the end of the response. It is possible for distinguished response time to be extremely fast. However, this is not the regular criticism. Response time = Finisht-Arrivalt+Transmission delay, 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.

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

**Scalability:** Scalability = (totAll/(succRes\*100)).

**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.

**Reliability:** (succRes/(totAll\*100)), succRes+ = entry. getAllocatedMips(); totReq+ = entry.getRequeste Mips(); totAll = totReq+succRes.

**Energy-consumption:** EnergyconsumptionCostRs.: %.4f, energy\*energyCostdatacenter.getPower().

## RESULTS AND DISCUSSION

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

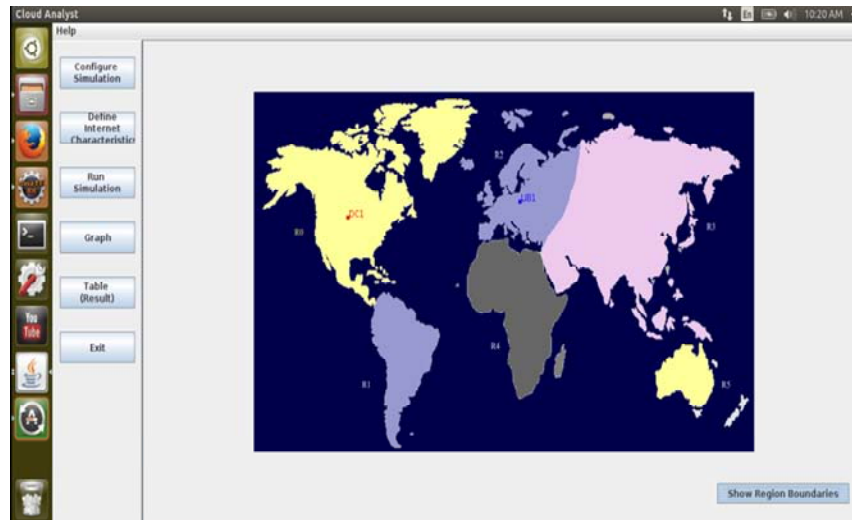


Fig. 8: Cloud analyst regions

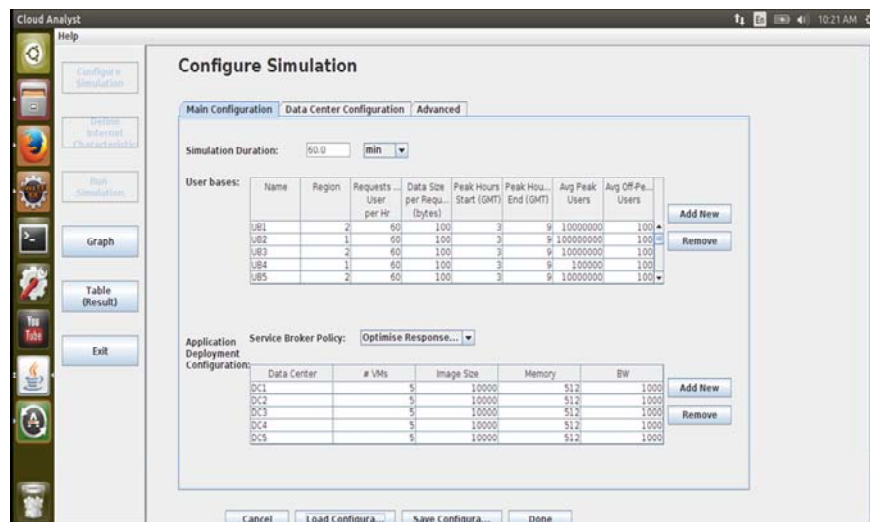


Fig. 9: Simulation configurations for userbase and virtual machines

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 Fig 8-11.

Here, in this Fig. 10, we had set up the data center configuration to be specific physical hardware details of data focus at that point. We will get the physical hardware details of data center at that time move to advanced configurations, here client base which designs an association of clients and provides activity going to the

clients and application deployment configurations (cloudlets). After this service broker policy, i.e., service broker configuration at that point save this configuration too in the event that you require to take advantage of it subsequently. It is focused away as Sim record. XML information is offered and saved as Sim document. The spared arrangement can be loaded whenever effectively into Cloud Analyst. So, we have to begin information each time you require to run the simulation. Then configured data centers in different regions. After that, we have performed 5 data centers in the different region, each with 38 user base each with 5 virtual machines. Virtual

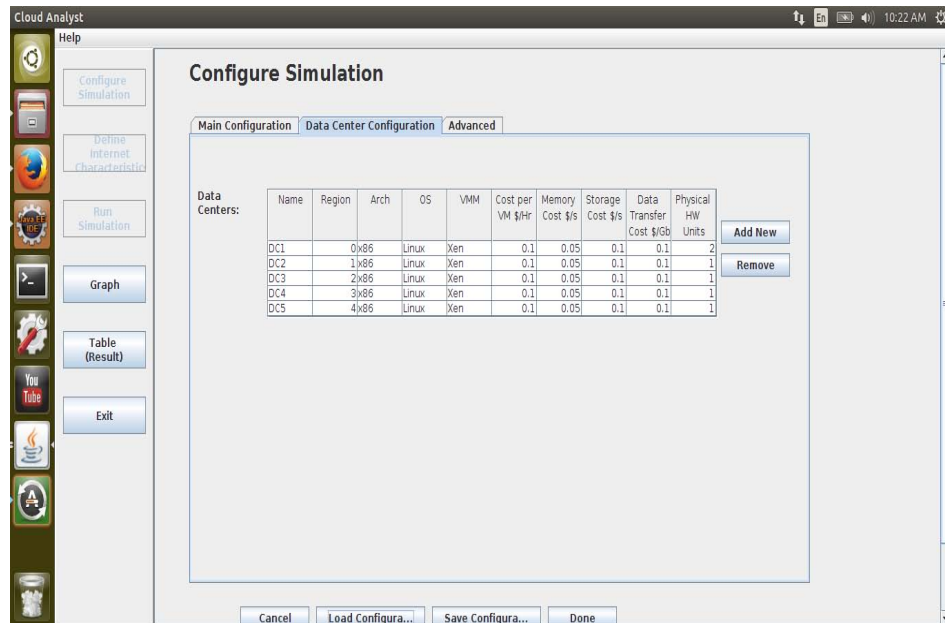


Fig. 10: Simulation configuration for datacenter and hypervisor

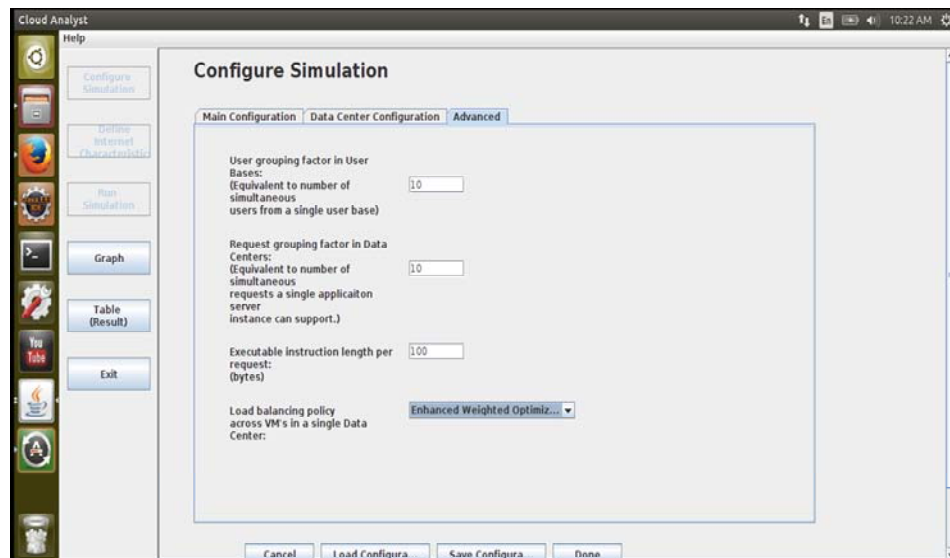


Fig. 11: Simulation configuration for algorithm selection

Machine manager specifically Xen is utilized and service broker policy utilized to be specific optimize response time. Thus, configurations of round robin load balancing policy 38 user bases and each in individual area, 5 virtual machines are empowered to satisfy requests from all client bases from both the data center same action done with the configuration of equally spread current execution policy, weighted optimization policy and

enhanced weighted optimization policy that is the proposed one approach. The specific situations have been drawn up using the configurations in the below mentioned images. Figure 12-18 shows the result of Round robin, equally spread current execution, weighted optimization and enhanced weighted optimization algorithms analogous to the matrices migration-time, responsetime, throughput, scalability, availability,

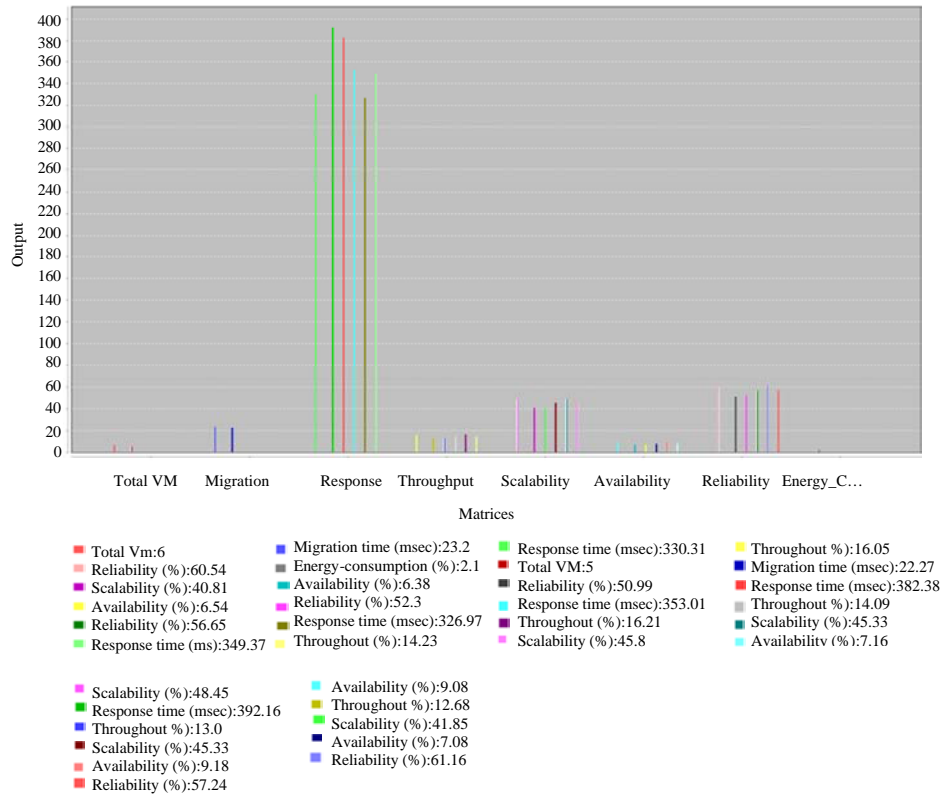


Fig. 12: Different matrices result of RR algorithm; Round robin

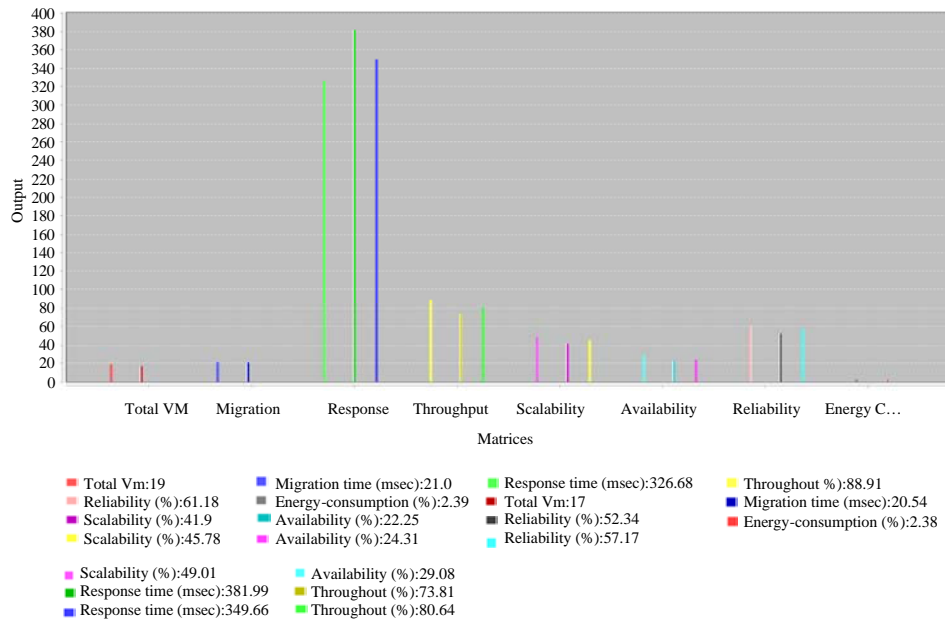


Fig. 13: Different matrices result of ESCE algorithm; Equally speed current execution load

reliability and the energy consumption. In Fig. 19, we have determined the all matrices results in coupled form

which demonstrates that the migration time is 22.27 msec in Round robin and weighted optimization policy when

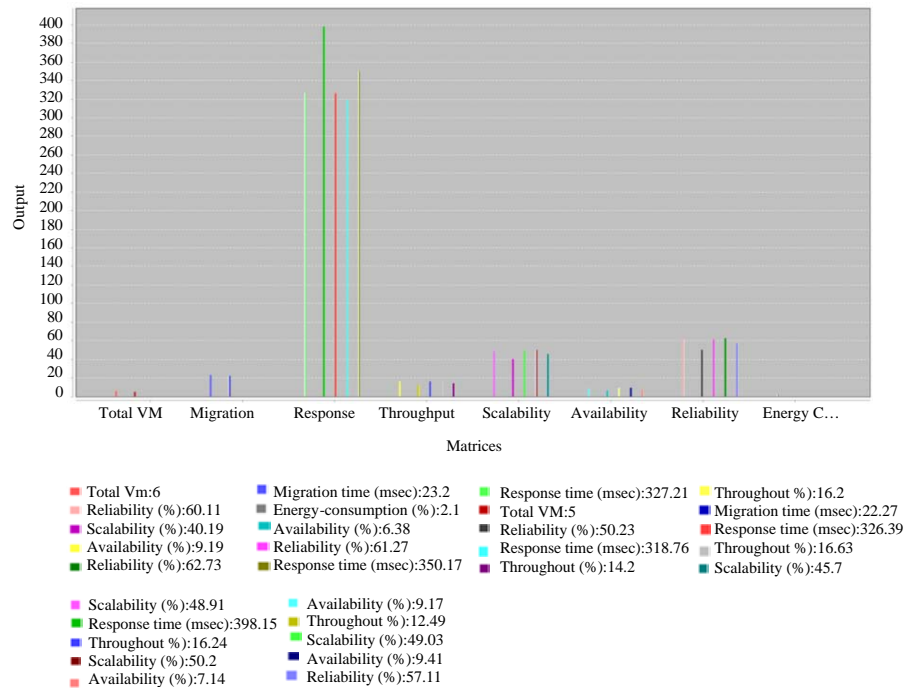


Fig. 14: Different matrices result of WOA algorithm; Weight optimization algorithm

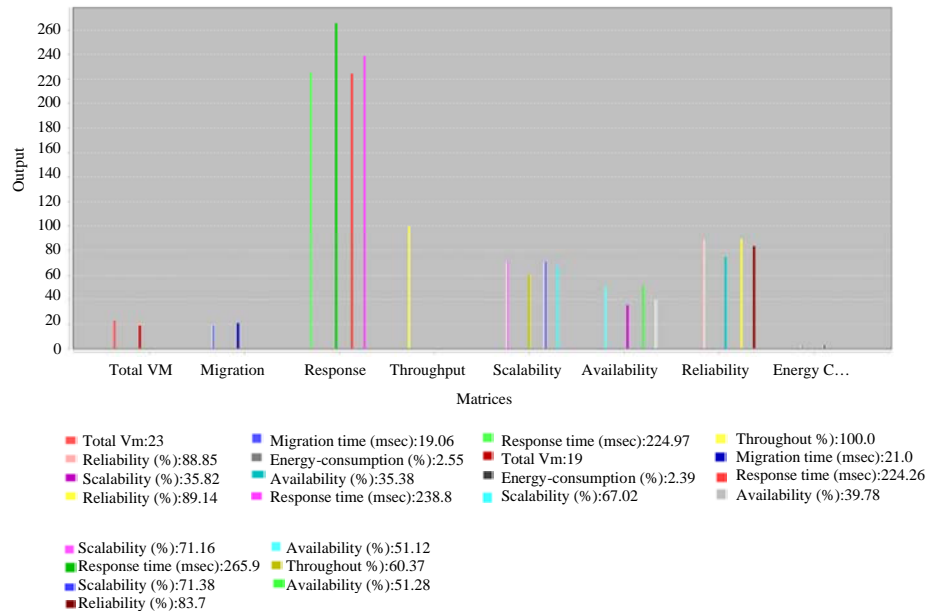


Fig. 15: Different matrices result of EWOA algorithm; Enhanced weight optimization algorithm

number of virtual machines are 5 in both the cases. But 21 msec in enhanced weighted optimization policy when number of virtual machines are 19. So, it is the leading difference and proposed policy is offering better proceeding in aspect of migration time. So, all

other parameters response-time, throughput, scalability, availability, reliability and energy consumption are likewise offering stronger result in the proposed policy that is enhanced weighted optimization algorithm.

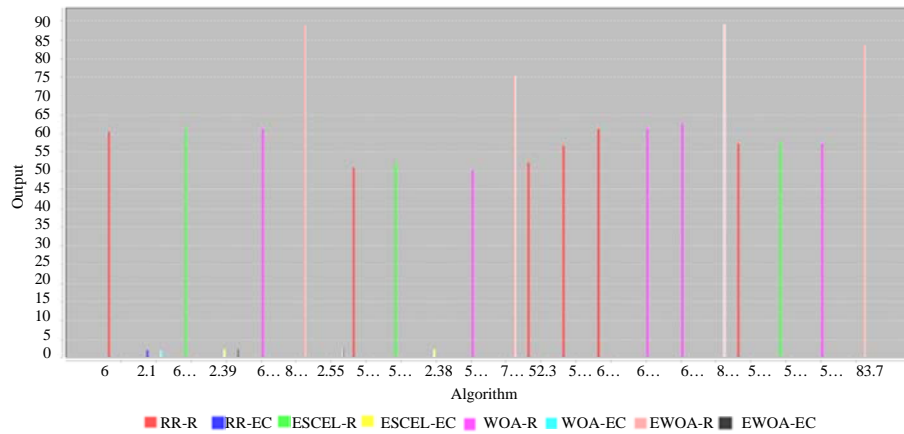


Fig. 16: Combined matrices result of RR, ESCE, WOA, EWOA, R and EC algorithm

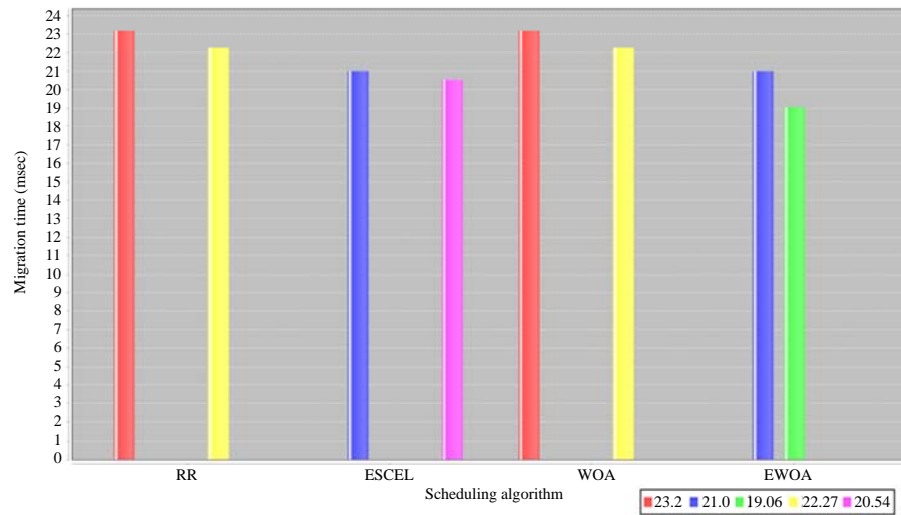


Fig. 17: Migration time result of RR, ESCE, WOA, EWOA and MT algorithm

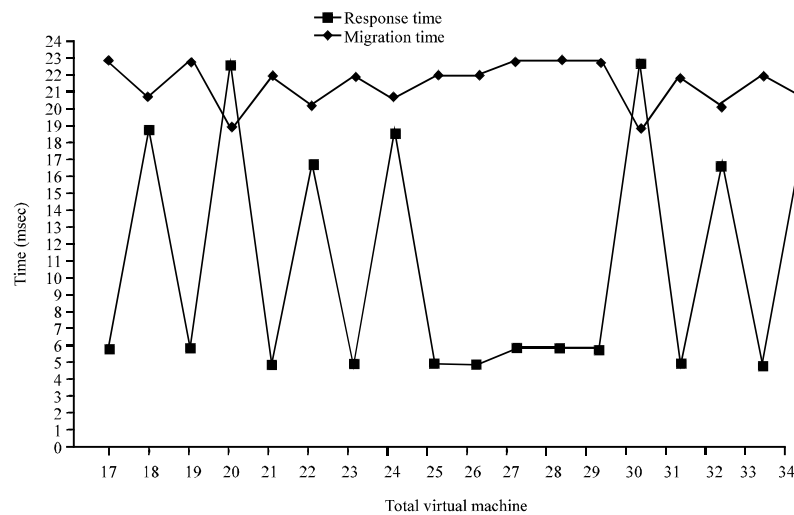
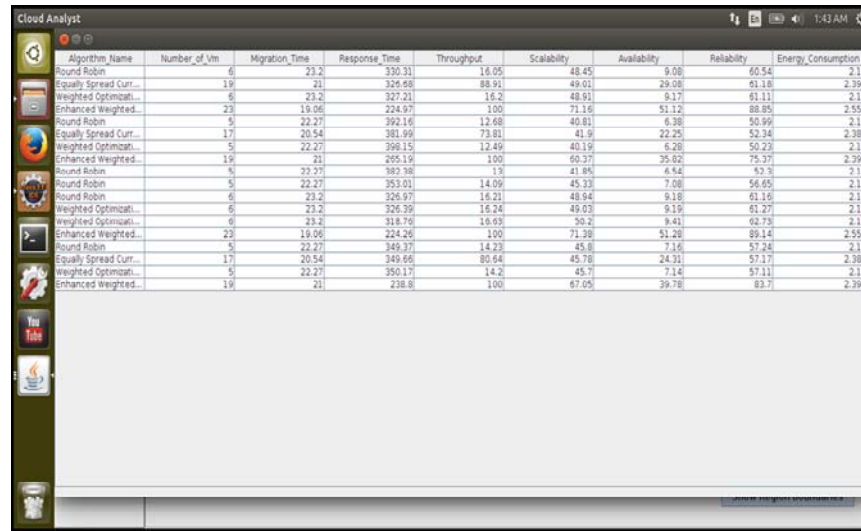


Fig. 18: Response time and migration time comparison depends on total No. of VMs



Algorithm Name	Number of Vm	Migration Time	Response Time	Throughput	Scalability	Availability	Reliability	Energy Consumption
Round Robin	6	23.2	330.31	16.05	48.45	9.08	60.54	2.11
Equally Spread Curr...	19	21	326.68	88.91	49.01	29.08	61.18	2.39
Weighted Optimizat...	6	23.2	327.21	16.2	48.61	9.17	61.11	2.1
Enhanced Weighted...	23	19.06	224.67	100	71.16	51.12	88.95	2.55
Round Robin	5	22.27	392.16	12.68	45.81	6.38	50.99	2.11
Equally Spread Curr...	17	20.54	381.99	73.81	41.9	22.25	52.34	2.38
Weighted Optimizat...	5	22.27	398.15	12.49	40.19	6.28	50.23	2.1
Enhanced Weighted...	19	21	255.19	100	60.27	35.62	75.37	2.39
Round Robin	5	22.27	382.38	13	41.84	4.44	53.3	2.1
Round Robin	5	22.27	353.01	14.09	45.33	7.08	56.65	2.11
Equally Spread Curr...	6	23.2	326.97	16.21	48.94	9.18	61.16	2.1
Weighted Optimizat...	6	23.2	326.39	16.24	48.03	9.19	61.27	2.1
Enhanced Weighted...	6	23.2	318.76	16.69	50.2	9.41	62.73	2.1
Round Robin	23	19.06	224.26	100	71.38	51.28	89.14	2.55
Equally Spread Curr...	5	22.27	349.37	14.23	45.8	7.16	57.24	2.1
Weighted Optimizat...	17	20.54	349.66	80.64	45.78	24.31	57.17	2.38
Enhanced Weighted...	5	22.27	350.17	14.2	45.7	7.14	57.11	2.1
Round Robin	19	21	238.6	100	67.05	39.78	83.7	2.39

Fig. 19: RR, ESCE, WOA, EWOA result shows on different matrices in tabular form on different-different SIM files

## CONCLUSION

Load changing is a champion among the crucial issues secured nearby cloud enlisting. The present load modifying and arranging figurings in cloud enlisting Earth have a segment deficiency what's more this may impact. Those execution and hence, we prescribed a blend computation on enhance those cloud enrolling. Execution of those blend calculation in perspective of randomization what's greedier calculation, they make those. Purposes of enthusiasm for both unpredictable, furthermore, ravenous computations and consider the present resource information and the focal preparing unit constrain figure on accomplish the goals. The examinations were executed in the Cloud Analyst test framework without recognizing those machine capacity factor, those results bring showed that those greedy, round robins. What's more ESCE required abetter realize deficiencies over those blend estimations and the unpredictable calculation. This may have been a direct result of those equivalent conveying about burdens between each and every one of VMs. Additionally, we found that the round robin. Calculation may have been better than the ESCE also, greedy calculation in light it might be essential, also, needn't bother with. The overhead figuring concerning delineation ESCE what's greedier. On the immense holders kept all at those VMs passed on the hosts concurring those hosts capability furthermore. Concurring those machine capacity, those host that bring best capability will have a more prominent sum VMs over various. Hosts, so, at we select k center points aimlessly from the VMs furthermore, choose those base stacked one beginning with. Those picked VMs, the response time

will an opportunity to be upgraded on most of VMs picked will an opportunity to be in the qualified host. 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

- Almutairi, A., M. Sarfraz, S. Basalamah, W. Aref and A. Ghafoor, 2011. A distributed access control architecture for cloud computing. *IEEE. Software*, 29: 36-44.
- Behal, V. and A. Kumar, 2014. Cloud computing: Performance analysis of load balancing algorithms in cloud heterogeneous environment. *Proceedings of the 2014 5th International Conference-Confluence on Next Generation Information Technology Summit (Confluence)*, September 25-26, 2014, IEEE, Noida, India, pp: 200-205.
- Buyya, R., R. Ranjanand and R.N. Calheiros, 2009. Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: Challenges and opportunities. *Proceedings of the 2009 International Conference on High Performance Computing and Simulation HPCS'09*, June 21-24, 2009, IEEE., Leipzig, Germany, pp: 1-11.

- Cormen, T.H., C.E. Leiserson, R.L. Rivest and C. Stein, 2009. Introduction to Algorithms. 3rd Edn., MIT Press, Cambridge, Massachusetts, USA., ISBN:9780262533058, Pages: 1292.
- Dave, S. and P. Maheta, 2014. Utilizing round robin concept for load balancing algorithm at virtual machine level in cloud environment. *Intl. J. Comput. Appl.*, 94: 23-29.
- Deepika, D.W. and N. Kumar, 2014. Performance analysis of load balancing algorithms in distributed system. *Adv. Electron. Electr. Eng.*, 4: 59-66.
- García-Valls, M., T. Cucinotta and C. Lu, 2014. Challenges in real-time virtualization and predictable cloud computing. *J. Syst. Archit.*, 60: 726-740.
- Jararweh Y., Z. Alshara, M. Jarrah, M. Kharbutli and M.N. Alsaleh, 2012. TeachCloud: A cloud computing educational toolkit. Proceedings of the 1st International Conference on IBM Cloud Acad, April 19-20, 2012, Research Triangle Park, North Carolina, USA., pp: 1-16.
- Kaleeswari and N.M. Juliet, 2014. Dynamic resource allocation by using elastic compute cloud service. *Int. J. Innovative Res. Sci. Eng. Tech.*, 3: 12375-12379.
- Kaushik, V.K., H.K. Sharma and D. Gopalani, 2013. Load balancing in cloud computing using high level fragmentation of dataset. Proceedings of the International Conference on Cloud, Big Data and Trust, November 13-15, 2013, Bhopal, Madhya Pradesh, India, pp: 118-126.
- Khanghahi, N. and R. Ravanmehr, 2013. Cloud computing performance evaluation: Issues and challenges. *Intl. J. Cloud Comput. Serv. Archit.*, 3: 29-41.
- Kim, J.K., S. Shiple, H.J. Siegel, A.A. Maciejewski and T.D. Braun *et al.*, 2007. Dynamically mapping tasks with priorities and multiple deadlines in a heterogeneous environment. *J. Parallel Distrib. Comput.*, 67: 154-169.
- Kulkarni, G., J. Gambhir and R. Palwe, 2012. Cloud computing-software as service. *Intl. J. Cloud Comput. Serv. Sci.*, 1: 11-16.
- Kumar, V.S. and M. Aramudhan, 2013. Performance analysis of cloud under different virtual machine capacity. *Intl. J. Comput. Appl.*, 68: 1-4.
- Kuyoro, S.O., F. Ibikunle and O. Awodele, 2011. Cloud computing security issues and challenges. *Intl. J. Comput. Netw.*, 3: 247-255.
- Mehta, R., P. Yask and T. Harshal, 2012. Architecture for distributing load dynamically in cloud using server performance analysis under bursty workloads. *Intl. J. Eng. Res. Technol.*, 1: 1-5.
- Mishra, R. and A. Jaiswal, 2012. Ant colony optimization: A solution of load balancing in cloud. *Intl. J. Web Semant. Technol.*, 3: 33-50.
- Mohapatra, S., K.S. Rekha and S. Mohanty, 2013. A comparison of four popular heuristics for load balancing of virtual machines in cloud computing. *Intl. J. Comput. Appl.*, 68: 33-38.
- Mohialdeen, I.A., 2013. Comparative study of scheduling algorithms in cloud computing environment. *J. Comput. Sci.*, 9: 252-263.
- Mollah, M.B., K.R. Islam and S.S. Islam, 2012. Next generation of computing through cloud computing technology. Proceedings of the 2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), April 29-May 2, 2012, IEEE, Montreal, Quebec, Canada, pp: 1-6.
- Pathak, K.K., P.S. Yadav, R. Tiwari and T.K. Gupta, 2012. A modified approach for load balancing in cloud computing using extended honey bee algorithm. *Intl. J. Res. Rev. Eng. Sci. Technol.*, 1: 12-19.
- Rajoriya, S., 2014. Load balancing techniques in cloud computing: An overview. *Intl. J. Sci. Res.*, 3: 1616-1623.
- Sajid, M. and Z. Raza, 2013. Cloud computing: Issues & challenges. Proceedings of the International Conference on Cloud, Big Data and Trust Vol. 20, November 13-15, 2013, Rajiv Gandhi Proudhyogiki Vishwavidyalaya, Bhopal, India, pp: 35-41.
- Sareen, P., 2013. Cloud computing: Types, architecture, applications, concerns, virtualization and role of it governance in cloud. *Intl. J. Adv. Res. Comput. Sci. Software Eng.*, 3: 533-538.
- Sethi, S., A. Sahu and S.K. Jena, 2012. Efficient load balancing in cloud computing using fuzzy logic. *IOSR J. Eng.*, 2: 65-71.
- Shameem, P.M. and R.S. Shaji, 2013. A methodological survey on load balancing techniques in cloud computing. *Intl. J. Eng. Technol.*, 4: 3801-3812.
- Sharma, T. and V.K. Banga, 2013. Efficient and enhanced algorithm in cloud computing. *Int. J. Soft Comput. Eng.*, 3: 385-390.
- Singh, A., R. Bedi and S. Gupta, 2014. Design and implementation of an efficient scheduling algorithm for load balancing in cloud computing. *Intl. J. Emerging Trends Technol. Comput. Sci.*, 3: 106-109.
- Tiwari, M.N., K.K. Gautam and R.K. Katore, 2014. Analysis of public cloud load balancing using partitioning method and game theory. *Intl. J. Adv. Res. Comput. Sci. Software Eng.*, 4: 807-812.
- Vouk, M.A., 2008. Cloud computing-issues, research and implementations. *J. Comput. Inform. Technol.*, 16: 235-246.



- Yao, J. and J.H. He, 2012. Load balancing strategy of cloud computing based on artificial bee algorithm. Proceedings of the 8th International Conference on Computing Technology and Information Management, Volume 1, April 24-26, 2012, Seoul, South Korea, pp: 185-189.
- Zhang, Q., L. Cheng and R. Boutaba, 2010. Cloud computing: State-of-the-art and research challenges. J. Internet Serv. Applic., 1: 7-18.
- Zhang, Z., J.D. McCalley, V. Vishwanathan and V. Honavar, 2004. Multiagent system solutions for distributed computing, communications and data integration needs in the power industry. Proceedings of the IEEE General Meeting on Power Engineering Society, June 6-10, 2004, IEEE, Denver, Colorado, USA., pp: 45-49.