Optimizing and Testing Load Balancing Algorithms Using CloudSim

Prof. Amol Gaikwad
Department Of Information Technology
Yeshwantrao Chavan College Of Engineering
Nagpur, India
amolgaikwad.ag@gmail.com

Sheetal Vishwakarma
Department Of Information Technology
Yeshwantrao Chavan College Of Engineering
Nagpur, India
19030036@ycce.in

Girish Makarwar
Department Of Information Technology
Yeshwantrao Chavan College Of Engineering
Nagpur, India
18010613@ycce.in

Arppit Madankar
Department Of Information Technology
Yeshwantrao Chavan College Of Engineering
Nagpur, India
18010822@ycce.in

***Abstract*— Rapidly enhancement in cloud computing technology makes the world easier. The effect makes the cloud computing generation extra smarter. The cloud centers and give up users are geographically allotted across the globe in a big-scale cloud computing surroundings. the most important undertaking for cloud records centers is how to manage and service the millions of requests that are arriving very regularly from quit users effectively and correctly. In cloud computing, load balancing is needed to distribute the dynamic workload evenly throughout all the nodes. Load balancing is an crucial key in cloud computing. Load balancing helps to obtain a excessive consumer pride and resource usage ratio by way of ensuring an efficient and honest allocation of every computing aid. we are going to simulate all of the effects of load balancing set of rules the use of CloudSim three.zero.three. At final we are going to evaluate all results and shown which one it better for load balancing. Cloudsim is a simulator which allows to simulate the result of load balancing algorithm. proper load balancing aids in minimizing useful resource consumption, imposing fail-over, allowing scalability, avoiding bottlenecks and over-provisioning and so forth. right here we are comparing, the burden balancing algorithms to balance the burden among digital machines in cloud statistics middle. Results shown that our set of rules can obtain better load balancing in a big-scale cloud computing surroundings in comparison to previous load balancing algorithms.**

Key Phrases—simulator, allocation, utilization, balancing (key words**)**

I. INTRODUCTION

In cloud ecosystem, application provider vendors or software companies can cognizance solely on building their software program, rather than dealing with server infrastructure and platform themselves. Cloud computing takes away this burden from software companies by way of providing handy and cost effective mechanism of deploying the software program. cutting-edge software vendors broaden their software program on cloud environment, installation them on some of rented server times and deliver the software program as a service. whilst those rented server instances supply them instantaneous transport mechanism for their software program, inefficient load distribution amongst those server instances may result in poor performance for his or her brought software. therefore the load balancer comes into the photograph.

Cloud Computing has become one of the famous era adopted by way of both enterprise academia supplying a flexible and green way to shop and retrieve documents . The essential problem is scheduling of the incoming request so mlmmum reaction time is received, green aid usage.

Cloud Computing device are heavily depend on time period virtualization that improves the strength performance of information centers and enables virtual machines to unmarried physical server. Many algorithms FCFS, honeybee primarily based load balancing approach, round Robin, active Clustering, lively monitoring Load Balancer, Throttled Load Balancer, active Clustering, WCAP, no, CLBVM, Random sampling have been designed to carry out the consumer's request toward the cloud nodes but to ensure powerful usage of resources and reaction time minimum the term load balancing comes into effect.

Load balancing is the system of making a set of servers participate within the same service and do the equal paintings. the general reason of load balancing is to boom availability, enhance throughput, reliability, hold stability, optimize resource utilization and provide fault tolerant functionality. because the range of servers grows, the chance of a failure everywhere will increase and such disasters ought to be treated cautiously. The capability to keep unaffected service at some stage in any variety of simultaneous disasters is termed as high availability.

Load balancing may be very critical in allotted computing systems to enhance the nice of service with the aid of coping with customer masses that are changing through the years. The needs of incoming requests are optimally distributed amongst to be had machine resources to avoid useful resource bottlenecks in addition to to absolutely utilize available sources. Load balancing also presents horizontal scaling e.g., adding computing sources so as to deal with extended masses.

I.1. PROBLEM STATEMENT

With the fast growth in technology, there's a massive proliferation in our on-line world for its efficient management and minimizing the proliferation errors. Allotted record machine plays a vital role in the control of cloud storage which is sent a number of the numerous servers. in many instances some of those servers get overloaded for coping with the patron requests and others re-principal idle. Huge variety of patron requests on a specific storage server may in-crease the burden of the servers and will cause gradual down of that server or dis-card the purchaser requests if no longer attended well time.

This state of affairs degrades the over-all systems overall performance and will increase the response time. We have proposed an approach that balances the burden of garage servers and efficaciously utilizes the server abilties and resources. From our experimental results and overall performance assessment of proposed set of rules with least loaded set of rules we can finish that our method balances the load, successfully make use of the server abilties and leverage the overall gadget overall performance.

I.2. OBJECTIVES

- **No capital funding worried**. With a simulation tool like CloudSim there may be no installation or protection value.
- **Smooth to use and Scalable.** you can exchange the necessities inclusive of including or deleting resources via converting only a few traces of code.
- **Dangers can be evaluated at an in advance stage**. In Cloud Computing utilization of real testbeds limits the experiments to the dimensions of the testbed and makes the reproduction of effects an extremely tough project. With simulation, you may take a look at your product towards test cases and remedy issues earlier than real deployment with none limitations.
- **No need for strive-and-blunders strategies**. rather than counting on theoretical and vague opinions which can result in inefficient provider overall performance and sales generation, you may take a look at your services in a repeatable and managed environment free of price with CloudSim.

II. LITERATURE REVIEW

Diverse load balancing algorithm have been proposed for cloud computing to provide efficient distribution of load among to be had machines. a number of techniques proposed for load balancing are primarily based on live digital device migration.

Zhao et al. [11], proposed a dispensed load balancing algorithm COMPARE_AND_BALANCE primarily based on sampling to reach an equilibrium solution. They designed and applied a easy version which decreases the migration time of virtual machines with the aid of shared storage and fulfills the zero-downtime relocation. Stay virtual system migration has specifically performance troubles:1) total migration time: it's miles total time taken emigrate virtual machines from its host gadget to the goal machine. 2) Down time: Down time is period of time at which services are not to be had to the customers.

Ma et al. [15] proposed a new version for dispensed load balancing allocation of digital gadget in cloud records center the usage of the TOPSIS method that's one of the most green Multi standards decision Making Function. This technique can discover the most appropriate bodily device in the information middle for the migrated VMs. MCDM approach try and keep away from the stay virtual gadget migration.

Zhang el al. [14], added an method (Statistic based Load stability, SLB) that makes use of the statistical prediction and to be had resource evaluation mechanism to make online aid allocation decisions.

Bhathiya et al. [4] proposed two virtual machine load balancing algorithms, which have been used for load balancing in cloud records center. First algorithm is energetic tracking Load Balancer, which distributes the burden equally to available digital machines in the manner that every virtual machine consist identical range of responsibilities. 2nd algorithm is Throttled Load Balancer, which ensures best a pre-defined number of venture/request are allotted to a single VM at any given time. If requests are gift extra than pre-described range of VM's at a facts middle, than some of the requests will must be queued until the following VM will become to be had. Load balancing algorithms active tracking Load Balancer [4] and Throttled Load Balancer [4] labored properly whilst all of the digital machines of records middle had similar hardware configurations. The main problem takes place whilst the hardware configuration of digital machines is extraordinary and it creates the under load and over load conditions in digital machines. the key undertaking is to increase a load balancing algorithm, so that you can attain the better load balancing among virtual machines that had exceptional hardware configurations in cloud records middle.

Liu et al. [9], cautioned a load balancing digital storage approach that gives a large scale internet statistics garage model and garage as a provider model primarily based on garage of Cloud

Y. Fang et al. [10], proposed a two-level venture scheduling mechanism based on load balancing to satisfy dynamic necessities of customers and obtain a high resource usage.

Wang et al. [12], proposed scheduling algorithm which combines OLB (Opportunistic Load Balancing) and LBMM (Load stability Min-Min) scheduling algorithms that could utilize better executing efficiency and maintain the weight balancing of device. Bhadani et al. [13], endorsed significant Load Balancing coverage for virtual Machines (CLBVM) to balance the load flippantly in a disbursed virtual machine/cloud computing surroundings.

III. ARCHITECTURE

1. Cloudsim : It is an open-source framework, that's used to simulate cloud computing infrastructure and services. it's miles advanced by the CLOUDS Lab employer and is written absolutely in Java. it's miles used for modelling and simulating a cloud computing surroundings as a means for evaluating a hypothesis previous to software program improvement so that it will reproduce exams and results.

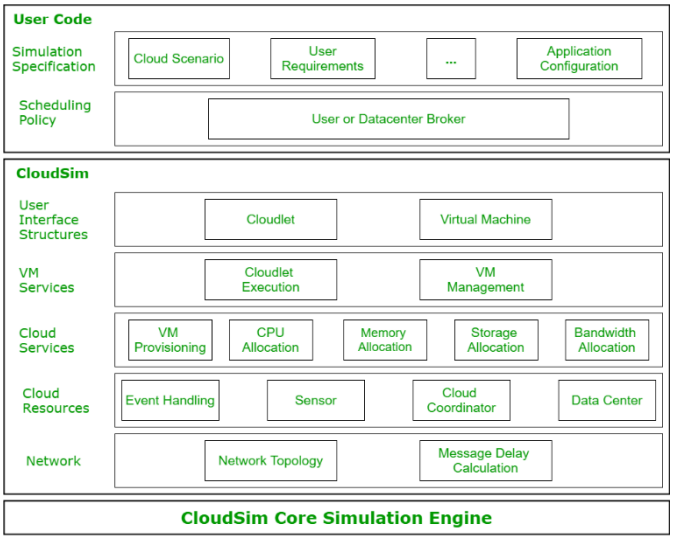

Fig. CloudSim Layered Architecture

Layers of CloudSim Architecture

• CloudSim middle Simulation Engine gives interfaces for the control of resources together with VM, reminiscence and bandwidth of virtualized Datacenters.
• CloudSim layer manages the creation and execution of middle entities which include VMs, Cloudlets, Hosts and so forth. It also handles community-associated execution at the side of the provisioning of resources and their execution and management.
• User Code is the layer managed by way of the user. The requirements can be wriiten by the developer of the hardware specs in this layer according to the state of affairs.

Classes used during simulation are :

- **Datacenter:** used for modelling the foundational hardware device of any cloud surroundings, this is the Datacenter. This magnificence provides strategies to specify the practical necessities of the Datacenter in addition to techniques to set the allocation regulations of the VMs and so on.
- **Host:** this magnificence executes moves related to management of digital machines. It also defines policies for provisioning reminiscence and bandwidth to the virtual machines, as well as allocating CPU cores to the virtual machines.
- **VM:** this elegance represents a digital system via supplying statistics participants defining a VM's bandwidth, RAM, mips (million instructions in step with 2d), length at the same time as additionally supplying setter and getter techniques for these parameters.
- **DatacenterBroker:** is an entity acting on behalf of the user/client. it's miles answerable for functioning of VMs, consisting of VM creation, management, destruction and submission of cloudlets to the VM.
- **CloudSim:** that is the elegance liable for initializing and beginning the simulation environment after all of the necessary cloud entities were defined and later preventing after all of the entities have been destroyed.
- **Cloudlet:** A cloudlet elegance represents any venture this is run on a VM, like a venture of processing, or a memory get right of entry to mission, or a document updating assignment etc. It shops parameters defining the characteristics of a project consisting of its period, length, mi (million commands) and offers techniques in addition to VM elegance whilst additionally presenting strategies that outline a venture's execution time, reputation, price and history.

2. Load Balancer : Load balancing refers back to the procedure of dispensing a hard and fast of duties over a set of assets (computing devices), with the intention of creating their basic processing more efficient. Load balancing can optimize the response time and avoid inconsistently overloading a few compute nodes whilst different compute nodes are left idle.

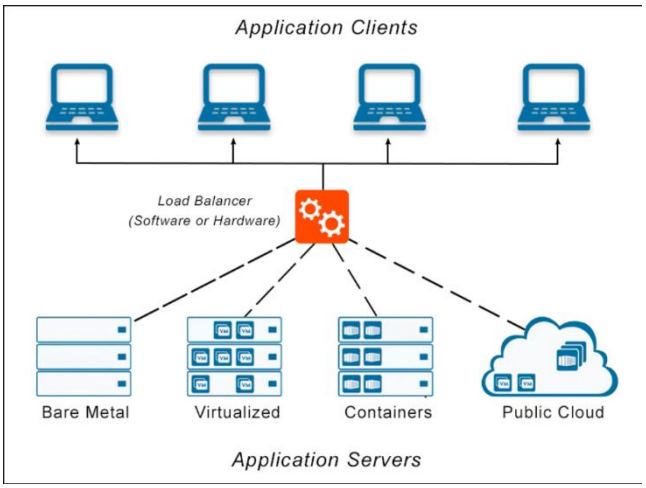

Fig. Load Balancing Architecture

Algorithm Design

Step 1: receives incoming service requests from diverse clients
Step 2: Calculates asked load size of the incoming load request from clients and builds a queue of request.

Step 3: assessments the cutting-edge load repute of the servers inside the server pool periodically the usage of a server display daemon.
Step 4: makes use of a load balancing strategy/algorithm/heuristic to
select appropriate server.

FLOWCHART :

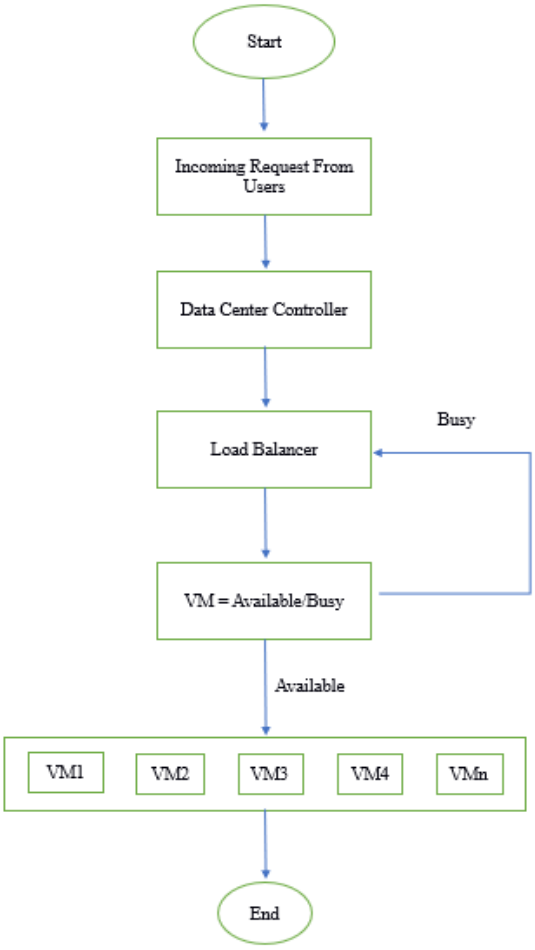

Fig. Flow Chart of Load Balancing

IV. PROPOSED SOLUTION

Approach towards designing the simulation:

The goal of this path assignment is to simulate cloud environments in CloudSim Plus the usage of distinctive load balancers in a networked cloud structure. The assessment of the weight balancers is primarily based at the time and fee of processing the cloudlets which are submitted dynamically in the simulation.

Method closer to designing the simulation:

• **Random:** Random algorithm uses a random wide variety generator to assign cloudlets to VMs.It is a Static algorithm. We use the Random algorithm because the baseline for comparing the simulation.

• **Round Robin**: spherical robin is another static load balancing set of rules wherein each cloudlet might be assigned to VM in a round - robin series. as an instance, the primary request receives VM1, the second request receives the VM2, and so on, with request beginning again at VM1 whilst all VM have been assigned on get entry to request throughout a cycle.

• **Horizontal VM Scaling:** Horizontal VM Scaling is a dynamic load balancing algorithm that allows defining the circumstance to identify an overloaded VM, which in our case is primarily based on modern CPU usage exceeding 70%, a good way to result in the introduction of additional VMs.

V. EVALUTION OF THE SIMULATION

As a part of our assessment, we count on the beneath Null and exchange Hypotheses:

• **Null hypothesis :** Horizontal VM Scaling set of rules plays higher than the Random and spherical Robin algorithms

• **Alternate speculation :** No improvement in overall performance may be seen with Horizontal VM Scaling set of rules in comparison with the Random and RR algorithms.

We evaluated our load balancers with the aid of retaining cloudlet execution time and cost of going for walks of our general workload as our number one criterion.

VI. RESULT AND ANALYSIS

T_test scores for execution time

Random vs Horizontal VM Scaling

t-Test: Paired Two Sample for Means

	Variable 1	*Variable 2*
Mean	13.068	13.012
Variance	96.9172104	96.4728016
Observations	500	500
Pearson Correlation	-0.0137699	
Hypothesized Mean Difference	0	
df	499	
t Stat	**0.08943064**	
P(T<=t) one-tail	0.46438778	
t Critical one-tail	1.64791298	
P(T<=t) two-tail	0.92877556	
t Critical two-tail	1.96472939	

Round Robin vs Horizontal VM Scaling

t-Test: Paired Two Sample for Means

	Variable 1	*Variable 2*
Mean	13.068	13.012
Variance	96.9172104	96.4728016
Observations	500	500
Pearson Correlation	0.29395631	
Hypothesized Mean Difference	0	
df	499	
t Stat	**0.1071618**	
P(T<=t) one-tail	0.45735185	
t Critical one-tail	1.64791298	
P(T<=t) two-tail	0.91470369	
t Critical two-tail	1.96472939	

Random vs Round Robin

t-Test: Paired Two Sample for Means

	Variable 1	*Variable 2*
Mean	13.068	13.068
Variance	96.9172104	96.9172104
Observations	500	500
Pearson Correlation	-0.0160109	
Hypothesized Mean Difference	0	
df	499	
t Stat	**0**	
P(T<=t) one-tail	0.5	
t Critical one-tail	1.64791298	
P(T<=t) two-tail	1	
t Critical two-tail	1.96472939	

We determined a bigger distinction inside the T-take a look at value between Random vs Horizontal VM scaling and round Robin vs Horizontal VM Scaling with the aid of walking T-tests at the corresponding expenses of the burden balancers. This end result is statistically substantial as it proves the fact that round Robin LB algorithm and Horizontal VM Scaling algorithm plays higher than simply randomly assigning these cloudlets to VMs in different dc's.

Furthermore, T-test score of 0.21 and zero.27 for jogging charges among Random vs Horizontal VM Scaling and round Robin vs Horizontal VM Scaling signify that Horizontal VM scaling outperforms the latter algorithms by using a enough margin. however, the difference of zero.016 among T-take a look at scores of Random and round Robin give a belief that Random LB is as suitable as round Robin LB.

However, this isn't the case as Randon LB runs for nearly 3000 seconds for executing 500 cloudlets inside the worst case while multiple cloudlets get assigned to the same VM whereas spherical Robin finishes the simulation in less than 60 seconds. The change-off between walking cost and execution time should be taken into consideration and no longer just each parameter alone.

T_test scores for costs

Random vs Horizontal VM Scaling

t-Test: Paired Two Sample for Means

	Variable 1	*Variable 2*
Mean	10.10558	10.00678
Variance	54.4709506	54.707385
Observations	500	500
Pearson Correlation	-0.0141111	
Hypothesized Mean Difference	0	
df	499	
t Stat	**0.20995728**	
P(T<=t) one-tail	0.41689335	
t Critical one-tail	1.64791298	
P(T<=t) two-tail	0.83378671	
t Critical two-tail	1.96472939	

Round Robin vs Horizontal VM Scaling

t-Test: Paired Two Sample for Means

	Variable 1	*Variable 2*
Mean	10.11322	10.00678
Variance	54.8778131	54.707385
Observations	500	500
Pearson Correlation	0.29361813	
Hypothesized Mean Difference	0	
df	499	
t Stat	**0.27051667**	
P(T<=t) one-tail	0.39343732	
t Critical one-tail	1.64791298	
P(T<=t) two-tail	0.78687464	
t Critical two-tail	1.96472939	

Moreover, T-test rating of zero for execution time among Random vs spherical Robin LB show that the approach of these two Load Balancers are equal. The T-take a look at rating of zero.eleven for execution instances of spherical Robin vs Horizontal VM Scaling LB is valid as spherical Robin takes much less time to execute for 500 cloudlets than Horizontal

VM Scaling. Random vs Horizontal VM Scaling indicates a T-check score 0f 0.09 for execution time. This result concludes that Horizontal VM Scaling takes much less time to execute 500 cloudlets than Random Load Balancing.

Random vs Round Robin

t-Test: Paired Two Sample for Means

	Variable 1	*Variable 2*
Mean	10.11322	10.10558
Variance	54.8778131	54.47095056
Observations	500	500
Pearson Correlation	-0.0171117	
Hypothesized Mean Difference	0	
df	499	
t Stat	**0.01619896**	
P(T<=t) one-tail	0.49354107	
t Critical one-tail	1.64791298	
P(T<=t) two-tail	0.98708214	
t Critical two-tail	1.96472939	

We found that the p-values for Horizontal VM Scaling LB to be around ~ 0.40, when compared to RoundRobin and Random load balancers. This shows statistical significance as a significantly higher p-value (typically >= zero.05) indicates strong proof for helping the null hypothesis.

VII. CONCLUSION & FUTURE SCOPE

Random LB set of rules took higher time to execute when compared to spherical Robin and Horizontal VM Scaling LB algorithms. It took > 300 seconds to complete executing 500 cloudlets, at the same time as spherical Robin LB took much less than 60 seconds and Horizontal VM scaling took round one hundred ten seconds.

Based totally on our overall performance evaluation, we see an development in overall performance in terms of value and processing time when Horizontal VM Scaling is used vs different easy static algorithms for load balancing like Random LB and spherical Robin LB.

Inside the destiny, we can paintings to optimize the cloud resources similarly and enhance cloud-based totally software performance, together with considering more functions. for instance, the algorithm could be tested based on the variety of violations and the migration rely for higher overall performance. additionally, the set of rules can be comprehensively compared to other present algorithms.

REFERENCES

Website

[1]https://en.wikipedia.org/wiki/Load_balancing_(computing)

[2] https://www.geeksforgeeks.org/what-is-cloudsim/

[3]https://journalofcloudcomputing.springeropen.comJournal

[4] Kaur, J. “evaluation of load balancing algorithms in a Cloud”, international magazine of Engineering research and packages(IJERA), Vol. 2, trouble three, may also-Jun. 2012, pp.1169-1173.

[5] Nusrat Pasha, Dr. Amit Agarwal1 and Dr. Ravi Rastogi2, “spherical Robin approach for VM Load Balancing algorithm in Cloud Computing environment”, ijarcsse, volume four, difficulty five, may also 2014

Conference

[6] M. Randles, D. Lamb, and A. Taleb-Bendiab, “A Comparative examine into distributed Load Balancing Algorithms for Cloud Computing,” in WAINA’10: Proc. of the 2010 IEEE twenty fourth Int. convention on advanced facts Networking and packages Workshops, Perth, Australia, Apr. 2010.

[7] M. Randles, D. Lamb, and A. Taleb-Bendiab, “A Comparative look at into distributed Load Balancing Algorithms for Cloud Computing,” in WAINA’10: Proc. of the 2010 IEEE 24th Int. conference on superior records Networking and packages Workshops, Perth, Australia, Apr. 2010.

[8] M. Randles, D. Lamb, and A. Bendiab, “A comparative examine into disbursed load balancing algorithms for cloud computing”, IEEE 24th worldwide conference on advanced statistics Networking and programs Workshops (WAINA), pp.551-556, April 2010.

Tech recordTech Report

[9] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia, “Above the clouds: A berkeley view of cloud computing,” UC Berkeley Technical report UCB/EECS-2009-28, February 2009.

Thesis

[10] Y. Teo and R. Ayani, “evaluation of load balancing techniques on cluster-based web servers, Simulation”, the magazine of the Society for Modeling and Simulation, vol. seventy seven, pp. 185-189, November- December 2001.

Book

[11]Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing a practical technique, TAT A McGRAW-HILL edition 2010.