

# Comparative Analysis of Load Balancing Approaches in Cloud Computing

NAMAN MUNET<sup>1</sup>, MANISH SHARMA<sup>2</sup>

<sup>1,2</sup>Department of Computer Engineering, Poornima College of Engineering, Jaipur, Rajasthan, India

*Abstract -- Cloud computing is the following stage in advancement of Internet. It is help of IT as an administration through Internet. Cloud in cloud computing gives the methods through which everything from registering energy to figuring infra, applications, business procedures to individual joint effort instruments can be conveyed to the clients as an administration at whatever point and wherever they require. Load adjusting is isolating the measure of work that a PC needs to do between at least two PCs so no single gadget is overpowered. Web servers utilize stack adjusting to uniformly part the activity stack among a few unique servers, and in this manner it enhances the execution and unwavering quality of the sites, applications and different administrations. We utilize hereditary calculation approach also called as genetic algorithm for load adjusting in distributed computing. Our proposed work is more fitting than the present strategies work, as we have compared different load balancing algorithm based on different parameters.*

*Index Terms – Cloud computing, Load balancing, Stack Adjusting, Genetic algorithm*

## I. INTRODUCTION

Cloud is an arrangement of equipment, programming, stockpiling, administrations and interfaces that empower the conveyance of programming, foundation and capacity over the web in view of the client request. It chips away at pay as you go show and gives on-request administrations to the clients. The fundamental empowering innovation for distributed computing is virtualization. Virtualization programming isolates a physical figuring gadget into at least one "virtual" gadgets, every one of which can be effectively utilized and figured out how to perform processing errands.

A load balancer goes about as the "movement cop" sitting before our servers and directing customer request for over all servers

equipped for satisfying those solicitations in a way that expands speed and limit use and guarantees that no server is exhausted, which could debase execution. On the off chance that a solitary server goes down, the load balancer diverts activity to the other online web servers. At the point when another server is added to the server gathering, the load balancer consequently begins to send demands to it. In this way, a load balancer performs the following functions:

- Appropriates customer demands or system stack productively over numerous servers.
- Ensures high accessibility and unwavering quality by sending demands just to servers that are online.
- Provides the adaptability to include or subtract servers as request directs.

## II. ABOUT GENETIC ALGORITHM

A Genetic Algorithm (GA) is a pursuit calculation in view of the standards of advancement and regular hereditary qualities. GA consolidates the abuse of past outcomes with the investigation of new territories of the inquiry space. By utilizing survival of the fittest methods joined with an organized yet randomized data trade, a GA can mirror a portion of the imaginative style of a human hunt. An age is an accumulation of counterfeit animals (strings). In each new age, an arrangement of strings is made utilizing data from the past ones.

Genetic algorithm is a class of optimal search algorithms which simulate biological evolution and genetic mechanism. After the underlying populace is created, it develops better and better estimated arrangements in view of wellness from age to age. During every generation, the individual is chosen based on the fitness of the different individuals in a certain problem domain. Then the individuals

combine, cross and vary by the genetic operators in natural genetics and then a new population representing a new solution set is produced. The best arrangement will be chosen, and the poor arrangements will be wiped out after a few ages of development. Based on the real situation of cloud computing this paper proposes a comparison strategy to find out which algorithm is perfect.

Genetic Algorithm (GA) is much well known for taking care of NP-Complete issues. GA is one of system which has a place with the class of developmental calculations which produces arrangements propelled by normal advancement. The Simple GA Concepts are as per the following:

Population- It is an arrangement of possible solutions for proposed issue.

Chromosome- The individuals in the population.  
Gene- A variable in a chromosome.

Fitness Function- A type of an objective function used to find out how close the arrangement is accomplishing the set point.

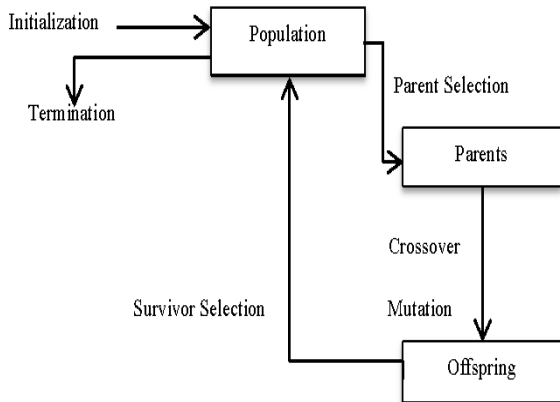


Fig 1: Flowchart of Genetic Algorithm

The GA evolves these three operators:-

Selection- The thought is to offer inclination to the individuals with great wellness scores and enable them to pass their genes to the progressive ages.

Crossover- This shows mating between the individuals. Two individuals are chosen using selection operator and crossover sites are picked

arbitrarily. At that point the genes at these crossover sites are traded in this way making a totally new individual (crossover).

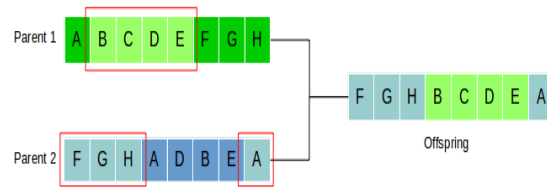


Figure 2: Crossover

Mutation- The key thought is to embed irregular genes in the offspring to keep up the decent variety in populace to dodge the untimely convergence.

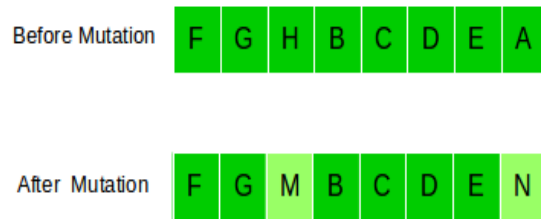


Figure 3: Mutation

### III. LITERATURE REVIEW

There are many different techniques given by different author for load balancing in cloud Computing, from which some of them are discussed as below

Tingting Wang, Zhaobin Liu, Yi Chen, Yujie Xu, Xiaoming Dai [3] has provided the idea of using two fitness function such that first fitness function calculates the time performance of VM and Second fitness calculates the intensity of different node’s load using Variance. Numbers are used for chromosome representation which can be called as permutation encoding. A single point crossover and Mutation is performed after selecting two best fitness chromosomes. Algorithm end when terminating condition is reached. Authors had compared there proposed method with original GA and shown the results. Also all the requests are of same priority was the consideration done by authors.

Neha Gupta, Parminder Singh [9] has proposed a method in which the tasks are divided in

to sub tasks. The reason behind the division is the parallel execution of jobs on multiple machine. The GA works for distribution of sub tasks to the multiple processors such that execution of one full task will be faster compared to working on one system. Initial population consist of two array consists of VMs and request respectively. Author did not provide any knowledge about use of Crossover. Mutation is applied to fittest chromosome to find the best VM for the job. The stimulation of proposed system is done in Cloud Sim 2.0.

Mayur S. Pilavare, Amish Desai [7] has proposed the method in which the VMs are prioritized by using Logarithmic Lease Mean Square Matrix. This Gives priority to the VMs and initial population is made such that VMs having high priority gets the Jobs first. After these steps, GA operators are applied to obtain optimum output. Here there is no priority given to the jobs.

#### IV. CLASSIFICATION OF LOAD-BALANCING APPROACHES

Cloud is comprised of gigantic resources. Administration of these resources requires capable arranging and proper outline. While designing an algorithm for resource provisioning on cloud the designer must think about exceptional cloud situations and must be aware of the issues that are to be dictated by the proposed calculation.

Distributed computing can have either static or dynamic condition in light of how designer arranges the cloud requested by the cloud provider.

In static algorithm the activity is isolated equitably among the servers. This algorithm requires a past learning of system assets, with the goal that the decision of moving of the heap does not rely upon the present state of system. A Static algorithm is legitimate in the framework which has low discrepancy in stack. Some of the static loads balancing algorithms are:

1) Round Robin Load Balancer: In round robin settled quantum time is given to the occupation. Basic complement in round robin is on sensibility and time imperative. It uses the ring to orchestrate the assembled errands. It uses rise to period to finish each errand. On the off chance that there

ought to be an event of overpowering weight, round robin takes a whole deal to finish all the given assignments. On the off chance that there ought to be an event of greater errands it requires longer speculation for wrap up.

- 2) OLB + LBMM: It is a two-phase scheduling algorithm that combines OLB (Opportunistic Load Balancing) and LBMM (Load Balance Min-Min) scheduling algorithm to use better executing proficiency and keep up the load balancing of the framework. This consolidated approach helps in a productive usage of assets and upgrades the work effectiveness.
- 3) Min-Min: It starts with an arrangement of every unassigned task. Above all else, least completion time for all task is found. At that point among these base circumstances the base esteem is chosen which is the base time among all the undertakings on any resources. At that point as indicated by that base time, the errand is booked on the relating machine. At that point the execution time for every single other undertaking is refreshed on that machine by including the execution time of the appointed undertaking to the execution times of different errands on that machine and relegated assignment is expelled from the rundown of the undertakings that are to be allotted to the machines. On the other hand a similar method is taken after until all the undertakings are relegated on the assets. In any case, this approach has a noteworthy downside that it can prompt starvation.
- 4) Max-Min: Max-Min is relatively same as the min-min calculation with the exception of the accompanying: in the wake of discovering least execution times, the most extreme esteem is chosen which is the greatest time among every one of the assignments on any assets. At that point as indicated by that most extreme time, the undertaking is planned on the relating machine. At that point the execution time for every single other undertaking is refreshed on that machine by including the execution time of the relegated errand to the execution times of different errands on that machine and doled out assignment is

expelled from the rundown of the assignments that are to be allotted to the machines.

In dynamic algorithm the lightest server in the entire system or framework is hunt and favoured down adjusting a stack. For this real time communication with network is necessary which can increase the traffic in the system. Here current situation with the framework is utilized to make decisions to handle the heap.

- 1) Throttled load balancing: This count depends on the theory of fitting request of virtual machine. The undertaking supervisor makes a rundown of virtual machines. By using the once-over, client request allotted to the related machine. If the size and capacity of the machine is proper for request, by then the occupation is given to that machine. This computation is better than round robin count.
- 2) Honey bee foraging algorithm: The principle thought behind the calculation is gotten from the conduct of bumble bees for finding and harvesting sustenance. M. Randles et al. proposed a decentralized bumble bee based load adjusting strategy that is a nature-propelled calculation for self-association. For this situation the servers are gathered under virtual servers (VS), every VS having its own virtual administration lines. Every Server preparing a demand from its line figures a Comparative Analysis of Existing Load Balancing Techniques in Cloud Computing benefit or reward, which is practically equivalent to the quality that the honey bees appear in their waggle move. On the off chance that this benefit was high, at that point the server remains at the current virtual server generally then the server comes back to the scrounge. The calculation executes as the framework decent variety increments. In any case, it has a major impediment that it doesn't build the throughput as the framework estimate increments.
- 3) Ant Colony: Z Zhang et al. proposed a load balancing algorithm based on ant colony and complex network theory (ACCLB) in an open cloud alliance this technique brings into utility little world and scale free qualities of a mind

boggling system to accomplish upgraded stack adjusting this strategy conquers heterogeneity it adjusts to dynamic conditions, has great adaptation to internal failure and is extremely steady and improves the general execution of the framework.

- 4) Carton: R Stanojevic et al. proposed this system it works by binding together the utilization of LB (Load Balancing) and DRL (dispersed Rate Limited) , stack adjusting is utilized to allot similarly the diverse undertakings/employments to different servers DRL guarantees that the assets are conveyed in a way to keep up an impartial and reasonable distribution of assets DRL likewise changes with the limit of servers for dynamic workloads to guarantee comparable level of execution at all servers the fundamental factor handled by this calculation is use control without interminable transfer speed suspicion the calculation is basic, simple to actualize and has low correspondence and calculation overhead.

## V. COMPARATIVE ANALYSIS OF DIFFERENT APPROACHES

There are some subjective measurements that can be improved for enhanced load adjusting in distributed computing.

Throughput - It is the total number of processes that have completed execution for a given size of time. It is important to have high through put for better operation of the system.

Overhead - It portrays the measure of overhead amid the execution of the load balancing algorithm. It is a composition of progress of tasks, inter process communication and inter processor. For stack adjusting procedure to work legitimately, most reduced overhead ought to be there.

Fault Tolerance - We can characterize it as the ability to execute stack adjusting by the reasonable calculation without discretionary connection or hub disappointment. Each load adjusting calculation ought to have great adaptation to non-critical failure approach.

Response Time - In Distributed framework, it is the time taken by a specific load adjusting strategy to react. This time ought to be limited for better execution.

Resource Utilization - It is the parameter which gives the data inside which introduce the resource is utilized. For productive load adjusting in system, optimal resource ought to be used.

Scalability - It is the capacity of load adjusting calculation for a framework with any limited number of processor and machines. This parameter can be enhanced for better framework execution.

Performance - It is the general proficiency of the framework. On the off chance that every one of the parameters is upgraded then the general framework execution can be moved forward.

Migration Time - It is the measure of time for a procedure to be exchanged starting with one framework hub then onto the next hub for execution. For better execution of the framework this time ought to be constantly less.

So based on the above parameters comparing different load balancing algorithm we have come to

a point to find out which algorithm is suitable for load adjusting in distributed computing. The relationship of these computations exhibit positive and negative results and we depict this as high and low term.

In Max-Min, essentials are prior known. So it works better and gives high throughput. Close by this, dynamic load modifying requires simply current state of the structure and has more overhead and adjustment to inside disappointment. Honey bee has high throughput and low response time. It has low overhead and execution since high need errands can't work without VM machine. Insect province is basic calculation and less perplexing. Table 1 gives a definite examination of various calculations over various parameters like decency, execution, speed, multifaceted nature. We suggest that, Genetic Algorithm is more proficient as per following truths, Genetic Algorithm consider reasonable for circulate the heap, it has high throughput, great reaction time and less mind boggling than different calculations. The noteworthy ideal position of Hereditary Algorithm is time limitation and use measure up to period to complete each endeavor.

Table 1: Comparison of Existing Load Balancing Technique

Parameters	Round Robin	Min-min	Min-max	OLB +LB MM	Throttle	Carton	Honey Bee	Ant Colony	Genetic algorithm
Nature	Static	Static	Static	Static	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic
Throughput	High	High	High	High	Low	High	High	High	High
Overhead	High	High	High	Low	Low	Low	Low	High	Low
Fault Tolerance	No	No	No	No	Yes	No	No	No	No
Migration	No	No	No	No	Yes	No	Yes	Yes	Yes
Response Time	Fast	Fast	Fast	Slow	Fast	Fast	Slow	Slow	Fast
Resource Utilization	High	High	High	High	High	High	High	High	High
Speed	N/A	Fast	Slow	Slow	Fast	Fast	Fast	Fast	Fast
Complexity	Low	Low	Low	High	Low	High	Low	No	Low

## VI. CONCLUSION AND FUTURE SCOPE

In this paper, we have exhibited examination of various load adjusting calculations for distributed computing, for example, round robin (RR), Min-Min, Max-Min, Ant state, Genetic, honey bee and so forth. We depicted points of interest and constraints for these calculations indicating demonstrating realizes different conditions. The key bit of this paper is examination of different estimations considering the traits like sensibility, throughput, and adjustment to inward disappointment, overhead, execution, and response time and resource utilization. The imperative of existing work is that each appropriated processing count does not address the related issues like conventionality, high throughput and value.

Load adjusting is one of the primary difficulties in distributed computing. It is required to disperse the dynamic neighborhood workload equally over every one of the hubs to accomplish a high client fulfilment and asset use proportion by ensuring that each registering asset is appropriated proficiently and decently. So in this paper we have looked at different algorithm of load adjusting in Cloud Computing. Also, we have reasoned that we can utilize a specific algorithm per our prerequisite/require. Yet, as we realize that the

Cloud Computing covers an extremely tremendous zone, it is appropriate to both small and large scale region. So there is a need to build up a versatile algorithm which is appropriate for heterogeneous condition and ought to likewise diminish the cost.

## REFERENCES

- [1] Hussain A Makasarwala , Prasun Hazari, "Using Genetic Algorithm for Load Balancing in Cloud Computing" in ECAI 2016-International Conference-8th Edition, Electronics, Computers and Artificial Intelligence, 30 June-02 July, 2016, Ploiesti, ROMANIA.
- [2] Monika Lagwal, Neha Bhardwaj, "Load balancing in Cloud Computing using Genetic Algorithm" in International Conference on Intelligent Computing and Control Systems ICICCS 2017.
- [3] Tingting Wang, ZhaobinLiu, Yi Chen, Yujie Xu, Xiaoming Dai, "Load Balancing Task Scheduling based on Genetic Algorithm in Cloud Computing" in 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing.
- [4] Priyanka Gonnade, Sonali Bodkhe, "An Efficient load balancing using Genetic algorithm in Hierarchical structured distributed system" in International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970) Volume-2 Number-4 Issue-6 December-2012.
- [5] Md. Shahjahan Kabir , Kh. Mohaimenul Kabir and Dr. Rabiul Islam, "PROCESS OF LOAD BALANCING IN CLOUD COMPUTING USING GENETIC ALGORITHM" in Electrical & Computer Engineering: An International Journal (ECIJ) Volume 4, Number 2, June 2015.
- [6] Parveen Kumar , Er.Mandeep Kaur, "LOAD BALANCING IN CLOUD USING ACO AND GENETIC ALGORITHM" in International Journal of Scientific Research Engineering & Technology (IJSRET), ISSN 2278 – 0882 Volume 4, Issue 7, July 2015.
- [7] Mayur S. Pilavare, Amish Desai "A Novel Approach towards Improving Performance of Load Balancing Using Genetic Algorithm in cloud computing" in IEEE sponsored 2nd International Conference on Innovations in Information Embedded and Communication Systems ICIECS. IEEE 2015
- [8] D. Saranya, L. Sankara Maheswari, "Load Balancing Algorithms in Cloud Computing: A Review", ISSN: 2277 128X/Volume 5, Issue 7, July 2015
- [9] Neha Gupta, Parminder Singh "Load Balancing Using Genetic Algorithm in Mobile Cloud Computing" in International Journal of Innovation in Engineering & Technology (IJIET) 2014.
- [10] Nayandeep Sran, Navdeep Kaur "Comparitive Analysis of Existing Load Balancing Techniques in Cloud Computing" in International Journal of Engineering Science Invention, ISSN: 2319-6734 Volume 2 Issue 1, January 2013.
- [11] Amritpal Singh "A Review of Existing load Balancing Techniques in Cloud Computing" in International journal of Advanced Research in Computer Engineering & Technology(IJARCET) Volume 4 Issue 7, July 2015.