Genetic Load Balancing Algorithms in Cloud Environment

Neha Sharma, Jaspreet Singh, Rohit Bajaj

*Abstract***:** ***Cloud services are broadly used in accomplishment, logistics, and computerized applications. It is not an easy technology, it consists of lots of issues like virtual machine management, scheduling of virtual machines, data security, providing resources (like hardware and software) and load balancing. The issue of load balancing arises in abundant applications but essentially they play an essential role in the application of cloud environment. Load balancing distributes a task into subtasks that can be performed together and mapping each of these programs to computational resources like a computer or a processor, the complete processor time will be decreased with upgrade processor usage. To solve the issue of load balancing various algorithms are proposed by authors in the recent past and one of them is genetic algorithms. The paper describes insight survey some genetic load balancing algorithms used in a cloud environment by taking into consideration different factors, further we have analyzed and correlated all these factors in order to do a comparative assessment based upon different parameters so as to identify the proficiency of different genetic algorithms.***

Keywords: - Cloud computing, Load balancing, Genetic Algorithm for Load Balancing.

I. INTRODUCTION

Cloud computing is playing a very important role in computing that makes adoption of real-time exchanging of data with the help of internet network for large-scale of cloud computing with the help of virtualization approach [1]. The main benefits of cloud computing are cost efficiency and others include rapid development, scalability, better IT resource management, flexibility, greater mobility and better performance [13]. According to user requirement, it provides distinct IT services and resources. The services provided by cloud computing are Software as a service (SaaS), Platform as a service (PaaS) and Infrastructure as a service (IaaS) [2]. The services that are proposed by the third party with the help of the internet through pay per user use that is known as a public cloud [3]. The organization which does not support services Publically that is called private cloud [1]. The cloud which is shared by several associations is known as community cloud. Hybrid cloud contains the advantages of public and private cloud. The various component of cloud computing are [14]:

a) On demand access: - Cloud computing supply on demand services to the customer as per their requirements.

b) Rapid elasticity: - According to the user demand, the amount of resources are increased at any time.

c) Broad network access: - The resources of cloud computing area broadly accessed through distinct mechanisms.

d) Resource pooling: - In cloud computing resources are pooled by using different models that are provided by the providers to their users. According to the customer requirements, all the resources are dynamically assigned and reassigned.

The overview of cloud computing paradigm is shown below in figure [1]:

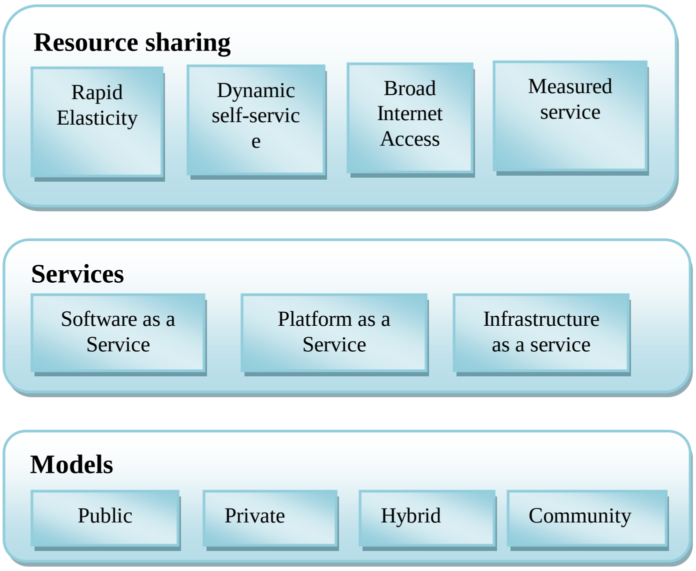

Figure 1: Overview of Cloud Computing [11].

(a) Role of Load balancing in Cloud computing

Load balancing approach is a technique which is used to distribute tasks between two or more resources. In a load balancing, high throughput is accomplished by using possible resources [4]. Load balancing method can be generally classified as dynamic or static, periodic or non-periodic and centralized and decentralized.

As demand from the customer is handled by randomly on the basis of nodes selection, thus the workload on the nodes are varied such as some links are under the burden and some are overburden which precisely affects the cloud services standard. The Importance's of load balancing are [15]:

Revised Manuscript Received on July 10, 2019.

Neha Sharma, M.tech (CSE) Scholar Chandigarh University, Mohali, Punjab, India.

Jaspreet Singh, Department of CSE, Chandigarh University, Mohali, Punjab, India.

Rohit Bajaj, Department of CSE, Chandigarh University, Mohali, Punjab, India.

i) On demand resources are quickly accessible.
ii) Under the circumstances of low and high load resources are efficiently utilize.
iii) The cost of used resources is reduced.
iv) In case of low load energy is saved.

b) Benefits of Load balancing for cloud services

Cloud services are defined as the services that are accessible to customer on demand with the help of internet from a cloud computing. It is designed to support scalable, easy, access to services, application and resources, and are completely handle by a cloud service provider. The benefits of load balancing for cloud service provider are:

i) Cost-effectiveness: - Load balancing provides great performance at a low price.

ii) Flexibility and Scalability: - The load balancing algorithm is flexible and scalable because the system for which load balancing techniques are executed may be altered in size after some time. These types of situations must be handled by load balancing techniques.

iii) Priority: - Higher priority of the resources gets a superior chance to executed [14]. The basic architecture of load balancing is shown below:

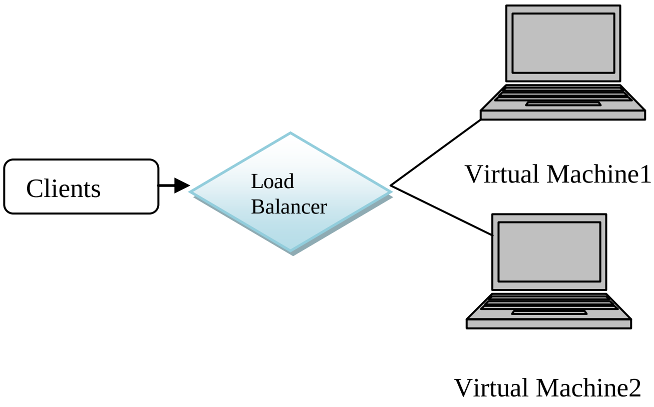

Figure 2: Architecture of Load Balancing [12].

(c) Load balancing approaches in cloud computing

Load balancing approaches broadly fall under two categories such as:
i) Static load balancing
ii) Dynamic load balancing.

i) Static Load Balancing: - The static load balancing approach uses the past knowledge and prior information load is shared among the virtual machines. These approaches are appropriate for the limited workload in the cloud. Different types of static approaches are Round Robin algorithm, Randomized algorithm, Max-Min approach, and Opportunistic Load balancing algorithm [2]. Objective of Static load balancing is it divides the traffic equally to whole server [14].

ii) Dynamic load balancing:-In dynamic approach, decisions are taken on the basis of present state of the system. It can be divided into centralized and Semi distributed load balancing techniques. Load balancing problems can be deal with nature inspired intelligent approaches like Particle Swarm Optimization (PSO), Artificial-bee Colony (ABC), Genetic Algorithm (GA) and Ant Colony System (ACS) [2].

(d) Load Balancing matrix

Load balancing matrices are discussed below [17]:

i) Throughput: - It is used to determine the all jobs whose implementation has been executed. When the System performance is improved then the throughput is high.

ii) Fault Tolerance:-It is defined as the recovery from the failure node.

iii) Migration Time: - It is the time to migrate the task from one node to another node. Migration time should be reduced to enhance the performance of the system.

iv) Response Time: - Time is taken by the load balancing technique to response a job in a system.

In the below section various benefits of load balancing in cloud computing, various genetic algorithms and comparison based on some parameters are discussed.

II. VARIOUS GENETIC ALGORITHMS FOR CLOUD ENVIRONMENT

Nowadays, research and application of genetic algorithm has been quickly developed and broadly utilize. It can be used to determine the complex problem in science and engineering [16].The advantages of genetic algorithm using load balancing in cloud computing are discussed below [16]:-
a) Genetic algorithm examines parallel from a population of points, not a single point.
b) It uses probabilistic selection rule, not deterministic.
c) Genetic algorithm uses fitness function, which is access from objective functions.
d) Genetic algorithm support multi objective optimization.
e) It is used is to solve the current issue of resource deficiency.

2.1 Simple Genetic Algorithm [5]: - Simple genetic algorithm is a method of natural-selection approach. The procedure of genetic algorithm is simple; it is not more complicated than imitate a string and exchanging limited strings. Genetic algorithm performance depends upon a suitable mixture of exploitation and exploration. There are three processes that are followed by genetic approach are Selection, crossover, and mutation. Population from which two chromosomes is selected that are known as selection. In a crossover, we crossover the probability of the parent's to construct a new child.

In the case of mutation the best value is selected. Every new creation of chromosomes is created by using previous information. Chromosomes are chosen based on their fitness values. On the preferred solution mutation or crossover is applied and a genetic approach produces a solution and advance result. Computational working of a simple genetic algorithm as specified in [5] can be explained in below mention steps:

Step1. Initial Population: The execution time and failure rates of every virtual machine which help in performing task execution define the initial population. The virtual machine resource that is used for task execution is considered as the initial population.

Step2. Cross over value calculation: Every chromosome x fitness value is explored under the populace. Two parent chromosomes are chosen from a populace depending upon their fitness values. Generally, the fitness value increases with increment in the populace which also results in increasing the crossover possibility due to which a new offspring is created from the parent node. Offspring become an exact replica of parents in case if no crossover is performed. New offspring at each locus are recreated with the possibility of offspring.

Step3. Best value Calculation: From the calculated crossover value, the best value is chosen. To run an algorithm for longer time duration, the newly generated populace is utilized.

2.2 Job spanning time and load balancing genetic algorithm [6]:- In this algorithm when the initial population is created, it develops better and better exact result based on fitness value from one generation to other. The node is chosen based on their fitness value on a distinct domain. Then the nodes combined, cross and vary by genetic operators in natural genetic and then new offspring is produced. The best result is chosen and the poor result will be eliminated after various generation of calculation. Computational working of genetic adaptive search algorithm as specified in [6] can be explained in below mention steps:

Step1. Representation of chromosomes: - The chromosomes are represented by its length and gene value. Length is equal to its number of task and gene value is equal to the number of a node on which job is accomplished.

Step2. Population Initialization: - Simple genetic algorithm randomly establishes initial population and every single node have low performance.

Step3. Fitness Value: - It is calculated on the basis of the quality of the nodes in the population. Operators involved are Selection, Crossover, and mutation based on fitness value.

Step4. Selection: - The variety of the population is ensured by the mutation and crossover operation the node with good performance is demolished by these operations. The nodes with good fitness value have a large probability to be selected.

Step 5.Adaptive probabilities of crossover and mutation operation: Crossover operation is to create new offspring by replacement and improving part of two finally chosen parent node. Each and every node has a certain probability to exchange with another node. There are one point and multi-point crossover.

2.3 Genetic algorithm with multi parent's recombination [7]:- To produce new offspring it uses two or more parents. In this, two mechanisms are introduced such as dialogue crossover and gene scanning. In dialogue crossover, generalize reliable, commonly crossover at n point and in case of gene scanning, it creates one offspring from n>1 parent in such a way firstly it assigns a number of marker one for every parent and one for offspring. The position of the offspring from left to right is traverse by child maker one at a time. For every position, parents maker are renewed so that when selecting a value for gene presently marked in the offspring, the choice indicated by parents maker [7]**.**

Two types of genetic algorithm with multi parent's recombination are used in this approach that is explained in the below mention steps [7]:

Step1. Gene Scanning technique: - Gene scanning technique has done on the basis of below-mentioned scanning approaches:

i) Uniform scanning: - In this, two children and two parents are traverse from left to right and for each position first child is selected randomly whether to drive from parents 1 and parents 2(the next offspring is established by reversing the decisions).

ii) Occurrence-based scanning: - It is based on the assumption that the value which exists most often in a precise position in the parents (selected on the basis of their fitness) is apparently best possible value to be chosen. If no value exists more than any other one (The majority function is not definite) then the value which comes first is chosen.

iii) Fitness based scanning: - It uses roulette wheel selection when determining which parent an allele will be inherited.

iv) Adapting scanning to different representation types: - It is attainable to define scanning process for distant representation types by altering the marker update mechanism. All the position are independent, the marker modified updates are simple the parent marker all are initialized to the first position in every parent and each step of the marker is increased by one (traversing parent left to right).

Step2. Adjacency based crossover: - After scanning technique, the second step is adjacency based crossover and it is a special case of scanning that is designed for order based representations where respective positioning of the value is essential.

2.4 Proposed Parallel Genetic Algorithm [8]:- In a parallel genetic algorithm, it breaks the large task into subtasks and to execute every subtask it uses distinct processors. There are many distinctive ways in the genetic algorithm where Divide and conquer technique can be applied. It is also achievable for multi populations. The essential characteristic of a parallel genetic algorithm for multi-deme is known as coarse gained Gas, are various huge subpopulation and migration. They offer basic changes in GA operations and very difficult to recognize [8].
Computational working of genetic adaptive search

algorithm as specified in [8] can be explained in below mention steps:

Step1. The master scheduler that is the in charge of scheduling selects the slaves. This selection is based upon the transmission overhead involved and computational potential of the slave processor. The processor which is too slow will not be used as a slave.

Step2.The master has the population of the chromosome for which the fitness function is to be calculated.

Step3. Each slave calculates the fitness of a fraction of the population in the master scheduler and returns the value.

2.5 Modified Genetic Algorithm [9]:- In modified genetic algorithm, the initial population is created by improvement in Max-min technique. Two operations are performed on initial population such as mutation and crossover and have added new creates schedules to the population. In the enhanced max-min technique, the abundant task is scheduled to moderate resources and using speedy resources rest all tasks are executed. This is because the makespan value is determined by expected finishing time of an abundant task. This dependency is removed only when the makes span can be optimized for a particular case. Instead of appointing an abundant task to moderate resources, if task schedule, having ordinary finishing time to moderate resources than this can optimize the result. Computational working of genetic modified genetic algorithm as specified in [9] can be explained in below mention steps:

Step1.The initial population is created randomly by using enhanced Max-Min approach.

Step2. By using fitness function, chromosomes are calculated which is based on makes span and then finest chromosomes are selected and the distinct crossover is used on chromosomes pair.

Step3. Additionally, exchange and progress mutation operation is enforced to create new offspring and then these new result is also figured out for the condition that if they are good to perform in additional operations.

2.6 Genetic Adaptive Search [10]:- It uses a mixture of a real and binary coded genetic algorithm. It depends on the nature of the architecture such as the variables are differently coded and for every variable differently accomplished the genetic phases (crossover and mutation).In the binary coded genetic algorithm, the variables are depicted in binary string create with 0 and 1. The variables that are represented by continuous variables are known as a real-coded genetic algorithm. Variable by variable crossover and mutation phase have to be covered. The binary genetic activities are accomplished in the case of binary-coded variables and real-genetic activities are achieved in the case of real-coded variables. The constraint and fitness action are determined as normally [10]. It uses two types of the genetic algorithm such as real Coded genetic algorithm and binary coded genetic algorithm. Computational working of genetic adaptive search algorithm as specified in [10] can be explained in below mention steps:

Step1. Binary coded genetic algorithm: - The variables are performed in the binary string such as 0 and 1. It performs two operations such as one point crossover and Binary mutation. An individual crossover point on both parents string is chosen. For example variables can be represent as (x_1 x_2 x_3 x_4...$x_{n)}$,x_1 is either 0 or 1.

Step2. In real-coded genetic algorithm: - In this algorithm, the variables are performed in the continuous variable. The procedure of binary-coded GA cannot be used for real-coded GA. To handle real-coded variables genetic operations have to be changed.
For example, variables can be represented as (y, a number in [y^l, y^u]) y^l is lower bound and y^u is upper bound. Depending on its behavior every variable is coded in gene.

2.7 Multi-agent genetic algorithm (MAGA) [16]:- Multi-agent genetic algorithm is a hybrid algorithm of genetic algorithm and multi agent approach. It improves the performance and convergence time as compare to traditional genetic algorithms. It has evident superiority, particularly when handling very high dimensional, large scale, Dynamic and complex optimization issue. Computational working of multi-agent genetic algorithm as specified in [16] can be explained in below mention steps:

Step1. Agent Genetic algorithm: -In agent genetic algorithm, every node act as an agent. This agent is efficient to limited approach, self learning, cooperation, competition and embracing the function of global development over the communication between both agent and environment, and agent and agent.

Step2. Individual Survival Environment: - Each and every node act as an agent, able to modification, impacting and sensing environment individually, thus acquire its own characteristics.

Step3. Genetic Operator: - It consist the neighborhood orthogonal crossover operator that accomplish combination among the agents, neighborhood competition operator execute the action of competition between all agents, and self-learning and the mutation operators proficient the nature that agents achieve their own information.

III. COMAPRISION OF VARIOUS EXISTING GENETIC ALGORITHMS

An analysis on above load balancing algorithms and the distinct load balancing parameters that are approve by them, their comparison based on some parameters are combined in the Table I; given below.
Resource Utilization (RU),
Optimization (OP),

Execution Time (ET),
Migration Time (MT),
Performance (PR),
Reduce Makespan (RMS),
Task Scheduling (TS)

Table1. Comparison of various existing Genetic algorithms

Algorithms Name	RU	OP	ET	MT	PR	RMS	TS
Simple Genetic Algorithm [5]	No	No	Yes	Yes	Yes	No	No
Job spanning and Load balancing Genetic Algorithm [6]	No	Yes	No	No	Yes	No	Yes
Genetic Algorithm With Multi Parent's Recombination [7]	No	Yes	No	No	Yes	No	No
Proposed Parallel Genetic Algorithm [8]	No	No	Yes	No	No	No	No
Modified Genetic Algorithm [9]	Yes	Yes	No	No	Yes	Yes	No
Genetic Adaptive search[10]	No	Yes	No	No	No	No	No
Multi-agent genetic algorithm (MAGA) [16]	No	Yes	No	No	Yes	No	No

Table2. Illustrate the Techniques of Genetic Algorithms.

Algorithms Name	Load Balancing Objectives	Findings	Matrix used	Tool used
Simple Genetic Algorithm [5]	To execute cloudlet in less time and improve performance	More waiting time and availability	Finish Time	Cloud-sim
Job spanning and Load balancing Genetic Algorithm [6]	Task scheduling and improve performance	Improve resource utilization and performance.	ETC and DTC randomly	MAT LAB
Genetic Algorithm With Multi Parent's Recombination [7]	Improve performance.	Improve performance	Optimal number of parents and scanning	De-Jong funcTion
Simple Genetic Algorithm [5]	To reduce the scheduling time without affecting the makespan.	To reduce the scheduling time.	Makespan and Time Required	De-Jong funcTion
Job spanning and Load balancing Genetic Algorithm [6]	Improve performance	Improve performance	Processing Speed and Makespan	Cloud-sim
Genetic Algorithm With Multi Parent's Recombination [7]	To solve Optimization problem	Reduce optimization and improve performance	Optimal Solution	SBX
Multi-agent genetic algorithm (MAGA) [16]	To achieve better performance of load balancing.	Improve performance	Single user Optimal Solution, CPU consumption and memory.	MAT LAB

IV. CONCLUSION

Load Balancing is one of the key issues to distribute the load among various nodes. In this paper, various load balancing techniques based upon genetic algorithms are discussed and the comparison of these approaches are done on the basis of some parameters like resource utilization, optimization, execution time, migration, performance, reduce make span and task scheduling. Further, we have illustrated the algorithms based upon load balancing objectives, findings, the matrix used and tool used. Establishing a better algorithm by considering the combination of some significant parameters together always result in a good load balancing algorithm which can be used in a cloud environment for providing better cloud services to the users and designing a new technique by reducing the flaws of the current approach could be a compelling future work.

.

REFERENCES

1. Monika, L. and Bhardwaj, N. (2017). Load Balancing in cloud computing using Genetic Algorithm. *International Conference on Intelligent Computing and Control system ICICCS*.
2. M., P. and Mahaboob, B. (2017). Dynamic And Elasticity ACO Load Balancing for cloud computing. *International Conference on Intelligent Computing and Control system ICICCS*.
3. Jaspreet Singh and Deepali, G. (2017). Energy Efficient Heuristic Base Job Scheduling Algorithms in Cloud Computing. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 19, pp.50-55.
4. Khusboo, K., Megha, R. and Dishant, R. (2017). Dynamic Priority Based Load Balancing Technique for V M Placement in cloud computing. *International Conference on computing Methodologies and communication*.
5. Albert, Y. and Yee, H. (2001). Observation on Using Genetic Algorithms for Dynamic Load-Balancing. *IEEE Transactions on parallel and distributed systems*, 12(9).
6. Tingting, W. and Xiaming, D. (2014). Load Balancing Task scheduling based on Genetic Algorithm in cloud computing. *12th International conference on Dependable, Autonomic and secure Computing*.
7. A.E.Eiben, P-E Roué and Zs, Ruttkay (n.d.). Genetic Algorithm with multi parent recombination.
8. R.Nedunchelian, K.Koushik, N.Meiyappan and V.Raghu (n.d.). Dynamic Task scheduling using parallel genetic algorithm for heterogeneous distributed computing.
9. Shekhar, S. and Mala, K. (2014). Scheduling of Independent task in cloud computing using Modified Genetic Algorithm. *Sixth International conference on Computational Intelligence and communication Networks*.
10. Kalyanmoy, D. and Mayank, G. (n.d.). A combined genetic Adaptive Search (Gene As) for Engineering Design. *journal of computer science and informatics*, 26, pp.30-45.
11. Abhishek, G., H S, B., Annapurna, S. and Jagdesh, C. (2015). A Theoretical Comparison of job scheduling algorithm in cloud computing Environmnet. *1st International Conference on Next Generation computing Environment*.
12. Deepa, T. and Cheelu, D. (2017). A comparative study static and dynamic load balancing algorithms in cloud computing. *International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*.
13. Mariana, C. and Paula, K. (2011). Secure Cloud Computing Benefits, Risks and Controls. *IEEE*.
14. Rajwinder, K. and Luthra, P. (2014). Load Balancing in Cloud Computing. *Association of Computer Electronics and Electrical Engineers, 2014*.
15. Mayank, K. and Atul, M. (2013). A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment. *International Journal of Distributed and Cloud Computing*, 1(2).
16. Zhu, K., Song, H., Liu, L., Gao, J., and Cheng, G. (2011).Hybrid Genetic Algorithm for Cloud Computing Applications. *In Proceedings of IEEE Asia-Pacific* Services Computing Conference, 182-187.
17. P. Geetha and C.R, Rene Robin (2017). A Comparative study of Load cloud Balancing Algorithms in Cloud Environments. *International Conference on Energy, Communication, Data Analytics and soft computing*.

AUTHORS PROFILE

Er. Neha Sharma is a student of M.tech in computer science and engineering from Chandigarh University, Gharuan Mohali, Punjab, India. I have done my B.tech from Global Institute of engineering & management Amritsar. Her research area of interest is Cloud Computing.

Dr. Jaspreet Singh is working with Department of Computer Science and Engineering, Chandigarh University, Gharuan, Mohali, (Punjab) India. He received M.tech degree in Computer Science and Engineering from Punjab Technical University, Jalandhar, Punjab, India and obtained Ph.D. degree from Department of Computer Science and Engineering, Maharishi Markandeshwar University, Ambala, (Haryana),India. He also has published several research papers in national as well as international Journals and conferences of repute. He has research contributions in the area of cloud computing, networking, security and cryptography and image processing.

Dr. Rohit Bajaj received his B-Tech degree from GJUS&T, Hisar and M-Tech degree from MDU, Rohtak. He received Ph.D. degree from Sai Nath University, Ranchi. Presently, he is an Associate Professor in the Department of Computer Science & Engineering, Chandigarh University, Gharuan Mohali, Punjab, India. He has 10 years of experience of Teaching and Research as well as guiding Post Graduate & Ph.D. Students. He has several papers in referred journals of national and international repute to his credit. His research area of interest is Data Science, Cloud Computing, Wireless Sensor Networks, Soft Computing.