Load Balancing using Improved Genetic Algorithm (IGA) in Cloud Computing

Sukhpreet Kaur, Dr. Jyotsna Sengupta

***Abstract*— In cloud computing, the applications and services are provided over the internet. The various users send requests for accessing resources and applications. Such large number of requests increase the load on the cloud. For balancing the load, an Improved Genetic Algorithm (IGA) has been introduced in this paper for allocating the user's tasks to the virtual machines (VMs). The aim of this proposed algorithm is to maximize resource utilization and to minimize the task execution cost and energy consumption. All the tasks are assigned to the available VMs in such a way that load is balanced by distributing the dynamic workload across multiple virtual machines to ensure that no single VM is either over utilized or underutilized. The proposed algorithm is simulated using CloudSim toolkit.**

***Index Terms*—Cloud computing, Load Balancing, Improved Genetic Algorithm, VM allocation.**

I. INTRODUCTION

Cloud computing is a model where the users can remotely store their data in the cloud and can access the services from shared pool of configurable computing resources. It is a model in which the users can access the resources over the internet. Various hardware and software services are provided to the users by the cloud service providers. This is very beneficial as it reduce the burden of maintaining resources. The organizations need not to buy all the resources, instead they can use the shared resources available over the internet by paying just little amount to the service provider. Cloud computing provide various advantages over the traditional computing in terms of security, reliability, scalability, fault-tolerance. Basically cloud computing provides three types of services, which are application, platform and infrastructure. These services can be categorized into three types, namely, software-as-a-service (SaaS), infrastructure-as-a-service (IaaS) and platform-as-a-service (PaaS).

In cloud computing the users access the resources available over the internet. The users can access these resources by sending requests. Such large number of requests increases the load on the cloud. This load has to be balanced in such a way that it distributes the dynamic workload across multiple nodes to ensure that no single resource is either overwhelmed or underutilized

Load balancing is a method to distribute workload across multiple computers, network links, CPUs or any kind of other resources. Load balancing is done in order to get maximum throughput and minimum execution cost. It also provides efficient resource utilization. Load balancing tries to maximize CPU utilization, minimize response time, improve the performance of the system, utilize resources under condition of low/high load, increase the throughput achieve, reduce the cost for accessing resources and maintain the stability of the system in order to improve the performance and efficiency of the cloud computing
Various types of load balancing algorithms are used to enhance the speed and performance of the system and to make the system more efficient.

II. LITERATURE SURVEY

Kousik Dasgupta et al [1] developed a genetic algorithm based load balancing strategy for Cloud Computing to provide an efficient utilization of resource in cloud environment. Analysis of the results, indicates that the proposed strategy for load balancing not only outperforms a few existing techniques but also guarantees the QoS requirement of customer job. Though it has been assumed that all the jobs are of the same priority which may not be the actual case, this can be accommodated in the JUV and subsequently taken care in fitness function.

K. Gai et al [2] proposed a paper which contains a review of technical literature relating the definitions, characteristics, operations, security management, service governance, and development trends of cloud computing. The advantages and disadvantages of cloud computing are respectively described to represent the impacts of cloud computing in various fields. The paper also explains the structure of clouds in service-oriented architectures and summarizes six major development trends, which can be utilized as a reference for entrepreneurs and researchers.

N.J. Kansas et al [3] consider most of the running process, which are directed to bringing down the associated overhead, service response time and bettering performance of the technique. Existing load balancing techniques that have been analyzed in this paper generally focused on minimizing service response time and overhead. The paper also furnishes details about several parameters, used to equate the existing methods. This paper gave an outline of load balancing in

cloud computing, categorization based on system weight and system form.

J. Hu et al [4] proposed a paper on scheduling strategy on load balancing of resources based on genetic algorithm. According to historical data and current state of the system and through genetic algorithm, this strategy computes ahead the influence it will have on the system after the deployment of the needed resources and then chooses the least-affective solution, through which it achieves the best load balancing and reduces or avoids dynamic migration. At the same time, this paper brings in variation rate to describe the load variation of system virtual machines, and it also introduces average load distance to measure the overall load balancing effect of the algorithm.

Kundan Pagar et al [5] proposed a survey on the existing load balancing techniques in a cloud based environment. Load Balancing is an important aspect of cloud computing environment. In this paper, different load balancing techniques for cloud computing has been surveyed. The key function of load balancing is to satisfy the customer requirement by distributing load dynamically along with the nodes as well as to make highest resource utilization by reassigning the total load to individual node. Existing load balancing techniques that have been analyzed in this paper generally focused on minimizing service response time and overhead.

GE Junwei et al [6] presents the genetic algorithm (MGA) which comprehensive consideration of the total task completion time, average task completion time and cost constraints, it can obtained not only total task completion time and average task completion time is short, but also the required costs is low.

Dr. Prashanth C.S.R et al [7] proposed a paper that has focused on the issue of load balancing. This paper illustrates various aspects pertaining to domain of cloud computing, its evolution, its generic issues, and particularly to issues related to load balancing. Various techniques adopted in the past research work have been analyzed and the findings were illustrated in this paper. With proper load balancing, resource consumption can be kept to a minimum which will further reduce the amount of energy consumed.

Jing Liu et al [8] established a scheduling model for cloud computing based on MO-GA algorithm to minimize energy consumption and maximize the profit of service provides under the constraint of deadlines. They first propose a job scheduling architecture under the environment of cloud computing, which contains several components to analyze the application, and allocate the suitable resources to the applications to improve the effectiveness and efficiency of the computing; then, the MO-GA based scheduling algorithm is proposed, at last, several experiments are conducted to validate our scheduling models.

Richa Garg et al [9] studied Genetic algorithms that are based on evolutionary ideas of natural selection and genetics. Genetic algorithms solve the problems step by step and produce next generation. All evolutionary algorithms including Genetic Algorithm can find near optimal solution. A set of test functions including unimodal and multimodal benchmark functions is employed for optimization. In this paper they discuss the dejong function 4 (rastrigin function) it is highly multimodal function.

Safwat A. Hamad et al [10] proposed a paper in which a task scheduling algorithm based on Genetic Algorithm (GA) has been introduced for allocating and executing an application's tasks. The aim of this proposed algorithm is to minimize the completion time and cost of tasks, and maximize resource utilization. The performance of this proposed algorithm has been evaluated using CloudSim toolkit.

III. EXISTING GENETIC ALGORITHM

A simple Genetic Algorithm (GA) [1] consists of three operations: initial population generation, mutation, and crossover. These operations are explained below [1]:

1. Initial population generation: GA works on fixed bit string representation of individual solution. So, all the possible solutions in the solution space are encoded into binary strings. From this an initial population of ten chromosomes is selected randomly.

2. Crossover: The objective of this step is to select most of the times the best fitted pair of individuals for crossover. The fitness value of each individual chromosome is calculated using the fitness function as given in 3. This pool of chromosomes undergoes a random single point crossover, where depending upon the crossover point, the portion lying on one side of crossover site is exchanged with the other side. Thus it generates a new pair of individuals.

3. Mutation: Now a very small value (0.05) is picked up as mutation probability. Depending upon the mutation value the bits of the chromosomes, are toggled from 1 to 0 or 0 to 1. The output of this is a new mating pool ready for crossover.

In the existing algorithm the problem of load balancing is defined for allocating N number of user tasks to M number of virtual machines. Two vectors are considered: one is processing unit vector (PUV) which represents processing unit's current status and other one is job unit vector (JUV) which represents the user tasks. PUV vector contains MIPS which stand for million instructions executed by that machine per second, α, cost of execution of instruction and delay cost L [1]. The cost function is calculated using the following equation:

Cost=w1*ProcessingCost()*(NIC/mips)+w2*Delaycost();

Where, NIC denotes the number of instructions present in the

job, w1 and w2 are predefined weights. The weights are considered as w1 = 0.8 and w2 = 0.2 such that their summation is 1[1]. This algorithm allocates the best suitable virtual machine to a job.

IV. THE PROPOSED IMPROVED GENETIC ALGORITHM (IGA)

The existing Genetic Algorithm balances the load in cloud computing by assigning tasks to the virtual machines. But it is not effective in resource utilization which means it fails to utilize all the available virtual machines. It repeatedly assign tasks to some of the VMs. Due to which some machines remains idle while some machines are overloaded. The resources are not properly utilized. So this problem is tackled by modifying the existing algorithm. The proposed Improved Genetic Algorithm keeps track of all the free virtual machines. When a new task arrives, first it is checked that whether a free machine is available or not and if a machine is available then task is assigned to that particular machine. If no free virtual machine is available then the task is assigned to that machine whose current task is going to be completed in lesser time as compared to other machines.

In this way all the VMs are properly utilized and no VM remains idle and also no VM is overutilized. The proposed algorithm gives better output in terms of energy efficiency, cost and also all the VMs are allocated tasks. The cost is improved by making the following changes in the cost function:

Cost=w1*ProcessingCost()*(FreePesGenetic()
/NumberOfPes())+w2*Delaycost()

Where the values of the weights w1 and w2 are the same as of the base Genetic Algorithm which are w1 = 0.8 and w2 = 0.2. The Improved Genetic Algorithm gives better output in terms of energy efficiency, cost and also all the VMs are allocated tasks in such a way that the load is properly balanced.

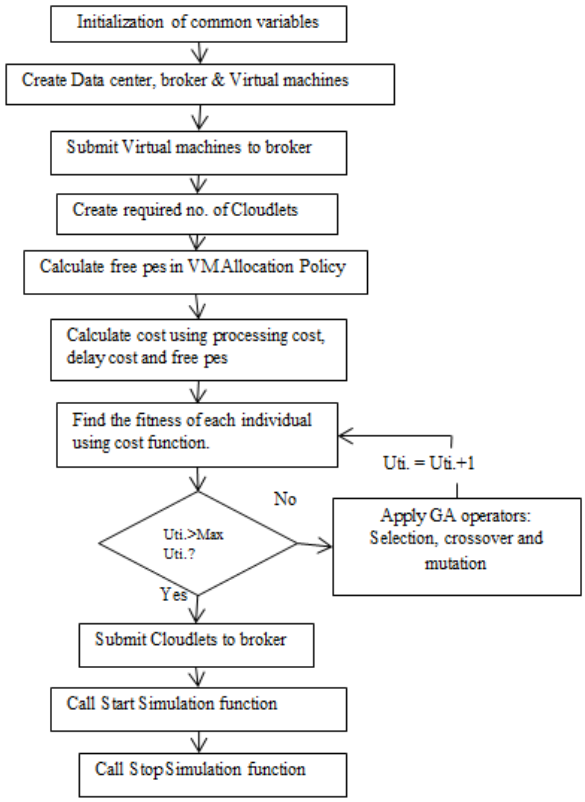

Figure 1: Flowchart of Simulation Steps for Improved Genetic Algorithm

Cloudlets are the user tasks.
Uti: Utlization
pes: processing elements

V. PERFORMANCE EVALUATION

A. *Comparison of Allocated VM Id for Cloudlets*

This graph shows the relationship between Cloudlets and VMs. Cloudlet basically contains the tasks. These tasks are to be assigned to the VMs available. This graph shows that all the VMs are allocated to the cloudlets in Improved Genetic algorithm while in Genetic Algorithm some VMs are not allocated to any of the Cloudlet. Hence in Improved Genetic algorithm, the load is equally balanced on all the VMs whereas in existing Genetic algorithm some VMs are not assigned any Cloudlet while some are overloaded.

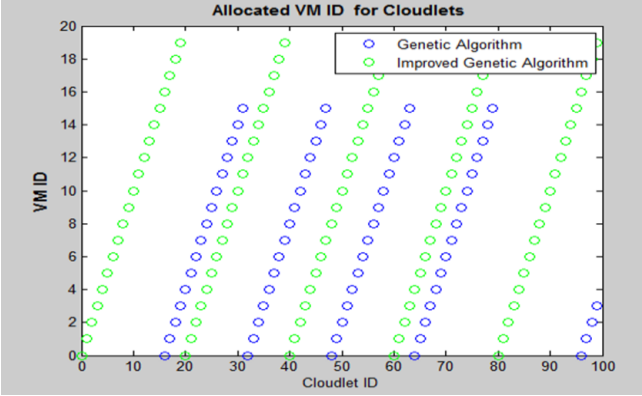

Figure 2: Comparison of Allocated VM Id for Cloudlets

B. *Comparison of Energy Consumption*

This graph shows the relationship between energy consumption and cloudlets. Cloudlets are the user tasks. These tasks are to be assigned to the VMs available for further processing. Energy is consumed when these tasks are being executed on the VMs. This graph shows that less energy is consumed while executing the user tasks using the Improved Genetic algorithm whereas comparatively more energy is consumed while executing the user tasks using the existing genetic algorithm. Hence in Improved Genetic algorithm provides better performance than the existing Genetic algorithm as comparatively less energy is consumed in it during the execution of the user tasks.

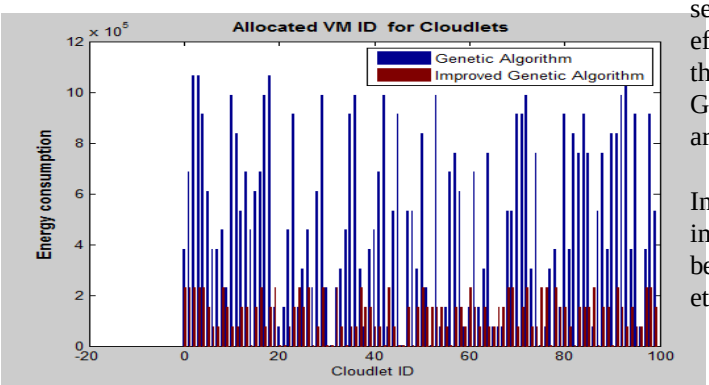

Figure 3: Comparison of Energy Consumption

C. *Comparison of Cost*

This graph compares the cost of execution of user tasks for Improved Genetic algorithm and existing Genetic Algorithm. In this graph number of cloudlets is taken along the horizontal axis. This graph shows the cost calculated for both the Improved Genetic algorithm and existing Genetic Algorithm using same number of cloudlets. The cost of execution of user tasks for Improved Genetic algorithm is less than the cost of execution for Genetic algorithm.

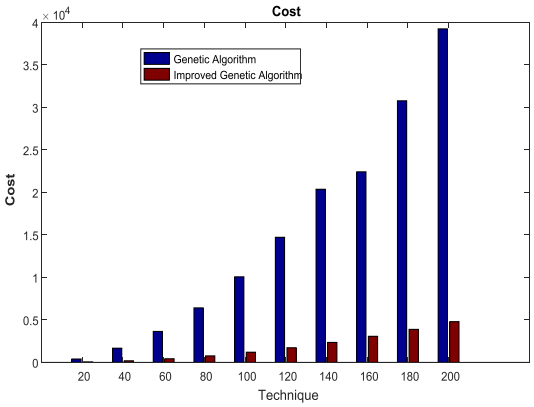

Figure 4: Comparison of Cost

VI. CONCLUSION AND FUTURE SCOPE

The existing Genetic Algorithm balances the load in cloud computing by assigning tasks to the virtual machines. But it is not effective in resource utilization because it fails to utilize all the available virtual machines. This problem is tackled by proposing an Improved Genetic Algorithm (IGA).

The Improved Genetic Algorithm gives better output in terms of energy efficiency, cost and also all the VMs are allocated tasks in such a way that the load is properly balanced. The graphical representation of the simulation results show that the Improved Genetic Algorithm is more efficient than the existing Genetic Algorithm in terms of several parameters used. These parameters include energy efficiency and cost. The graphical outcome clearly depicts that all the VMs are allocated to the cloudlets in Improved Genetic algorithm while in Genetic Algorithm some VMs are not allocated to any of the Cloudlet.

In future, Improved Genetic Algorithm should be implemented in real time environment. This algorithm can be further improved in terms of response time and finish time etc.

REFERENCES

[1] Dasgupta, K., Mandal, B., Dutta, P., Mandal, J. K., & Dam, S. (2013). A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing. Procedia Technology, 10, 340-347.

[2] Gai, K., & Li, S. (2012). Towards Cloud Computing: A Literature Review on Cloud Computing and Its Development Trends. 2012 Fourth International Conference on Multimedia Information Networking and Security.

[3] Kansal, N. J., & Chana, I. (2012). Existing Load Balancing Techniques In Cloud Computing: A Systematic Review. J, 3(1), 87-91.

[4] Gu, J., Hu, J., Zhao, T., & Sun, G. (2012). A New Resource Scheduling Strategy Based on Genetic Algorithm in Cloud Computing Environment. Journal of Computers, 7(1).

[5] Pagar, K., & Patil, S. (2012). A Survey on Load Balancing Techniques in Cloud Computing. International Journal of Science and Research.

[6] Ge, J., & Yuan, Y. (2013). Research of cloud computing task scheduling algorithm based on improved genetic algorithm. Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013).

[7] C.S.R, P., & Begum, S. (2013). Review of Load Balancing in Cloud Computing. International Journal of Computer Science Issues (IJCSI), 10(1), 2nd ser.

[8] Liu, J., Luo, X., Zhang, X., Zhang, F., & Li, B. (2013). Job Scheduling Model for Cloud Computing Based on Multi- Objective Genetic Algorithm. International Journal of Computer Science Issues (IJCSI), 10(1), 3rd ser.

[9] Garg, R., & Mittal, S. (2014). Optimization by Genetic Algorithm. International Journal of Advanced Research in Computer Science and Software Engineering, 4(4).
[10] A., S., & A., F. (2016). Genetic-Based Task Scheduling Algorithm in Cloud Computing Environment. International Journal of Advanced Computer Science and Applications, 7(4).

Sukhpreet Kaur is currently pursuing M.Tech. in Computer Science and Engineering at Department of Computer Science, Punjabi University, Patiala. She is carrying out her research work under the guidance of Dr. Jyotsna Sengupta. Her main research interest is cloud computing.

Dr. Jyotsna Sengupta is currently working as a Professor in the Department of Computer Science at Punjabi University, Patiala, Punjab, India. Her research interests include Cloud Computing, ADHOC Networks, Wireless Networks and Distributed Systems. She has published a number of research papers in national and international journals. She has also published two books and edited one book in her area of interest.