A Novel Hybrid Approach To Load Balance Optimization based on Task Scheduling in Cloud Computing Environment

Shalini Shukla[1], Dr. Nitesh Bharot [2]
[1, 2] Dept of Computer Science
[1, 2] Shriram College of Engineerimg & Management

Abstract- *Cloud Computing (CC) is a type of technology which integrates distributed computing, parallel computing, virtualization technology & web services technology. Due to excess traffic & service requests in the cloud datacenter, load balancing is a serious challenge in cloud environments owing to failure in energy consumption, node failure & machine failure. So policy is needed for load balancing between datacenters, & researchers have proposed different solutions for load balancing (LB). Load distribution is a mechanism for distributing load amongst different nodes depend on definite parameters e.g.under loaded (node) & overloaded (node). There are two main tasks of resource allocation, second is scheduling tasks in a cloud environment. Mainly attention is given to task scheduling (TS). Cloud computing is a widespread stage for performing tasks using virtual machines (VMs) as processing components. In this paper, we are introduced Load Balancing Optimization Improved Genetic Algo (GA) with Simulated Annealing (LBIGASA)algo based on task scheduling in CC environments to decrease performance time, as well as cost of performing autonomous tasks on cloud resources, has reduced. The simulating parameters on which we have performed simulation are execution time, memory usage, fitness function (or cost), load balance and finishing time of VMs. Simulation has performed on the Cloud Sim tool, which shows the efficiency of the proposed job in relation to execution time & efficient resource allocation but variations in memory usage. The overall execution time and finishing time of VMs take minimal time in compare of an existing method.*

Keywords- Cloud Computing (CC), Load Balance, Optimization, Task Scheduling, Improved Genetic Algo, Simulated Annealing.

I. INTRODUCTION

Cloud computing has emerged as a short-term "pay-as-you-go" in IT infrastructure. By CC, establishments can expand to massive capacity at any time without having to pay for new infrastructure, training issues or licenses for new software. CC profits for entire types of businesses (small or medium) that required enlarge its resources without paying new servers[1].

LB is one of the most challenging issues that we have faced today. LB can be distinct as uniform distribution of workloads at all nodes to avoid a condition where some nodes are overloaded, whereas other nodes are overloaded or idle. LB system importantly greatly affects the performance of the system. It is supposed that agents have modern computer programs that work automatically for consumers once there is too much work to do. In most belongings, multiple agents are frequently needed after there are more uses toward meet user needs efficiently[2].

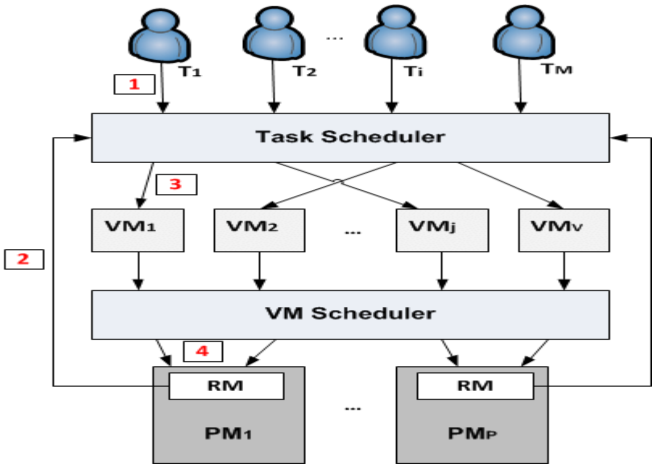


Fig.1. Scientific diagram of Task & VM scheduling in cloud computing

Resource management & TS are the main issues that need to be addressed in CC. Therefore, cloud providers need to define services & implement scheduling policies that allow VMs to deploy &assign VMs to users' tasks. These2categories of TS in CC; Static & dynamic TS, static scheduling allow data pre-fetching & pipelines needed for various phases of the executed task. Static scheduling executes less runtime above. In the item of dynamic scheduling, job mechanisms/task data is not call in advance. Therefore, the executed time of task may not be identified, &distributions of tasks on the fly when applications are running[3].

The four major contributions of such a paper are described below:

- We used a powerful two-point crossover operator that does not require exploitation of the preceding restraints in the strategies generated. 2-point crossover is quite different from one-point crossover except that 2 cut-points are generated randomly rather than one.
- In order to handle either minimization or maximization problems without any structural changes, tournament technique by unusual selections is approved. Tournament Selection also works for negative fitness values.
- Data analyses are conducted on the various randomly-generated graphs to ensure consistency of the information obtained from implementation.
- The suggested method was converted into a verifiable behavioral model as well as the performance model was introduced.

The reaming part of this paper is organized as following: works of numerous scheduling algos were analyzed in Section 2 for load balancing mechanisms. Section 3 introduces the LBIGASA algo proposed as well as the system testing model. Section 4 explains the proposed LBIGASA algo in detail. Experimental or analytical findings were listed in Section 5. Analysis is finally discovered in Section 6.

II. LITERATURE SURVEY

Sreelakshmi & S. Sindhu [2019] propose multi-objective particle cluster optimization for TS. Target time, makespan time & communication cost. It has been exposed which proposed method helped to reduce the time & communication cost of the makespan & to complete the task on time [4].

Nitesh Bharot et al. [2019] The new tool for the identification of DDoS attacks is introduced. A studies are done as follows: I Originally DDoS attack is identified in determining max no. of network connections, (ii) otherwise te targeted VM or non-attacked virtual machines are clustered utilizing Neural Network (NN) dependent on Self Organized Mapping (SOM). Experimental data indicate which DDoS attacks & cluster attacks or non-attack VMs can be successfully detected by the presented process in attacked cloud networks. In addition, these outcomes show which suggested DDoS attacks 97.63 percent accuracy and 95.4 percent accuracy and is better than the current technique [5].

M. Ashouraei et al. [2018] presents a method based on a parallel Genetic Algo (GA) for scheduling tasks using priorities. The goal is to use resources efficiently &decrease resource expenditure in cloud environments. This can be attained by refining the LB rate, while low task failure rate selects the best resources to complete arrival tasks in a short time. Toward evaluate the proposed method, it is simulated through MATLAB, hybrid agent colony-honey technique& round-robin (RR) based LB technique, comparing 2 existing methods. Proposed method shows 9% - 31% lower energy consumption, 14% - 37% lower migration rate & 13% - 17% improved service level agreement (SLA) than hybrid & RR method [6].

Y. Samadi et al. [2018]propose a method to obtain a balanced load in VM though trying to reduce the makespan of specific workflow applications by improvement of Heterogeneous Earliest Finish Time (E-HEFT) algo below user-specified financial constraints. Toward assess performance of enhancement algo, we associate our algo by specific existing scheduling algo. The outcomes demonstrate that this algoout performs other algo by dropping makespan & refining LB in VM[7].

M. Padmavathi& S. M. Basha [2017] uses the services of remote computers & CC servers. In this regard, the load is distributed between existing systems in data centers; some systems are overloaded &specific under loaded. Load balancing can be used to balance the load between different systems in a datacenter using scheduling techniques. CC is not a modest technology. This contains problems such as VM management, provision of software & hardware resources, scheduling virtual machines, & data security. Load balancing is the main challenge. We current an original, dynamic & elastic algo for performing load balancing &ant colony optimization LB between existing systems in data centers[8].

NiteshBharot et al. [2017]Use the feature method of selecting or Intensive Care Request Processing Unit (ICRPU) to suggest a DDoS attack detection or mitigation system. In the suggested project, traffic is initially analyzed using the Hellinger distance function, &unless some separation is detected, then all packets will be evaluated and categorized into 2sets as DDoS & valid demand states based on the identification function selected. Both appropriate requests are sent to the Standard Request Processor where they will be completed. ICRPU's strength would be that the attacker would never know that they are stuck in the demand sent to maximize resources, so the attacker will not do some reflex action, &the perpetrator will be easily tracked. Data show which approaches proposed provides the best success rate, accuracy & false alarm compared to existing filter methods & other such solutions proposed [9].

A. Gupta & R. Garg [2017]chose the meta-heuristic method of ACOalgo to resolve task scheduling problems ina cloud environment, focusing primarily on 2 objectives, namely makespan / computational time reduction & optimal load balancing. Relative analysis shows that proposed LB-ACO gives better results than NSGA-II algo, providing better LB& less makespan. Simulation has been performed by the Cloud Sim toolkit [10].

Y. Fang & X. Li [2017] proposes VM real-estate improvementant colony algo (VM-ACO) to resolve the difficulty of task scheduling in cloud environments. Algo takes time to complete the load balancing of tasks. Experiments with CloudSim suggest that VM-ACO algo executes better than task latency, task fulfillment time, resource status & polling-based anti-colony algo& better scheduling in cloud environments [11].

Nitesh Bharot et al. [2016] Using a threshold-based approach to reduce DDoS attack on the cloud network. Based on their age during the Turing test, a list of defective IP addresses was prepared in the suggested solution and designated as blacklist. When the application is from the blacklist, it will be forwarded straight to the next level. In either, the second stage test when the amount of available resources is greater than the request made and also the demand for resources is less than resource m's threshold value than resource allocated to that request is denied. One of the 3 defense mechanisms, i.e., can protect cloud services from the DDoS attack. Prevention of DDoS attacks, prevention of DDoS attacks and mitigation and recovery of DDoS attacks. However, Attack prevention is considered to be the simplest way of protecting against the DDoS attack related to tools that are easily available. The introduction of minimal cost and overhead is also very straightforward. Proposed work could be carried out in some cloud networks to avoiding malicious requests by being lost. [12].

S. Aslanzadeh& Z. Chaczko[2015] proposesa new LB procedure. Our proposed algo load balancing accomplishes framework load adjusting by actualizing a self-organizing technique between VMs. This strategy depends on the correspondence between VMs. By utilizing an improved feed backing approach utilizing partial swarm - optimization (PSO), it helps under loaded VMs convert their extra capacities to other emptied VMs. To assess our proposed algo, we have protracted the Cloud Simulation Tool (ClsDim) created by the University of Melbourne. Experiment outcome demonstrates that our suggested LB method fundamentally lessens time interim contrasted with conventional load balancing strategies. In addition, it reduces the idle time of VMs & improves the quality of service (QOS) [13].

Dr. D. Singh Karaulia and Nitesh Bharot [2014]We were developing the current route repair process in AODV in this paper to obtain broadcasting or mitigate flooding. The improved version of the protocol initially establishes the mobile node party, then broadcasting can be completed, and if the connection breaks then it is possible to apply the route restoration technique. Several of the network's intermediate nodes are active in connectivity. Enabled AODV Route Repair (EAODV) protocol on the NS2 network model is introduced. To evaluate and evaluate the activity of the proposed protocol (EAODV) with the dynamic TTL value (fixed and varying) according to the forecast of network conditions, simulations are performed. The proposed protocol has been associated with available AODV in relation to routing load, & data delivery ratio [14].

W. Sun et al. [2013] to resolve the issue in distributed computing, intermittent ACO_based scheduling algo (PACO) is proposed in this paper. PACO utilizes ACO algo in distributed computing, improving the first proposed scheduling time technique and pheromone power update system. Experimental outcomes suggest that PACO performs best on the makespan & load balancing [11] of the entire cloud cluster [15].

K. Li et al. [2011] proposed LB Ant Colony optimization (LBACO) algo. The fundamental commitment of this work is to balance the whole framework load while attempting to limit the makespan of the particular task set. The new scheduling technique was simulated utilizing the CloudSim tool. Tentative results improve proposed LBACO algo outpaced FCFS (first cum 1st service) & basic ACO [16].

Author [year]	**Technique used**	**Description**	**limitation**
Sreelakshmi& S. Sindhu [2019]	multi-objective particle cluster optimization	It has been shown that the proposed method helped to reduce time & communication cost of the makespan & to complete the task on time	-
NiteshBharot et al. [2019]	SOM based on NN	The creative method for detecting attacks by DDoS is introduced. Studies are done as tries to follow: I Primarily DDoS attack is observed by defining that maximum of network connections, (ii) then invaded virtual machine or non-attacked VMs are clustered expending NN based on SOM.	cryptography algo not applied for mitigating the detected attacks
M. Ashouraei et al. [2018]	parallel GA	Presents method based on a parallel Genetic Algo (GA) for scheduling tasks using priorities. The goal is to use resources efficiently & decrease resource expenditure in cloud environments. This can be attained by refining the LB rate, while low task failure rate selects the best resources to complete arrival tasks in a short time.	lower migration rate
Y. Samadi et al. [2018]	LB& ant colony optimization	Propose a method to obtain a balanced load in VM though trying to reduce makespan of specific workflow application by improvement of Heterogeneous Earliest Finish Time (E-HEFT) algo below user-specified financial constraints.	Not run in real cloud environments
M. Padmavathi& S. M. Basha [2017]	LB & ant colony optimization	In this regard, the load is distributed amongst existing systems in data centers; specific systems are overloaded & specific underloaded. Load balancing can be used to balance the load between different systems in a datacenter using scheduling techniques. This contains problems such as VM management, provision of software & hardware resources, scheduling virtual machines, & data security.	LB
NiteshBharot et al. [2017]	Intensive Care Request Processing Unit & feature selection method	Well into the proposed study, traffic is initially analyzed bythe Hellinger spacing function, &unlessspecificspace is detected, then entirely packets will be studied& categorized into 2 sets as DDoS & valid request categories based on the classification aspects specified. Both valid requests are transferred to the Standard Request Processing Unit in which they can be completed.	resources are trapped
A. Gupta & R. Garg [2017]	LB-ACO	Chose the meta-heuristic method of ACO algo to resolve TS issues in a cloud environment, focusing primarily on 2 aims, namely makespan / computational time reduction & optimal load balancing.	increases the utilization of resources
Y. Fang & X. Li [2017]	VM-ACO	Proposes VM real-estate improvement ant colony algo (VM-ACO) to resolve the difficulty of task scheduling in cloud environments. Algo takes time to complete the load balancing of tasks.	Not user satisfaction
NiteshBharot et al. [2016]	threshold-based technique	Utilizing a threshold-based techniques to reduce DDoS attacks in the cloud network. Based on their income during the Turing test, a list of defective IP addresses was prepared in the proposed solution and designated blacklist. Whether the request is as of blacklist, it will be forwarded straight to the next stage. In 2nd phase test whether the amount of resources obtainable is greater than the statement made & demand for resources is below resource m's threshold value than resource allotted to a certain request is dismissed.	wasting the resources for malicious requests

There are various work has been done in the field of load balance optimization. With either the ample literature on LB strategies over the cloud environment, the current system is witnessed. Many of the strategies were found to follow an approach using increasingly complex and sophisticated architecture in which specific load balance optimization probabilities have not been effectively oriented. Therefore, it is important to implement a strategy that can minimize the execution time cost-effectively utilizing multi-tenancy technique, efficient scheduling or delivery, etc..

III. PROPOSED METHODOLOGY

A. Problem Description

In cloud computing environments, parallel processing is required to be distributed for large-scale computing tasks. We divide a large task into sub-tasks that are independent of each other, & then assign to mVM node to perform, & participants to independent sub-task set D={d1,d2,...,dn}(di:tasks) has assigned to determine. System usages optimal methods to handle tasks allocated for various resource nodes, & we usage R = {r1, r2, ..., r3} (ri: virtual node i) to determining resource nodes for participants in scheduling. MI represents the execution speed & task length instruction of every resource node signified by MIPS, & every separate task could be allocated to run on only one virtual node, &currently each resource node's execution time matrix represented by T. The initial load balancing server delivers the requests in a stable manner so which present computing resources are not idle to recover system use.

GA has a strong global search capability to solve such difficulties, nonetheless too defects e.g. premature & weak local search capability. Instead, SA does not have resilient local search ability& premature problems. Thus, the grouping of GA & SA can both overawed drawbacks & implement their advantages & improve their efficiency. This algo is called improved genetic algo with simulated annealing to optimize load balance (LBIGASA).

B. Procedure

In this work, we have applied the concept of simulated annealing with genetic algo to load balancing optimization problem. Firstly, task loading & computing resources ability measured.In each physical machine, the resource utilization ratio of computing resources is dispersed to computing properties so that the loading balance is large & relatively unbalanced. Task loading measurements are determined by predictive time &load of the task, nonetheless, the computing capacity of the physical machine (PM) is taken into account, such as CPU type, frequency, core no., & CPU size. Compute computing resource occupancy rate of physical machines. Compute average of computing resource occupancy rate for all physical machines. Then, calculate the variance of the computing resource occupancy rate. Device genetic algo. This involves randomly generated populations & executing individual coding. We start with loading every VM & computing capacity on every PM. Calculate the reciprocal of variance in resource take out as Fitness Function.

On the basis of fitness of dissimilar persons& the probability of every separate being selected will be resolved. Individuals selecting the activity will be based selected by tournament selecting method on pairs to interchange certain genes of the 2 individuals& generate novel chromosomes. The 2-point crossover functionality optimized the global search optimization process to eliminate the local optimal solution & specified the optimal solution for the local range. When the mutation function achieves local search optimization, it is gradually switched to optimum resolution. Here, the single-bit mutation function was performed at a random place. After coding the chromosome, the final problem is for every VM to load. That is, it analyzes the task of any user's using a single virtual machine. Therefore, decoding has to be done.

C. Proposed Algo

Step 1: Start

Step 2: Person K is randomly produced to be the initial population. Every individual is responsible for delivering VM scheduling for PM. PM = 20, VM = 100, population-scale K = 50. At the same time, start every task loading & computing capacity of every PM & K person. Population size N, evolutionary generation M, crossover probability pc, mutation probability pm, cooling coefficient temperature α& initial annealing temperature T_0.

Step 3: Calculate the individual fitness(i).

$f(i)$=1/VOR

Step 4: Select operation. Individuals are selected conferring to probability based on their Individual fitness.

Step 5: Crossover operation. After selection, designated Individuals will be in pairing crosses. It uses 2-point cross & crossover probability Pc=0. 90.

Step 6: Mutation operation. The mutation probability of randomly selected individuals mutated at random position to accept single-bit mutations is randomly assigned to the PM. Mutation Probability Pm = 0.005.

Step 7: Tournament Selection operation. In K-Way tournament selection, we choice K-individuals & run tournaments between them. Out of selected applicants, only the most suitable candidate will be selected & it will be given to the next generation. So there is a possibility to select the most suitable candidate aimed at then generation. Suppose best candidate by probability P is selected, succeeding best candidate through probability of p*(1-p) &following one by p*(1-p) ^{2}& so on.

Step 8: If $fi<fj$ ($j>i$), receive new person; Then, take new person with probability $p = \exp(-\Delta f/t)$;

Step 9: Terminating condition criteria. Based on the repetition time, update annealing temperature & determine if it satisfies expiration conditions. If this is unsatisfactory, step 2 starts the process down. Otherwise, the program will be terminated to get the best individuals.

Step 10: Exit

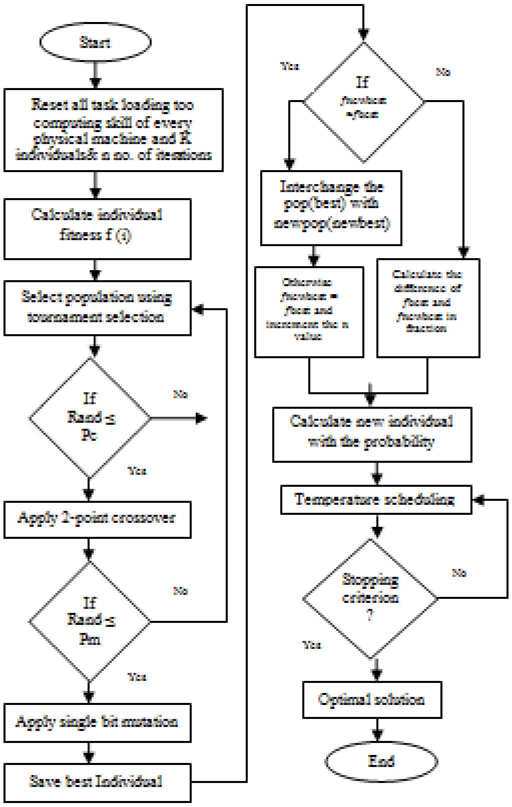

Start
Reset all task loading too computing skill of every physical machine and K individuals& n no. of iterations
Calculate individual fitness f (i)
Select population using tournament selection
If Rand ≤ Pc
No
Yes
Apply 2-point crossover
If Rand ≤ Pm
No
Yes
Apply single bit mutation
Save best Individual
If fnewbest <fbest
Yes
No
Interchange the pop(best) with newpop(newbest)
Otherwise fnewbest = fbest and increment the n value
Calculate the difference of fbest and fnewbest in fraction
Calculate new individual with the probability
Temperature scheduling
Stopping criterion ?
No
Yes
Optimal solution
End

Fig. 2. Proposed Model

IV. IMPROVED GENETIC ALGO WITH SIMULATED ANNEALING TO OPTIMIZE LOAD BALANCE (LBIGASA)

Load balancing is completed at the first level of task distribution. The task request is consistently dispersed to different physical servers conferring to required resources, so cloud computing server tasks are executed while maintaining high efficiency. Load balancing using VM distributed physical servers is accomplished at another level. Once the task is received, the server installs a virtual machine that runs each task independently. When a server is overloaded, the virtual machine is moved to the physical machine.

A. Load Balancing Optimization Model under Cloud Computing Environment

Task-based computation-based balance of resources plays a vital role in LB, & numerous LB approaches are efficient in computing the balance of resources. The framework created throughout this paper is focused on the combination of project scheduling computing resources. To beginning with, it measures the ability of task loading or computing resources. The capability to load or compute resources to shape structured climate models will be abstracted. The asset use proportion of computing resources in every physical machine will be conveyed to computing resources with the goal that the loading balance isn't huge and moderately adjusted. Task loading estimations are dictated by predicting time & task load but computing ability of physical machine is considered, for example, CPU type, recurrence, center number, and CPU size.

It calculates the computing source occupancy rate of physical machine j

$$OR_j = \frac{L_i}{c_j} \quad (1)$$

Here, Lj is a load of the virtual machine, &Cj is the computing power of *j*.

Average computing calculation of computing source occupied rate for all physical machines

$$AOR = \frac{\sum_{i=1}^{n} OR_i}{n} \quad (2)$$

Computing Equations of variance of Computing Resource occupied Rate

$$VOR = \frac{\sum_{i=1}^{n} (OR_i - AOR)^2}{n} \quad (3)$$

Here, n represents the number of physical machines. If the VOR is low, the load balancing effect is excellent. Consequently, the objective function of the LB model is min $\{VOR\}$.

B. Relevant Operations of GA

1) *Encoding.* GA is being developed to better solve the populace of candidate solutions (named individuals) for the optimization problem. With a set of attributes, each individual can alter &mutate its own chromosomes or genotype. Traditionally, the solution was represented in binary as strings of 0 & 1 s. The Q_i operator task is mapped to V_{mj}VM. Values of genes in chromosome characterize ID of VM, &the corresponding location of the pg gene is the ID of a user task. Q_i *(0≤i≤n-1)* means task ID &V_{mj}*(0≤j≤m-1)* 0 1) denotes VM IS. Once preparing colony, randomly no. in [0, *m-1*] will be produced to express gene value. By *n* times of loops, separate MAY be produced. Afterward, the chromosome is coding, concluding problematic is to obtain every VM loading. That is to say, which user's task is studied by one VM? So, decoding must be achieved.
2) *Fitness Function.(FF)*Fitness Function is an exceptional kind of objective function that is utilized to estimate the nature of spoke to the solution. In this investigation, FF is characterized as

$$f(i) = 1/VOR \quad (4)$$

3) *Selection.* In every successive generation, the quantity of the present population is designated for the different populations to grow. Individual solutions are chosen by fitness-based procedure, wherever fitness solutions (calculated as fitness function) are generally more prospective to be chosen. Using the tournament selection operator.
4) *Crossover.* Crossover is the process of taking more than one parent & making one of them a child. Crossover is utilized to change the programming of chromosomes or genesas of one generation to another. A two-point crossover is utilized in this paper.
5) *Mutation.* Mutation is utilized to preserve GA as of one population generation to another generation. Mutations in GAs are intended to develop & introduce diversity. Mutation algo should prevent populations of individuals from becoming identical to each other to slow down the local minimum or stopping evolution. Simple & efficient mutation functions, i.e., single-bit mutations are used.

C. Relevant Operations of SA

1) Annealing Process to receive SHALINI next Individual. Presently request to anticipate populace in local optimization, Metropolis acknowledgment standards in SA are connected in GA in this paper. We held the best parent separate in populace named old, & then chose best offspring individual in another populace termed new; old & new go into the next generation population through competition. Let

$$\Delta f = f_{new} - f_{old} \qquad (5)$$

if $\Delta f < 0$, then individual novel is expected, & hold it to following generation; then, individual new is usual thru probability

$$p = \exp(-\Delta f/t) > \mathit{random\ digit} \qquad (6)$$

Wherever t is annealing temperature.

2) Temperature Modified Criteria. An important step in the SA process is to conclude the temperature update function; function is utilized to continuously decrease temperature value, &after its temperature decreases toward almost 0, a final solution is measured global optimal resolution. Update function

$$t_{k+1} = \alpha t_k (7)$$

$K \geq 0$, $0 < \alpha < 1$; the nearer α is to 1, slower temperature decreases. Here t_{min}=0.0001.

D. Termination Criteria

Generally, the algo ends after either max no. of groups is produced or population acceptable fitness level is extended. Termination condition is which fitness has extended continuous plateau M repetitions(i.e. 100) no longer produce enhanced outcomes. Also, check the minimum temperature for simulated annealing which is 0.0001.

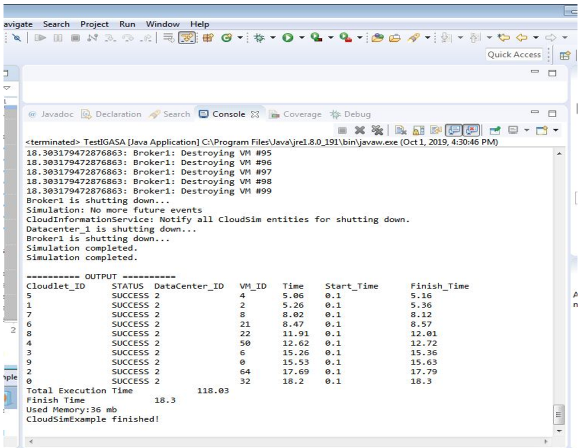
V. SIMULATION & RESULTS

Cloud Sim is the most effective simulation tool which may be utilized to achieving demonstrating of cloud condition. There are four fundamental segments in Cloudsim engineering, which are useful to set up essential distributed computing conditions. This condition is utilized to measure the adequacy of task scheduling algo. These segments are Datacenter, Datacenter Broker, VM, & Cloudlet. Datacenter is in charge of giving hardware-level services to cloud clients. VM Datacenter Broker makes and dispenses with the requirements of the task. This hides the organization of VM as of user. VM tasks are processed according to the policy issued by Cloudlet Scheduler. Cloudlet is a job that works in a virtual machine.

Table I. Simulation Parameters

Parameter	Value
No. of server machines	20
Machine CPU/MIPS	(2500,3000)
Server memory/GB	(4,8)
Bandwidth/Mbps	(50, 100)
No. of VMs	200
Dem& of VMs	200
Bandwidth of virtual machines/GB	10-15
Task size/MI	(10000,50000)
No. of tasks	(0,100)

This experimentation groups 200 physical service nodes &arbitrarily produces the CPU routine of every physical server in [2500, 3000] (unit MIPS). Temporarily, the presented task length is usual to [10000, 50000] (unit MI), & is used randomly to form each task length. At the same time, suppose that all the presented tasks come at a similar time, that is, the time of task is similar. Task deadlines will be randomly produced in [0,100] (units).



```
<terminated> TestIGASA [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (Oct 1, 2019, 4:30:46 PM)
18.303179472876863: Broker1: Destroying VM #95
18.303179472876863: Broker1: Destroying VM #96
18.303179472876863: Broker1: Destroying VM #97
18.303179472876863: Broker1: Destroying VM #98
18.303179472876863: Broker1: Destroying VM #99
Broker1 is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_1 is shutting down...
Broker1 is shutting down...
Simulation completed.
Simulation completed.

========== OUTPUT ==========
Cloudlet_ID     STATUS  DataCenter_ID   VM_ID   Time    Start_Time      Finish_Time
5               SUCCESS 2               4       5.06    0.1             5.16
1               SUCCESS 2               2       5.26    0.1             5.36
7               SUCCESS 2               8       8.02    0.1             8.12
6               SUCCESS 2               21      8.47    0.1             8.57
8               SUCCESS 2               22      11.91   0.1             12.01
4               SUCCESS 2               50      12.62   0.1             12.72
3               SUCCESS 2               6       15.26   0.1             15.36
9               SUCCESS 2               0       15.53   0.1             15.63
2               SUCCESS 2               64      17.69   0.1             17.79
0               SUCCESS 2               32      18.2    0.1             18.3
Total Execution Time            118.03
Finish Time             18.3
Used Memory:36 mb
CloudSimExample finished!
```

Fig. 2. Load balancing comparison of LBIGASA & LBIGA at 10 cloudlets

In fig. 2 depicts the number of cloudlets (i. e. 10 cloudlets) those are successful with various virtual machine. It also displays the started time & finished time. So, the total execution time for each cloudlet has displayed. Similarly, this will be performed for 20, 30, 40, 50, 60, 70, 80, 90 & 100 cloudlets. This is the comparison between improved genetic algo with simulated annealing to optimize load balance & improved genetic algo.

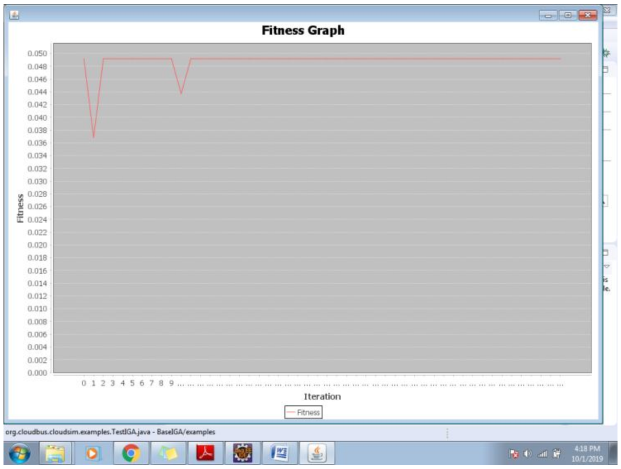


Fig. 3. Fitness function graph of LBIGA at 10 cloudlets

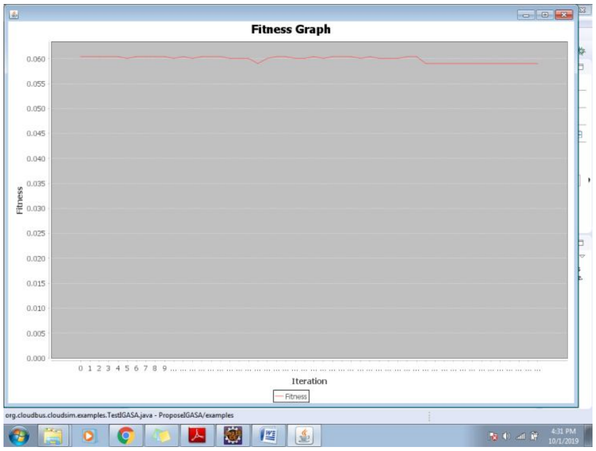


Fig. 4. Fitness function graph of LBIGASA at 10 cloudlets

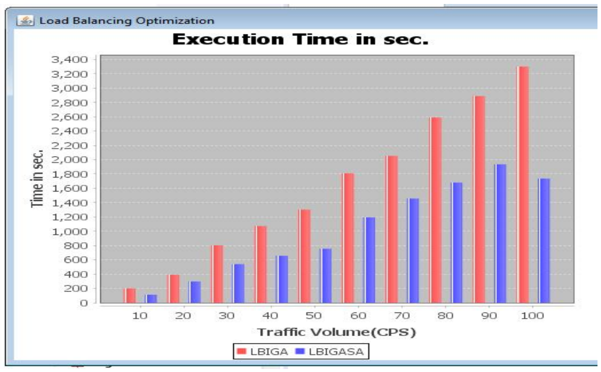


Fig. 5. Time-Span of LBIGA vs. LBIGASA

The user accepts task completion time, which is considered as beginning time to operating of task. The completion time of the user indicates the completion time of the task. The comparison of 2algo in cloud computing resources LB is shown in Figure 5.

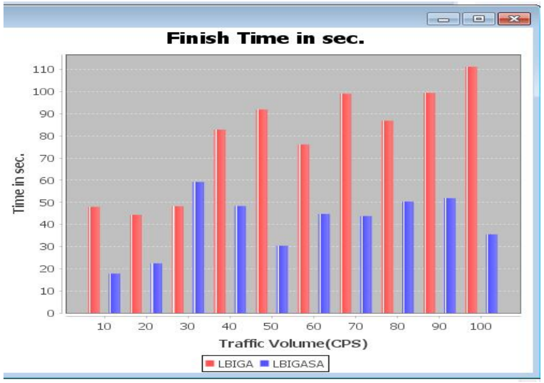


Fig. 6. Finishing time of Last VM

Fig. 6 shows the completion time for the last virtual machine. It represents the comparison of existing improved genetic algo for load balancing with improved genetic algo with simulated annealing for load balancing at various traffic volumes in cloudlets per second.

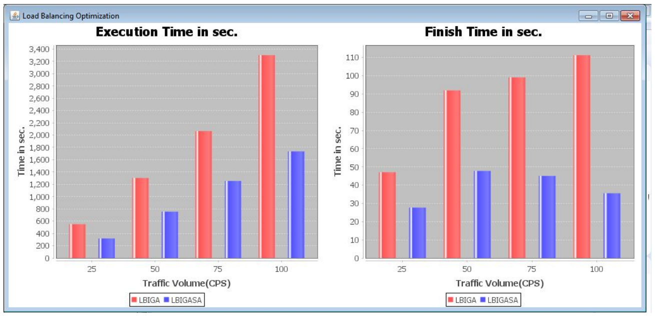


Fig. 7. Time comparison graph results with 25, 50, 75, 100 numbers of cloudlets

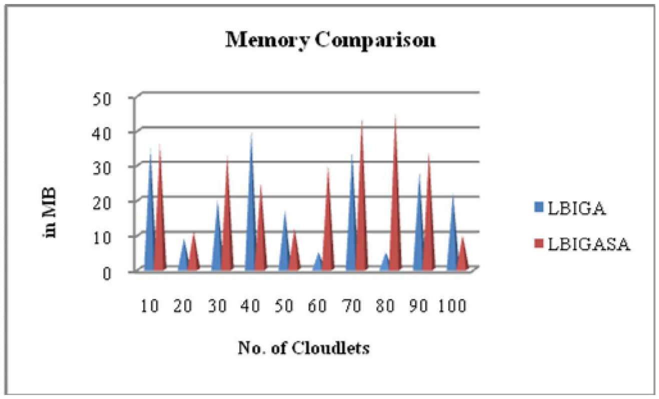


Fig. 8. Memory usage comparison of LBIGASA & LBIGA

Fig. 8 shows the used memory comparison for both algos at 10 to 100 numbers of cloudlets. Memory is measured in MB. There are variations in memory usage for both algos i.e. LBIGA and LBIGASA. Sometimes proposed LBIGASA uses less memory use in compare to LBIGA and sometimes LBIGA uses less memory in compare to LBIGASA.

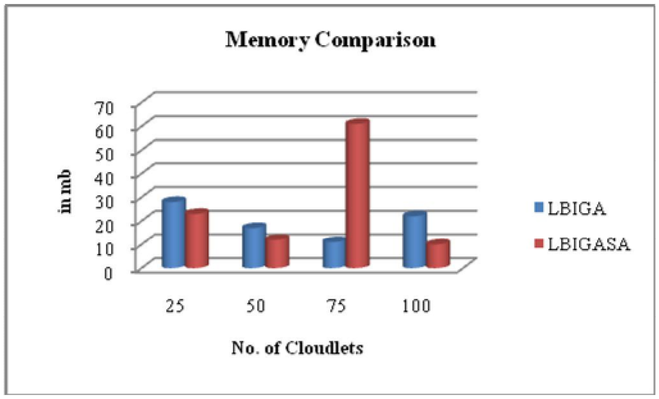


Fig. 9. Memory comparison of LBIGASA & LBIGA with 25, 50, 75, 100 numbers of cloudlets

VI. CONCLUSION

Due to the plethora of traffic & service requests in cloud datacenter, LB is a serious challenge in cloud environments owing to failure in energy ingesting, node failure & machine failure. Therefore a policy is needed to balance the load between datacenters. Paper discusses LBO in a cloud environment. Tournament selection strategy & simulated annealing have been presented to establish a fitness function model based on traditional genetic algo& to advance an enhanced genetic algo based resource loading optimization model. Formerly, we choose CloudSim as an ultimate simulating tool to pretend data center, virtual resources, & users' tasks in the cloud environment. The operator’s task completion period & LB are compared with simulated annealing to optimize load balancing using an improved genetic algo than an improved genetic algo. Experimental results suggest that our scheme is better & can be considered acurrent LB algo in the cloud environment. This takes less execution time than the previous method & is also useful for better resource allocation. There is also shows a comparison of memory usage and fitness function.

REFERENCES

[1] J. Tsai, J. Fang, J. Chou, “Optimized task scheduling & resource allocation on cloud computing Environment using improved differential evolution algo”, Computers &Operations Research 40, PP. 3045–3055,2013.

[2] Nair, N. K., Navin, K. S., &Ch&ra, C. S. S. (2015). A survey on load balancing problem & implementation of replicated agent-based load balancing technique. 2015 Global Conference on Communication Technologies (GCCT), pp. 897-901.

[3] Ibrahim, E., El-Bahnasawy, N. A., &Omara, F. A. (2016). Task Scheduling Algo in Cloud Computing Environment Based on Cloud Pricing Models. 2016

World Symposium on Computer Applications & Research (WSCAR).

[4] Sreelakshmi& S. Sindhu, "Multi-Objective PSO Based Task Scheduling - A Load Balancing Approach in Cloud," *2019 1st International Conference on Innovations in Information & Communication Technology (ICIICT)*, CHENNAI, India, 2019, pp. 1-5.

[5] Bharot, N., Suraparaju, V., & Gupta, S. (2019). DDoS Attack Detection and Clustering of Attacked and Non-attacked VMs Using SOM in Cloud Network. Advances in Computing and Data Sciences, 369–378.

[6] M. Ashouraei, S. N. Khezr, R. Benlamri& N. J. Navimipour, "A New SLA-Aware Load Balancing Method in the Cloud Using an Improved Parallel Task Scheduling Algo," *2018 IEEE 6th International Conference on Future Internet of Things & Cloud (FiCloud)*, Barcelona, 2018, pp. 71-76.

[7] Y. Samadi, M. Zbakh& C. Tadonki, "E-HEFT: Enhancement Heterogeneous Earliest Finish Time algo for Task Scheduling based on Load Balancing in Cloud Computing," *2018 International Conference on High-Performance Computing & Simulation (HPCS)*, Orleans, 2018, pp. 601-609.

[8] M. Padmavathi& S. M. Basha, "Dynamic & elasticity ACO load balancing algo for cloud computing," *2017 International Conference on Intelligent Computing & Control Systems (ICICCS)*, Madurai, 2017, pp. 77-81.

[9] Bharot, N., Verma, P., Sharma, S., &Suraparaju, V. (2017). Distributed Denial-of-Service Attack Detection and Mitigation Using Feature Selection and Intensive Care Request Processing Unit. Arabian Journal for Science and Engineering, 43(2), 959–967

[10] A. Gupta & R. Garg, "Load Balancing Based Task Scheduling with ACO in Cloud Computing," *2017 International Conference on Computer & Applications (ICCA)*, Doha, 2017, pp. 174-179.

[11] Y. Fang & X. Li, "Task Scheduling Strategy for Cloud Computing Based on the Improvement of Ant Colony Algo," *2017 International Conference on Computer Technology, Electronics & Communication (ICCTEC)*, Dalian, China, 2017, pp. 571-574.

[12] NiteshBharot, Priyanka Verma, VeenadhariSuraparaju and Sanjeev Gupta, “Mitigating Distributed Denial of Service Attack in Cloud Computing Environment using Threshold-based Technique”, Indian Journal of Science and Technology, Vol 9(38), October 2016, pp. 1-7.

[13] S. Aslanzadeh& Z. Chaczko, "Load balancing optimization in cloud computing: Applying Endocrine-particle swarm optimization," *2015 IEEE International Conference on Electro/Information Technology (EIT)*, Dekalb, IL, 2015, pp. 165-169.

[14] Karaulia, D. S., &Bharot, N. (2014). Dynamic TTL Variance Foretelling Based Enhancement of AODV Routing Protocol in MANET. 2014 Fourth International Conference on Communication Systems and Network Technologies, pp. 244-248.

[15] W. Sun, N. Zhang, H. Wang, W. Yin & T. Qiu, "PACO: A Period ACO Based Scheduling Algo in Cloud Computing," *2013 International Conference on Cloud Computing & Big Data*, Fuzhou, 2013, pp. 482-486.

[16] K. Li, G. Xu, G. Zhao, Y. Dong & D. Wang, "Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization," *2011 Sixth Annual Chinagrid Conference*, Liaoning, 2011, pp. 3-9.