ANT-LOAD: A Proficient Meta-Heuristic Cloud Load Balancing

Ankita Taneja
Department of Computer Science and Engineering
P.I.E.T, Kurukshetra University,
Haryana 132102. India.

Deepti Dhingra
Department of Computer Science and Engineering
P.I.E.T, Kurukshetra University,
Haryana 132102. India.

ABSTRACT

The Cloud computing is an embryonic as an innovative hypothesis of gigantic distributed calculation. Load balancing, the main trial in cloud computing, requires to allocate the vibrant workload uniformly across all of the machines. Burden balancing leads to a high user satisfaction and resource utilization ratio by confirming a proficient and fair allocating of all of the resources. Burden Balancing additionally supports ranking users by applying suitable method for scheduling. This paper concludes the counseled algorithm, Ant colony optimization, to resolve the setback of burden on the nodes in the cloud web, making the nodes burden free to work. This paper displays the drawbacks of Genetic Algorithm are resolved employing ACO for balancing the burden in the cloud network.

Keywords

ACO, Cloud Computing, Load Balancing, Pheromone table.

1. INTRODUCTION

Cloud Calculating is the novel trend growing in IT environment alongside huge infrastructural and resources requirements. It has voluminous benefits alongside a little of the critical concerns to be fixed in order to increase dependability of cloud system. The “cloud” in cloud computing can be delineated as the set of hardware, webs, storage, services, and interfaces that link to grasp aspects of computing as a service[1][2].The main concern in cloud computing are connected alongside the burden balancing, obligation tolerance and countless kinds of protection setbacks in cloud system. The Three Cloud models defines the Cloud services:

Cloud Software as a service (SAAS): Users use the provider’s requests on a cloud groundwork employing web browser. The users cannot grasp the fundamental infrastructure of the cloud.

Cloud Platform as a Service (PAAS): Users arrange/install onto the cloud groundwork user crafted or acquired applications generated employing the software design tongues and tools. User cannot grasp or manipulation the fundamental infrastructure of cloud, but has the manipulation above the installed applications.

Cloud Infrastructure as a Service (IAAS): User has provision to procedure, store, webs, and supplementary fundamental computing resources whereas the user is capable to mount and run/compile the arbitrary multimedia, consists of applications and working systems. User cannot grasp or manipulation the fundamental groundwork of the cloud, but has manipulation over the storage, installed requests, working arrangements, and probably partial manipulation of select networking components. The datacenter hardware and multimedia is what we will call a Cloud. After a Cloud is made obtainable in a pay-as-you-go manner to the finished area, we call it an Area Cloud; the ability being vended is Utility Computing [7] [8]. Features of IaaS contain vibrant scaling, Desktop virtualization, utility computing ability and billing model, Policy-based services and Automation of official tasks.

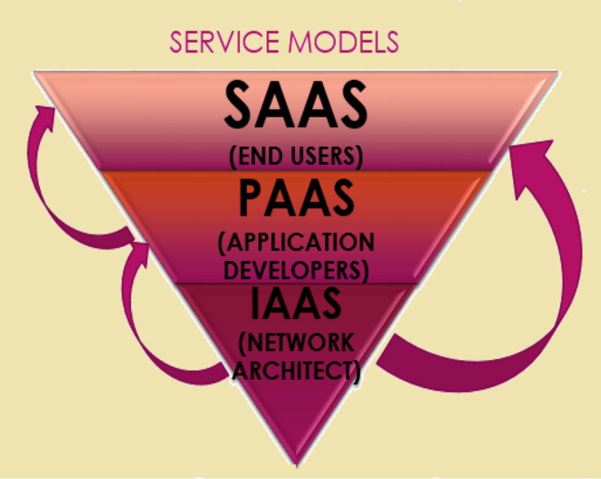

Figure 1: Cloud Service Model

The subject of IAAS burden balancing can be resolved by redistribution of burden amid the processors that increases the system’s performance. The benefits of proper load balancing are minimizing resource consumption, enabling scalability, requesting fail-over, circumventing overprovisioning and bottlenecks and so on.

The public setback arises due to burden balancing are:

- ❑ The jobs staying in the queue.
- ❑ Arrival rate of Jobs.
- ❑ Physical Machine’s processing rate.

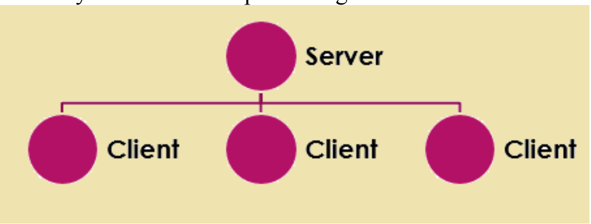

Figure 2: Client’s appeal on alike Server leads to Load

Load Balancing uses the method to allocate the stream of traffic amid the servers or the nodes in the cluster or the network. Burden on the main server can be distributed by generating the cluster server head, can be uttered as virtual machine, in the network. These Cluster server head may further distributes the burden to disparate severs and client requests are consented by these servers, that finished balance the burden of the network.

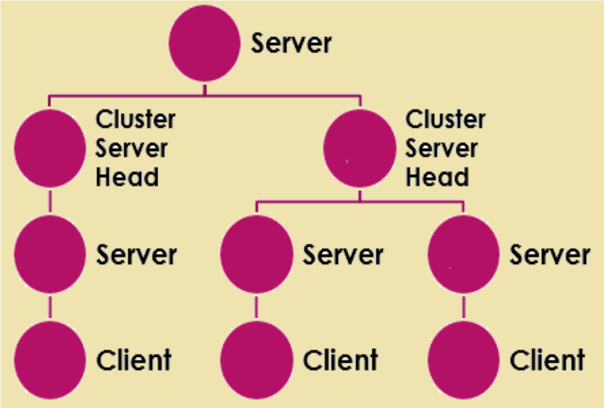

Figure 3: Client's appeal allocation on disparate servers. (VM Burden Balance)

The main goal of burden balancing is as follows:

- ❑ Even allocation of burden to every single resource
- ❑ Minimization of processing period for every single job
- ❑ Maximum utilization of every single resource
- ❑ To enhance the presentation substantially.
- ❑ To have a backup design in case the arrangement fails even partially.
- ❑ To uphold the arrangement stability.
- ❑ To accommodate upcoming modification in the system.

2. RELATED WORK

The burden balancing methods that are extensive in cloud computing, stated:

MIN-MIN Algorithm: Set of laws initialized alongside a set of unassigned jobs. Early of all, we discovered lowest period for the accomplishment for all the jobs. Later that, least number of times, the least worth is selected in that the minimum times among the jobs on the assets. Then, afterward this, according to that least period, the job arranging is completed on the corresponding arrangement or machine. Next the notifying, of execution period, is completed for all supplementary jobs on that contraption by the supplement of the period of killing of all the allocated job to the number of period of killing of supplementary tasks for the same machine and all the allocated jobs are removed from the catalog of the jobs that are to be allocated to the machineries. Once more the same arrangement is trailed till resources hold all the allocated jobs. On the supplementary hand, the main disadvantage of this method is that it can lead to the setback of starvation [9] or it cannot design the long period jobs. Min-Min algorithm processes the smaller task first. Min-Min doesn't produces optimal schedule. Min-min's setback is overcome by the Min-Max algorithm. [10]

MIN-MAX Algorithm: Set of laws are requested in two player logical games, such as tic-tac-toe, chess and so on. The algorithm directs the computer contestant to make the next possible move in the game. The algorithm sketches the "full informative" games in that every single contestant is acquainted with everything concerning the probable moves of the opponent. The Algorithm generates the find tree to resolve the alternative possible game move of the players. These set of laws are only applicable, or functional for, to the logical games. In this approach, the maximum value is selected to get to the solution [9]. The Min-Max approach is used for the pheromone trail updation by the iteration's best ant to get the optimal solution [6].

Nishant, K. et al, [11]. In this paper, they counselled an algorithm for burden allocation of workloads amid nodes of a cloud by the use of Ant Dominion Optimization (ACO). This is an adjusted way of ant dominion optimization that has been requested from the outlook of cloud or grid web arrangements alongside the main target of burden balancing of nodes. This adjusted algorithm has an frontier above the early way in that every single ant craft their own individual consequence set and it is afterward on crafted into a finished solution. Though, in their way the ants unceasingly notify a solitary consequence set rather than notifying their own consequence set. Further, as they understand that a cloud is the collection of countless nodes that can prop assorted kinds of request that is utilized by the clients on a basis of wage each use. Therefore, the arrangement, that is incurring a price for the user ought to purpose smoothly and ought to have algorithms that can tolerate the proper arrangement working even at top custom hours.

Ajit, M. et al, [12]. As Cloud Calculating is spreading globally and number of users demanding extra cloud services and larger aftermath are producing quickly, cloud burden balancing come to be an extremely interesting and vital scrutiny area. Generally, cloud is established on influential datacentres that grasp colossal number of users, so it have to be showcased alongside burden balancer to accomplish reliability that depends on the method it grips the load. Cloud burden balancing helps to enhance the finished cloud performance. Countless algorithms were counselled for allocating the users demands to Cloud resources to furnish services efficiently. This paper presents the scrutiny of three present algorithms in cloud analyst instrument to ascertain the subject of cloud burden balancing as an arranging period for new burden balancing technique. A Weighted Signature established burden balancing (WSLB) algorithm is counselled to minimize user's reply time. Further, this paper additionally provides the anticipated aftermath alongside the implementation of the counselled algorithm.

Pacini, E. et al, [13]. Scientists and builders normally need huge numbers of computing manipulation for giving their experiments. Precisely, Parameter Brush Examinations (PSE) permit these kind of users to present simulations by running the alike logical program alongside disparate input data, that aftermath in countless CPU-intensive jobs and therefore computing settings such as Clouds have to be used. They delineate two Cloud schedulers established on two accepted swarm intellect (SI) methods, namely ant dominion optimization (ACO) and particle swarm optimization (PSO), to allocate adjacent mechanisms (VM) to physical Cloud resources. The main presentation metrics to discover are the number of serviced users by the Cloud -i.e., the number of Cloud users that the cheduler is able to prosperously serve- and the finished number of crafted VMs, in vibrant (non-batch) arranging scenarios. Simulated examinations gave by employing CloudSim and real PSE job data counsel that their schedulers, across a weighted metric, present competitively alongside respect to the number of serviced users and accomplish an competent assignment of VMs contrasted to a scheduler established on Genetic Algorithms.

Silva, M. et al, [14]. The development in the adoption of cloud computing is driven by different and clear benefits for both cloud clients and cloud providers. Though, the rise in the number of cloud providers as well as in the collection of offerings from every single provider has made it harder for clients to choose. At the alike period, the number of options to craft a cloud groundwork, from cloud association periods to disparate interconnection and storage technologies, additionally poses a trial for cloud providers. In this context, cloud examinations are as vital as they are labour intensive.

Chun-Wei Tsai et al, [15]. Cloud computing has come to be an increasingly vital scrutiny case given the forceful progress and migration of countless web services to such computational environment. The setback that arises is connected alongside efficiency association and utilization of the colossal numbers of computing resources. This paper begins alongside a brief retrospect of established arranging, pursued by a methodical study of metaheuristic algorithms for resolving the arranging setbacks by allocating them in a fused framework. Armed alongside these two technologies, this paper surveys the most present works concerning metaheuristic arranging resolutions for cloud. In supplement to requests employing metaheuristics, a little vital subjects and open inquiries are gave for the reference of upcoming researches on arranging for cloud.

Hung, P.P. et al, [16]. To accomplish elevated presentation, thousands of servers in cloud datacentres coordinate tasks to furnish reliable and exceedingly obtainable cloud computing services, exceptionally, in words of multitasking. Competent mechanisms are nowadays needed to design for a wreck of such computing nodes. A number of studies have been completed to address this setback, but it cannot always promise a satisfactory performance. In this paper, they present an arranging algorithm, established on price and bandwidth that makes effectual recovery probable on heterogeneous computing environments. The algorithm not merely considers the web bandwidth, but additionally seizes into report the monetary price as well. They validate their counselled work across comprehensive simulations and difference their work alongside the continuing studies. The aftermath can enhance the possible benefit of their approach.

3. ANT COLONY OPTIMIZATION

Initially, Macro Dorigo has proposed an algorithm to explore the optimum route in the graph, based on the ant's way of moving looking for the track sandwiched between the colony and source of food, in his PhD thesis in 1992.Ant Dominion Optimization (ACO) [1] is currently recommended as metaheuristic tactic for cracking the inflexible combinatorial optimization problems. The motivational foundation of ACO is the pheromone stream assigning and trailing deeds of factual ants that use pheromones as a connexion medium.

Initial instance of such an algorithm is Ant Arrangement (AS) [2]. AS was counselled employing as example requests the well-recognized Voyaging Salesman Setback (TSP) [3].Although reassuring early aftermath, AS might not contest alongside state-of-the-art algorithms for the TSP. Nevertheless, it had the vital act of invigorating more scrutiny on algorithmic variants that attain far larger computational presentation, as well as on requests to a colossal collection of disparate problems. In fact, there exists nowadays a substantial number of requests obtaining globe class presentation on setbacks like the quadratic assignment, vehicle routing, sequential arranging, arranging, routing in Internet-like webs, and so on [4]. Motivated by this accomplishment, the ACO metaheuristic has been counselled [5] as a public framework for the continuing requests and algorithmic variants. Algorithms that pursue the ACO metaheuristic will be shouted in the pursuing ACO algorithms.

ANT MECHANISM:

The frank believed of the ant dominion optimization Meta heuristic is seized from the food hunting deeds of real ants. After ants are on the practice to find for food, they onset from their nest and stroll in the direction of the food. After an ant reaches an intersection, it has to choose that division to seize next. As walking, ants deposit pheromone that marks the path taken. The compression of pheromone on a precise trail is an indication of its usage. With period the compression of pheromone cuts due to diffusion effects. This property is vital because it is incorporating vibrant aspect into the trail hunting process. Suddenly, an obstacle gets in their method so the early ants randomly select the subsequent division amid the two branches: the higher and lower branches. As the higher path is shorter than the lower one, the ants that seize this shorter trail will grasp the food locale first. On their method back to the nest, the ants once more have to select a path. Later a short period the pheromone compression on the shorter trail will be higher than on the longer trail, because the ants employing the shorter trail will rise the pheromone compression faster. The shortest trail will therefore be recognized and in the end all ants will merely use this one. This deeds of the ants can be utilized to find the shortest trail in webs.

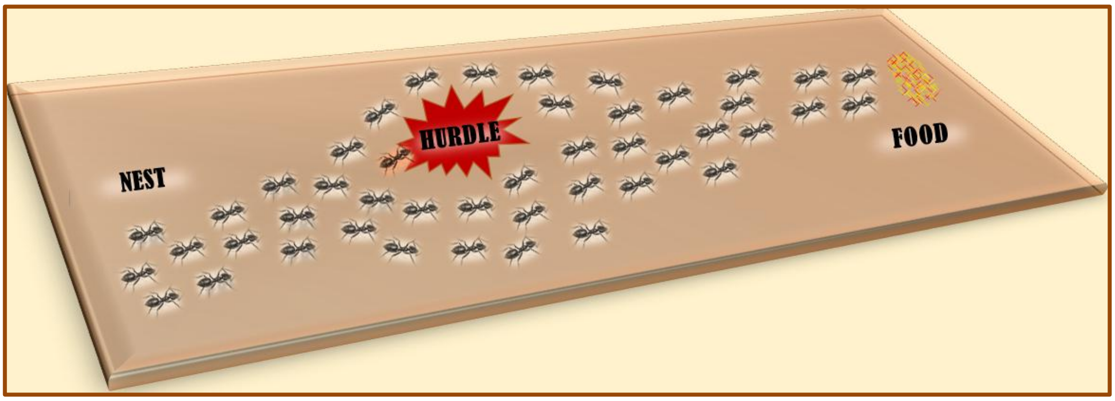

Figure 4: Ants Behaviour

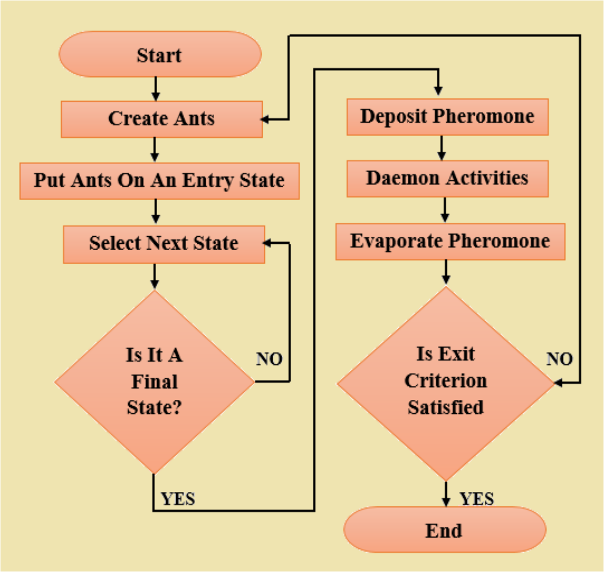

Figure 5: Flow Construction of Ant Behavior

4. PROPOSED WORK

The limitations of Genetic algorithm are vanquish by implementing ACO for burden balancing. The Genetic algorithm leaves the drawbacks as:

- ❑ GENETIC Algorithm is period consuming that makes them improper for online applications.
- ❑ Bulky to work on vibrant data sets, as convergence is the main issue.
- ❑ GA's have tendency to encounter towards innate optima rather than the globe optimum of the problem.

These limitations can be resolved by the Algorithm established on Ant behavior, i.e., Ant Dominion Optimization.

Let us ponder the minimization setback, (S, f, Ω) whereas S is the set of candidate resolutions, *f* is the goal function which assigns to every single candidate resolution an goal function (cost) value *f*(s, t)^4 and Ω is a set of constraints.
The aim is to find a globally optimal resolution s_{opt} ∈ s that is, a minimum cost resolution that gratifies the constraints Ω.

The setback representation of a combinatorial optimization problem (S, f, Ω), that is exploited by the ants, can be characterized as follows:

A finite set C = {c1, c2...c_{Nc}} of machines is given. The states of the setback are described in words of sequences |x= (ci,cj,…..,ck,…) above the agents of C. The set of all possible sequences is denoted by X. The length of a sequence x, that is, the number of constituents in the sequence, is expressed by |x|.

The finite set of constraints Ω defines the set of feasible states $X^{\tilde{}}$, alongside $X^{\tilde{}}$ ⊑ X. The length of a sequence x, that is, the number of constituents in the sequence, is expressed by |x|.

A set S* of feasible resolutions is given, alongside S* ⊆ $X^{\tilde{}}$ and S* ⊆ *S*.

A cost *f* (s, t) is associated to every single candidate resolution s ∈ S.

In some of the cases, the cost, or the guesstimate of a cost, |J (x_i, t) can be associated to states supplementary than solutions. If x_i can be obtained by adding resolution constituents to a state x_i then J (x_i, t) < J (x_j, t). Note that *J(s,t)* ≡ *f(s,t)*.

Given this representation, imitated ants craft resolutions by moving on the assembly graph G= (C, $\mathcal{L}$) whereas the vertices are the constituents and the set $\mathcal{L}$ fully links the components C (elements of $\mathcal{L}$ are shouted connections). The problem constraints Ω are requested in the strategy followed by the manmade ants, as clarified in the subsequent section. The choice of requesting the constraints in the construction policy of the manmade ants permits a precise degree of flexibility.

In fact, reliant on the combinatorial optimization problem considered, it could be extra reasonable to implement constraints in a hard method permitting ants to craft merely feasible solutions, or in a soft method, in that case ants can build infeasible resolutions (that is, candidate resolutions in S/S*) that will be penalized, in dependence of their degree of infeasibility.

ANT'S BEHAVIOR:

Ants can be described as stochastic construction procedures that craft resolutions advancing on the construction graph G = (C, $\mathcal{L}$). Ants do not travel randomly on G, but rather follow an assembly strategy that is a purpose of the problem constraints Ω. In finish, ants endeavor to craft feasible solutions, but, if vital, they can produce infeasible solutions. Constituents c_i ∈ *C* and connections l_{ij} ∈ $\mathcal{L}$ can have associated a pheromone trail τ (τ_i if associated to components τ_{ij}, if associated to connections) encoding a long-term memory about the finished ant find procedure that is notified by the ants themselves, and a heuristic worth η (n_i and n_{ij} respectively) representing a priori data concerning the setback instance definition or run-time data endowed by a source different from the ants. In countless cases η is the cost, or an estimate of the cost, of spreading the present state. These values are utilized by the ant's heuristic law to make probabilistic decisions on how to move on the graph.

More precisely, every single ant k of the dominion has the following properties:

- ❑ It exploits the graph G = (C, $\mathcal{L}$) to find for feasible solutions s of minimum cost. That is, resolutions such that *f*ˆs = $\min_6$ *f*(s, t).
- ❑ It has a recollection M^k that it uses to store information about the trail it pursued so far. Recollection can be used (i) to craft feasible resolutions (i.e., to implement constraints Ω), (ii) to assess the resolution discovered, and (iii) to retrace the trail retrograde to deposit pheromone.
- ❑ It can be allocated an onset state x_s^k and one or more termination conditions e^k normally, the onset state is expressed as a constituent length sequence, that is, a single component or an empty sequence.
- ❑ After in state x_r= ($x_{r-1,\ i}$) it attempts to move to each node j in its feasible area N_i^k that is to a state (x_r, j) ∈ $\tilde{X}$. If this is not probable, next the ant can move to a node j in its impossible area IN_i^k, generating in this method an impossible state (i.e., a state (x_r, j) ∈ X /$\tilde{X}$}.
- ❑ It selects the move by requesting a probabilistic decision rule. Its probabilistic decision law is a purpose of (i) locally obtainable pheromone tracks and heuristic values, (ii) the ant's confidential recollection storing its past history, and (iii) the setback constraints.
- ❑ The assembly procedure of ant k stops after at least one of the termination conditions e^k is satisfied.
- ❑ After adding a constituent c_i to the present resolution it can notify the pheromone trail associated to it or to the

corresponding connection. This is shouted online step-by-step pheromone update.

- ❑ After crafted a resolution, it can retrace the alike path backward and notify the pheromone tracks of the used components or connections. This is shouted online delayed pheromone update.

It is vital to note that ants move concurrently and independently and that every single solitary ant is convoluted plenty to find a (probably poor) resolution to the setback below consideration. Typically, good quality resolutions materialize as the consequence of the collective link amid the ants that is obtained via indirect link mediated by the data ants read/write in the variables storing pheromone trail values. In a method, this is a distributed discovering procedure in that the solitary agents, the ants, are not adaptive themselves but, on the contrary, they adaptively adjust the method the setback is embodied and noted by supplementary ants.

ACO FOR LOAD BALANCING:

The easy ant dominion optimization meta-heuristic displayed in earlier disparate reasons why this kind of algorithms could perform well in networks.

- ❑ ACO is instituted on agent arrangements and works alongside individual ants. This permits an elevated adaptation to the present vibrant topology of the network;
- ❑ ACO is instituted merely on innate data, that is, no routing tables or supplementary data blocks have to be sent to acquaintances or to all nodes of the network;
- ❑ It is probable to incorporate the connection/link quality into the computation of the pheromone concentration;
- ❑ Every single node has a routing table alongside entries for all its acquaintances that encompasses additionally the pheromone concentration. Thus, the method supports multi-path Burden Balancing.

Hereby, we delineate the disparate procedures utilized to grasp the disparate events of the counselled protocol. The events to be grasped are acquaintance connectivity, trail formation appeal, trail formation answer, trail expiry, connection defeat and innate repair. Supplementary proper features of these constituents are given below. The pursuing data

1. **A pheromone matrix Ak**: a matrix coordinated as in vector distance routing table algorithms, but alongside probabilistic entries which is the pheromone worth of every single probable route. **Ak** defines the probabilistic routing strategy presently adopted at node **k**: for every single probable destination d and for every single neighbor node n, **Ak** stores a pheromone worth **Pnd** expressing the goodness (desirability), below the present network-wide routing strategy, of selecting **n** as subsequent node after the destination node is **d**;
2. **A Burden table Lk**: a table coordinated as in vector distance algorithms, but alongside a pheromone entry added. **Tk** defines the pheromone worth as well the usual routing entries at node **k**: for every single probable destination **d**, **Tk** stores a pheromone value **Pnd** expressing the best (most desirable) of selecting n as the next node after the destination node is **d**.

5. RESULTS

The generated results for Genetic Algorithm and Ant Colony Optimization, as follows:

GENETIC ALGORITHM BASED LOAD BALANCING:

The **table 1**, for Genetic algorithm, below shows that the various parameters such as number of Iterations, number of Cloud machines, number of Tasks, and Genetic Algorithm constraints, like number of Generations, Mutation States, Solution Population Size, have been specified. Note that, the number of Cloud machines and number of Tasks for both Genetic Algorithm and Ant Colony Optimization are kept same for the accurate comparison.

Table 1: GA Parameters

Number of Iterations	1000
Number of Cloud Machines	10
Number of Tasks in the Set	20
Number of Generations	100
Mutation States	2
Solution Population Size	100
Crossover Probability	0.95
Mutation Probability	0.05

The below graph, **figure 6**, shows the variation of Execution time with respect to the Iterations take place. The Execution time of Genetic Algorithm for each Iteration is more as compare to Ant Colony Algorithm.

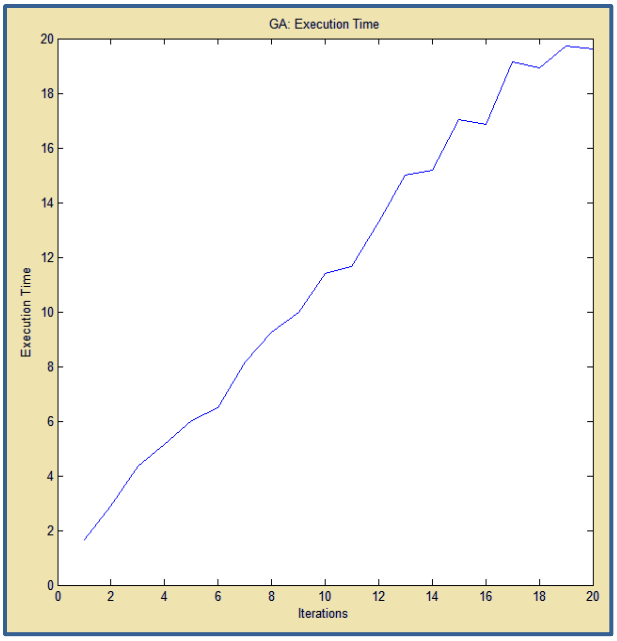

Figure 6: GA Execution time-Iteration Graph

The **figure 7** describes the maximum estimated cost for the number of task allotted to the number of cloud machines for each iteration. The evaluated cost using Genetic Algorithm depletes with respect to the iterations takes place. The Cost depletes at the slower rate, as compare to Ant colony optimization algorithm.

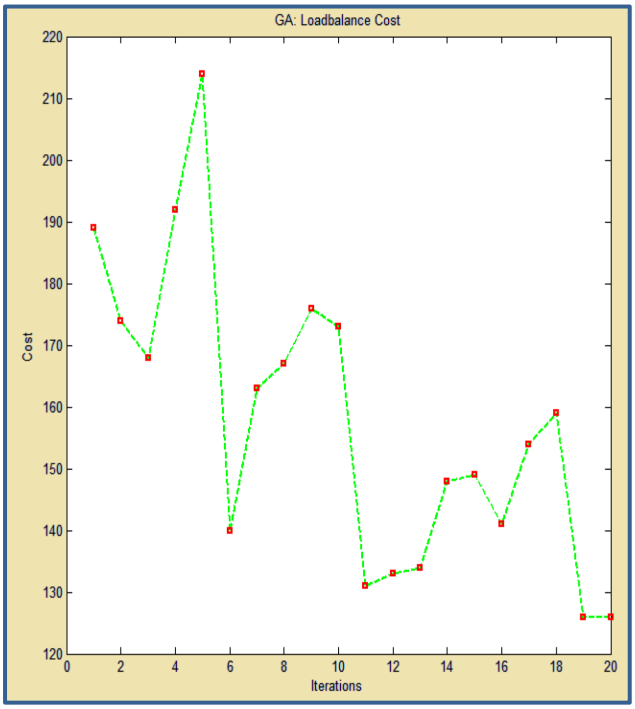

Figure 7: Estimated Cost of all Tasks by the Genetic Algorithm for Execution in given iteration.

ANT COLONY OPTIMIZATION BASED LOAD BALANCING:

The **table 2**, for Ant Colony Optimization, below shows that the various parameters such as number of Iterations, number of Cloud machines, number of Tasks, number of Ants and Ant Colony Optimization Algorithm constraints, like Pheromone Table Size, Pheromone Update, have been specified. Note that, the number of Cloud machines and number of Tasks for both Genetic Algorithm and Ant Colony Optimization are kept same for the accurate comparison.

Table 2: ANT Colony Algorithm Parameters

Number of Iterations	1000
Number of Cloud Machines (m)	10
Number of Tasks in the Set	20
Pheromone Table Size	m * m
Pheromone Update	0.04
Number of ANTS	20

The below graph, **figure 8**, shows the variation of Execution time with respect to the Iterations take place. The Execution time of Genetic Algorithm for each Iteration is more as compare to Ant Colony Algorithm, which makes ACO more suitable for balancing the load or the traffic of client requests. The Execution time for Genetic Algorithm is shown with the Blue plane line and Execution time for Ant Colony Optimization Algorithm is shown with the Red Dashed line in the graph.

The **figure 9** depicts the maximum estimated cost for the number of task allotted to the number of cloud machines for each iteration. The evaluated cost using Ant Colony Optimization Algorithm depletes, with respect to the iterations, faster as compared to Genetic Algorithm.

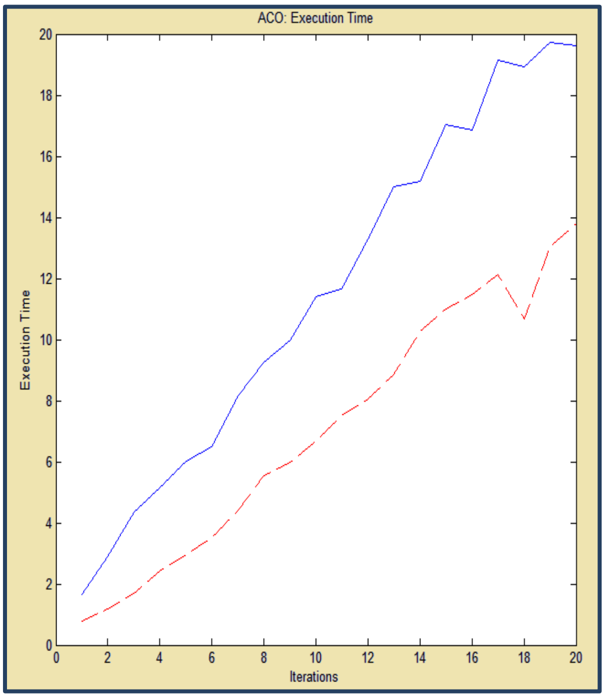

Figure 8: ACO Execution time-Iteration Graph

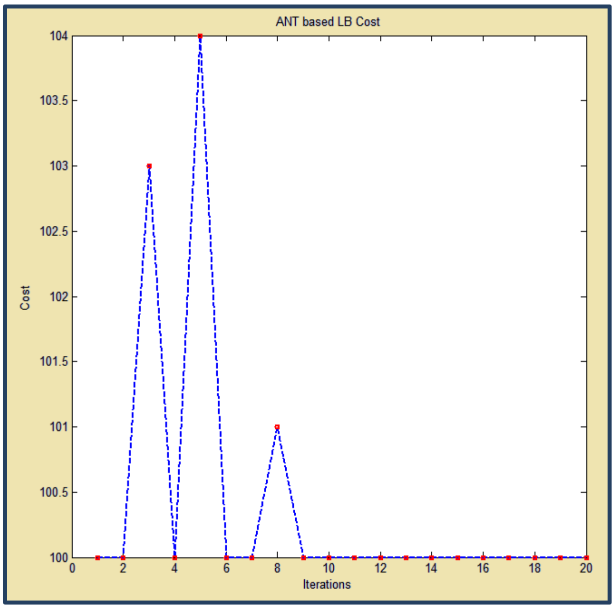

Figure 9: Estimated Cost of all Tasks by the Ant Colony Optimization Algorithm for Execution in given iteration.

OBSERVED RESULTS:

For a given number of cloud machines and number of tasks, the Ant Colony Optimization was found to be far superior then Genetic Algorithm. According to our observations, the Ant Colony Optimization benefits in two ways:

- ❑ Maximum Cost in Ant Colony Optimization is less than that of Cost of Genetic Algorithm.
- ❑ Time required to execute the load balance process for a given number of iterations is less for Ant Colony Optimization as compare to Genetic Algorithm.

This makes Ant Colony Optimization Algorithm more Energy Efficient Algorithm in comparison to Genetic Algorithm.

6. CONCLUSION

Cloud Computing has broadly been accepted by the industry or organization though with many existing issues like Load Balancing, Virtual Machine Consolidation, Energy Management, etc. Central of all is the issue of balancing the load that is required to allocate the additional dynamic local workload uniformly to all the nodes in the Cloud which leads to high user satisfaction and every computing resource is distributed proficiently and equally.An analogy was be made amid disparate strategies encompassing Genetic Algorithm and Ant Colony. In this work we have describe a cloud arrangement, as the Power of a cloud arrangement is described to purpose as the pair of traffic loads below that the queues in the contraption and is be stabilized ACO that is employing based. The load-balancer receives data concerning can accord data considering the Task from an individual and accordingly balance the cost. ACO established resolution aftermath in less computationally intensive load-balancers alongside regards to the instance whereas the load-balancer ought to somehow guesstimate the chances of giving the discretional agents, but needs contact that is extra. The cost, nevertheless, is tremendously restricted.

6.1 Future Scope

Load Balancing is a task that is vital Cloud Calculating nature to attain maximum utilization of resources. The request ideal utilized in this paper assumes a finite number of users that way load-balancer that is industrialized maximize the percentage of discretional content served. Though, after a request that is disparate is believed, optimizing the definite number of demands assisted alongside discretional content is one more probable goal, that ought to be investigated in upcoming work. Also, presentation of the cloud computing be more maximized if dependencies amid tasks are modeled employing ACO established workflows.

7. ACKNOWLEDGMENT

We, the writers, would like to show the gratitude to the anonymous arbitrators for their respected annotations that enriched the content and the staging of this research and the paper.

8. REFERENCES

[1] Yang, Yubin, Hui Lin, and Jixi Jiang. "Cloud analysis by modeling the integration of heterogeneous satellite data and imaging." *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 36, no. 1 (2006): 162-172.

[2] Kaewpuang, Rakpong, Putchong Uthayopas, Ganid Srimool, and Juta Pichitlamkhen. "Building a service oriented cloud computing infrastructure using microsoft CCR/DSS system." In *Computer Sciences and Convergence Information Technology, 2009. ICCIT'09. Fourth International Conference on*, pp. 812-817. IEEE, 2009.

[3] Wu, Tao, and Kun Qin. "Inducing Uncertain Decision Tree via Cloud Model." In*Semantics, Knowledge and Grid, 2009. SKG 2009. Fifth International Conference on*, pp. 85-91. IEEE, 2009.

[4] Zhao, Yi, and Wenlong Huang. "Adaptive distributed load balancing algorithm based on live migration of virtual machines in cloud." In *INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on*, pp. 170-175. IEEE, 2009.

[5] Zhang, Zehua, and Xuejie Zhang. "A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation." In*Industrial Mechatronics and Automation (ICIMA), 2010 2nd International Conference on*, vol. 2, pp. 240-243. IEEE, 2010.

[6] Feller, Eugen, Louis Rilling, and Christine Morin. "Energy-aware ant colony based workload placement in clouds." In *Proceedings of the 2011 IEEE/ACM 12th International Conference on Grid Computing*, pp. 26-33. IEEE Computer Society, 2011.

[7] Kruber, Nico, Mikael Hogqvist, and Thorsten Schutt. "The Benefits of Estimated Global Information in DHT Load Balancing." In *Proceedings of the 2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 382-391. IEEE Computer Society, 2011.

[8] Li, Kun, Gaochao Xu, Guangyu Zhao, Yushuang Dong, and Dan Wang. "Cloud task scheduling based on load balancing ant colony optimization." In *Chinagrid Conference (ChinaGrid), 2011 Sixth Annual*, pp. 3-9. IEEE, 2011.

[9] Kokilavani, T., and Dr DI George Amalarethinam. "Load balanced min-min algorithm for static meta-task scheduling in grid computing." *International Journal of Computer Applications* 20, no. 2 (2011): 43-49.

[10] Parsa, Saeed, and Reza Entezari-Maleki. "RASA-A New Grid Task Scheduling Algorithm." *JDCTA* 3, no. 4 (2009): 91-99.

[11] Nishant, Kumar, Pratik Sharma, Vishal Krishna, Chhavi Gupta, Kuwar Pratap Singh, and Ravi Rastogi. "Load balancing of nodes in cloud using ant colony optimization." In *Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on*, pp. 3-8. IEEE, 2012.

[12] Ajit, M., and G. Vidya. "VM level load balancing in cloud environment." In*Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on*, pp. 1-5. IEEE, 2013.

[13] Pacini, Elina, Cristian Mateos, and Carlos Garcia Garino. "SI-based scheduling of scientific experiments on Clouds." In *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2013 IEEE 7th International Conference on*, vol. 2, pp. 699-704. IEEE, 2013.

[14] Silva, Marcio, Michael R. Hines, Daniele Gallo, Qi Liu, Kyung Dong Ryu, and Dilma Da Silva. "CloudBench: experiment automation for cloud environments." In *Cloud Engineering (IC2E), 2013 IEEE International Conference on*, pp. 302-311. IEEE, 2013.

[15] Tsai, Chun-Wei, and Joel JPC Rodrigues. "Metaheuristic scheduling for cloud: A survey." *Systems Journal, IEEE* 8, no. 1 (2014): 279-291.

[16] Hung, Pham Phuoc, Mui Van Nguyen, Mohammad Aazam, and Eui-Nam Huh. "Task scheduling for optimizing recovery time in cloud computing." In*Computing, Management and Telecommunications (ComManTel), 2014 International Conference on*, pp. 188-193. IEEE, 2014.