International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 07 | July -2017 www.irjet.net p-ISSN: 2395-0072

Hybrid of Ant Colony Optimization and Gravitational Emulation Based Load Balancing Strategy in Cloud Computing

Jyoti Yadav[1], Dr. Sanjay Tyagi[2]

[1]M.Tech. Scholar, Department of Computer Science & Applications, Kurukshetra University, Haryana, India [2]Assistant Professor, Department of Computer Science & Applications, Kurukshetra University, Haryana, India

---

Abstract- *The distributed architecture of cloud computing set up the resources distributively for delivering the services to cloud consumers. In this paper, a novel hybrid ACO and gravitation emulation based strategy considering load balancing has been implemented for solving the load balancing issue in cloud environment efficiently. Moreover, this hybrid ACO-GELS algorithm uses the physics concept of gravitational attraction between objects. GELS algorithm is powerful for local search in searching space. CloudSim has been used as a simulation tool for proposed hybrid load balancing strategy. The proposed ACO-GELS algorithm has been compared with the existing GA-GELS algorithm. It has been compared on the basis of three important factors of load balancing: resource utilization, makespan and load balancing level.*

Key Words: Ant Colony Optimization, Cloud Computing, CloudSim, GELS, Gravitational Emulation, Load balancing.

I. INTRODUCTION

Internet technologies are growing rapidly and cloud computing has become a hot topic to meet the user's needs. Cloud computing is a distributed computing mechanism. The cloud based system provides services on demand from anywhere in the world.

Cloud computing has a splendid future, even though there are some important problems that need to be solved for minimizing response time, minimizing cost, maximizing throughput, etc.

Load Balancing is one of the main issues that need to be considered in cloud computing as it plays a vital role in cloud computing. As the name suggests, load balancing means to distribute the workload evenly among the virtual machines (VMs) [1].

For load balancing, some points need to be kept in mind: communication between the nodes, expected load, stability of different system, selection of nodes, arrangement of system and nature of work to be transferred [2]. The main objective of load balancing is to distribute the local workload evenly to ensure that no VM is overloaded or idle. Load balancing reduces the makespan and response time and increases the utilization of resources. The basic scenario of load balancing has been represented in fig-1:

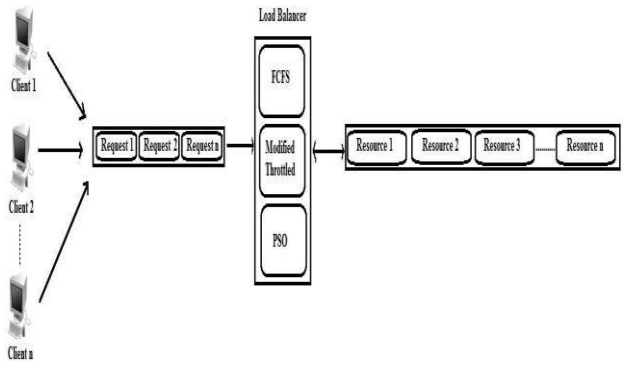

Fig-1: Load Balancing Scenario

There are many static and dynamic load balancing algorithms like FCFS, Round-Robin, ACO, GA, PSO, etc.

In this paper, for load balancing in VMs, a hybrid of Gravitational Emulation Local Search (GELS) and ACO has been proposed. ACO is an optimization algorithm which uses basic foraging behavior of ants and uses the pheromone values & GELS is powerful for local search and weak for global search. It is based on the basic concept of gravitational attraction in physics.

For simulation and analysis of the proposed hybrid ACO-GELS algorithm, CloudSim simulation tool has been used.
The rest of the paper has been organized as follows:
Section II- Related Work
Section III- Load Balancing of VMs using ACO & GELS
Section IV- Proposed ACO-GELS Algorithm
Section V- Simulation Results and Analysis
Section VI- Conclusion

II. RELATED WORK

A cloud task scheduling based ACO approach was presented by Medhat A. Tawfeek et. al. [3] for allocation of

incoming tasks to the VM's in an efficient manner to minimize the makespan. Therefore, the available resources could be utilized optimally to achieve a high user satisfaction and minimize the resource consumption. This approach was performed using CloudSim simulation tool. The simulation results show that ACO algorithm is better than Round-Robin and FCFS algorithms. Also, the best values of parameters have been experimentally obtained for ACO algorithm.

An advanced reservation approach was developed to provide users with the guaranteed quality of service (QoS) [4]. Advanced reservation is a kind of process which allows the resources to be allocated on the basis of increase in number of accepted users request and the need of QoS in Grid system. In this paper, GELS algorithm, a heuristic method was used for scheduling and advanced reservation of resources. The algorithm proposed was named as GELSAR and compared with GA. The experiments showed that GELSAR has better execution time in comparison with GA which reduced to 50 percent. Also, it increased the jobs reservation.

Hybrid algorithm of GA and GELS was proposed for load balancing of VMs in cloud [5]. GEL is a local search technique which uses the concept of gravitational attraction. Therefore, it is powerful for local search and feeble for global searches, while GA is inspired by natural evaluation method for existence. A new set of strings was generated in each generation. The crossover and mutation operations were used in GA. GA is strong for global searches. Cloud Analyst simulation tool has been used for the implementation of this proposed algorithm. The proposed GA-GELS tried to reduce the makespan and improved the response time of VMs. Also, this algorithm was compared with existing techniques like ACO, GA, FCFS and guaranteed the QoS requirement of user's request.

Particle Swarm Optimization (PSO) and Gravitational Emulation based Hybrid approach has been presented by Rakshanda et. al. [6], which was named as PSO-GEL algorithm. The mechanism of PSO is inspired by swarm of insects, bird flocks and fish schooling. Each particle represents a feasible solution and have a position and a velocity. In this paper, the results showed that PSO-GEL was better than GA-GEL algorithm and proposed PSO-GEL algorithm has less response time than GA-GEL.

III. LOAD BALANCING OF VMs USING ACO & GELS

3.1 Ant Colony Optimization Algorithm (ACO)

ACO is a random optimization technique which is inspired from the food searching behavior of real ants. When ants move from their nest to food, they deposit the chemical substance called pheromone on their way [7]. An ant can follow the trail of other ants by sensing the pheromone on the ground. This pheromone concentration is used to find the best path to the source. The more deposition of pheromone leads to a positive feedback effect. And then the pheromone value is updated and more ants follow that path.

Two types of pheromones are used in ACO:
Foraging Pheromone (FP) is used for movement towards overloaded nodes, and
Trailing Pheromone (TP) is used for tracing underloaded nodes [8].

Disadvantages of this algorithm are overhead, stagnation phenomenon and this algorithm converges to local optimal solution.
Pseudocode of ACO can be represented as:
Initialize parameters;
while(termination criterion not satisfied)
  Construct Solution;
  Apply Local Search;
  Global Pheromone Update;
  Self-Adaptive Mechanism;
end
return best solution;

3.2 Gravitational Emulation Local Search Algorithm (GELS)

GEL algorithm was given by Voundaris and Tesong in 1995, for searching in a search space. In 2004, more powerful algorithm was proposed by Barry Webstar called GELS (Gravitational Emulation Local Search Algorithm) [4].

This algorithm is inspired from the basic concept of gravitational attraction. The objects are pulled towards each other due to gravity. Also, more closer the two objects, stronger the gravitational force between them. This algorithm introduced the concept of randomization along the two primary parameters: velocity and gravity.
Newton's formula of gravitational force is:

$$F = \frac{G\, m1.m2}{R^2} \qquad (1)$$

Here, m1 & m2 are mass of first & second object, G is gravitational constant which has value 6.672 and R is distance between two objects.

GELS algorithm includes a pointer that moves through the search space and GELS allows movement by two methods.

In first method, a candidate solution is selected from the neighborhood of the current response. In second method, movements are allowed outside the neighborhood of current response [5].

In GELS, formula in equation 1 is modified and the gravitational force has been calculated as follows:

$$F = \frac{G\,(CU-CA)}{R^2} \qquad (2)$$

The mass is replaced by difference of Current response and Candidate response.
Here, CU= Current response
CA= Candidate response
R= Constant or may change on each iteration.
G= Gravitational Constant (6.672)
Makespan can be defined as the maximum completion time, i.e.
Makespan = max(Finish_time [i,j])
Finish_time[i,j] indicates the time at which task i ends on VM j.
For VM, load balance can be obtained from eq 3 as
CPU_LB= makespan / avgET (3)
Here, avgET is the average execution time for all user tasks.
So, the fitness function can be calculated as
Fit1=1/ CPU_LB (4)

IV. PROPOSED ACO-GELS ALGORITHM

Proposed ACO-GELS algorithm uses the best quality of ACO and GELS algorithm. The basic steps of algorithm are given below:

1) Initial population generation: GELS algorithm is used to select the total population and high primary velocity is taken into concern.

2) Force Calculation: Gravitational force of current response and candidate response is calculated using equation 2 and then it is added to the velocity of that dimension of the candidate response.

3) Pheromone Updation: Global pheromone updation is done using acceleration.
Acceleration[i][j] = Force[i][j] / mass[i]

4) Termination condition: Either primary velocity equals to zero or maximum number of iteration is reached.

Pseudocode of proposed ACO-GELS algorithm:
begin
-**Initialize** the population of VM randomly
-**Evaluate** the fitness for each ant
-Assign a predefined starting soln as current soln
-Assign an initial velocity randomly in the dimension
-Calculate an initial vector velocity sum
-Best_soln = current soln
-**while** (velocity_sum != 0 OR i <= max_ITER)
Do
velocity_sum=0
Generate candidate response
Calculate Mass and Acceleration
Calculate the difference in GF between current soln and candidate soln using eq 2
if (fit(candidate soln) > fit(current soln))
then
Best_soln = candidate soln
Update Velocity (v,force)
end
-Global pheromone update using Acceleration
-return best solution found
-end

V. SIMULATION RESULTS AND ANALYSIS

CloudSim has been used for simulation of proposed algorithm. CloudSim is an extensible simulation framework that allows experimentation and seamless modeling of emerging cloud computing application services. The main advantages of using CloudSim are:
(i) Time effectiveness
(ii) Flexibility and applicability

The proposed ACO-GELS algorithm has been simulated by using three parameters: makespan, resource utilization and load balancing level.

Makespan- Makespan can be defined as the time difference between the start and finish time of tasks or it is the completion time of tasks. Also the makespan must be low for better performance [9].

Resource Utilization- Resource utilization means to serve more users during specific operation time [10]. The more the utilization of resources means more the balancing of load i.e. it must be large.

Load Balancing Level- It includes the balancing level of load on the VMs or how the load is distributed evenly on the machines [11]. The load is distributed efficiently, if this parameter is high.

The experimental results of ACO-GELS algorithm have been compared with the existing GA-GELS algorithm [5] on the basis of above three parameters and the results are

shown below in tabular form as well as graphically with different number of tasks.

Here, the numbers of virtual machine are 5 for all the experiments and numbers of tasks have been varied like 50, 100 and 200.

Table-1: Makespan Analysis (in ms)

Number of Tasks	GA-GELS	ACO-GELS
50	94.1	38.1
100	104.1	41.43
200	307.1	214.1

Table-1 shows the makespan metric values for GA-GELS and proposed ACO-GELS algorithms with three different scenarios. First simulation has been done with 50 tasks on 5 VMs, then with 100 and 200 tasks on 5 VM's.

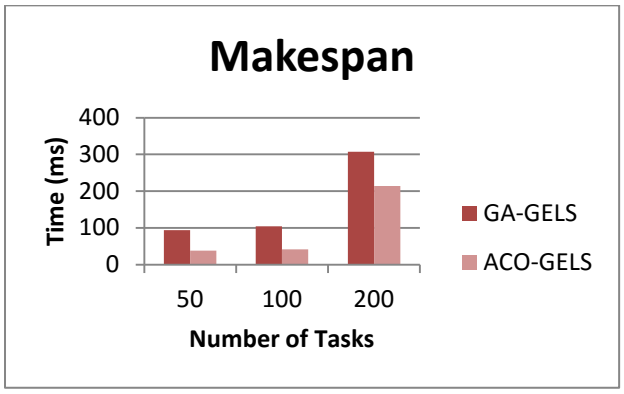

Fig-2: Performance analysis of ACO-GELS with existing GA-GELS using five VMs and 50, 100 & 200 tasks

Fig-2 represents the simulation results of comparison of makespan between ACO-GELS and GA-GELS in graphical form. It is clear from the figure that proposed ACO-GELS algorithm performs better than GA-GELS and reduces the makespan even with the large number of tasks.

Table-2: Average Resource Utilization Analysis

Number of Tasks	GA-GELS	ACO-GELS
50	44	93.96
100	46.79	96.21
200	69.03	99.02

The average resource utilization rate for GA-GELS and ACO-GELS has been shown in Table-2. The readings are taken for different number of tasks i.e. 50, 100 and 200 with five VMs.

Proposed ACO-GELS algorithm has been compared with GA-GELS based on average RU metric in fig-3. It shows that ACO-GELS is better than GA-GELS for this factor in load balancing of cloud system.

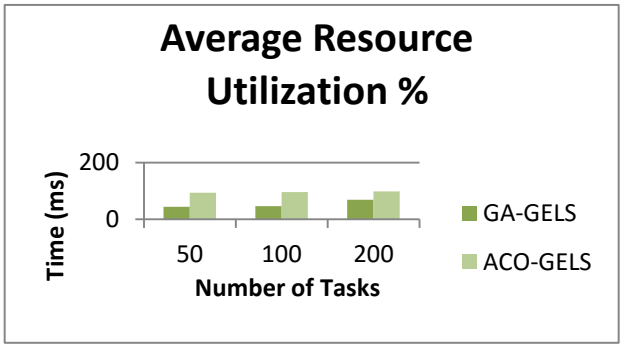

Fig-3: Performance analysis of ACO-GELS with existing GA-GELS using five VMs and 50, 100 & 200 tasks

Table-3: Load Balancing Level Analysis

Number of Tasks	GA-GELS	ACO-GELS
50	33.84	95.19
100	42.08	96.83
200	74.35	99.01

Table-3 represents the load balancing level for GA-GELS and proposed ACO-GELS for 50, 100 and 200 tasks using five VMs.

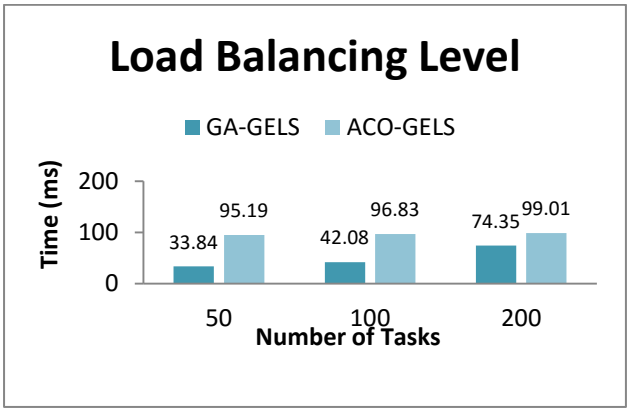

Fig-4: Performance analysis of existing GA-GELS and proposed ACO-GELS using five VMs for 50, 100 and 200 tasks

Fig-4 represents the comparison of ACO-GELS and GA-GELS graphically for load balancing level (LBL) metric. The

results showed that proposed algorithm increases the LBL as compared to the GA-GELS algorithm.

VI. CONCLUSION

This paper presents a novel load balancing algorithm based on hybrid of ACO algorithm and GELS algorithm. The proposed hybrid algorithm is named as ACO-GELS. ACO-GELS algorithm provides the better result to solve cloud computing load balancing problem to minimize the makespan. GELS algorithm provides better result for local searching and ACO is an optimization technique. Experimental results conclude that ACO-GELS provides the better results than the GA-GELS algorithm. It is also better than FCFS, GA, SHC and ACO algorithms, since GA-GELS results have shown it better than all these algorithms [5]. ACO-GELS reduced the makespan, increased the resource utilization and also increased the load balancing level as compared with GA-GELS.

Though priority of tasks and fault tolerance issues has not been considered here, this may be considered in further research. Researcher can include the priority issue and fault tolerance issue for future research work.

REFERENCES

[1] N. Pasha, D. A. Agarwal and D. R. Rastogi, "Round Robin Approach for VM Load Balancing Algorithm in Cloud Computing Environment," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 4, no. 5, pp. 34-39, May 2014.

[2] B. L. Webster, "Solving Combinatorial Optimization Problems Using a New Algorithm Based on Gravitational Attraction," Florida Institute of Technology, Melbourne, USA, 2004.

[3] M. A. Tawfeek, A. El-Sisi, A. E. Keshk and F. A. Torkey, "Cloud Task Scheduling Based on Ant Colony Optimization," in *8th International Conference on Computer Engineering & Systems (ICCES),IEEE*, 2013.

[4] B. Barzegar, A. M. Rahmani, K. Z. far and A. Divsalar, "Gravitational Emulation Local Search Algorithm for Advanced Reservation and Scheduling in Grid Computing Systems," in *Fourth International Conference on Computer Sciences and Convergence Information Technology, IEEE*, 2009.

[5] S. Dam, G. Mandal, K. Dasgupta and P. Dutta, "Genetic Algorithm and Gravitational Emulation Based Hybrid Load Balancing Strategy In Cloud Computing," in *Third International Conference on Computer, Communication, Control and Information Technology (C3IT), IEEE*, 2015.

[6] Rakshanda, D. K. Garg and Vinod, "Particle Swarm Optimization and Gravitational Emulation Based Hybrid Load Balancing Strategy in Cloud Computing," *International Journal for Scientific Research & Development,* vol. 4, no. 4, pp. 983-986, 2016.

[7] Y. Zhou and X. Huang, "Scheduling Workflow in Cloud Computing Based on Ant Colony Optimization," in *Sixth International Conference on Business Intelligence and Financial Engineering*, 2013.

[8] S. Kumar, D. S. Rana and S. C. Dimri, "Fault Tolerance and Load Balancing Algorithm in Cloud Computing," *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 4, no. 7, pp. 92-96, 2015.

[9] N. Sharma, S. Tyagi and S. Atri, "A Comparative Analysis of Min-Min and Max-Min Algorithms based on the Makespan Parameter," *International Journal of Advanced Research in Computer Science,* vol. 8, no. 3, pp. 1038-1041, 2017.

[10] A. Al-Shaikh, H. Khattab, A. Sharieh and A. Sleit, "Resource Utilization in Cloud Computing as an Optimization Problem," *International Journal of Advanced Computer Science and Applications (IJACSA),* vol. 7, no. 6, pp. 336-342, 2016.

[11] P. A. Pattanaik, S. Roy and P. K. Pattnaik, "Performance Study of Some Dynamic Load Balancing Algorithms in Cloud Computing Environment," in *2nd International Conference on Signal Processing and Integrated Networks (SPIN)*, 2015.