Particle Swarm Optimization and Gravitational Emulation Based Hybrid Load Balancing Strategy in Cloud Computing

Rakshanda[1] Dr. Kanwal[2] Garg Vinod[3]
[1]Research Scholar [2]Assistant Professor [3]Professor
[1,2,3]Department of Computer Science & Applications
[1,2,3]Kurukshetra University, Kurukshetra, Haryana, India

Abstract— Cloud computing is a rising technology adopted by both industry and the academic world, providing an elastic way to store and recover the data files. Today's data is increasing rapidly so load balancing is the essential parameter required for efficient operation of various components in a cloud computing to minimize system load and provide the resources at a rapid rate. The Genetic Algorithm (GA) along with its many versions has been popular mainly because of their intuitiveness, ease of implementation. PSO is a current heuristic search method whose mechanics is inspired by the swarming or common behavior of natural populations. GEL algorithm depends on gravitational attraction. The present research paper conducted an experiment to ensure that a hybrid combination of PSO-GEL has the better result than GA-GEL.

Key words: Cloud Computing; Cloud Sim; Load Balancing; Genetic Algorithm; Particle Swarm Optimization

I. Introduction

Internet technologies are fast growing and are being used extensively, with that Cloud Computing became a hot topic of business and academic world as an emerging new computing mechanism. It is supposed to grant computing as the utility to meet the daily needs of the general community [1] [2]. Its infrastructure is used by businesses and users to right to use application services from anywhere in the world on demand. Thus, it represents as a new standard for the dynamic provisioning of computing services, normally supported by state-of-the-art data centers containing ensembles of networked Virtual Machines [3]. It is the distributing computing mechanism that utilizes the high speed of the internet to move job from personal PC to the remote computer clusters (big data centers owned by the cloud service providers) for data processing. Even if there is a splendid future of Cloud Computing, many essential Problems still need to be solved for the realization of cloud computing. Load balancing is also the big problem; it plays a very key role in the grasp of Cloud Computing. Load balancing in the cloud computing is to distribute the local workload uniformly to the whole cloud. In fact, it has become obligatory for cloud computing. It is used by cloud service providers (CSP) in its own cloud computing stage to provide a high proficient solution for the user. Also, an inter CSP load balancing mechanism is needed to put up a low cost and the infinite resource pool intended for the user. This Load balancing in cloud computing provides an organization with the ability to dispense application requests across any number of application deployments situated in data centers and through cloud computing providers. For developing strategy for load balancing the main points to be well thought-out are expected of load, comparison of load, stability of different system, the arrangement of system, communication between the nodes, nature of work to be transferred, selecting of nodes [8]. This load used can be in terms of CPU load, quantity of memory used, holdup or Network load. Load balancing is a process of reassigning the total load to the individual nodes of the computing environment, this facilitates the network and resources and further improving the system performance. The important parts of this process judgment and comparison of the stability, load and performance of the system, inter nodes traffic optimization. To construct a load balancing mechanism many techniques and strategies are used. Hence load balancing serves two important needs, mainly to prop up the availability of Cloud resources and secondarily to promote performance. Some presented scheduling techniques like the equally spread current execution algorithm is exist in literature [11]. An intelligent method for load balancing has been proposed by kaur [12]. Mala Kalra proposed load balancing algorithm in [13] called CLB (central load balancer). It is the advanced version of Throttled.

II. Load Balancing Using Gravitational Emulation Algorithm and Particle Swarm Optimization in Cloud Computing

A. Genetic Algorithm

Genetic Algorithms (GAS) are a stochastic global search method to facilitate the process of natural progression. It is one of the methods used for optimization. In a genetic algorithm, a population of string called chromosomes or which encode candidate solutions on the way to an optimization problem, evolve in the direction of the better solutions. That Solutions are represented in binary as strings of 0s and 1s, but other encodings are also capable. The growth generally starts from a population of randomly generated individuals and happens in generation. In each group, the strength of every entity in the population is evaluated, multiple individuals be stochastically selected from the current population (based happen their fitness), and modified recombined the possibly randomly mutated to form a new population. The new population is then old in the next iteration of the algorithm. If the algorithm has completed due to a maximum number of generations, a satisfactory output may or may not have been reached [1]. Holland legitimately introduced this method in the United States in the 1970 at the University of Michigan. The genetic algorithm starts with no knowledge of the accurate solution and depends entirely on responses commencing its environment and progress operator such as, selection crossover and mutation to reach your destination at the best solution [6]. By starting at several independent points and searching in parallel, the algorithm avoids local minima and converging toward sub best solutions.

A distinctive genetic algorithm requires:

- A genetic illustration of the solution area
- A fitness function to evaluate the solution area.

1) Implementation Algorithm: The genetic algorithm uses the chromosomes fitness value to create a new population consisting of the fittest members. The flow chart of the GA is given in Figure. 1.

B. Particle Swarm Optimization

The PSO was first designed to simulate birds seeking food which is defined as a “cornfield vector.” The bird would find food through social cooperation with other birds around it (within its region). It was then long drawn out to multidimensional search. Particle swarm optimization (PSO) is a computational method that optimizes a crisis by iteratively trying to improve a candidate solution with look upon toward a given measure of quality. Such methods are commonly known as metaheuristics as they make little or no assumptions about the problem being optimized and can search very huge spaces of candidate solutions. PSO can therefore also be used on optimization problems that are in some measure uneven, thunderous, change over time, etc. [2]. PSO optimizes a complexity by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae. The movements of the particles are guided by the best found position in the search-space which is updated as better positions are originate by the particles. The PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved transversely in the search-space according to a few simple formulae. The schedule of the particles is guided by their own best known position in the search space as well as the entire swarm's best identified location. When swarm enhanced positions are being revealed. Then it will guide the activities of the swarm. The process is repeated and satisfactory solution will be discovered. In the basic particle swarm optimization algorithm, particle swarm consists of “n” particles, and the position of each particle stands for the potential solution in D-dimensional space. The particles change its condition according to the following three principles: (1) to keep its inertia (2) to change the condition according to its most optimist position (3) to change the condition according to the swarm’s most optimist position.

The pseudo code of the procedure is as follows

```
For the each particle
    Initialize particle
END
Do
    For the each particle
        Calculate the fitness value
        If the fitness value is enhanced than the best fitness
value (current Best) in previous record
            Set current value as the new current Best
    End
    Choose the particle with the best fitness value of all the
particles as the global Best
    For each particle
        Calculate the particle velocity according equation
        Then Update particle position
    End
While maximum iterations or minimum error criteria is not
examined.
```

C. Gravitational Emulation Local Search (Gels)

Voudouris and his colleagues [7] for the first time in 1995 were suggested that the GELS algorithm for searching in a searching space and NP-hard solution. In 2004, Vaster [8] presented it as a strong algorithm and called it as GELS algorithm. This algorithm based on gravitational attraction and by using of this process is imitate for searching within searching space. Obtained neighbors in every one’s neighbor’s group called measurement. For each measurement, was defined a primary velocity and each measurement has much primary velocity and more evident response for problem. This algorithm introduced randomization conception alongside with two of four primary parameters i.e. velocity and gravity in physics through switch over in terms of groups by using random number in the existing local search algorithm. GELS take as its basis the natural principles of gravitational attraction. Gravity works in natural world to drive the objects to be pulled towards each one other. The more massive the object, the more gravitational pull it exert on other objects. Also, the closer two objects are to each other, the stronger the gravitational forces are between them.

1) *Fitness function*

With respect to the basic purpose of load balancing, minimizing the Response time. Our proposed algorithm PSO-GEL is static, we assume expected completion time of user task are indomitable and set in the Expected Time of Completion table (ETC) where ETC [i,j] represents each task on each VM. Also, the ready time (Ready[j]) means when a VM is ready to take the next task. Make span indicates maximum completion time**.**

Make span = Maximum (Completion Time [i, j]) (3)

$\{1 \leq i \leq N,\ 1 \leq j \leq M\}$

Completion Time [i, j] is the time at which task i ends on VM j and is calculated like to Equation (4)

Completion Time [i, j] = Ready[j]+ ETC[i,j] (4)

2) *Initial population generation*

We use GEL algorithm to select the total population here high primary velocity taken into concern.

3) Crossover

Our proposed algorithm uses a two- point crossover operator. Which are selected at random from the earlier phase?

4) *Mutation*

Now a point on each chromosome from preceding step is selected randomly the bits of the chromosomes, are toggled from 1 to 0 or vice -versa.

5) Force Calculation

After crossover and mutation the gravitational force of the in progress chromosome and candidate chromosome are deliberate using Equation 1, after the gravitational strength is added just before the velocity of that dimension.

6) Terminating Conditions

The algorithm terminates when the primary velocity is equal to zero for all dimensions or the maximum number of iterations has been reached.

III. Proposed Algorithm

1) *Create a M*N Dimension swarm with p particle*
2) *Initialize the M*n Dimension*
3) *Calculate the each participant fitness function.*
4) *initialize Velocity _vector (V)*
5) *{*

```
6)  Calculate the P best and G best
7)  // (P best of each particle)
8)  // (G best of population)
9)  Update The G best of Swarm and P best of Particle
10) }
11) Do while
12) Current- solution = G best
13) Best- solution = current- solution
14) Repeat until stop criterion is mee or satisfied
15) {
16) candidate _ solution = make neighbour (direction)
17) Calculate the force
18) Calculate mass and acceleration
19) If ( fitness Of (candidate solution) > fitness (current
    solution)
20) Then Best- solution = candidate- solution
21) Update velocity _ velocity (v, force)
22) Direction = selection Direction (v)
23) }
24) If repeat
25) Return best solution
26) End
```

IV. Implementation Experimental Results

In present paper the proposed algorithm is simulated in clouSim-3.0.3[14]. Simulation conclusions about comparison within suggested algorithm (GA-GEL) and PSO- GELS algorithm has been shown in Figures 1to Figure 5. In Figure 1 shown a diagram which was scheduled the number of tasks 50 and VM's 5. In Figure 2 shown a graph which was listed the number of tasks having 100 and VM's 5. In Figure 3 makes a diagram which was examined the number of tasks is 150 and VM's 5. In Figure 4 exposed a schedule that is listed the number of tasks within 200 and VM's 5. In Figure 5 researcher examined the proposed PSO-GEL algorithm produced less Response time than GA-GEL. Also examined, as the number of tasks increases the response time also increased.

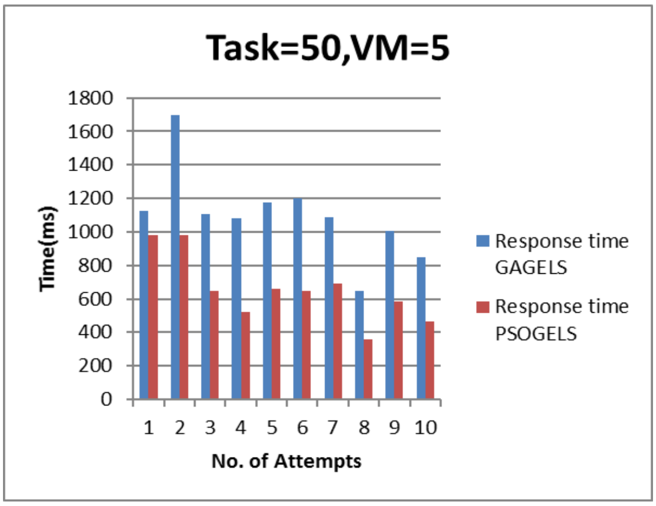

Fig. 1: Performance analysis of proposed PSO-GEL and GA-GEL with 50 tasks and 5VM

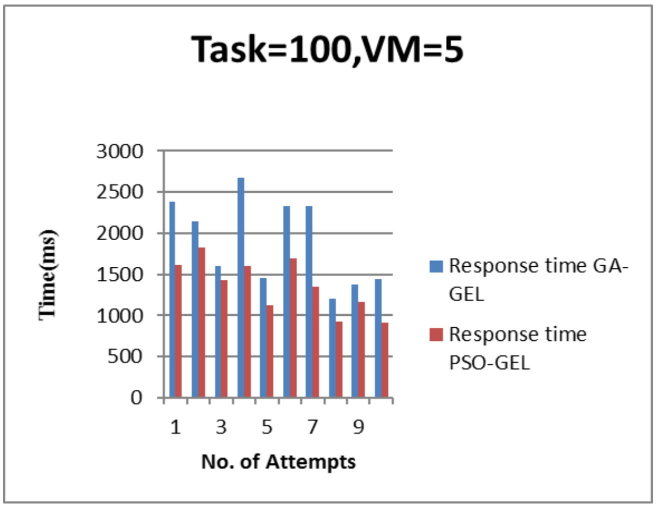

Fig. 2: Performance analysis of proposed PSO-GEL and GA-GEL with 100 tasks and 5VM

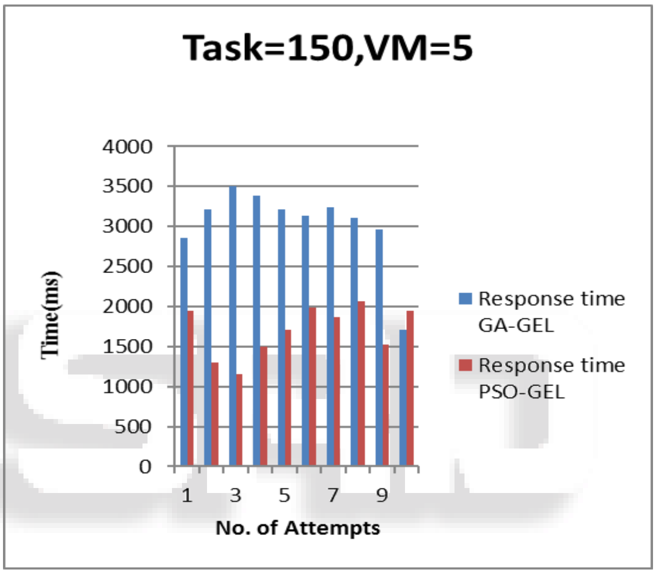

Fig. 3: Performance analysis of proposed PSO-GEL and GA-GEL, with 150 tasks and 5VM

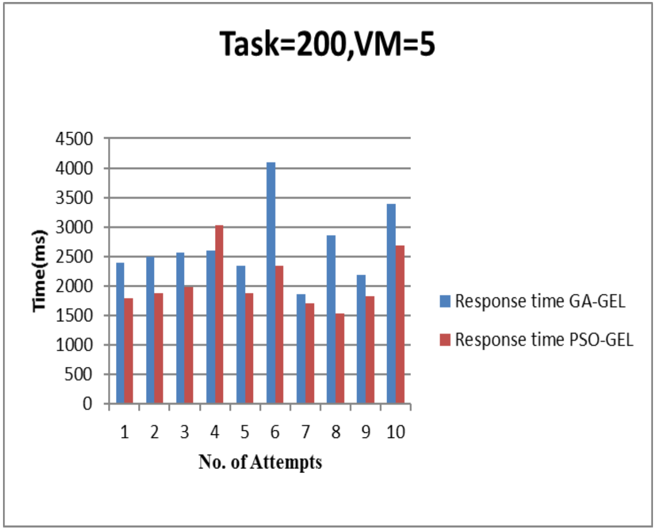

Fig. 4: Performance analysis of proposed PSO-GEL and GA-GEL, with 200 tasks and 5VM

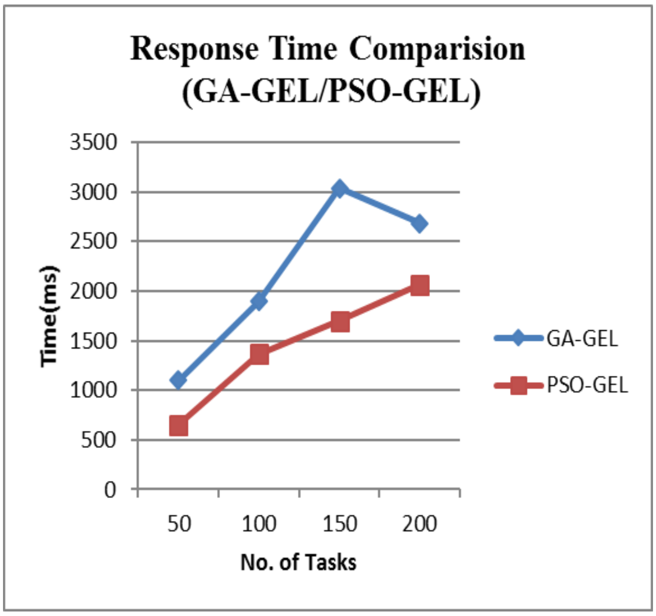

Fig. 5: Comparison analysis of proposed PSO-GEL and GA-GEL

V. CONCLUSION

This paper presents a novel task scheduling method based on hybrid PSO-GELS algorithm to solve cloud computing task scheduling problem to minimize the Response time. Each particle represents a feasible solution. The position vector is transformed from the continuous values to the discrete values based on round off real values. The hybrid PSO-GEL performs better than the local search. Because GELS algorithm rather than the other local searching algorithms such as GA, PSO, GA-GEL searching space problems and find better solutions. The performance of the proposed method is compared with existing method. From the simulated experiment, the result of GEL-PSO algorithm is better than GA-GEL algorithms.

REFERENCES

[1] Goldberg and D.E, “Genetic Algorithms in Search, Optimization and Machine Learning”, Addison-Wesley, 1989.

[2] Kennedy and J., “Particle Swarm Optimization,” Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia 1995, pp. 1942-1945.

[3] Kennedy, J. and Eberhart, “Swarm Intelligence”, Academic Press, 1st ed., San Diego, CA, 2001.

[4] Chen Yon gang, Yang and Fengjie, Sun Jigui, “A new Particle swam optimization Algorithm” Journal of Jilin University, pp: 181-183, In Chinese, 2006

[5] R. Pole. “Analysis of the publications on the applications of particle swarm optimization”, Journal of Artificial Evolution and Applications, Article volume ID685175, pp: 10, 2008.

[6] lan griffin and Jennifer bruton "On-Line Pid controller tuning using genetic algorithm" URL; www.eeng.dcu.ie/-brutonjiReportsIlGriffinMEng 03.pdf

[7] Voudouris, Chris and Edward Tsang, “Guided Local Search”, Technical Report CSM-247, Department of Computer Science, University of Essex, UK, August 1995.

[8] Barry Lynn Webster, “Solving Combinatorial Optimization Problems Using a New Algorithm Based on Gravitational Attraction”, Ph.D. Thesis, Florida Institute of Technology Melbourne, FL, USA, May 2004.

[9] C. Voudouris, E. Tsang, “Guided Local Search”, European Journal of Operational Research, Technical Report CSM-247, UK, pp: 1-18, August 1995.

[10] B.Webster, “Solving Combinatorial Optimization Problems Using a New Algorithm Based on Gravitational Attraction”, Ph.D. thesis, Melbourne, Florida Institute of Technology, , pp:1-250, May 2004.

[11] M. Aruna and D. Bhanu, “A Survey on Load Balancing Algorithms in Cloud Environment”, International Journal of Computer Application, Volume 82 – No 16, pp: 39-43, November 2013.

[12] Amandeep Kaur Sidhu and Supriya Kinger, “Analysis of Load Balancing Techniques in Cloud Computing”, International Journal of Computers & Technology, Vol. - 4 No. 2, pp: 737-741, March-April, 2013.

[13] Gulshan Soni and Mala Kalra, “A Novel Approach for Load Balancing in Cloud Data Center”, International Advance Computing Conference (IACC) IEEE, pp: 807-812, 2014.

[14] Rajkumar Buyya, “Cloud Sim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms”, Software Practice and Experience, Wiley publishers, 2010.