Proposed model for DE-GA based Load Balancing in Cloud Computing

[1]Pooja Mangla, [2]Dr. Sandip Kr. Goyal
[1]Ph.D Scholar, [2]Head of Department
[1]CSE Department,
[1]Maharishi Markandeshwar University, Mullana, Ambala, Haryana, India

***Abstract* - Cloud Computing is one of the major aspect in the field of Computer Science. Load balancing is one of the major issues in the Cloud Computing. It is the proficiency that allows workload to be disseminated across various number of available resources, to make effective resource employment. It also enhances response time by dealing in a situation in which some of the nodes are extremely loaded while some others are under loaded. The intention of load balancing is to optimize usage and throughput while cutting down the reaction time. In this model, we are going to plan a model for Differential Evolution based Genetic Load Balancing in Cloud environment.**

***Index Terms* - Cloud Computing, Load Balancing, Differential Evolution, GA, PSO**

I. INTRODUCTION

Distributed network computing environments have become a cost effective and popular choice to achieve high performance and to solve large scale computation problems. Unlike past supercomputers a cloud or cluster or grid system can be used as multipurpose computing platform to run diverse high performance parallel applications. Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a utility (like the electricity grid) over a network (typically the Internet). The term Cloud refers to a Network or Internet. Cloud can provide services over network, i.e., on public networks or on private networks, i.e., WAN, LAN or VPN. Applications such as e-mail, web conferencing, customer relationship management (CRM), all run in cloud.

A 'cloud' is an elastic execution environment of resources involving multiple stakeholders and providing a metered service at multiple granularities for a specified level of quality (of service). Cloud Computing refers to manipulating, configuring, and accessing the applications online. It offers online data storage, infrastructure and application. Cloud Computing is high utility software having the ability to change the IT software industry and making the software even more attractive. It has also changed the way IT companies used to buy and design hardware. The elasticity of resources without paying a premium for large scale is unprecedented in the history of IT industry. The increase in web traffic and different services are increasing day by day making load balancing a big research topic.

II. LOAD BALANCING

Load balancing [1] is a generic term used for distributing a larger processing load to smaller processing nodes for enhancing the overall performance of system. Load Balancing is a process of reassigning the total load to the individual nodes of the collective system to make more resource utilization effective and to improve the response time of the job, simultaneously removing a condition in which some of the nodes are overloaded while some others are under loaded.

An ideal load balancing algorithm should avoid overloading or under loading of any specific node. But, in case of a cloud computing environment the selection of load balancing algorithm is not easy because it involves additional constraints like security, reliability, throughput etc. So, the main goal of a load balancing algorithm in a cloud computing environment is to improve the response time of job by distributing the total load of system. The algorithm must also ensure that it is not overloading any specific node.

Load Balancing is done with the help of load balancers where each incoming request is redirected and is transparent to client who makes the request. Based on predetermined parameters, such as availability or current load, the load balancer uses various scheduling algorithm to determine which server should handle and forwards the request on to the selected server [5]. To make the final determination, the load balancer retrieves information about the candidate server's health and current workload in order to verify its ability to respond to that request. Load Balancing helps in [2]:

a. Improving the performance substantially.
b. Having a Reverse up plan in case the system fails even partially.
c. Maintenance of system stability.
d. Accommodation of future modification.
e. Efficient load distribution.
f. Cost effectiveness.

III. PROPOSED MODEL:

a. **Differential Evolution**: One extremely powerful algorithm from evolutionary computation due to its excellent convergence characteristics and few control parameters is Differential Evolution (DE). Differential Evolution (DE) is an efficient optimization approach. Storn and Prince in 1995 is the inventor of Differential evolution algorithm. Differential Evolution is a simple yet very efficient optimization approach in solving a variety of task scheduling problems with many real-world applications [6]. Differential evolution combined with evolution strategies (ES"s) and evolutionary programming (EP) can together be categorized into a class of population –based, with Derivative free methods known as Evolutionary algorithm. All these approaches mimic Darwinian evolution and evolve a population of individuals from one generation to another generation by analogous evolutionary operational factors such as mutation, crossover and selection.

```
Pseudo Code  for Differential Evolution:
      Step 1: Random initialization of agents in the parent Population.
      Step 2: If stopping criteria is not met than do
            2.1 For each agent from Population repeat until condition is not met
            a) Create Candidate from parent.
            b) Evaluate the Candidate fitness values.
            c) If Candidate is pareto optimal than Candidate replaces the
               Parent.
            d) If Parent is pareto optimal than the Candidate is discarded.
               Otherwise, Candidate is added into the Population.
```

Figure 1 Pseudocode for DE algorithm

The basic pseudocode for the DE algorithm is shown above. For each iteration, new individuals are generated in the population matrix through operations performed among individuals of the matrix (mutation - F), with old solutions replaced (crossover - CR) only when the Fitness value of the objective function is better than the current one. A population matrix with optimized individuals is obtained as output of the algorithm. The best of these individuals are selected as solution close to optimal for the objective function of the model.

b. **Genetic Algorithm**: GA was first introduced by Holland in 1975 and represents a population based optimization method based on a metaphor of the evolution process observed in nature. In GA, each chromosome (individual in the population) represents a possible solution to a problem and is composed of a string of genes. The initial population is taken randomly to serve as the starting point for the algorithm. A fitness function is defined to check the suitability of the chromosome for the environment. On the basis of fitness value, chromosomes are selected and crossover and mutation operations are performed on them to produce offsprings for the new population. The fitness function evaluates the quality of each offspring. The process is repeated until sufficient offspring are created [3, 4]. Pseudo code of GA algorithm for optimization of scheduling problem in cloud is shown in Fig. 4.

```
Procedure GA
   1. Initialization: Generate initial population P consisting of chromosomes.
   2. Fitness: Calculate the fitness value of each chromosome using fitness
      function.
   3. Selection: Select the chromosomes for producing next generation using
      selection operator.
   4. Crossover: Perform the crossover operation on the pair of chromosomes
      obtained in step 3.
   5. Mutation: Perform the mutation operation on the chromosomes.
   6. Fitness: Calculate the fitness value of these newly generated chromosomes
      known as offsprings.
   7. Replacement: Update the population P by replacing bad solutions with better
      chromosomes from offsprings.
   8. Repeat steps 3 to 7 until stopping condition is met. Stopping condition may
      be the maximum number of iterations or no change in fitness value of
      chromosomes for consecutive iterations.
   9. Output best chromosome as the final solution.
End Procedure
```

Figure 2 Pseudo code of GA.

c. **DE-GA MODEL:** In the proposed model, hybrid DE and GA are used. In this, Differential Evolution starts upto the point where the trial vector is generated. If that vector satisfies the equation, then it is included in the population otherwise algorithm enters the Genetic algorithm phase and generates a new candidate solution. The pseudo-code for efficient load balancing using DE & GA is as below:

1. Sampling the search space at multiple, randomly chosen initial points i.e. a population of individual vectors.
2. Differential evolution is a nature derivative-free continuous function optimizer, it encodes the parameters as a floating-point numbers and manipulates them with simple arithmetic operations For this differential evolution it mutates a (parent) vector in the population with a scaled difference of the other randomly selected individual vectors.
3. The resultant mutation vector is a crossed over with corresponding parent vector to generate a trial or a offspring vector.
4. Then, finally it takes a decision in a one-to-one selection process of each pair of offspring and parent vectors.
5. If the best population is generated, it is taken into consideration; otherwise the population is generated by using Genetic algorithm.
6. In Genetic algorithm, crossover and mutation operations are applied on the candidates and new candidate generation is achieved.
7. The one with a better fitness value survives and enters the next generation.

Figure 3 Pseudocode of DEGA model.

It creates new candidate solutions (called agents) by combining the parent individual and several other individuals of the same population. These agents are moved around in the search-space by using mathematical formulae to combine the positions of existing agents from the population. If the new position of an agent is improved than it is accepted and forms part of the population, otherwise the new position is easily throw away. The series of act ion is repetition until achieve the results and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered. This is a greedy selection scheme that often outperforms traditional EAs (Evolutionary Algorithms).

IV. CONCLUSION

Here I would like to conclude that as this is the proposed model for hybrid DEGA and various parameters will be calculated based on which the readings will be considered and graphs will be generated. In next paper, the results for this model will be definitely shown.

V. ACKNOWNLEGMENT

I hereby acknowledge that all the information provided here is absolutely corrected and analyzed by us. I would like to thank my guide, Dr. Sandip Kr. Goyal, for providing me the best resources, with the help of which I am able to design the work and which will be implemented soon. Also, I would like to thank the institution for their support and help.

REFERENCES

[1] S. S. Moharana, R. D. Ramesh & D. Powar , ―Analysis Of Load Balancers In Cloud Computing , International Journal Of Computer Science And Engineering (IJCSE) Vol.2(2), pp. 101-108, 2013.

[2] A. M. Alakeel, ―A Guide to Dynamic Load Balancing in Distributed Computer Systems‖, International Journal of Computer Science and Network Security (IJCSNS), Vol.10(6), 2010.

[3] Moraga RJ, DePuy GW, Whitehouse GE. Metaheuristics: a solution methodology for optimization problems. Handb Ind Optim Probl Handb Ind Syst Eng AB Badiru 2006. http://dx.doi. org/10.1201/9781420038347.

[4] Pop F, Dobre C, Cristea V. Genetic algorithm for DAG scheduling in grid environments. In: 5th IEEE int conf intell comput commun process; 2009. p. 299–305.

[5] T. Sharma, V. K. Banga, ―Efficient and Enhanced Algorithm in Cloud Computing‖, International Journal of Soft Computing and Engineering (IJSCE), Vol. 3(1), 2013.

[6] Ahluwalia et al., International Journal of Advanced Research in Computer Science and Software Engineering 6(6), June-2016, pp. 340-347.
[7] Anu Rani, Dr. Kanwal Garg,"Multi-objective Differential Evolution for Task Scheduling in Cloud Computing", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 5, May 2016.
[8] Mitchell, Melanie, ―An Introduction to Genetic Algorithms‖, International Journal of Communication Network Security ISSN: 2231 – 1882, Vol. 1(4), 2012.
[9] Parveen Sharma, Neha Khurana, "Study of Optimal Path Finding Techniques", International Journal of Advancements in Technology, Vol. 4(2), 2013.