IMPLEMENTATION OF A HYBRID LOAD BALANCING ALGORITHM FOR CLOUD COMPUTING

Gill Sukhjinder Singh[1], Thapar Vivek[2]

[1]*PG Scholar,* [2]*Assistant Professor, Department of CSE, Guru Nanak Dev Engineering College, Ludhiana, Punjab, (India*)

ABSTRACT

In Cloud computing day by day number of users are increasing enormously with large amount of data processing requirements through various sources like social networking sites , business organizations and other. Therefore to solve this issue of growing demands for the resources usage which are heterogeneous and dynamic in nature, resource sharing is done for the achievement of rapid elasticity and consistency, where sharing of resources require an optimal approach in terms of load balancing. In this Paper, the proposed algorithm is a Hybrid algorithm (HACOBEE) which is the optimal solution for load balancing among nodes in a dynamic cloud environment that uses combination of 2 approaches i.e. Ant colony optimization and Artificial Bee Colony approach for workloads distribution among nodes of a cloud. In Ant Colony (ACO), ants continuously update a pheromone table while moving forward and backward from nest to food source and vice versa. Best solution corresponds to maximum pheromone deposit on the path. In Artificial Bee Colony (ABC), bees help in searching food sources & best source is found calculating the fitness value. In Hybrid approach both ACO and ABC provide a combined optimal solution for load distribution based on the pheromone values and bee fitness values. Hybrid approach smoothens the system functioning at normal as well as during high workloads hours. Both ACO and HACOBEE algorithms have been used on similar data sets where HACOBEE showed better response times.

Keywords: Ant Colony Optimization, Artificial Bee Colony, Cloud Computing, Load Balancing, Optimization, Task shifting.

I. INTRODUCTION

Today, Cloud computing is a field that is rapidly advancing in industry as well as Research. Due to the advancement in Cloud, the service industries are offering many services to end users such as distributed, virtualized and elastic resource for realizing computing as a utility.

As cloud service usage is increasing rapidly, it becomes difficult to provide quality services in an effective and efficient manner. In this concern Load Balancing is important field for research as revealed by various comparison studies[1][2][3], therefore with better load balancing , performance improves & hence the services. So, efficient algorithms are needed for load balancing among nodes by task scheduling, where unused virtual machines are put in use in cloud environment. Thus Swarm intelligence techniques like Ant colony and Bee colony optimization are used for load balancing issue in cloud computing environment.

1.1 Load Balancing

Load Balancing is a technique to distribute the load equally among all the nodes of the network. It helps the overloaded and under loaded nodes i.e. if any node is having more load than the threshold value, then its load is shifted to the node with less load. Thus an optimal solution for Load balancing is a major challenge of cloud computing. In most situations it may infeasible and unaffordable, if some of the servers become idle in a datacenter while others are heavily loaded in response to user demands. It means jobs lack the proper assignment to servers in a datacenter thus incurring maintenance cost of idle servers also. Therefore proper load balancing techniques need to be deployed for extra cost cutting in a wide and complex cloud environment.

The goals of load balancing [4] are to:

- For better performance.
- For achieving Stable System state.
- For construction of fault tolerant system.
- For accompanying further modifications.

Load balancing in cloud clusters is different than previous schemes on load balance as there are multiple reference and deployment models to serve different types of clients efficiently. It's a new area where multiple commodity servers participate for load balancing through various cloud clusters. Thus cloud provides economies-of-scale, elasticity and much more[5].

1.2 Load Balancing on the basis of Cloud Environment

Cloud computing can be classified under static or dynamic environment based upon required configuration [6].

1.2.1 Static Environment

In static environment, the cloud provider installs inelastic resources of similar type. In such a case, the cloud has prior information about the capacity of nodes, computing power, storage, performance and statistics of user requirements, where user requirements can't change at run-time. Static Algorithms do not adapt to the dynamic load change. Static environment can be simulated with ease but is not suitable for heterogeneous clouds .

1.2.2 Dynamic Environment

In dynamic cloud environment, the elastic resources of dissimilar type are installed. Therefore run time knowledge about the resources is taken into account rather than prior knowledge unlike static environments. Here abrupt user demands during peak load hours can be accommodated affordably and smoothly. Dynamic environment can't be simulated with ease but its adaptability is high to run time changes in load. There are various policies used in dynamic load balancing like transfer policy, selection policy, location policy & information policy [7].

II. OPTIMIZATION TECHNIQUES FOR LOAD BALANCING IN CLOUD ENVIRONMENT

There are many existing load balancing techniques used in dynamic cloud environment. Below are the two most popular load balancing techniques, which have been considered for proposed work.

2.1 Ant Colony Optimization

M. Dorigo, proposed a scheme based on Ant Colony which is used for various optimization problems, where the ants of various colonies make solutions to a problem by sharing the quality information. ACO algorithm's working is similar to the real ants in which they try to find a shortest path between nest and food source. In ACO, multiple artificial ants make solutions to the optimization problem and share quality information that is, pheromone concentration on the path traversed by ants.

ACO takes the real ant behavior as a basis. Thus the deposition of the pheromone on the path that ants traversed make sense for others to be followed. More frequent traversed Paths contain the higher pheromone concentration whereas the less frequently used paths lose their importance because of less concentration of pheromone on that path. This is based on the assumption that the pheromone concentration fades away after a regular interval of time. Hence the newly arriving ants intending to follow the paths with higher concentration take the advantage to traverse through the shortest path from source to destination.

Ants move forward and backward following two ways[8]:

1) Forward move **–** While moving forward, ants extract the food, or search food sources.

2) Backward move **–** Whereas in backward move, ants pick up food from the food sources and span out back for food storage in the nest.

2.2. Artificial Bee Colony Optimization

Karaboga, proposed a foraging behavior of honey bee swarm in 2005. Artificial Bee Colony Optimization technique was inspired by the above mentioned behavior of bee swarm. In ABC model[9], there are three sorts of bees to inquiry nourishment sources, which incorporate:

- Scout bees: Arbitrarily Search for food source.
- Employed bees: Gather the information about food source and exchange the information gathered with onlooker bees.
- Onlooker bees: Determine the best food source by Calculating the fitness.

Scout bees on their return to the hive after search, examine the food quality by doing waggle dance. While dancing, the food sources are compared on the basis of distance from the hive and current nectar amount of food source. Finally, the onlookers choose best source of food after evaluating from the information provided by scouts during dance and scout bees continue the process of extracting food and take it back to hive.

III. PROPOSED HYBRID ALGORITHM

In our proposed algorithm , the characteristics of both the algorithms Ant Colony and Artificial Bee colony optimization are being utilized for distribution of loads among various nodes in cloud environment. Both ACO and ABC algorithms execute simultaneously. In ACO, ants originate from the head node, where a random node is selected as a head node. Ants span over the cloud network, step by step for selection of nodes following the pheromone concentration on the paths from source to destination nodes. The amount of the pheromone can vary depending upon various elements such as food source quality, distance of food source, etc. Ant's during their traversal update a pheromone table[10], which keeps the record on resource utilization by each node and the distance between nodes. Pheromone is used to move forward and backward i.e. from underloaded node to overloaded node & vice versa for load shifting purpose. Hence in ACO next node is searched depending upon the distance between two nodes and the pheromone concentration[11] on the edges between the same nodes.

On the other hand, Bee colony is used for finding the bee with best fitness value, which will help the task to be allocated to that particular virtual machine based on best fitness value. Employed bees will be carrying the information about the virtual machines capacity in terms of HDD, RAM, Processor and Bandwidth. The knowledge acquired by the employed bees is shared with onlooker bees in the hive, and then the onlooker bees decide the bee with the best fitness value or a virtual machine with similar capacity as that of task length to be executed. Finally , the task is allocated to that virtual machine found with the help of BCO using ACO.

3.1 Programming Steps of the Proposed Hybrid Algorithm

Step1: Initialize population with random solutions or it is generated using Cloud Analyst.

Step2: Initialization of Pheromone Trails.

Step3: Declare threshold level of nodes (between 0 and 1).

Step4: While (no of agents! = null) do

Step5: Prioritize the Population (SJF)

Step6: Evaluate the fitness of the population of the Bees using given formula:

$$fit_{ij} = \frac{\sum_{i=1}^{n} cloudlet\_length_{ij}}{Vm_j\_mips} \quad (1)$$

Where, Vm_j_mips is defined by millions of instructions per second for each processor of Vm_j, **n** is the total no of scout foragers, $\boldsymbol{fit}_{ij}$ defines the fitness function of population of bees (i) for Vm_j or say capacity of Vm_j with i^{th} bee number, **cloudlet_length** is defined as the task length that has been submitted to Vm_j.

The virtual machine (Vm_j) capacity is being calculated using the following parameters[12]:

$$Capacity_{Vm_j} = Vm_j\_cpu * Vm_j\_size + Vm_j\_bandwidth \quad (2)$$

Where, Vm_j_cpu is the number of processors in a virtual machine Vm_j, Vm_j_size is the virtual machine memory size, $Vm_j\_bandwidth$ is the network bandwidth ability of Virtual Machine Vm_j.

Step7: Try searching for new nodes and select sites for neighbourhood search.

Step8: Calculate the Probability and find the Next Probable Node using the Pheromone values using equation:

$$P_k(r,s) = \frac{[\tau(r,s)][\eta(r,s)]^{\beta}}{[\tau(r,u)][\eta(r,u)]^{\beta}} \quad (3)$$

Where,

r = Current node,
s = Next node,
τ = Pheromone concentration of the edge,
η = The desirability of the move for the ant (if the move is from an under loaded node to overloaded node or vice-versa the move will be highly desirable),
β = Depends upon the relevance of the pheromone concentration with the move distance.

Step9: Send bees for the chosen sites and calculate the fitness function.

$$fit_{ij} = \frac{\sum_{i=1}^{n} cloudlet\_length_{ij}}{Vm_j\_size} \quad (4)$$

International Journal of Advanced Technology in Engineering and Science
Vol. No.3, Special Issue No. 01, September 2015
www.ijates.com
ijates
ISSN 2348 - 7550

Where, Vm_j_size is the virtual machine memory size.

Step10: Selection of best fit bee is done in each patch and select the Node whose Pheromone is highest on the basis of condition i.e. threshold of ACO and fitness value of ABC. At each algorithm iteration, the fittest bee will be chosen to assign tasks in Vm_j

Step11: Calculate Load Balance Check if the load on selected node less than or greater than threshold Update Foraging or trailing Pheromones .

Step12: Assign remaining bees and remaining ants to search randomly and evaluate their fitnesses and pheromones.

3.2 Flowchart (Hybrid(HACOBEE) Algorithm)

The flowchart in the Figure1, describes the above-mentioned procedure.

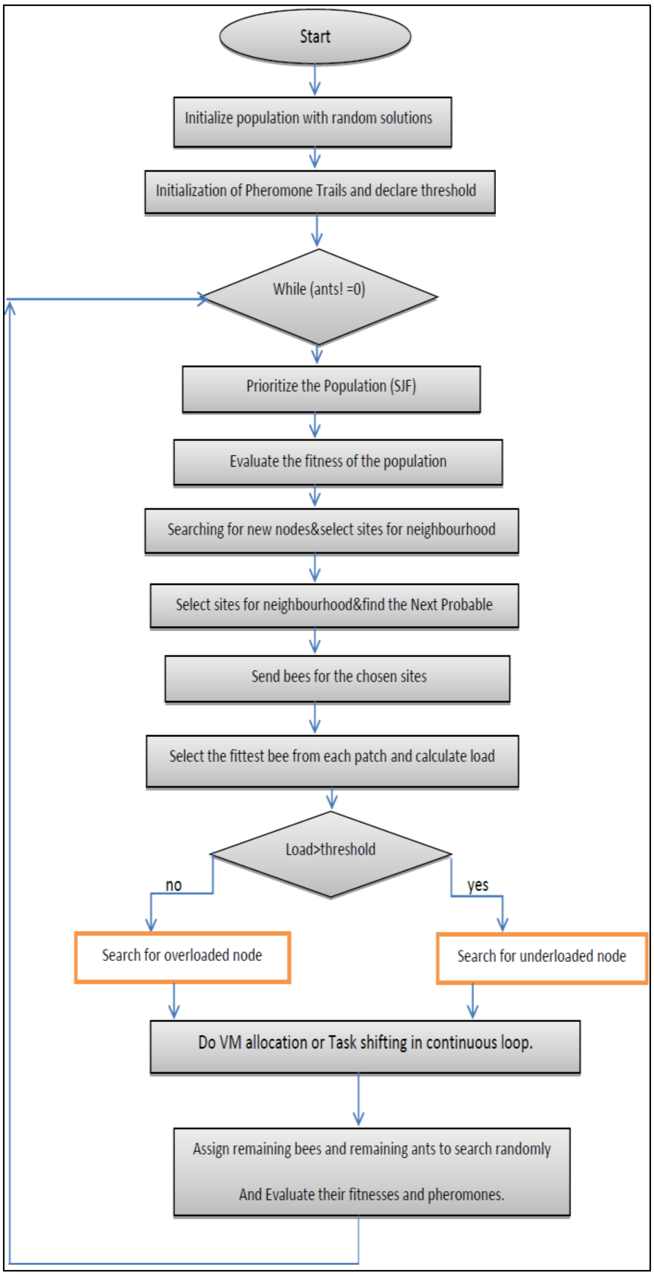

Figure 1. Flowchart (Hybrid(Hacobee) algorithm)

IV. RESEACH SETUP AND RESULTS

The proposed algorithm is implemented through simulation package Cloud Analyst tool [13]. Java language is used for development and implementation of Hybrid algorithm for load balancing. Assuming the application is deployed in 3 data centers having virtual machines 30 in each data center running on each physical host , then the **parameter values** are as under:

4.1 Table Below Shows Parameter Values

Table 1 Parameter Setting of Cloud Analyst Tool (Simulation duration - 60 mins)

PARAMETERS	VALUE
Number of Datacenters	3
User Bases	6
Service Broker Policy	Closest Data Center
Physical H/w units (Physicals hosts) in each data center	10
Total No. of VMs	DC1-30,DC2-30, DC3 - 30
Processor Speed	1000 - 3000
VM Memory (RAM)	512 - 1024
VM Policy	Time Shared

4.2 Below are the configuration screens in cloud analyst tool

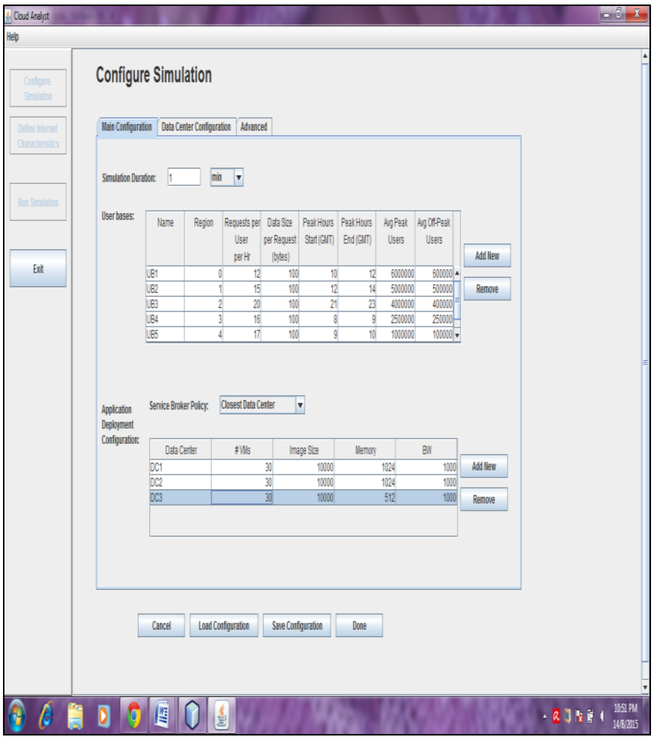

Figure 2. Main Configuration Window

ijates
ISSN 2348 - 7550

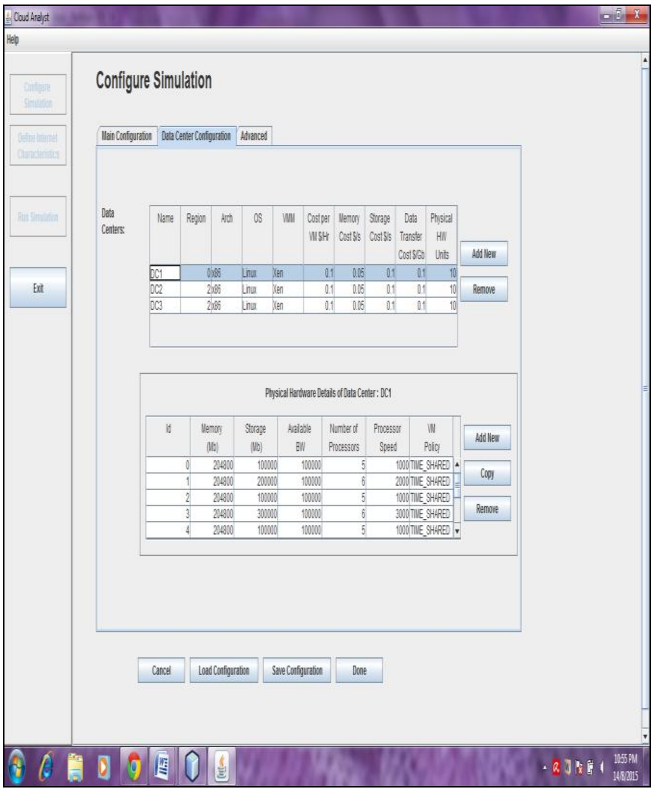

Figure 3. data center configuration window

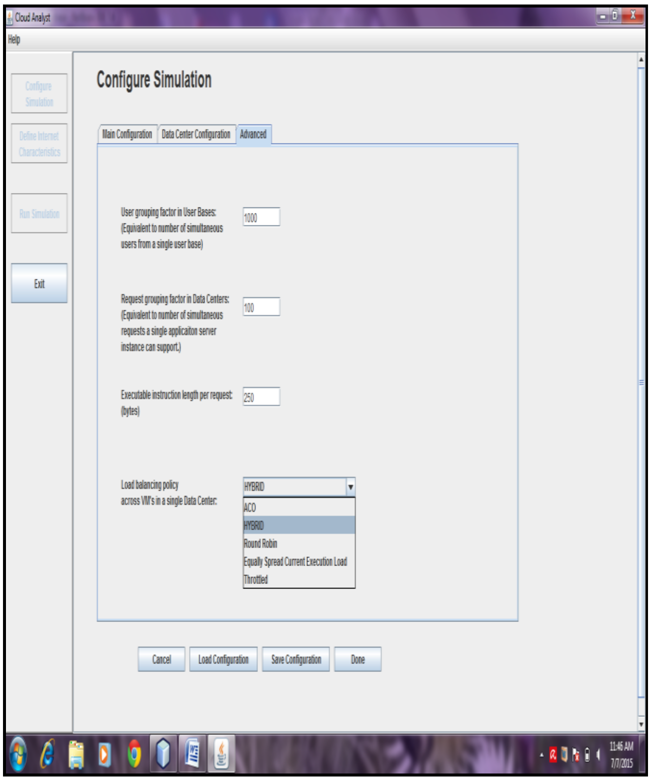

Figure 4. configure simulation window

4.3 Following are the experimental Results based on ACO and Hybrid Algorithms

Overall Response Time Summary

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	2102.67	58.46	5596.83
Data Center processing time:	1950.65	8.57	5316.82

Figure 5. - Results (Using ACO)

Overall Response Time Summary

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	2099.83	58.67	5597.59
Data Center processing time:	1946.59	8.42	5316.21

Figure 6 - results (using Hybrid(Hacobee) algorithm)

In above figures, the complete simulation configuration has been shown in terms of various parameters, using Cloud Analyst tool. After configuration, the results of both the algorithms were analyzed to check Overall Response time and Data processing time.

V. CONCLUSION

In this paper , a Hybrid Algorithm is proposed and then implemented in Cloud Computing environment using Cloud Analyst tool. In this algorithm, VM's with different resource types are used in such a way that the capacities of servers are utilized in an efficient manner. Virtual machines are mapped on to physical resources in different data centers as per userbase requests, thus avoiding overloading of nodes even at peak load hours.

The proposed hybrid algorithm shows that response time with this algorithm is better than the ACO in specific situation's but not in all cases. Also the cost may not be a favorable factor in some situations while performing this algorithm in a dynamic cloud computing environment.

REFERENCES

[1] Goyal Akhil and Bharti, "A Study of Load Balancing in Cloud Computing using Soft Computing Technique", International Journal of Computer Applications(0975– 8887), Volume 92 – No.9, 2014, 29-32.

[2] Kumar K. Prasanna, Kumar S. Arun and Dr. Jagadeeshan, "Effective Load Balancing For Dynamic Resource Allocation in Cloud Computing", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, Issue 3, ISSN 2320-9801, 2013, 3427-3431.

[3] Uma J., Ramasamy V. and Kaleeswaran, "Load Balancing Algorithms in Cloud Computing Environment - A Methodical Comparison", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) , Volume 3, Issue 2, ISSN 2278-1323, 2014, 272-275.

[4] Desai Tushar and Prajapati Jignesh, "A Survey Of Various Load Balancing Techniques And Challenges In Cloud Computing", International Journal Of Scientific & Technology Research Volume 2, Issue 11, ISSN 2277-8616, 2013, 158-161.

[5] Mishra Ratan and Jaiswal Anant, "Ant colony Optimization: A Solution of Load balancing in Cloud", International Journal of Web & Semantic Technology (IJWesT) Vol.3, No.2, DOI : 10.5121/ijwest.2012.3203, 2012, 33-50.

[6] Katyal*, Mayanka and Mishra Atul, “A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment”, International Journal of Distributed and Cloud Computing, Volume 1, Issue 2, 2013, 5-14.

[7] Gupta Ruhi, “Review on Existing Load Balancing Techniques of Cloud Computing”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 2, ISSN: 2277 128X, 2014, 168-171.

[8] W. Ngenkaew, S. Ono and S. Nakayama, “Pheromone-Based Concept in Ant Clustering”, Proc. 3rd IEEE Conf. on Intelligent System and Knowledge Engineering, 2008, 308-312.

[9] Kruekaew, B. and Kimpan, W., “Virtual Machine Scheduling Management on Cloud Computing Using Artificial Bee Colony", Proc. International Multi Conference of Engineers and Computer Scientists 2014, Vol I, IMECS 2014, 2014, .

[10] Kumar Nishant, Sharma Pratik, Krishna Vishal, Gupta Chhavi , Singh KunwarPratap, Nitin and Rastogi Ravi, “Load Balancing of Nodes in Cloud Using Ant Colony Optimization”, Proc. 14th IEEE Conf. on Modelling and Simulation, 2012, 3-9.

[11] Arabi E. Keshk, El-Sisi Ashraf, Medhat A. Tawfeek and F. A. Torkey, “Intelligent Strategy of Task Scheduling in Cloud Computing for Load Balancing”, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 2, Issue 6, ISSN 2278-6856, 2013, 12-22.

[12] L.D. Dhinesh Babu and Krishna P. Venkata, “Honey bee behavior inspired load balancing of tasks in cloud computing environments”, Applied Soft Computing table of contents archive, Volume 13, Issue 5, 2013, 2292-2303.

[13] Wickremasinghe Bhathiya, Rodrigo N. Calheiros, and Buyya Rajkumar , Cloud Analyst: A CloudSim-based Visual Modeller for Analysing CloudComputing Environments and Applications, Distributed Computing Project, CSSE Dept., University Of Melbourne, 2009.