An Ant Colony Based Load Balancing Strategy in Cloud Computing

Santanu Dam [1], Gopa Mandal[2], Kousik Dasgupta[3], Paramartha Dutta[4]

[1]Future Institute of Engineering & Management
Kolkata-700 150, India

[2,3] Kalyani Govt. Engineering College
Kalyani-741 235, India

[4]Visva- Bharati University
Shantiniketan-731 235, India
sntndm@gmail.com

Abstract.

Cloud computing thrives a new supplement of consumption and delivery model for internet based services and protocol. It provides large scale computing infrastructure defined on usage and also provides infrastructure services in a very flexible manner which may scales up and down according to user demand. To meet the QoS and satisfy the end users demands for resources in time is one of the main goals for cloud service provider. For this reason selecting a proper node that can complete end users task with QoS is really challenging job. Thus in Cloud distributing dynamic workload across multiple nodes in a distributed environment evenly, is called load balancing. Load balancing can be an optimization problem and should be adapting its strategy to the changing needs. This paper proposes a novel ant colony based algorithm to balance the load by searching under loaded node. Proposed load balancing strategy has been simulated using the CloudAnalyst. Experimental result for a typical sample application outperformed the traditional approaches like First Come First Serve (FCFS), local search algorithm like Stochastic H*ill Climbing (SHC) ,another soft computing approach Genetic Algorithm (GA) and some existing Ant Colony Based strategy.*

Keywords: Cloud Computing; CloudAnalyst; Ant Colony Optimization; Load Balancing.

1 INTRODUCTION

A new paradigm of large scale distributed computing is "Cloud". It utilizes the high speed of the internet to disperse the job from private PC to the remote computer clusters (Data Center owned by the cloud service providers). Cloud computing has become very popular for industry as well as academia for its sophisticated on demand services offered by its service providers like Google, Amazon [1]. Due to exponential

growth of the internet in this decade computing infrastructure provided by its service providers may be used by industry or individuals from anywhere of the world. In future it has full potentiality to serve as computing as utility by the help of distributed virtualized elastic resource for end user [2]. Cloud service provider offers computing, software and storage as service. On demand provisioning and de-provisioning helps organization to reduce capital costs of software and hardware for this reason it has been adopted widely. As the size of the cloud may scale up and down the service providers have to provide computing power as lease to the users, in form of virtual machines (VM's)[3]. That makes Cloud computing a promising technology to provide resource on demand and to service the received request within time. Therefore high availability of resources is required and moreover management of resources is a big challenge to ensure QoS to end user and accelerates business performance of cloud service provider [4]. The primary challenges for the Cloud service provider is to scale up the performance or keep same. Cloud computing has a glorious future but many crucial problem still need to be realized. Load balancing is one of these problems where we have to distribute the local workload evenly to the whole cloud and ensures that at any instant of time all the processor or resources in the cloud does approximately the equal amount of work. This avoids the situation where some resources are heavily loaded while other are idle or doing very little amount of work (under loaded). To meet the criteria a good load balancing algorithm should be dynamic and adapt the environment [5].

In this paper a basic Ant Colony Optimization (ACO) has been proposed for load balancing of VMs in Cloud. ACO is a random search algorithm imitating the behavior of ant colonies. Ants are trailing from their nest to food and connect each other by pheromone which is volatile substance laid on paths traveled. CloudAnalyst a CloudSim based visual modeler used here for simulation and analysis of the proposed technique. The experimental result remarkably optimizes the entire system load.

The rest of paper is organized as follows. Section2 Introduces the CloudAnalyst toolkit. Section 3 Load Balancing of VM's using Ant Colony Algorithm in Cloud Computing. Section 4 details the proposed ACO algorithm. Section 5 presents the simulation results. Finally, Section VI concludes this paper.

2 CLOUD ANALYST

Sometime it's very difficult and time consuming to measure the performance of the application or proposed policies in real world environment. In this consequence simulation is very much helpful to allow users or researchers with practical feedback without having real environment. This section portray the simulation in cloud to support application level infrastructure, services arises from cloud computing paradigm such as modeling of on demand virtualized resources, which supports cloud infrastructure. Different simulators are available today to adapt the real world situation like CloudSim [6] and CloudAnalyst [7]. CloudAnalyst has been used in this paper as simulation tool. CloudAnalyst is a GUI based visual modeling and simulation tool based on the functionalities of CloudSim . Large scale appli-

cations that can be deployed on cloud infrastructures. CloudAnalyst enables developers to evaluate the large scale application in terms of geographic distribution of both computing servers and user's workload. A snapshot of the GUI of CloudAnalyst simulation toolkit is shown in figure 1(a) and its architecture in depicted figure 1(b). CloudAnalyst [7] developed as a visual modeler tool on CloudSim [8].

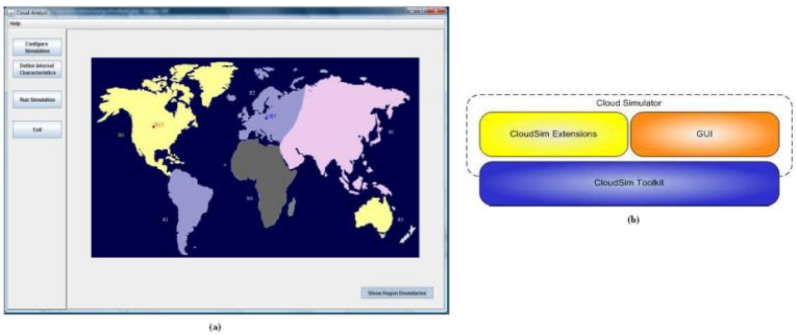

Fig. 1. Snapshot of CloudAnalyst (a) GUI of CloudAnalyst (b) Architecture of CloudAnalyst builds on CloudSim

3 LOAD BALANCING OF VM'S USING ANT COLONY ALGORITHM IN CLOUD COMPUTING

Ant Colony Optimization is basic foraging behavior of an ant that encouraged them to find the optimal shortest path from their nest to food introduced by Dorigo. M [8]. When ants are moving from their nest to food or vice versa they deposit a chemical substance called pheromone on their path. Paths are randomly chosen by ants initially. Chance of an isolated ant to follow a particular path among several possibilities always based on previously laid trail [10]. High concentrated pheromone helps an ant to choose a path and more ants are also attracted due to this high pheromone. By this way trail are reinforced with its own pheromone. Probability of an ant can separate the best optimal path from different set of paths is proportion to the concentration of a way's pheromone. As a result denser pheromone attracts more ants. It's a basically positive feedback mechanism that helps ants to find an optimal path finally.

3.1 A. **The proposed method**

As and when a job/request comes to the cloud service provider, they are allocated VMs in First Come First Serve manner and an index table is maintained to keep account about their current allocation. As the process continues a time will come due to vastness of Cloud when free available VMs are going to exhaust. In the situation artificial ants are created and dispersed to wander across the network to search under

loaded VM's. Such an artificial ant is trying to choose a path from pheromone trail intensity that is initially assigned as given in equation 1.

$$\tau_{ij}(t=o) = f(MIPSJ, L, BWJ) \tag{1}$$

where, τ_{ij} (t=0) is the pheromone value in between two node i and j at turn t=0, MIPSJ (Million Instructions per Second) is the maximum capacity of each processor of VMJ and the parameter BWJ is related to the communication bandwidth ability of the VMJ. L is the delay cost is an estimate of penalty, which cloud service provider needs to pay to customer in the event of job finishing actual time being more than the deadline advertised.

Thus any ant randomly choose VM's to find underloaded VM, as the ants starts its trip across the networks from a node, at each move of the kth ant traverse from node i to node j, the probability function for an ant at node i to choose a neighbour node j as its next stop at time t pkij(t) is given by equation 2.

$$p_{ij}^{k}(t) = \{ \frac{[\tau_{ij}(t)]^{\alpha} \quad [\eta_{ij}(t)]^{\beta}}{\sum_{k \in allowed_k} [\tau_{ik}(t)]^{\alpha} \quad [\eta_{ik}(t)]^{\beta}}, if \ j \in \text{allowed}_k \tag{2}$$

Where, $allowed_k$ means the pheromone value updating due to the tour of the kth ant on its tabu (memory) list. The tabu list of the kth ant defined by $tabu_k$. α, β are two parameters for controlling the relative weight of the pheromone trail and heuristic value. τ_{ij} (t) is the pheromone value in between two node i and j, this value defined attractiveness. η_{ij} is the heuristic value given by equation 3.

$$\eta_{ij}(t) = 1/d_{ij} \tag{3}$$

Where, d_{ij} is the hop distance between node i and node j.

Finally, the trip of an ant helps to identify the effectively underloaded VM within optimal distance. The information is updated in the index table globally. Correspondingly the pheromone values are updated as given in equation 4.

$$\tau_j(t+1) = (1-\rho) * \tau_j(t) + \Delta\tau_j \tag{4}$$

Where, τ_j(t+1) is pheromone value of node j at time (t+1), ρ is the pheromone trail decay coefficient. If the value of ρ is greater, that shows less the impact of past solution. $\Delta\tau_j$ is local pheromone updating on the visited VMs when an ant completes its tour is given by equation 5.

$$\Delta\tau_j = 1/T_{ik} \quad (5)$$

Where, T_{ik} be the optimal path distance that searched by k^{th} ant at the i^{th} iteration.

4 PROPOSED ALGORITHM

Step 1: Maintain an index table which contains VmId and its corresponding requests (that are allocated for execution). Initially all VMs have current request 0.
Step 2: Schedule new request to VMs according to FCFS scheduling policy.
Step 3: Make corresponding change in the index table.
Step 4: If VMs are not available to allocate next job.
Step 4a: Create random number of ant with same pheromone value and place them randomly to traverse.
Step 4b: For m numbers of VMs and n numbers of random ants do
Step 4b-1: If an ant choose a VM then check whether the ant completes its tour or not.
Step 4b-2: If tour completed then update the pheromone value.
Step 4b-3: Check whether the solution is optimal and go to Step 5,
Step 4b-3: Else for non optimal solution, check whether all the ants have completed its tour. For non completion go to step 4b-2, else step-5.
Step 5: Store the current optimal solution and update pheromone value globally in the table.
Step 6: If all ants complete their tour then compare every local pheromone updates to output best possible solution.

5 SIMULATION RESULTS AND ANALYSIS

The proposed algorithm is simulated in CloudAnalyst[8] by considering the scenario of "social networking site like Facebook". Suppositional configuration generated partitions the world into six "Regions" that is nothing but six continents as given in Table 1.

Table 1. Configuration of simulation environment

S.No	User Base	Region	Simultaneous Online Users During Peak Hrs	Simultaneous Online Users During Off-peak Hrs
1	UB1	0–N. America	4,70,000	80,000
2	UB2	1–S. America	6,00,000	1,10,000
3	UB3	2 – Europe	3,50,000	65,000
4	UB4	3 – Asia	8,00,000	1,25,000
5	UB5	4 – Africa	1,15,000	12,000
6	UB6	5 – Oceania	1,50,000	30,500

A single time zone is set for all user bases(UB) and for each UBs a sample online user during peak hour and off peak hour has been considered. Of the entire online user only one tenth approximately is available during off peak hours.

Each simulated data centre host has a particular amount of virtual machines (VMs) dedicated for the application. For simulation each of the Machines has been considered of 4GB of RAM, 100 GB storage and 1000MB of available bandwidth. Each Datacenter (DC) is assumed to be having 4 CPUs with a capacity of 10000 MIPS. Simulated hosts have x86 architecture, virtual machine monitor Xen and Linux operating system. Each user request (jobs) has been considered to be requiring 100 instructions to be executed.

The proposed algorithm is executed in several setups as tabulated in Table 2, where one DC is considered having initially 25, 50 and 75 VMs in each Cloud Configurations (CCs). Simulation scenario of Table 3 consists of two DCs with a variation of 25, 50 and 75 VMs. In Table 4, 5, 6 and 7 considers three, four, five and six DCs respectively with a mixture of 25, 50 and 75 VMs for all DCs. Average response time of the jobs are calculated for the proposed algorithm and tabulated. The performance of proposed algorithm is compared with some existing load balancing algorithm like .Genetic Algorithm (GA)[10], Stochastic Hill Climbing Algorithm (SHC)[11], Existing ACO[12] strategy and First Come First Serve (FCFS)[8]. Figures 2, 3,4,5,6 and 7 make a comparative analysis of the proposed technique for the different scenarios and techniques. The comparative analysis confirms the novelty of the work.

6 CONCLUSION

In this paper, soft computing based algorithm on ant colony optimization has been proposed to initiate the load balancing under cloud computing architecture. Detail analysis of the results, indicates that the proposed strategy for load balancing not only outperforms a few existing techniques but also guarantees the QoS requirement of customer job. Though fault tolerance issues does not consider and all jobs are predicted with same priority here, which may not be the actual scenario. Researchers can proceed to include the fault tolerance and different function variation to calculate the pheromone value can be used for further research work.

REFERENCES

1. Rajkumar Buyya, James Broberg and Andrzej Goscinski CLOUD COMPUTING Principles and Paradigms, Jhon Wiley & Sons, 2011
2. “Cloud Task scheduling based on Load Balancing Ant Colony Optimization presented on” 2011 Sixth Annual ChinaGrid Conference IEEE DOI 10.11.09 /ChinaGrid.2011.17 by Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong, Dan Wang
3. M. D. Dikaiakos, G. Pallis, D. Katsa, P. Mehra, and A. Vakali, “Cloud Computing: Distributed Internet Computing for IT and Scientific Research”,in Proc. of IEEE Journal of Internet Computing, Vol. 13, No. 5,pp. 10-13, 2009.M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.
4. ”An Enhanced Load Balancing Technique for Efficient Load distribution in Cloud-Based IT Industries” Rashmi Krfishna Iyenger Srinivasam, V. Suma, and Vaidehi Nedu In proc of Intelligent Informatics,AISC182,pp479-485

5. “Cloud Task scheduling based on Load Balancing Ant Colony Optimization presented on” 2011 Sixth Annual ChinaGrid Conference IEEE DOI 10.11.09 /ChinaGrid.2011.17 by Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong, Dan Wang
6. R. N. Calheiros, R. Ranjan, A. Beloglazov, C. Rose and R. Buyya, “Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, in Software: Practice and Experience (SPE), Volume 41,Number 1, ISSN: 0038-0644, Wiley Press, New York, USA., pp. 23-50,2011.
7. B.Wickremasinghe, R. N. Calheiros and R. Buyya, “Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications”, in Proc. of Proceedings of the 24th International Conference on Advanced Information Networking and Applications (AINA2010), Perth, Australia, pp.446-452, 2010.
8. M. Dorigo, L.M. Gambardella, “Ant colony system: A cooperative learning approach to the traveling salesman problem”, in IEEE Transactions on Evolutionary Computation (1997), DOI: 10.1109/4235.585892, pp.53–66, 1997.
9. Sowmya Suryadevera, Jaisi Chourasia, Sonam Rathore & Abdul Jhummarwala,“Load Balancing in Computational Grids Using Ant Colony Optimization”in Proc. Of (IJCCT),ISSN0975-7449, vol-3,Iss-3,2012
10. [10]“A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing” in Proc. of CIMTA-2013 Elsevier, Procedia Technology(2013), Vol 10,2013, Pages 340-347, ISBN978-93-5126-672-3 Kousik Dasgupta , Brototi Mandal, Paramartha Dutta, Jyotsna Kumar Mondal, Santanu Dam.
11. ”Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach", Brototi Mondal,Kousik Dasgupta and Paramartha Dutta, in Proc. of (C3IT-2012),Elsevier, Procedia Technology 4(2012), pp.783-789, 2012.
12. Kumar Nishant, Pratik Sharma, Vishal Krishna, Nitin Rastogi and Ravi Rastogi; "Load Balancing of Nodes in Cloud Using Ant Colony Optimization"; Procedings of the 14th International Conference on Modelling and Simulation, pp. 1-9, 2012.

Table 2.Simulation scenario and calculated overall average response time (RT) in (ms) using one DC

Sl. No.	Cloud Configuration	Data Center specification	RT in ms for proposed ACO	RT in ms for existing ACO	RT in ms for GA	RT in ms for SHC	RT in ms for FCFS
1	CC1	One DC with 25 VMs	328.98	329.01	329.01	329.02	330.11
2	CC2	One DC with 50 VMs	327.63	328.63	328.97	329.01	329.65
3	CC3	One DC with 75 VMs	238.12	248.43	244	329.34	329.44

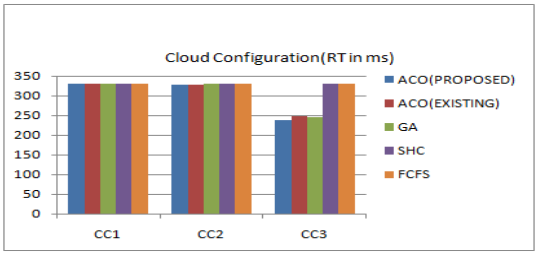

Fig. 2. Performance analysis of proposed ACO with GA, SHC and FCFS Result using one Datacenter.

Table 3. Simulation scenario and calculated overall average response time (RT) in (ms) using Two DC

S.No	Cloud Cofigura-tion	Data Center specification	RT in ms for proposed ACO	RT in ms for existing ACO	RT in ms for GA	RT in ms for SHC	RT in ms for FCFS
1	CC1	Two DCs with 25 VMs each.	354.72	358.97	360.77	365.44	376.34
2	CC2	Two DCs with 50 VMs each.	349.89	354.21	355.72	360.15	372.52
3	CC3	Two DCs with 75 VMs each.	348.68	352.66	355.32	359.73	370.56
4	CC4	Two DCs with 25, 50 VMs.	346.57	348.64	350.58	356.72	368.87
5	CC5	Two DCs with 25, 75 VMs.	347.86	348.12	351.56	357.23	367.23
6	CC6	Two DCs with 75, 50 VMs.	350.47	351.45	352.01	357.04	361.01

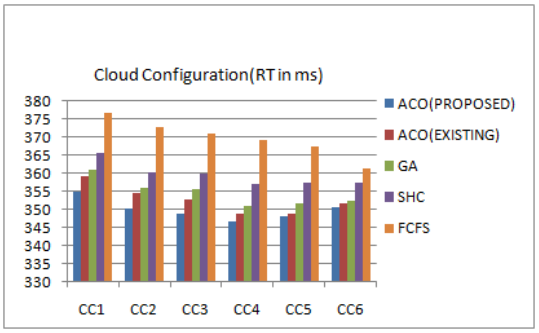

Fig. 3. Performance analysis of proposed ACO with GA, SHC and FCFS Result using Two Datacenter.

Table 4. Simulation scenario and calculated overall average response time (RT) in (ms) result using Three Data Centers

Sl No	Cloud Configura-tion	Data Center speci-fication	RT in ms for proposed ACO	RT in ms for existing ACO	RT in ms for GA	RT in ms for SHC	RT in ms for FCFS
1	CC1	Each with 25 VMs .	345.68	348.57	350.32	356.82	363.34
2	CC2	Each with 50 VMs .	344.86	346.83	350.19	355.25	363.52
3	CC3	Each with 75 VMs	340.62	343.89	346.01	350.73	361.56
4	CC4	Each with 25, 50 ,75VMs.	340.86	343.53	345.98	350.01	360.87

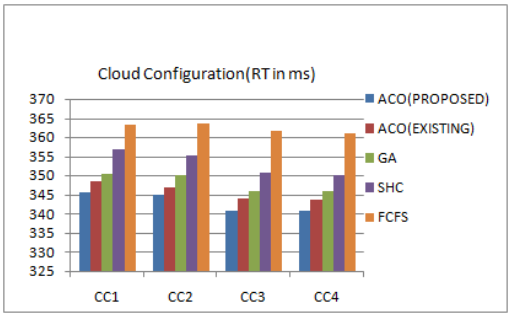

Fig. 4. Performance analysis of proposed ACO with GA, SHC and FCFS Result using Three Datacenter.

Table 5. Simulation scenario and calculated overall average response time (RT) in (ms) result using Four Data Centers

Sl No	Cloud Configuration	Data Center specification	RT in ms for proposed ACO	RT in ms for existing ACO	RT in ms for GA	RT in ms for SHC	RT in ms for FCFS
1	CC1	Each with 25 VMs.	341.46	346.57	348.85	354.35	360.95
2	CC2	Each with,50 VMs .	339.78	343.84	345.54	350.71	359.97
3	CC3	Each with 75 VMs	336.56	339.78	340.65	346.46	358.44
4	CC4	Each with 25, 50 ,75VMs.	334.32	335.43	337.88	344.31	355.94

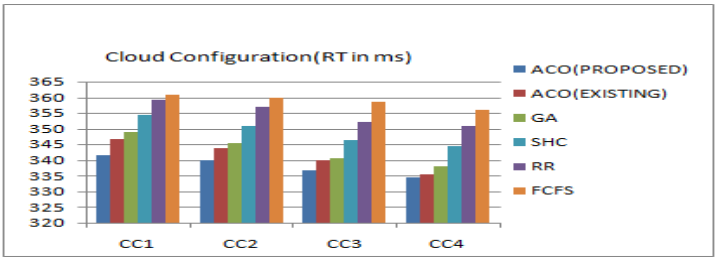

Fig. 5. Performance analysis of proposed ACO with GA, SHC and FCFS Result using Four Datacenter.

Table 6. Simulation scenario and calculated overall average response time (RT) in (ms) result using Five Data Center

Sl .No	**Cloud Configu-ration**	**Data Center specification**	**RT in ms proposed ACO**	**RT in ms for existing ACO**	**RT in ms for GA**	**RT in ms for SHC**	**RT in ms for FCFS**
1	CC1	Each with 25 VMs.	331.45	334.80	335.64	342.86	352.05
2	CC2	Each with 50 VMs.	321.12	325.59	326.02	332.84	345.44
3	CC3	Each with 75 VMs	319.89	321.48	322.93	329.46	342.79
4	CC4	Each with 25, 50 , 75 VMs.	317.65	319.04	319.98	326.64	338.01

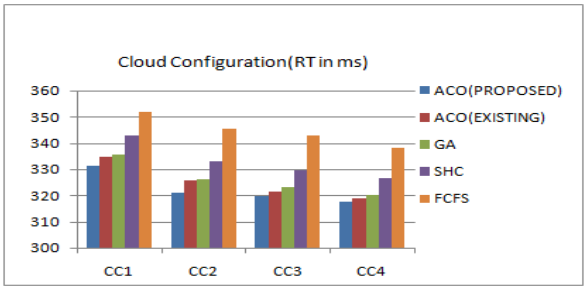

Fig. 6. Performance analysis of proposed ACO with GA, SHC and FCFS Result using Five Datacenter.

Table 7. Simulation scenario and calculated overall average response time (RT) in (ms) result using Six Data Center

Sl. No.	Cloud Con-figuration	Data Center specification	RT in ms for pro-posed ACO	RT in ms existing ACO	RT in ms GA	RT in ms SHC	RT in ms FCFS
1	CC1	Each with 25 VMs .	323.98	326.36	330.54	336.96	349.26
2	CC2	Each with 50 VMs .	316.48	321.73	323.01	331.56	344.04
3	CC3	Each with 75 VMs.	313.56	318.64	321.54	327.78	339.87
4	CC4	Each with 25, 50 ,75VMs.	309.66	312.32	315.33	323.56	338.29

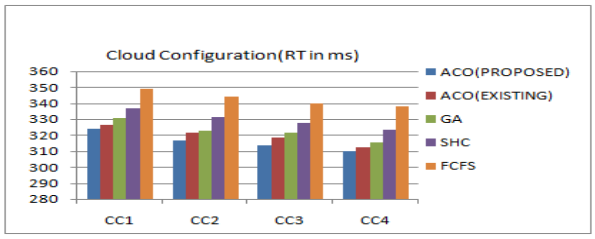

Fig. 7. Performance analysis of proposed ACO with GA, SHC and FCFS Result using Six Datacenter.