ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

Effective Load Balancing for Cloud Computing using Hybrid AB Algorithm

[1]N. Sasikala and [2]Dr. D. Ramesh

PG Scholar, Department of CSE, University College of Engineering (BIT Campus), Tiruchirappalli, Tamil Nadu, India[1]

Asst. Professor, Department of CSE, University College of Engineering (BIT Campus), Tiruchirappalli, Tamil Nadu, India [2]

ABSTRACT: Problem Statement: Cloud Computing is the fast growing technology, which shares the resource to achieve consistency and economies of scale similar to a utility over a network. Resource sharing requires more optimized algorithm. **Approach:** In traditional many static algorithms are proposed, but all these algorithms do not produce optimal job scheduling and load balancing. In order to reduce the waiting time and execution time, the hybrid AB algorithm is proposed. The hybrid AB algorithm is a combination of two dynamic algorithms, Ant Colony Optimization and Bees Life algorithm. **Results:** Ant Colony algorithm solves load balancing problem in cloud and Bees Life algorithm is used for optimal job scheduling. **Conclusion:** The result of the proposed study shows that the hybrid AB algorithm will effectively balances the load than other existing systems.

KEY WORDS: Cloud Computing, Load Balancing, Ant Colony Optimization, Bees Life algorithm

I. INTRODUCTION

Cloud makes us to access information from anywhere at any time. Cloud computing is a variety of computing concepts that involve a large number of computers connected through a network such as Internet. Cloud providers offers services as Infrastructure as a services (IaaS), Platform as a services (PaaS) and Software as a services (SaaS). Infrastructure as a services supply these resources on-demand from their pools installed in data centres. Platform as a service typically includes operating system, programming language execution environment and web server. Software as a service provides access to application software and database.

In distributed system, the load balancing is a process of distributing the load among various node to improve the resource utilization and job response time. Load balancing is to assign the workload equally as soon as possible. The traditional job scheduling algorithm works for minimum number of jobs alone as static nature. But, now-a-days user needs dynamic environment for getting fast job allocation and quick response time. In existing there are many dynamic algorithms such as Ant Colony, Bee Colony, Bees Life algorithm, Particle Swarm Optimization and job scheduling algorithm.

Ant colony optimization algorithm is used to determine the load balancing problem in cloud. Ant finds the optimal path i.e. destination based on the behaviour of ant. When ant goes for searching, it leaves pheromone when moving. All incoming ant must update on the pheromone table for scheduling. Pheromone table row contain routing preference for each destination and column represents the probability of choosing a neighbour as the next hop. After ant reaches the destination it must update in pheromone table. According to the probability in table, the highest probability is taken as a destination path; it will be shortest path than other path.

Bees life algorithm is an optimization algorithm used for job scheduling. Bees colony contains single breeding female bee called queen and male known as Drones. Bees start with scout bees with initial population first the bees choose randomly in space. Then, fitness is calculated for bees. The highest fitness is chosen as “selected bees” and remaining bees are workers. The selected bees alone visit the site by choosing neighbour search.

II. RELATED WORK

Jing Liu et al. (2013) proposed Multi Objective Genetic Algorithm (MO-GA) for dynamic job scheduling. This algorithm combines both random and greedy initialization methods. In genetic algorithm the fitness is calculated by energy consumption and profits of the service providers. The best fitness is selected and stored in pareto. Then selection operation can be done by two strategies: elitism and crowding. After the selection process crossover operation can be done by two individuals. Mutation swaps their position to generate new individuals This MO-GA is used to minimize the energy consumption and to maximize the profit of service providers.

Linan Zhu et al. (2012) proposed Ant Colony Optimization algorithm to solve travelling sales man problem. Ant colony algorithm chooses the target path through the pheromone strength so task amount spent is less than other algorithm. This ant colony algorithm achieves QOS requirement and shortest path. Thus ant colony algorithm gives more efficient result for calculating node distribution and load balancing.

Tasquia Mizan et al. (2012) proposed modified task scheduling algorithm based on the concept of bees life algorithm and greedy algorithm to get a optimal service in hybrid cloud. Bee life algorithm are residential for task scheduling with greedy method will selects the randomly for one data centre. In modified job scheduling the jobs are processed in the queue using non-premptive priority queue. First the tasks enter to the BLA algorithm and to greedy method to get optimal solution. The makespan can be reduced by this hybrid algorithm.

Yuvaraj Kumar et al. (2012) proposed fuzzy bee colony technique, which works in localized and self-motivated manner. The honey bee colony is a division of labour between the forgers, they work in the field of collecting the nectar at random search, after getting the nectar it returns back to the hive and it starts waggle dance. This dance is closely correlated with the search time experienced by dancing bees. By using this technique resource utilization can be maximized.

III. PROPOSED WORK

In proposed system, the ant colony and bees life algorithm are combined to improve the effectiveness of load balancing and cloud scheduling. The bees' life algorithm searches neighbour node and collect information (bandwidth, execution time, free memory space) from all neighbour nodes, and maintains as a table. If user gives any job as an input to cloud, the BLA first gets the job details and calculates the fitness for each job. The fitness value can be calculated by
Fitness(DCTasks)= Σ(Fitness(JTaskij, DCj))

Where,

Fitness(JTaskij, DCj) = JTaskij.TimeToExe

JTaskij.TimeToExe – execution time of job 'i' need to run in DCj.

From the above fitness value, the best fitness value alone is taken to next generation. The BLA table information's are given to UN (Updation Node), this node also contains ANT COLONY table information such as past history of each node, available resource and best path to resource allocation. The UN node chooses the best path for resource allocation and shares the workload all VMs.

GRID MONITOR NODE

The proposed Cloud Monitor node helps in effective scheduling of workloads. The cloud Monitor node is nothing but one of the active node of the cloud. The Cloud Monitor node is elected from the list of available cloud nodes based on the computational power, storage availability, RAM etc.,

The functionalities of the Cloud Monitor Node are ,
a) It receives the query from the user
b) It assigns the job scheduling effecively,
c) It splits the job as tasks whenever required.

In this system design the user schedules the job to cloud through BLA and ANT algorithm. Both give their calculation to control node. The BLA calculates the resource available in cloud and gives to control node. This ANT COLONY checks for available resource in cloud by BLA result and makes perfect scheduling for user to cloud.

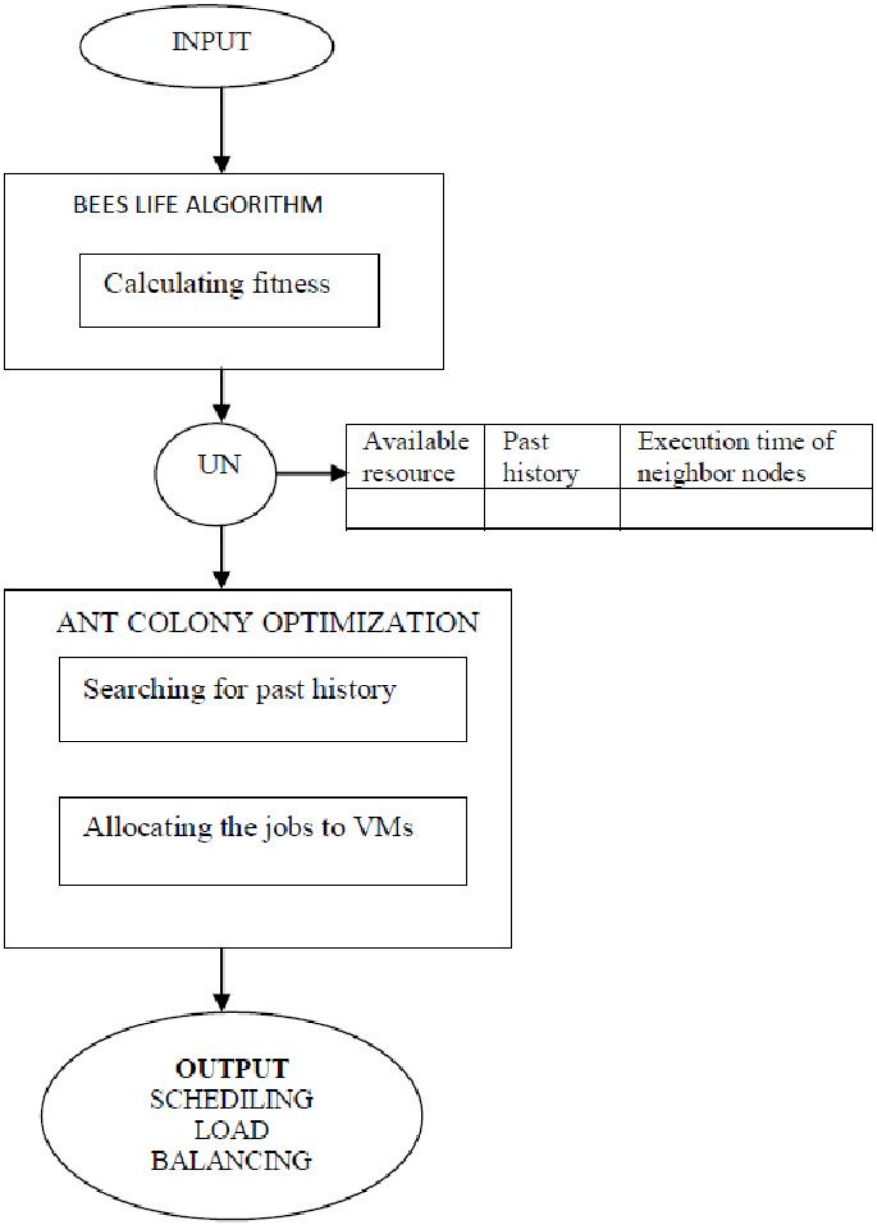

Fig 1: System Architecture

The node is selected in cloud by its performance. If the node fails then another node is selected by which node having priority to become a master node. Thus improves the performance of scheduling than other algorithm.

Algorithm

```
// pseudo code for Bees life algorithm
Initialize population (N bees) at random
Evaluate fitness of population (fittest bee is the queen, D fittest following bees are drones, W fittest remaining bees are workers)
While stopping criteria are not satisfied (Forming new population)
/* reproduction behaviour */
 Generate N broods by crossover and mutation
 Evaluate fitness of broods
 If the fittest brood is fitter than the queen then replace the queen for the next generation
 Choose D best bees among D fittest following broods and drones of current population (Forming next generation drones)
Choose W best bees among W fittest remaining broods and workers of current population (to ensure food foraging)
/* food foraging behaviour */
Search of food source in W regions by W workers
Recruit bees for each region for neighbourhood search (more bees (FBest) for the best B regions and (FOther) for remaining regions)
Select the fittest bee from each region
Evaluate fitness of population (fittest bee is the queen, D fittest following bees are drones, W fittest remaining bees are workers)
End while
```

```
//pseudo code for updation node
If node has high priority than other node
Then
Choose the highest priority node as master node
Priority_option
Execution time storage capacity and performance level
If master node fails
Then select an alternative node which has next priority to become a master
If the other slave nodes fails
Then
Use a Gossip protocol to get a alive message from all nodes
Which node gives alive message
Then
Select all alive paths and find the shortest path by ant colony optimization algorithm
```

```
// pseudo code for ant colony optimization
Initialize parameters
Initialize pheromone trails
Create Ants
While stopping criteria is not reached do
Let all ants construct their solution
Update pheromone trails
Allow Daemon actions
End while
```

ACO is framed from the optimal search capability of an ant colony that searches for a common source of food. It has been observed that ants are able to find the shortest path to such a source under certain circumstances by indirect communication. This communication is done by the so-called pheromone values. The behaviour of ants is put into an algorithmic framework to obtain solutions for a given problem. Solutions are constructed by random moments of artificial ants on a so-called construction graph, which has weights of the pheromone values on the edges. Larger pheromone values leads to higher probability of the edges being traversed in the next walk. In addition, the random walk is usually influenced by heuristic information about the problem.

IV. PERFORMANCE ANALYSIS

For better performance comparison we have formulated several test cases for bee's algorithm. Each test case can be formed based on the range of execution time. Test case can be assigned as data centres for cloud allocation. For ex. If job execution time is 150 then it can be allocated to T3 data centres.

S.NO	TESTCASE	RANGE(ms)
1	T1	0-50
2	T2	51-100
3	T3	101-150
4	T4	151-200
5	T5	201-250
6	T6	251-300
7	T7	301-350
8	T8	351-400
9	T9	401-450
10	T10	451-500
11	T11	501-550
12	T12	551-600
13	T13	601-650
14	T14	651-700
15	T15	701-750

Table 1: Assuming Range Value to Test Case

Assuming range values to test cases from 0 to 750. Each test case have range values of 50. Here only 15 test cases are assigned for execution.

PROCESS NO	EXECUTION TIME(ms)	WAITING TIME(ms)	TOTAL EXECUTION TIME(ms)
1	25	0	25
2	30	25	55
3	35	55	90
4	15	90	105
5	10	105	115

Table 2 : Performance Analysis Under Test case 1

PROCESS NO	EXECUTION TIME(ms)	WAITING TIME(ms)	TOTAL EXECUTION TIME(ms)
1	63	0	63
2	51	63	114
3	70	114	184
4	98	184	282
5	66	282	348

Table 3: Performance Analysis Under Test case 2

Calculate performance analysis under test case up to 15 test cases. To calculate total execution time for each task. Let us consider T1, T2…T15 are tasks to be scheduled in cloud.

PROCESS NO	EXECUTION TIME (ms)	CLOUD DATA CENTERS	WAITING TIME (ms)	TOTAL EXECUTION TIME (ms)
1	138	T3	0	138
2	154	T4	0	154
3	68	T2	0	68
4	140	T3	138	278
5	350	T7	0	350
6	84	T2	68	152
7	580	T12	0	580
8	170	T4	154	324
9	46	T1	0	46
10	25	T1	46	71
11	236	T5	0	236
12	320	T7	350	670
13	423	T9	0	423
14	603	T13	0	603
15	443	T9	423	866

Table 4: Calculation of Total Execution Time For Each Task

After allocating job to particular test case, then we need to allocate job to VMs by selecting shortest path using Ant algorithm. So, we calculating available resource table for knowing the machines power consumption, resource availability and memory availability

NODE ID	POWER CONSUMPTION (W)	RESOURCE AVAILABILITY (GHZ)	MEMORY AVAILABILITY (GB)
12231	40	3	3
11234	58	3.5	1
11232	50	1.2	4
11232	65	2	6
11231	6	6	8
12421	8	5	10
13312	5	2.8	8

Table 5: Ant Colony Available Resource Table

From above table, the node id is given to ant system. And cost can be assigned between two nodes.

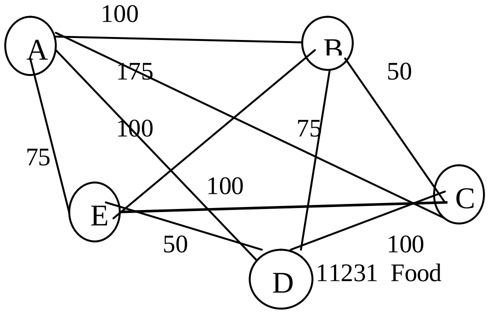

Fig 2: Ant System for Shortest path allocation

PATH	DISTANCE
A-D	100
A-B-D	175
A-B-C-D	250
A-E-D	125
A-C-D	275
A-B-E-D	275
A-E-C-D	300
A-B-E-C-D	450

Table 6: Calculating Shortest Path Distance

By comparing table 5 and table 6, we can take the maximum available memory in node 11231. And from above diagram assume nest is A and food D, calculating shortest path to reduce the job allocation time. Table 6 shows all available `paths from source to destination. From the distance calculation we found that path A-D direct connection will be a shortest path than other path.

V. COMPARISION AND RESULTS

The hybrid AB colony results can be compared with Max-Min algorithm, thus shows execution time and job allocation time are reduced.

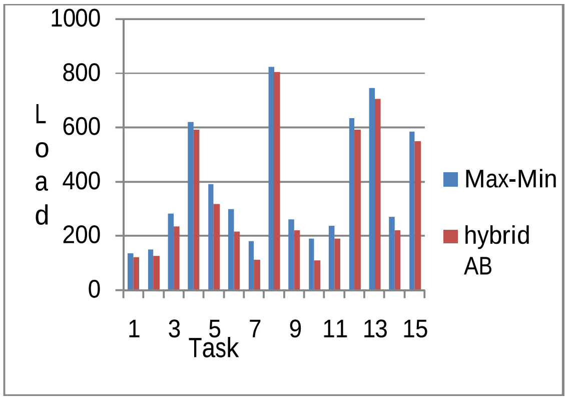

Fig 3: Comparison result of Max-Min and Hybrid AB colony algorithm

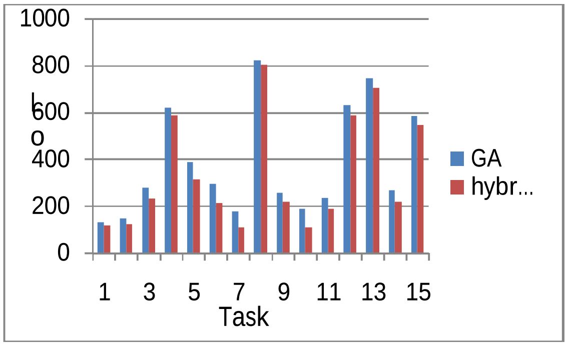

Fig 4: Comparison result of GA and Hybrid AB colony algorithm

VI. CONCLUSION

Evaluation is to combine VMs with different resource characteristics appropriately so that the capacities of servers are well utilized. The system multiplexes virtual to physical resources adaptively based on the changing demand. This system achieves overload avoidance while the number of resources are increased, thus the resources utilization are balanced for systems with multi resource constraints using hybrid AB colony algorithm.

REFERENCES

[1] Aarti Khetan ,Vivek Bhushan ,Subhash Chand Gupta, " A Novel Survey On Load Balancing In Cloud Computing", International Journal of Engineering Research & Technology (IJERT) ,ISSN: 2278-0181, Vol. 2 Issue 2, February- 2013

[2] Ali M Alakeel, "A Guide To Dynamic Load Balancing In Distributed Computer Systems", International Journal of Computer Science and Network Security, Vol. 10 No. 6, June 2010

[3] Alonso-Calvo. R, J. Crespo, M. Garcia-Remesal, A. Anguita, and V. Maojo, "On distributing load in cloud computing: A real application for very-large image datasets," Procedia Computer Science, Elsevier, 1, pp. 2669-2677, 2010.

[4] Furht. B, and A. Escalante, "Handbook of cloud computing," Cloud computing fondamentals chapter writen by B. Furht, Springer, 2010.

[5] Jing Liu1, Xing-Guo Luo2, Xing-Ming Zhang3, Fan Zhang4 and Bai-Nan Li5**,** "Job Scheduling Model for Cloud Computing Based on Multi- Objective Genetic Algorithm", IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 3, January 2013

[6] Linan Zhu1, Qingshui Li2， and Lingna He3, **"**Study on Cloud Computing Resource Scheduling Strategy Based on the Ant Colony Optimization Algorithm", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No. 2, September 2012

[7] M. Dorigo & T. Stützle, Ant Colony Optimization, MIT Press, 2004

[8] Pinal Salot, " A Survey Of Various Scheduling Algorithm In Cloud Computing Environment", ISSN: 2319 - 1163 Volume: 2 Issue: 2 131 – 135, IJRET | FEB 2013

[9] Tasquia Mizan, 2 Shah Murtaza Rashid Al Masud, 3Rohaya Latip, " Modified Bees Life Algorithm for Job Scheduling in Hybrid Cloud," International Journal of Engineering and Technology Volume 2 No. 6, June, 2012

[10] Yuvaraj Kumar, K. Govinda, "Resource Optimization for cloud environment using Fuzzy Bee Colony Technique",IJCA, ISSUE 2,vol , ISSN:2250-1797, Aug 2012