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. 3, Issue 12, December 2015

Load Balancing in Cloud Computing using Ant Colony Optimization

Harshada Raut[1], Kumud Wasnik[2]

[1]M.Tech. Student, Dept. of Computer Science and Tech., UMIT, S.N.D.T. Women's University, India

[2]Professor, UMIT, S.N.D.T. Women's University, Mumbai, Maharashtra, India

ABSTRACT: Cloud computing is sharing various computing resources rather than having local servers or personal devices to handle applications over internet. There is lots of data stored on cloud and multiple requests come for resources, due to this increase load on data servers and sometimes unable to provide resources on time. Therefore, to manage the load by scheduling task appropriately researchers have introduced some load balancing techniques that can improve the performance of cloud computing along with optimal resource utilization. The proposed load balancing technique is based on the ant colony optimization (ACO) which detects the overloaded nodes on data center and distributes it by taking threshold value. It distributes the load to the other nodes by finding the nearest underloaded node using travelling salesman problem(TSP). The proposed technique ensures availability, achieves efficient resource utilization, maximizes number of requests handled by cloud and minimizes time required to serve multiple requests. It reduces server's waiting time to respond request by storing data in temporary memory.

KEYWORDS: ACO, Cloud Computing, Load Balancing, TSP, Dynamic load balancing.

I. Introduction

Recently cloud computing has become most popular field for academic research area. Cloud computing is a central remote servers where a programs or applications run on server machine. Cloud computing is a system where services like IaaS (Infrastructure is provided as a Service), PaaS (Platform is provided as a Service), SaaS (Software is provided as a Service) are provided online using Internet Connection. These services are provided anywhere, anytime on the basis of "only pay for what you use". Cloud may be a Public, Private or Hybrid. Cloud Computing consists of Virtual machine, Host, Servers. User need not to worry about maintenance of any software, it is the responsibility of cloud service provider to provide the needed resources to user.

The cloud appears to be a single point of access for all the computing needs of consumers. It is difficult to manually assign tasks to computing resources in clouds because hundreds of thousands of virtual machines (VMs) are used[1] .Service provider should provide all cloud services to user efficiently ,but sometimes due to too many requests from users may get machine overloaded, In results it unable to responds to user and performance falls down.

In cloud computing load balancing is a major issue. If there are multiple requests at a same time from different users for resource utilization then it create a long queue on server and increases a response time and to avoid this, divide the task among all available machines and balance a system. If resources are less compared to requests then it may delay the response to clients and server may not be able to fulfil all demands of clients.

To avoid this, a good load balancing algorithm is required, that will help to provide resources to all users, use all machines and make sure there should not be under loading or overloading machines.

II. Related work

Ant colony optimization (ACO) algorithm is proposed by Marco Dorigo and his colleagues in 1992. It is inspired from real ants, when searching for a food ant travels randomly and in return trip they deposited some chemical pheromone. On the quantity of this pheromone other ants uses shortest path on which more pheromone value is deposited. Indirect communication between the ants via pheromone trails enables them to find shortest paths between their nest and food sources. In [2] Proposed technique is based on the ACO where redistribution of overloaded nodes is done based on the threshold value. If the load on current node is less than the threshold ant will then search for overloaded node among the neighboring nodes of the current node and move to the underloaded node by checking its Foraging Pheromone value. Here ants move only in one direction at a time. In [3] enhancement to ACO algorithm is

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. 3, Issue 12, December 2015

proposed. Artificial ants moves forward and backward direction to find the overloaded node and update value in pheromone table. Throughput, Response time, less energy consumption is achieved, but it gives less performance.

III. PROPOSED METHOD

There are various nodes in the cloud computing system. Each node has different capability to execute the requests received from user. Sometimes it is depends on the computer's processing power as how much time it can take to execute the task. Load balancing is done so that every virtual machine in the cloud system does the same amount of work throughout therefore increasing the throughput and minimizing the response time[4].

In our proposed work, we used Ant colony optimization algorithm with travelling salesman problem. TSP is one of the well-known and extensively studied problems in discrete or combinational optimization and asks for the shortest roundtrip of minimal total cost visiting each given node [5]. When server gets overloaded this method gives the solution for finding the minimum distance from one node to neighboring nodes. Here, we used temporary memory to store the previous executed requests; it reduces the time if the same request has already been executed on the same node. Fig.1 shows the architecture of proposed method.

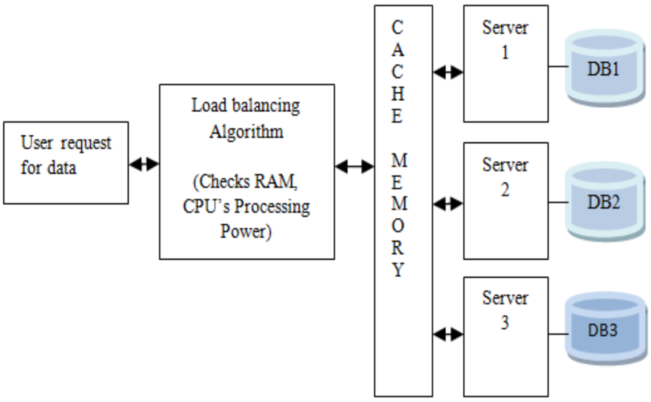

Fig.1. System architecture

Pheromone Updation

The pheromone updating formula is the change in amount of pheromone value because of incoming and outgoing number of ants includes the chemical pheromone every time they visited to the node. Its value is proportional to the quality of the solution built by the ants. Two moments of ants are Foraging Pheromone and Trailing Pheromone.

A. Foraging Pheromone:

The ants continuously move in the forward direction in the network encountering overloaded node or under loaded node. Equation to update value in foraging pheromone is,

$$FP(t+1) = (1 - \beta eva)\ FP(t) + \Delta FP$$

Where,
βeva = Pheromone evaporation rate
FP = Foraging pheromone of the edge before the move
FP(t+1) = Foraging pheromone of the edge after the move
ΔFP = Change in FP

B. Trailing Pheromone:

If an ant encounters an overloaded node in its movement then it will move back to the under loaded node which was previously encountered. It will check if the node is still under loaded or not and if it finds it still under loaded then it will redistribute the work to the under loaded node. The vice-versa is also feasible and possible.

Equation to update value in trailing pheromone is,

$$TP(t+1) = (1 - \beta eva)\ TP(t) + \Delta TP$$

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. 3, Issue 12, December 2015

Where,
βeva = Pheromone evaporation rate
TP = Trailing pheromone of the edge before the move
TP(t+1) = Trailing pheromone of the edge after the move
ΔTP = Change in TP

We assume ants as a requests and nodes as a data center server. Fig.2 show the flowchart of proposed load balancing algorithm based on ACO.

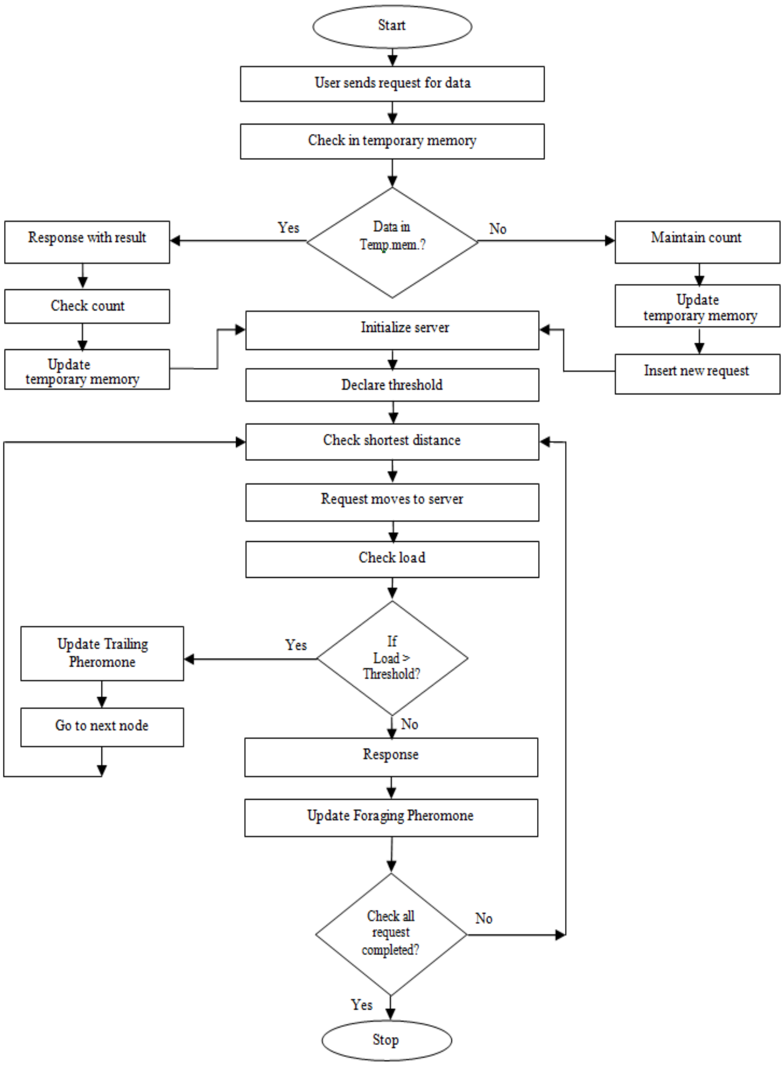

Fig.2. Flowchart of Proposed Algorithm

As shown in proposed algorithm user first request for data. Load balancing algorithm will check whether any data in temporary memory?, Request made by user is already processed? If yes then it will check temporary memory fetch data and check memory counter and update memory data else it will start with new counter update current data in memory and then algorithm will check for the available servers, calculates and declare threshold value for each server as = (number of request/available servers) and check for the shortest distance by sending single request to each server. In results, it will get server which is ready to execute request.

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. 3, Issue 12, December 2015

Request move to the server and check for the load by comparing with servers threshold value. According to overloaded or underloaded server update its Foraging or Trailing Pheromone. At the end algorithm will check whether all request has been processed or not and if yes it will stop the execution.

IV. SIMULATION RESULTS

We used Netbeans 7.0.1 for simulation. In request we retrieve a database query with using Mysql 5.0. We have created virtual servers using VMWare Workstation. Before initiating execution fetch the available RAM to calculate how much load a system can take and how fast it responds to the requests, also Simulation result shows how much RAM is used while performing load balancing. Fig.3 shows result of first request and Fig.4 shows result of second request.

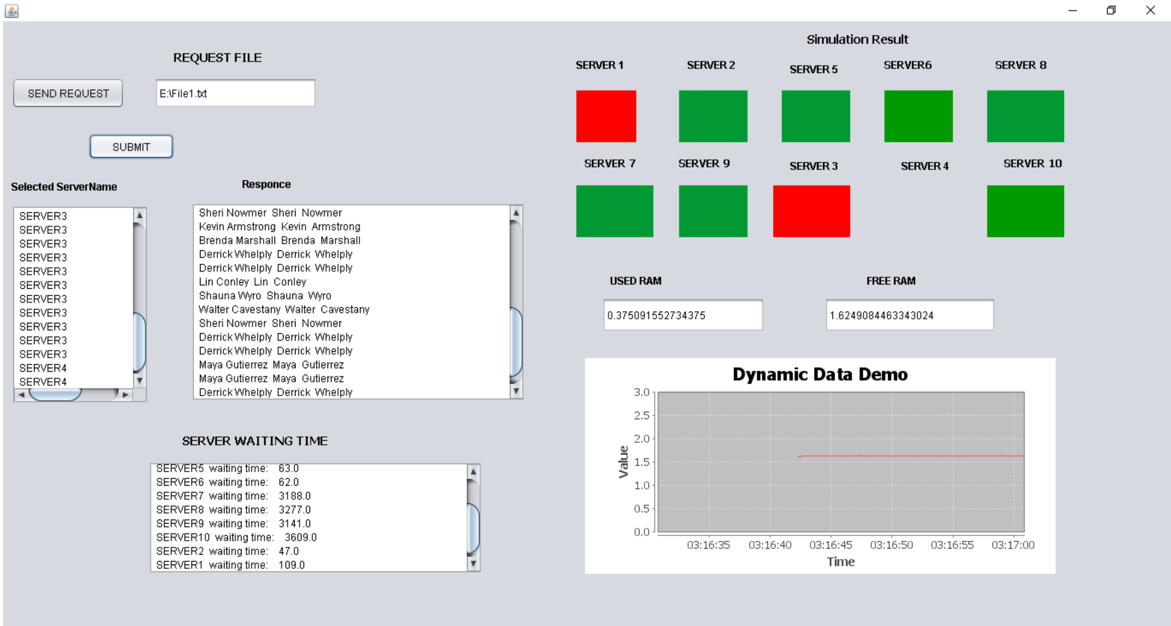

Fig 3: Simulation result (first request)

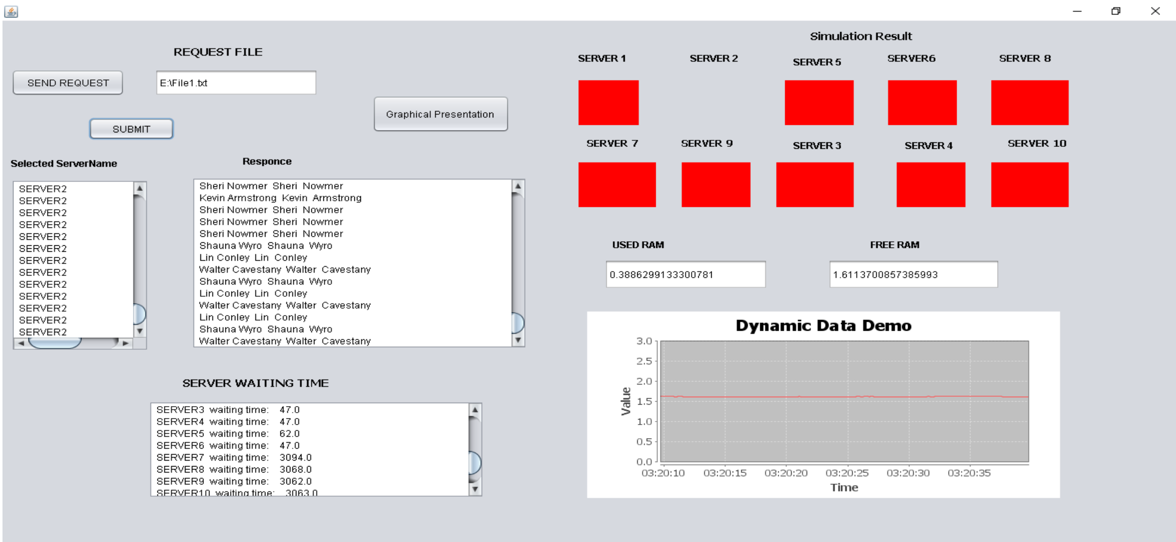

Fig 4: Simulation result (second request)

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. 3, Issue 12, December 2015

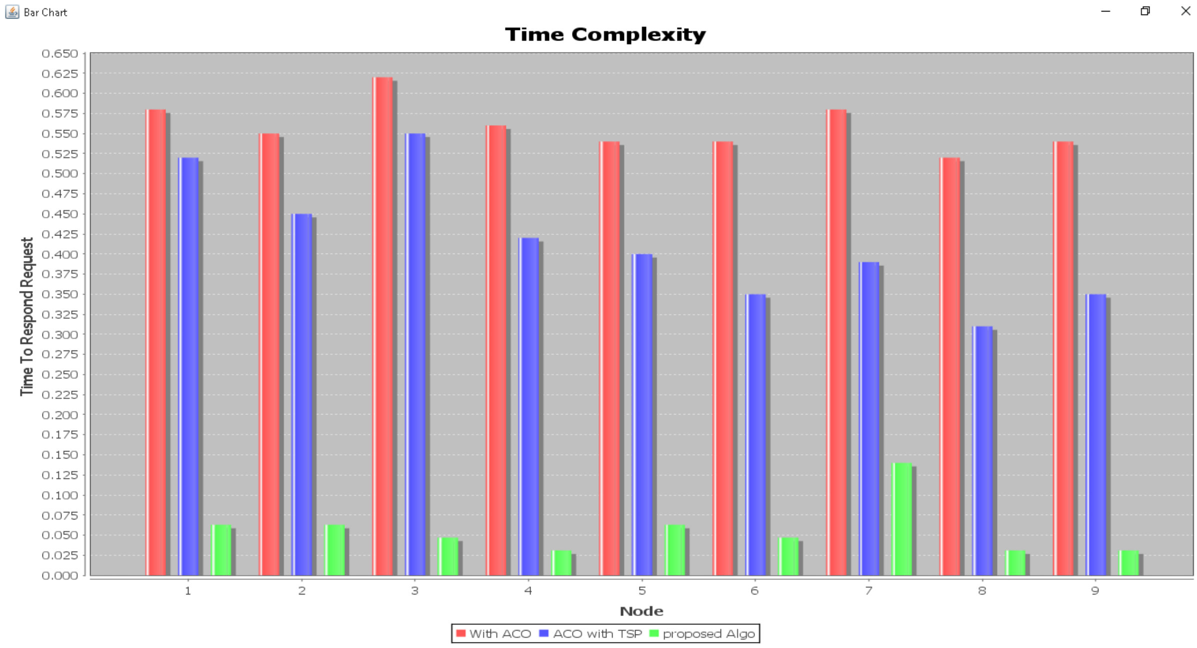

Fig 5: Comparison of waiting time of existing and proposed algorithm (by taking avg. Value)

As shown in simulation result first request will go to server 1, then it will distribute requests to nearest servers by calculating shortest distance among servers. The waiting time for Server 2 is 47ms, Server 3 is 94ms, Server 4 is 78ms, Server 5 is 63ms and so on as shown in Fig3. and Table 1. When we send same request for execution the waiting time is reduce which is shown in Fig 4 and comparison between them is shown in Fig 5 and Table 1.

Server name	Waiting time for 1st request(in ms)	Waiting time for 2nd request(in ms)
Server 1	109	109
Server 2	47	31
Server 3	94	47
Server 4	78	47
Server 5	63	62
Server 6	62	47
Server 7	3188	3094
Server 8	3277	3068
Server 9	3141	3062
Server 10	3609	3063

Table 1: waiting time for servers

V. CONCLUSION

For satisfactory service a user needs flexibility in service, easy availability, less waiting time to get resource. Incoming jobs can be scheduled according to virtual machine's capacity so that it may avoid the unavailability of service if there are multiple requests on single machine and this will get satisfactory service to cloud users. In conclusion, we have shown that our proposed algorithm gives a better response in less time. If the machine's RAM and Processor is good then this algorithm can perform a better load balancing compared to only using ACO. When node is overloaded then it searches for neighboring nodes with minimum distance. If the requests are already in temporary memory then it executes in less waiting time.

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. 3, Issue 12, December 2015

REFERENCES

1. Qiyi, H., Tinglei, H., "An Optimistic Job Scheduling Strategy based on QoS for Cloud Computing", IEEE International Conference on Intelligent Computing and Integrated Systems (ICISS), pp.673-675, 2010.
2. Ekta Gupta, Vidya Deshpande, "A Technique Based on Ant Colony Optimization for Load Balancing in Cloud Data Center", 13th International Conference on Information Technolog, 2014 IEEE.
3. Shagufta Khan, Niresh Sharma, "Effective Scheduling Algorithm for Load balancing (SALB) using Ant Colony Optimization in Cloud Computing", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 2, February 2014.
4. Miss. Neeta S. Nipane, Prof. Nutan M. Dhande, "ABC - LOAD BALANCING TECHNIQUE - IN CLOUD COMPUTING", International Journal of Innovative Research in Advanced Engineering (IJIRAE), Volume 1 Issue 2 (April 2014).
5. Zar Chi Su Su Hlaing, May Aye Khine, "An Ant Colony Optimization Algorithm for Solving Traveling Salesman Problem", International Conference on Information Communication and Management IPCSIT, vol.16 (2011).