Dynamic Load Balancing in Cloud Computing with Ant Colony Optimization using Random Selection

AlokTripathi[1], Dr. DeveshKatiyar[2], Mr. GauravGoel[3]

1-3Dr. Shakuntala Mishra National Rehabilitation University, MohaanRoad, Lucknow

---***---

Abstract-Now a day's population is on increasing and increasing regularly. Population leads increase transportation. On increase of transportation data loads in the Internet are increased, so we introduce new technology i.e. Load Balancing using dynamic Random selection over the cloud. Load Balance is a Process of dividing total workload to different Nodes, the work is divided and machine work efficiently. The aim of this study is to enhance the load balance & overall output of the overloaded and under loaded node with maximum throughput, minimum delay time and increased in execution.

Keywords: Transportation; load balancing; optimization.

I. INTRODUCTION

In this era a huge amount of customers or users, need to compute a large data over it but sometime due to lack of memory management Client cannot perform some operations. So, a storage system are introduce as "Cloud Computing". Cloud Computing can be demonstrated as operation of heavy computing resources such as software package & hardware a kind of services that are delivered to clients over an outsized scale network [1]. Cloud provides some services Like Online Storage (data,files,pictures) and Infrastructure. Cloud computing is very popular technology used in recent time. Cloud computing grave a lots of focused because of opportunities that it offers. It uses as a platform and application. It allows to user for creating, configuring and customizing corporate application online. It offers platform independency it means no requirement of installation in personal computers because of this features it is very helpful in extending business. User says that cloud is a very large database, which are used to store huge amount of information and provide different services to different server. Types of models are used in cloud computing-

i)Deployment Model

ii)Service Model

i) Deployment Model

In this model, it tells about how the cloud is actually locating at remote Position. In four ways we can access the cloud they are as follows-

a) Public Cloud

In this it is easily accessible by all the users such public. However, it is not much secure due to of its openness.

b) Private Cloud

In this, it is only accessible within or under a business enterprise. It is more secure then the Public by a reason its private nature.

c) Hybrid Cloud

It is combination of both public as well as private. In this some activities performed by public cloud and some

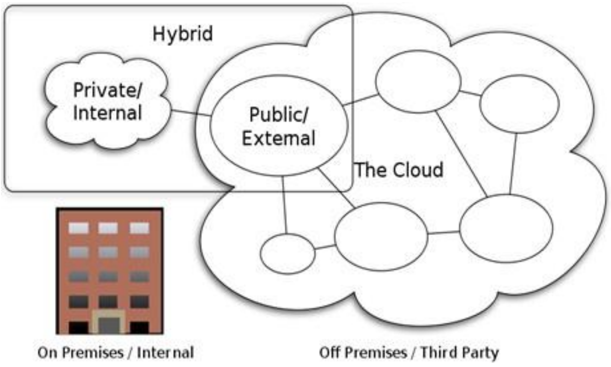

Fig.1.1 Cloud Computing Model [2]

ii) Service Model

It means different types of application provide different servers all across the cloud-

a) Software as a Service-

It allows user to use different type of software to different application. It allows to client for paying according to uses and there is no need of maintenance and updating from user side. Server needs to do all it's updating according to need of customers. It provides cheap cost and only a single application needed instead of buying whole software. Example- Salesforce, Zoho.

b) Platform as a Service

It allows user to build its own application by providing different resources that required to build a software and which run on the provider's infrastructure. It means it provide a platform including all system requirements. Example-Microsoft, Azure.etc.

c) Infrastructure as a Service

It provides users storage and infrastructure network to keep customers make its own platform to deploy software. The top companies that uses this service are Amazon (EC2), Teremark.

1.2 Virtualization

It means we perform task as like real but it happens virtually that is which not exist in real. It is form of machine that implement software and which will also execute different program as a real machine. The work is just like as real machine. Through these different users uses different application or services of the cloud. Types of virtualization are as follows-

a) Full Virtualization
b) Para Virtualization

In full virtualization, the whole machine is installed on the other machine but the virtual machine provides all the function, which work as original machine. In the case if real machine is not free when it busy then virtual machine is ready for performing execution.

In para virtualization that the hardware that allow user can run multiple operating system over a single machine. In this activity user can use system resources in efficient and effective manner.

1.3 Load Balancing

It is a technique of dividing a total workload to different nodes so that work is divided and machine works efficiently. Sometime it happens that at a time particular node is in under loaded condition because of this balancing is help in division of properworkload.

On the behalf of system state, the load balancing algorithm is divided in two parts such as –

a) Static Load Balancing
b) Dynamic Load Balancing.

This procedure is dynamic in spirit when it deems only current state, or current deeds of the module it does not depend upon the former state. Depending upon the who started the activity, load balancing procedure can be break into three types of sender started, receiver started and Symmetrical [6].

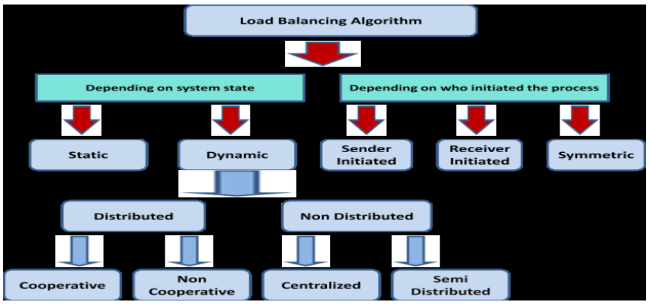

Fig.No.1.2 Types of Virtualization [3]

II. LITERATURE REVIEW

For Load Balance in the cloud there are different methods are used for balancing the load, which is defined as below which help in determining the different parameters, which is, used for balance the overloaded, and under loaded node.

BhawnaMallick [4] work on use of virtual load balancer, which check the status of current virtual machine and give this return id to virtual machine, which are least, loaded. If virtual machine is not found, then it returns the value of -1 then datacenter controller queues up the incoming request.

Nikita Haryani et.al [5] works on counter variable that start checking the variable counter of each server node and data center processor. Load balancing method is count on on least link mechanism and it is belonging to the dynamic organizing procedure. it counts the number of connections for each server dynamically to find the approximate the load.

RenGao et.al [6] works on the two strategies forward-backward technique and max-min protocols so that to find the candidate node and distribution of load according to it. In these pheromones, initialization and pheromones bring to update according to the physical resources. Using master slave architecture as like the first task is presented to the master node after the load distribution is decided by the master node.

Ekta Gupta et.al [7] algorithm is evoked by using ACO (ant colony optimization) that ant work togetherin foraging character of ant. The intensity of pheromone trail is varying on different factors such as quality of food sources; calculate distance of food from source etc. So, according to this pheromone updates the ant decides the next node. In this, the data center server is known as node. The Foraging Pheromone(FP) the ant moving from under loaded hope to overloaded hope, ant will update FP, and in Trailing pheromone(TP) the ant moving from overloaded to under loaded hope, ant will update TP.

III. RESEARCH OBJECTIVE AND RESEARCH METHODLOGY

OBJECTIVE

The main motive of Paper is to enhance load balance and overall performance of the under loaded node or overloaded node with maximum throughput, minimum delay time, and increased in execution time. The inward request that is coming at server side is totally balanced by the virtual machine.

IV. PROPOSEDWORK

The suggested algorithm whose objective is to balance maximum load with minimum average time and maximize throughput. The main work of this procedure is that it selects a station for every connection along the way by considering the two factors: firstly, channel is not already assigned within one-node neighbors. The target waits few volume of time to gain multiple response from separate paths. The destination estimates each path by electing the station on each link with tiniest vulnerability and avoiding the taking of same channel in the snoop range neighbor's. So, we are saying that the objective of this procedure is to balance the maximum overloaded node without any interferences of node.

This is happening because of calculating the average value of different types of parameters like as (delay time, execution time, throughput, energy consumption) by using random selection sequence method. The parameters have assign some maximum value and compare to that value upto its maximum limit, and calculate the final average value of all parameters and update to the routing tables after every calculation.

Proposed Ant Colony Optimization using Random Sequence Algorithm

Each node initiates Forward ant at regular time intervals, based on random sequence number.

Ants search a way to destination randomly based on current routing tables, route discovery phase.

The ant creates a stack, which is in forward position, find the trip time for each and every node which is reached.

If node is at target spot, then the stack congenital by the backward ant.

The backward ant pop the stack entries and inspects the path in opposite.

The routing table of each stayed node brings to update based on trip times.

Algorithm

```
Initialization Routing Table of each node with equal prospect.
Set time: Current time (C) & time interval (i1) between ant's node generations.
Start with main loop while C< = C max do
 If ((mod i1) = = 0) do select Destination Node launch forward ant
end if
for each forward ant do
while (current node != destination node) do
if pheromone info available do
$t_d \leq t_{\max}, t_e \leq t_{\max}, t_t \leq t_{\max}, t_{ec} \leq t_{\max}$
a<- select next hop else
a<- broadcast ants & select node which is first to reply back
End if
SK <- SK U a
End while
Launch backward ant Pass stack info and die End for
For each backward ant do
While (current node! = destination node) do Choose next hope by popping the stack Update traffic model
Update routingtable
Update Random Sequence Number End while
End for
```

Now, we take different parameters and calculate value of each parameter and on basis of these values, the node is selected for the execution.

The calculation of parameter are as given below-

t_d=delay time, t_e =execution time, t_h=throughput time, t_{ec}=energy consumption

Now physical resources of virtual machine is being calculateDelay time:

$$\tau_d = t_d \div t_{\max} \times 100\%$$

Execution time:

$$\tau_e = t_e \div t_{\max} \times 100\%$$

Throughput:

$$\tau_t = t_t \div t_{\max} \times 100\%$$

Energy Consumption:

$$\tau_{ec} = t_{ec} \div t_{\max} \times 100\%$$

Now,averagevalue of all these parameters are as follows-

$$\tau_{avg} = \tau_d(O) + \tau_e(O) + \tau_t(O) + \tau_{ec}(O)$$

Where τ is constant to calculate the average value.

After the calculation of these values, it is very simple to specifies that which VM has maximum capacity to handle their request & balancing load.

In the server, demand of services may be increase or decrease.

Path discovery: -

In this proposed approach, two phases are present path discovery phase and path maintenance phase. In Routing protocol, we need to found the way from source(s) to destination (D).

In Path Discovery, each hope is present in network sends "HELLO" message to all possible next node after certain time interval it reaches the correct destination node and each next hope sends "ACK" message. According the total time required to receive "ACK" message, start node calculates the cost on the track of all possible next hop node and stores it into the costtable.

The start node sends "HELLO" message after fixed time interval to check cost table.

Let start time1=t1 (when "HELLO" message is sent) and the adjacent node or neighbor node receives the message at time t2. So total time required to send "HELLO" message is T1=t2-t1.

If start time=t2when"ACK"messageissentbytheadjacent or neighbor node and source node receives the message at time t3. So total time required to send "ACK message is T2=t3-t2.

Total time T= (T1+T2)/2

Update Csn with Txk, where k is the cost per unit time.

If we are having total N no of path over the network, then we will select only N/2 no of shortest path for routing on the basis of cost (Csn).

Construct the ant for each source and generate sequences of ants according to the decision variable(X&) and select the best sequence.

Internal node selection Algorithm

Ideal state

When each of the internal node in ideal state then selection is done by priority scheduling, Sequence Number

Normal state (Balanced Ant Family algorithm)

Assigned the Threshold assessment for eachnode.

Initially the Ant began movement from the given model i.e. random waypoint mobility.

Calculate the pheromone value to the nodes based on the processing time of theserver.

L = time of low processing

H=time of high processing Life Time (lt) for the ant starts

Outcome set table is arranged by the ant for the nodes.

So according the these work we can control the server side load using these methods.

V. SIMULATION TOOLS AND RESULT

5.1 Stimulation Tool-

.Network Simulator Version 2, widely known as NS2, is an event driven simulation tool that is useful in studying the dynamic nature of communication networks. Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2. NS2 consists of two key languages: **C++ and Object-oriented Tool Command Language (OTcl).** While the C++ defines the internal mechanism (i.e.,a back end) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events (i.e.,a front end).

The C++ and the OTcl are linked together using **Tcl.**. Mapped to a C++ object, variables in the OTcl domains are sometimes referred to as *handles*. Conceptually, a handle (e.g., n as a Node handle) is just a string (e.g.,o10) in the OTcl domain, and does not contain any functionality. Instead, the functionality(e.g., receiving a packet) is defined in the mapped C++ object (e.g., of class Connector). In the OTcl domain, a handle acts as a front end which interacts with users and other OTcl objects.

It may defines its own procedures and variables to facilitate the interaction.. Note that the member procedures and variables in the OTcl domain are called instance procedures (instprocs) and *instance variables* (instvars) respectively. Before proceeding further, the readers are encouraged to learn C++ and OTcl languages and we include it on our previous posts.

NS2 provides a large number of built in C++ objects. It is advisable to use these C++ objects to set up asimulation using a Tcl simulation script. However, advance users may find these objects insufficient. They need to develop their own C++ objects, and use an OTcl configuration interface sto put together these objects.

After simulation, NS2 outputs either text-based or animation-based simulation results. To interpret these results graphically and interactively, tools such as **NAM (Network Animator) and XGraph** are used. To analyse a particular behaviour of the network, users can extract a relevant subset of text-based data and transform it to a more conceivable presentation.

Table-5.1: Configuration Parameters of in NS-2 Simulator

PARAMETERS	VALUES
Operating System	Linux (Ubuntu 12.04)
NS-2 version	NS-2.35
No. of Nodes	30, 40
Max Packet Length	50
Seed	1
Radio propagation model	Propagation/TwoRayGround
Network interface type	Phy/WirelessPhy
Packet Size	1000
Traffic Type	CBR
Simulation Time	100s
Antenna Type	Omni-Antenna
Transmission Range	1200*900 m
Routing Protocol (Proposed)	ACORP
Transmission Energy	0.175

We will compare our given algorithm selection of random node using ACO results with their corresponding results of existing algorithm ACO with comparison of taking 30 and 40 nodes and result as shown below using Network Simulator -

Throughput

In below graph we compared a different number of nodes i.e. (30 nodes and 40nodes) and it shows the total throughput of the node which varying when compare with the previous one. The throughput increases when we compare the graph.

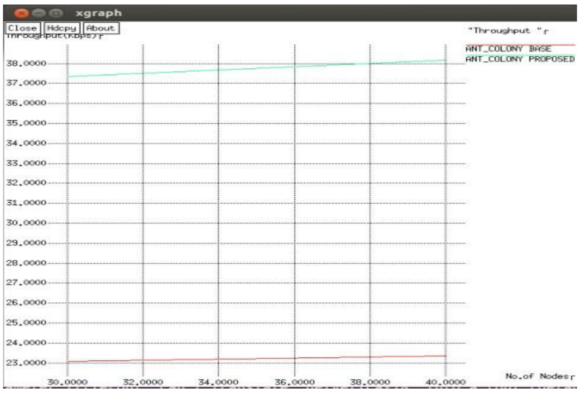

Fig 5.1 Throughput Graph

End Delay

In below graph we compared a different number of nodes i.e. (30 node and 40 nodes) and it display the total end to end delay of the node which varying when compare with the previous one.

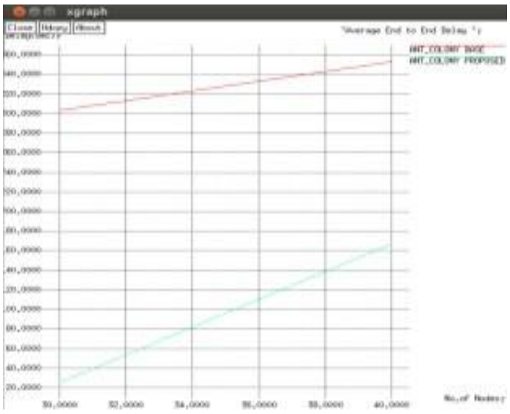

Fig 5.2 End-to-End Delay Graph

Energy Consumption

In below graph we compared a different number of nodes i.e. (30 nodes and 40 nodes) and it display the total energy consumption of the node which little varying when compare with the previous one.

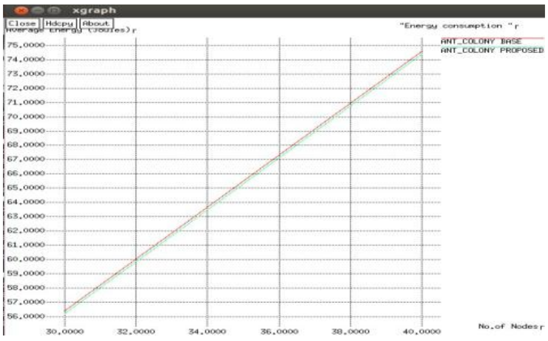

Fig 5.3 Energy consumption Graph

Packet Delivery

In below graph we compared a different number of nodes i.e. (30 nodes and 40 node) and it display the total packet delivery ratio of the node which varying when compare with the previous one.

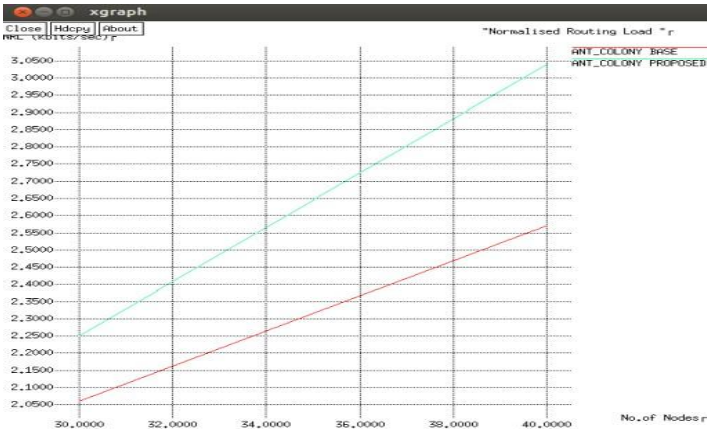

Fig 5.4 Packet delivery ratio Graph

VI. CONCLUSION AND FUTURE SCOPE

ACO technique in load balancing i.e. dynamic in nature. For the given system, it may increase the overall performance. Using this increase in throughput, increase Routing Load and decrease delay time and energy consumption time. Therefore, we can assume that given algorithm is more feasible and effective than the existing system. This system increases the performance and minimizes the load. By the utilization of resources, it increases the throughput.

In the Bhawna Mallick [4] work, they use the Virtual Load Balance i.e. check the current virtual machine status that are least loaded. If virtual machines are found then Load Balancer return1 value, otherwise return -1. Then Data Center Controller queue up the incoming request.

After this, Nikita Haryani [5] works on the Counter Variable for each server node and Data center processor. This algorithm is totally based on the Least Connection Mechanism, which is a component of Dynamic Scheduling Component.

RenGao et.al. [6] It works on two methods forward and Backward Mechanism and Max-Min rules. Therefore, by this method or strategy, he find the candidate node and Distribution of Load according to it. In this, they use Master Slave Architecture. In this Master Node, decide the Load Distribution. In this, pheromone we use to initialization and pheromone bring to update all other resources.

After this, Ekta Gupta et.al. [7] Invoked the ACO (Ant Colony Optimization).

We compare these above techniques with prospered work. Therefore, in future it may handle more load in comparison to proposed one and also with maximum throughput and with minimum energy consumption. In this, we should add more node as compare to suggested algorithm with less complexity.

VII. REFERENCES

[1] https:www.tutorialspoint.com/cloud computing [Last accessed:14-11-19]

[2]Kaur, Rajwinder, and PawanLuthra. "Load balancing in cloud computing.”Second Symposium on Cloud computing. 2012.

[3]Alok Tripathi and Dr, Devesh Katiyar,” A study of Cloud Computing & Load Balancing Algorithms” IJRASET, Vol. 8, Issue IV Apr 2020, www.ijraset.com.

[4]BhawnaMallick and ReenaPanwar ,IEEE,”Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm” 2015International Conference on Green Computing and Internet of Things (ICGCloT).

[5]NikitaHaryani,, DhanammaJagli, “Dynamic Method for Load Balancing in Cloud Computing“, IOSR Journal of Computer Engineering(IOSR-JCE), Volume 16, Issue 4, Ver. IV (Jul - Aug2014),www.iosrjournals.org.

[6]Gao, Ren, and Juebo Wu. "Dynamic Load Balancing Strategy for Cloud Computing with Ant Colony Optimization." Future Internet 7.4 (2015): 465- 483.

[7]Ekta Gupta and VidyaDeshpande. "A Technique Based on Ant Colony Optimization for Load Balancing in Cloud Data Center." Information Technology (ICIT) 2014 International Conference on. IEEE, 201.

VIII AUTHOR’S PROFILE

Mr. Alok Tripathi, He is a Student of MCA Department of Computer Science, Dr. Shakuntala Mishra National Rehabilitation University. Lucknow. Completed BCA from SRMU Deva Road Lucknow in 2017. His Area of Interest is Cloud Computing.

Dr. Devesh Katiyar, He is Assistant Professor at Department of Computer Science, DSMNRU Lucknow Uttar Pradesh, India. His Research Areas are: Software Engineering, Data Mining and Data Warehouse.

Mr. Gaurav Goel, He is Assistant Professor at Department of Computer Science, DSMNRU Lucknow Uttar Pradesh, India. His Research Areas are: Digital Image Processing, Machine Learning, Wireless Sensor Network, and Artificial Intelligence.