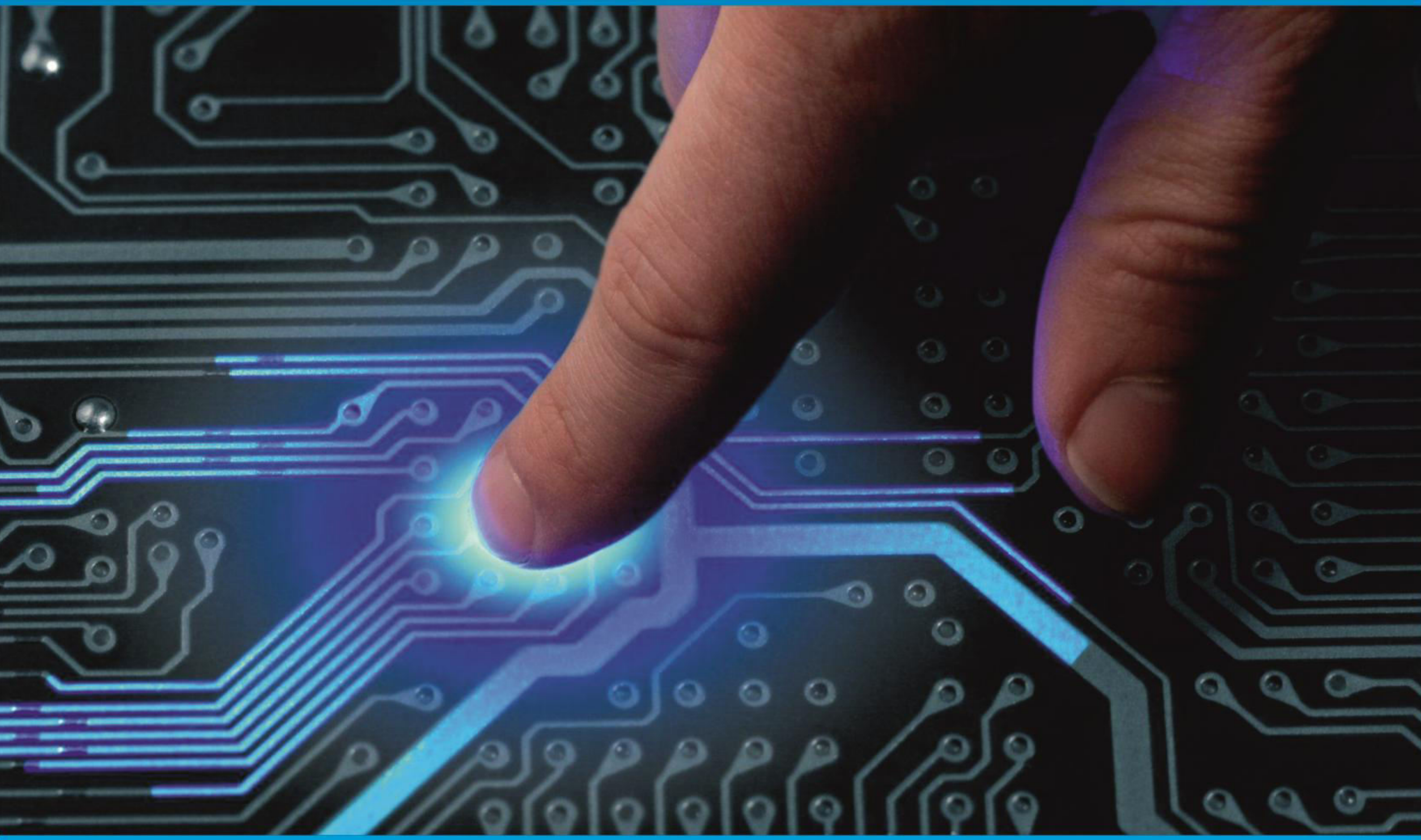
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 5, May 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.488

| DOI: 10.15680/IJIRCCE.2021.0905099 |

Load Balancing in Network using Ant Colony Optimization

Diksha Lohar,Tejas Jadhav, Shraddha Potdar, Raj Hingmire, Prof. Bhagyashali. V. Jadhav

Diploma Student, Department of Computer Engineering, Pimpri Chinchwad Polytechnic, Pune, India

Assistant Professor, Department of Computer Engineering, Pimpri Chinchwad Polytechnic, Pune, India

ABSTRACT: Workload and resource management are two essential functions provided in the service level of a Grid software infrastructure. Consistently, efficient load balancing algorithms are fundamentally important to improve the global throughput of these environments. Although previous works show that, ant colony algorithm works well for load balancing, the cost is a very important factor in this subject. In this paper, a grid load balancing algorithm is proposed by using an ant colony optimization which is able to consider shortest path, type of resource, and running speed of resource. The experimental results show that the proposed algorithm by using this ant colony optimization can reduce the cost of load balancing in comparison with standard algorithm DASUD.

KEYWORDS:Grid computing, Load balancing, Ant colony

I. INTRODUCTION

Cloud Computing is very hot topic in IT field. Many researches are going on Cloud Computing. This is basically "on-demand" service. It means whenever we need for some applications or some software, we demand for it and we immediately get it. We have to pay only that we use. This is the main motto of cloud computing. Our desired application will present in our computer in few moment. Cloud Computing has basically two parts, the First part is of Client Side and the second part is of Server Side. The Client Side requests to the Servers and the Server responds to the Clients. The request from the client firstly goes to the Master Processor of the Server Side. The Master Processor are attached to many Slave Processors, the master processor sends that request to any one of the Slave Processor which have free space. All Processors are busy in their assigned job and non of the Processor get Idle. The process of assigning job from Master processor to the Slave processor and after completion the job, then returning from the Slave processor to the Master processor is just like Ant takes their food and return to their nest. The real ants left out pheromone while travelling. A pheromone is a chemical used for communication. Now we are moving from real ants to artificial ants. The artificial ants have some special characteristics which is not found in real ants, such as they are not completely blind, they have some memory called tabu. Now the artificial ants are used in cloud computing. The cloud computing is composed of three service models, five essential characteristics, and four deployment models.
The three service models are as follows.

1)Software as a Service (SaaA).
2)Platform as a Service (PaaS).
3) Infrastructure as a Service (IaaS).

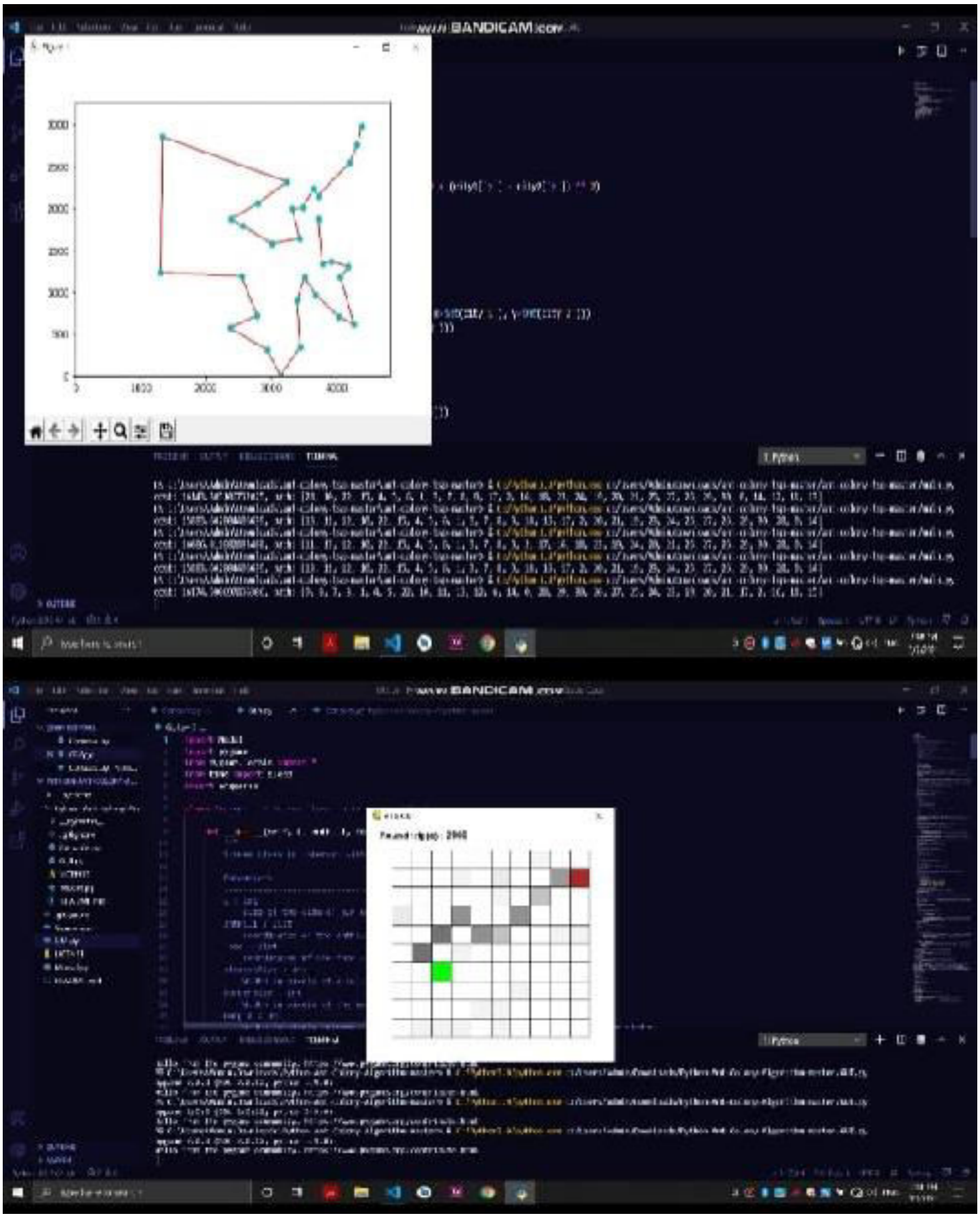
II. METHODOLOGY

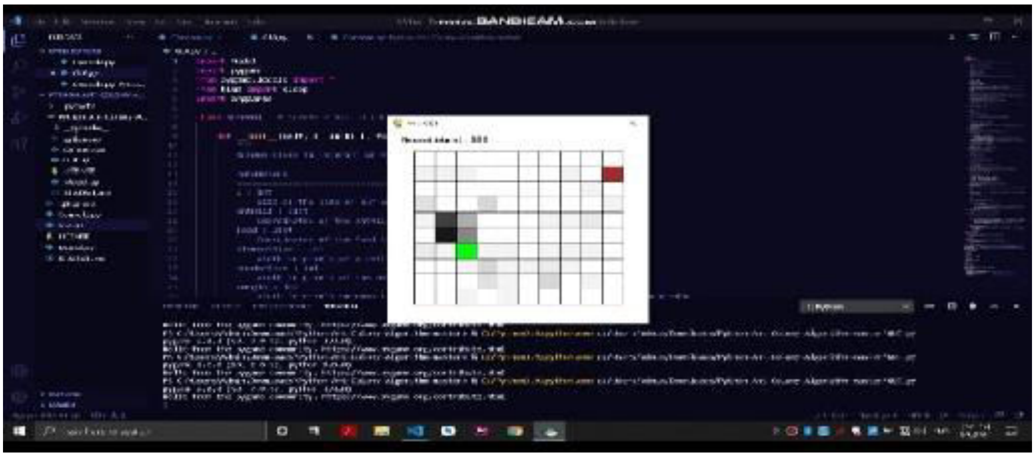
1.2 Cloud Service Models Cloud service delivery is divided into three models.
The three service models are :
1.2.1 Cloud Software as a service (Saas) The capability provided to the consumer is to use the providers applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser. The consumer does not manage the underlying cloud infrastructure.
1.2.2 Cloud Platform as a Service (Paas) The capability provided to the consumer is to deploy onto the cloud infrastructure consumer created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure, but has control over the deployed applications and possibly application hosting environment configurations.

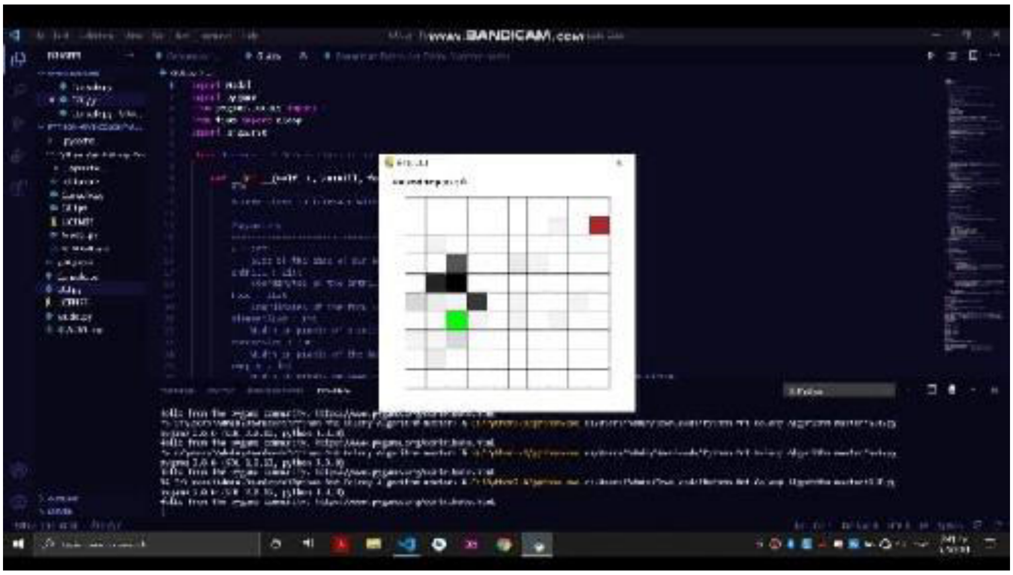
| DOI: 10.15680/IJIRCCE.2021.0905099 |

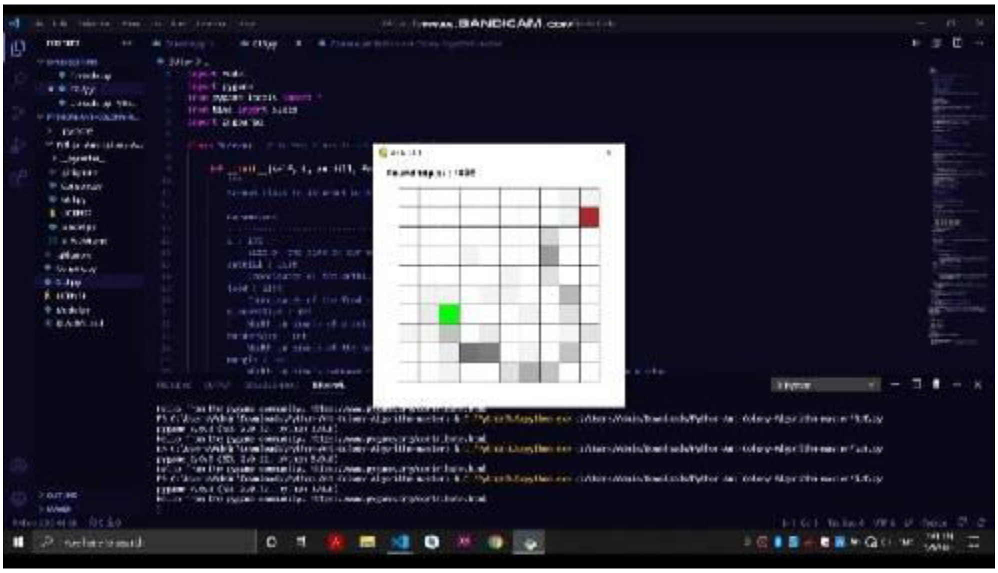
1.2.3 Cloud Infrastructure as a Service (Iaas) The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage.

III. MODELING AND ANALYSIS









Figures: Some Screenshots of the website under development.

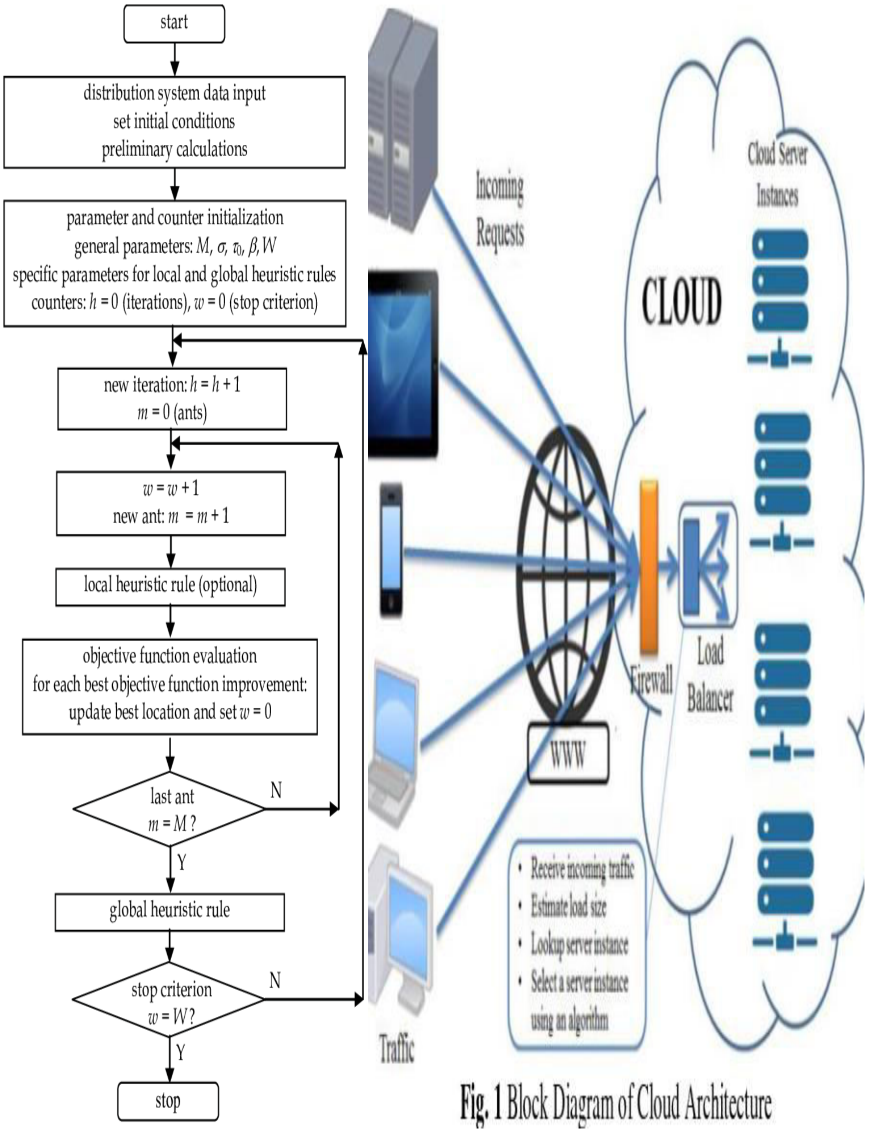

start
distribution system data input
set initial conditions
preliminary calculations
parameter and counter initialization
general parameters: $M, \sigma, \tau_0, \beta, W$
specific parameters for local and global heuristic rules
counters: $h = 0$ (iterations), $w = 0$ (stop criterion)
new iteration: $h = h + 1$
$m = 0$ (ants)
$w = w + 1$
new ant: $m = m + 1$
local heuristic rule (optional)
objective function evaluation
for each best objective function improvement:
update best location and set $w = 0$
last ant
$m = M$?
N
Y
global heuristic rule
stop criterion
$w = W$?
N
Y
stop
Incoming
Requests
CLOUD
Cloud Server
Instances
WWW
Firewall
Load
Balancer
Traffic
• Receive incoming traffic
• Estimate load size
• Lookup server instance
• Select a server instance
using an algorithm
Fig. 1 Block Diagram of Cloud Architecture

Figure: Block Diagram Displaying Internal Working.

Analysis: Working of the website inshort.

To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.
4. To accommodate future modification in the system.
To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.
4. To accommodate future modification in the system.
To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.
4. To accommodate future modification in the system.
To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.

3. To maintain the system stability.
4. To accommodate future modification in the system.
To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.
4. To accommodate future modification in the system.
To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.
4. To accommodate future modification in the system.
To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.
4. To accommodate future modification in the system.
To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.
4. To accommodate future modification in the system.
To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.
4. To accommodate future modification in the system.
To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.
4. To accommodate future modification in the system.
To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.
4. To accommodate future modification in the system.
To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.
4. To accommodate future modification in the system.
To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.
4. To accommodate future modification in the system.
To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.
4. To accommodate future modification in the system.
To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.
4. To accommodate future modification in the system.
To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.
4. To accommodate future modification in the system.
To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.
4. To accommodate future modification in the system.
To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.

4. To accommodate future modification in the system.
1. To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.
4. To accommodate future modification in the system.
5. Sharing a computer system among multiple users
6. Isolating users from each other and from the control program
7. Emulating hardware on another machine

IV. RESULTS

Our objective for this paper is to develop an effective load balancing algorithm using Ant colony optimization technique to maximize or minimize different performance parameters like CPU load, Memory capacity, Delay or network load for the clouds of different sizes. In this paper, a heuristic algorithm based on ant colony optimization has been proposed to initiate the service load distribution under cloud computing architecture. The pheromone update mechanism has been proved as a efficient and effective tool to balance the load. This modification supports to minimize the make span of the cloud computing based services and portability of servicing the request also has beenconverged using the ant colony optimization technique. This technique does not consider the fault tolerance issues. Researchers can proceed to include the fault tolerance issues in their future researches

V. CONCLUSIONANDFUTUREWORK

As we have seen that that present telecommunication networks suffer from network congestion which causes calls, put on the network , to fail . Better load sharing and routing algorithms are required to minimize the effect of congestion. In this thesis, we have simulated two algorithms ,ant colony algorithm and dijkstra algorithm for load balancing in telecommunication network. Performance of both algorithms is studied with respect to the average number of hops taken to complete a fixed no of calls in the network as shown in the graph .we found out that the average no of hops are less in case of ant colony based method as compared to the dijkstra. Hence ant colony optimization technique performs better that fixed-shortest path route algorithms.

REFERENCES

[1] Marco Dorigo, Luca Maria Gambardella " Ant Colonies for the travelling Salesman Problem", TR/IRIDIA, Vol.3, University Libre de Bruxelles, Belgium, 1996.
[2] Zehua Zhang and Xuejie Zhang "A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation", International Conference on Industrial Mechatronics and Automation, pp-240-243,2010.
[3] SarayutNonsiri and SiripornSupratid "Modifying Ant Colony Optimization", IEEE Conference on Soft Computing in Industrial Application, Muroran, Japan. Pp. 95-100. 2008.
[4] PatompornPremprayoon and ParamoteWardkein "Topological Communication Network Design Using Ant Colony Optimization", Department of telecommunication Engineering, King Mongkut's Institute of Technology LandkrabangBankok, Thailand. Pp. 1147-1151.
[5] Kun Li, GaochaoXu, Guangyu Zhao, Yushuang Dong and Dan Wang "Cloud Task scheduling based on Load Balancing Ant Colony Optimization", Jilin University, ChangChun, China, Sixth Annual ChinaGrid Conference. pp. 03-09. 2011.
[6] Shu-Ching Wang, Kuo-Qin Yan, Wen-Pin Liao and Shun-Sheng Wang " Towards a Load Balancing in a Three-Level Cloud Computing Network", Chaoyang University of Technology, Taiwan, R.O.C. pp. 108-113. 2010.
[7] ZenonChaczko, VenkateshMahadevan, ShahrzadAslanzadeh and Christopher Mcdermid " Availability and Load Balancing in Cloud Computing", International Conference on Computer and Software Modelling, IPCSIT, vol. 14, Singapore. pp. 134-140. 2011. [8] Ratan Mishra and AnantJaiswal" Ant Colony Optimization: A Solution of Load balancing in Cloud" International Journal of Web and Semantic Technology. Vol. 3, No. 2. pp. 33-50. April 2012.
[9] Kumar Nishant, Pratik Sharma, Vishal Krishna, Chhavi Gupta, KuwarPratap Singh, Nitin and Ravi Rastogi " Load Balancing of Nodes in Cloud Using Ant Colony Optimization", Department of CSE and ICT, Jaypee University of Information Technology, 14th International Conference on Modelling and Simulation. pp.03-08. 2012.
[10] "Swarm Intelligence from Natural to Artificial System" by Marco Dorigo and Eric Bonabeau, 1999.
[11] "Cloud Computing and SOA Convergence in Your Enterprise" by David s. Linthicum, 2011.
[12] "Cloud Computing Web Based Application" by Michael Miller, 2012.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING