International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, Issue 3, ISSN No. 2455-2143, Pages 76-79
Published Online December – January. 2016 in IJEAST (http://www.ijeast.com)

ANT COLONY OPTIMIZATION: A SOLUTION OF LOAD BALANCING IN CLOUD

Deepak Mahapatra
Dept. of CS

Gaurav Kumar Saini
Dept. Of CS

Himanshu Goyal
Dept. of CS

Amit Bhati
Dept. of CS

***Abstract* - As the cloud computing is a new style of computing over internet. It has many advantages along with some crucial issues to be resolved in order to improve reliability of cloud environment. These issues are related with the load management, fault tolerance and different security issues in cloud environment. In this paper the main concern is load balancing in cloud computing. The load can be CPU load, memory capacity, delay or network load. Load balancing is the process of distributing the load among various nodes of a distributed system to improve both resource utilization and job response time while also avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work. Load balancing ensures that all the processor in the system or every node in the network does approximately the equal amount of work at any instant of time. Many methods to resolve this problem has been came into existence like Particle Swarm Optimization, hash method, genetic algorithms and several scheduling based algorithms are there. In this paper we are proposing a method based on Ant Colony optimization to resolve the problem of load balancing in cloud environment**

Keywords - *Cloud computing, Load balance, Ant colony optimization, Swarm intelligence*

I. INTRODUCTION

1.1 Cloud Computing

Cloud computing has become very popular in recent years as it offers greater flexibility and availability of computing resources at very low cost. The major concern for agencies and organizations considering moving the applications to public cloud computing environments is the emergence of cloud computing facilities to have far-reaching effects on the systems and networks of the organizations. Many of the features that make cloud computing attractive, however, can also be at odds with traditional security models and controls. As with any emerging information technology area, cloud computing should be approached carefully with due consideration to the sensitivity of data. Planning helps to ensure that the computing environment is as secure as possible and is in compliance with all relevant organizational policies and that data privacy is maintained. It also helps to ensure that the agency derives full benefit from information

.

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 provider's 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.

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 or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components.

International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, Issue 3, ISSN No. 2455-2143, Pages 76-79
Published Online December – January. 2016 in IJEAST (http://www.ijeast.com)

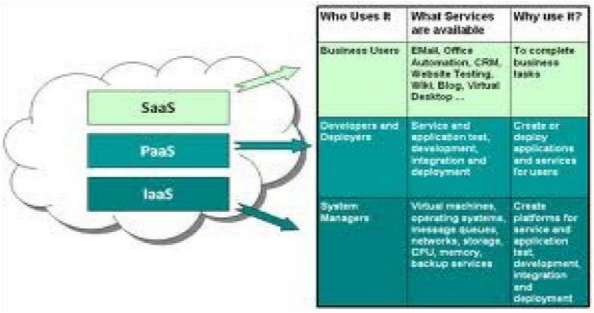

Figure 1

1.3Virtualization

It is a very useful concept in context of cloud systems. Virtualization means "something which isn't real", but gives all the facilities of a real. It is the software implementation of a computer which will execute different programs like a real machine. Virtualization is related to cloud, because using virtualization an end user can use different services of a cloud. The remote data center will provide different services in a fully or partial virtualized manner.

Two types of virtualization are found in case of clouds:

- Full virtualization
- Para virtualization

1.4 Load balancing in cloud computing

Load Balancing is a method to distribute workload across one or more servers, network interfaces, hard drives, or other computing resources. Typical datacenter implementations rely on large, powerful (and expensive) computing hardware and network infrastructure, which are subject to the usual risks associated with any physical device, including hardware failure, power and/or network interruptions, and resource limitations in times of high demand.

Load balancing is used to make sure that none of your existing resources are idle while others are being utilized. To balance load distribution, you can migrate the load from the *source nodes* (which have surplus workload) to the comparatively lightly loaded *destination nodes*.

When you apply load balancing during runtime, it is called *dynamic load balancing* — this can be realized both in a direct or iterative manner according to the execution node selection:

- In the iterative methods, the final destination node is determined through several iteration steps.
- In the direct methods, the final destination node is selected in one step.

Another kind of Load Balancing method can be used i.e. the Randomized Hydrodynamic Load Balancing method, a hybrid method that takes advantage of both direct and iterative methods.

1.4.1 Types of Load balancing algorithms

Depending on who initiated the process, load balancing algorithms can be of three categories as given in [15]:
Sender Initiated: If the load balancing algorithm is initialized by the sender.
Receiver Initiated: If the load balancing algorithm is initiated by the receiver.
Symmetric: It is the combination of both sender initiated and receiver initiated.
Depending on the current state of the system, load balancing algorithms can be divided into 2 categories as given in [15]:
Static: It does not depend on the current state of the system. Prior knowledge of the system is needed.
Dynamic: Decisions on load balancing are based on current state of the system. No prior knowledge is needed. So it is better than static approach. Here we will discuss on various dynamic load balancing algorithms for the clouds of different sizes.

1.5 Ant Colony Optimization

Individual ants are behaviorally much unsophisticated insects. They have a very limited memory and exhibit individual behavior that appears to have a large random component. Acting as a collective however, ants manage to perform a variety of complicated tasks with great reliability and consistency.

Although this is essentially self- organization rather than learning, ants have to cope with a phenomenon that looks very much like overtraining in reinforcement learning techniques.

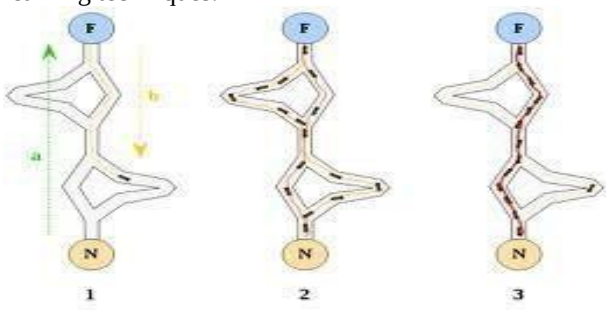

Figure 2

1. Proposed Work

Ant based control system was designed to solve the load balancing in cloud environment. Each node in the network was configured with:

1) Capacity that accommodates a certain.
2) Probability of being a destination.
3) Pheromone (or probabilistic routing) table.

International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, Issue 3, ISSN No. 2455-2143, Pages 76-79
Published Online December – January. 2016 in IJEAST (http://www.ijeast.com)

Each row in the pheromone table represents the routing preference for each destination, and each column represents the probability of choosing a neighbor as the next hop. Ants are launched from a node with a random destination.

In this approach, incoming ants update the entries the pheromone table of a node. For instance, an ant traveling from (source) to (destination) will update the corresponding entry in the pheromone table in. Consequently, the updated routing information in can only influences the routing ants and calls that have as their destination. However, for asymmetric networks, the costs from to and from to may be different. Hence, In this approach for updating pheromone is only appropriate for routing in symmetric networks.

If an ant is at a choice point when there is no pheromone, it makes a random decision However, when only pheromone from its own colony is present there is a *higher* probability that it will choose the path with the higher concentration of its own pheromone type. In addition, due to repulsion, an ant is *less likely* to prefer paths with (higher concentration of) pheromone from other colonies. Moreover, it is reminded that the degrees of attraction and repulsion are determined by two weighting parameters.

Procedure ACO Optimization

Initialize Variables;
Initialize Pheromone on the trail selected by GJAP;

While (*Value of Timer<T)* ***do***
Ants Construct Solutions;
Xnew=min{$f_{obj}(P_k)|k=1,2,...K$*};*
If Xnew<X
then X=Xnew;
Pheromone Update;

End
End

Fig 8: Pseudo Code of ACO [17]

Adopting the problem-solving paradigm of ACO, this example illustrates the use of two sets of mobile agents (that act as routing packets) for establishing call connections in a circuit- switched network. To establish connections between gateways 1 and 3, the two groups of mobile agents construct, manipulate and consult their own routing tables. In ACO, each group of mobile agents corresponds to a colony of ants, and the routing table of each group corresponds to a pheromone table of each colony. Even though the two groups of mobile agents (MAG1 and MAG2) may have their own routing preferences, they also take into consideration the routing preferences of the other group. (While the routing preferences of ants are recorded in their pheromone tables, the routing preferences of mobile agents are stored in their routing tables).

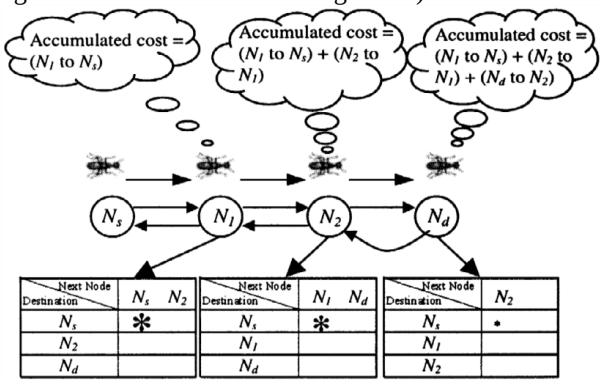

Figure 3

In constructing its routing table, MAG1 (respectively, MAG2) consults the routing table of MAG2 (respectively, MAG1) so as to avoid routing packets to those paths that are highly preferred by the other group. Doing so *increases the likelihood* that two different connections between gateways 1 and 3 may be established. This increases the chance of distributing data traffic between gateways 1 and 3 between the two connections and. In Fig. 16, for the same destination Gateway3, MAG1 is *more likely* to move along whereas MAG2 is *more likely* to move along. By adopting the MACO approach, it may be possible to *reduce the likelihood* that all mobile agents establish connections using *only* the optimal path. If MAG2 selects the optimal path, the idea of repulsion may *increase the probability* that MAG1 will select an alternative to. The advantage of using MACO in circuit-switched routing is that it is *more likely* to establish connections through multiple paths to help balance the load but does not increase the routing overhead. An on-going work implements *some* of the ideas of ACO (first proposed in [48]) in a test bed, and preliminary empirical results seem to suggest that using the same number of mobile agents (routing packets), it is *more likely* that ACO can establish connections through multiple paths while traditional ACO is *more likely* to establish connections through the optimal path

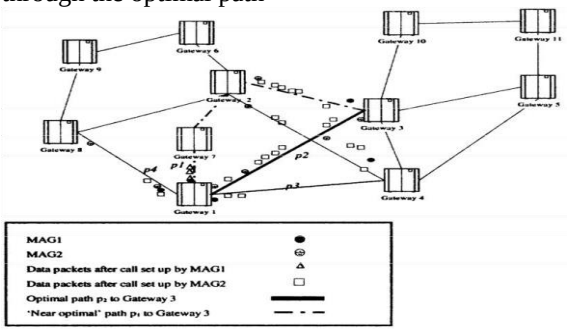

Figure 4

II. CONCLUSION AND FUTURE SCOPE

Till now we have discussed on basic concepts of Cloud Computing and Load balancing. In addition to that, the load balancing technique that is based on Swarm intelligence has been discussed. We have discussed how the mobile agents can balance the load of a cloud using the concept of Ant colony Optimization. The limitation of this technique is that it will be more efficient if we form cluster in our cloud. So, the research work can be proceeded to implement the total solution of load balancing in a complete cloud environment.

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 an 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 been converged 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.

III. REFERENCES

[1] Wayne Jansen, Timothy Grance, "Guidelines on Security and Privacy in Public Cloud Computing", National Institute of Standards and Technology Gaithersburg, January 2011.

[2] Jeep Ruiter, MartijnWarnier, "Privacy Regulations for Cloud Computing", Faculty of Sciences, VU University Amsterdam

International Journal of Web & Semantic Technology (IJWesT) Vol.3, No.2, April 2012

[3] DanchoDanchev,"Building and Implementing a successful Information Security Policy windowsecurity.com-WindowsSecurity Resources for IT admins.

[4] David Escalante and Andrew J. Korty, Cloud Services: Policy and Assessment, *EDUCAUSE Review*, vol. 46, no. 4 (July/August 2011)

[5] Richard N. Katz, "Looking at Clouds from All Sides Now", *EDUCAUSE Review*, vol. 45, no. 3 (May/June 2010): 32-45

[6] Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing A Practical Approach, TATA McGRAW-HILL Edition 2010.

[7] Martin Randles, David Lamb, A. Taleb-Bendiab, A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing, 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops.

[8] Mladen A. Vouk, Cloud Computing Issues, Research and Implementations, Proceedings of the ITI 2008 30th Int. Conf. on Information Technology Interfaces, 2008, June 23-26.

[9] Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed Computer Systems, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.

[10] ibm.com/press/us/en/pressrelease/22613.wss

[11] http://www.amazon.com/gp/browse.html?node=201590011

[12] Martin Randles, EnasOdat, David Lamb, Osama Abu- Rahmeh and A. Taleb-Bendiab, "A Comparative Experiment in Distributed Load Balancing", 2009 Second International Conference on Developments in eSystems Engineering.

[13] Peter S. Pacheco, "Parallel Programming with MPI", Morgan Kaufmann Publishers Edition 2008

[14] MequanintMoges, Thomas G.Robertazzi, "Wireless Sensor Networks: Scheduling for Measurement and Data Reporting", August 31, 2005

[15] Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed Computer Systems, IJCSNS International Journal of Computer Science and Network Security,VOL.10 No.6, June 2010.