IJIRMET ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 3, Issue 4, April 2018

Enhancing Load Balancing In Cloud Computing Using Ant Colony Method

[1] D.Meera, [2] Dr.k.P.Kaliyamurthie
[1] Mtech Student, Computer Science and Engineering, Bharath University
[2] Professor, Computer Science and Engineering, Bharath University.

Abstract: ***Cloud applications are deployed in remote data centers (DCs) where high capacity servers and storage systems are located. A fast growth of demand for cloud based services results into establishment of enormous data centers consuming high amount of electrical power. Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a public utility Cloud computing involves virtualization, distributing computing, platform, infrastructure, software &web based services. Load balancing is challenging issues in cloud environment. It is the process of distribution of the load among various nodes of distributed system so that the better utilization of resources while also avoiding the situation where some nodes are overloaded and some nodes are idle or under loaded. Many algorithm were suggested to provide the better utilization of resources on the basis of On-Demand services. All algorithm enhance the overall performance of the cloud. In this paper we investigated the various algorithm that are based on Ant Colony Optimization (ACO) aimed the load balancing of nodes. We discuss and compare the ACO algorithm to provide the overview of latest algorithm in cloud computing environment.***

Keywords: Cloud computing; Load balancing, Swarm intelligence, Ant Colony Optimization.

I. INTRODUCTION

Cloud computing is a newly progressing technique which offers online computing resources, storage and permits users to organize applications with enhanced scalability, availability and fault tolerance. Cloud computing is about storing the stuff on remote servers instead of on own computers or other devices. This information can be retrieved using the internet with any device, everywhere in the world as long as that device can support cloud computing systems. The cloud computing system is comprised of a front-end, which is the client side and a back-end which is a collection of the servers and computers owned by a third party which stores the data. A central server which is a fragment of the back-end follows protocols and uses middleware to communicate between networked computers. Cloud computing accumulates all the computing resources and manages them automatically [1]. Its characteristics describe a cloud computing system: on-need self-service, pooling of resources, access to the internet, the elasticity of service availability and measurement of services utilized by individual users. Cloud computing is everywhere with tools like Google Drives replacing Microsoft Office, Amazon Web Services replacing traditional enterprise data storage, banking websites replacing branch offices and Dropbox storing all our data and files. The cloud even provides different deployment models and service models.

The four deployment models present in cloud computing are:[2]

1. Public cloud: In the public cloud, the cloud provider provides resources for free to the public. Any user can make use of the resources; it is unrestricted. The public cloud is connected to the public internet for anyone to leverage

2. Private cloud: In a private cloud, the planning and provisioning of the cloud are operated and owned by the organization or the third party. Here the hosted services are provided to a restricted number of people or group of individuals.

3. Community cloud: These type of cloud infrastructures exists for special use by a group of users. These are a group of users who share a common mission or have specific regulatory requirements, and it may be managed by the third party or organizations.

4. Hybrid Cloud: Hybrid Cloud provides the best of above worlds. It is created by combining the benefit of different types of cloud (private cloud & public cloud). In these clouds, some of the resources are provided and managed by public cloud and others as a private cloud.

The three different service models present in cloud computing are:

1. Infrastructure as a Service (IaaS): IaaS model provides just the hardware and the network. It allows users to develop and install their operating system, software and run any application as per their needs on cloud hardware of their own choice.

2. Platform as a Service (PaaS): In PaaS model, an operating system, hardware, and network are provided to the user. It enables users to build their applications on cloud making use of supplier specific tools and languages

3. Software as a Service (SaaS):): In SaaS model, a pre-built application together with any needed software, hardware, operating system and the network is provided to the user

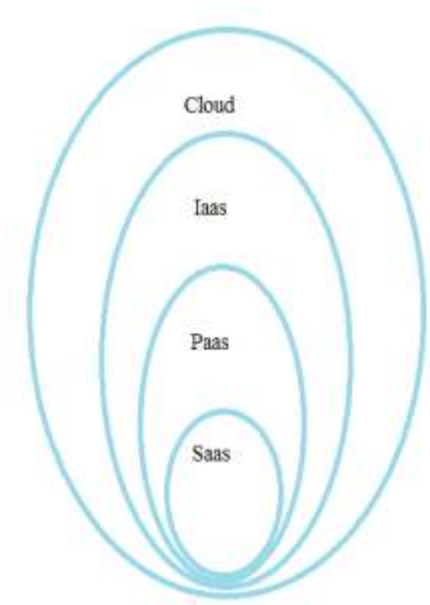

FIG: 1 Cloud service models

II. LITERATURE SURVEY

PrachiVerma, SonikaShrivastava, R.K.Pateriya(2017) Enhancing Load Balancing in Cloud Computing by Ant Colony Optimization Method. Load balancing is required to distribute the workload equally amongst all nodes in a network so that none of a node is overloaded or underloaded and each node does a similar amount of work in equal time. It minimizes the cost and time involved in the major computational models and helps to improve proper utilization of resources and system performance.

Deepak Mahapatra , Gaurav Kumar Saini, HimanshuGoyal,AmitBhati(2016) Ant Colony Optimization: A Solution of Load balancing in cloud.In this paper we are proposing a method based on Ant Colony optimization to resolve the problem of load balancing in cloud environment.

A. Selvakumar, Dr. G. Gunasekaran(2017)A Novel approach in Load balancing for dynamic cloud environment using ACO. This proposal, presenta unique method on load balancing using ant colonyoptimization (ACO), for workload balancing in acloud platform. Two approaches, max-min rules andforward-backward ant mechanism are familiarized todiscover the applicant nodes for load balancing. Thismethod articulates the pheromone initialization and update affording to physical properties under thecloud environment, including pheromone vanishing, inducement, and penalty rules, etc. Combined withtask execution estimation by defining the movementof ants in probability by two ways, that is, the forwardmovement of ant meets the backward movement ofant, or else in the adjacent node, with the goal oftracking searching processes. The proposed workprovides load balancing in dynamic with highnetwork performance.

Ratan Mishra and AnantJaiswal(2012) Ant colony Optimization: A Solution of Load balancing in Cloud. The ACO methods to resolve this problem hasbeen came into existence like Particle Swarm Optimization, hash method, genetic algorithms and severalscheduling based algorithms are there. In this paper we are proposing a method based on Ant Colonyoptimization to resolve the problem of load balancing in cloud environment.

III. LOAD BALANCING:

Load balancing is a serious concern in cloud computing. With the increase in attractiveness of cloud computing among users, the load on the servers and the quantity of processing done is surging drastically. There are multiple nodes in the cloud, and due to the random allocation of a request made by the client to any node, the nodes become unevenly loaded. So to avoid the condition where some nodes are either severely loaded or under loaded, the load balancer will evenly divide the workload among all the nodes [3]. Thus load balancing will equally distribute the workload among the nodes, and it can help in minimizing delays in communication, maximizing the throughput, minimizing execution time and maximizing resource utilization [3].

3.1 Goals of load balancing:

Some of the key purposes of a load balancing algorithm as pointed out by are:
1. It should possess fault tolerance.
2. It should be capable of modifying itself according to any change or expansion in the distributed system configuration 3.
3. Regarding system performance, it should give greater overall improvement at a minimal cost.
4. Regardless of the origin of job it must treat all jobs in the system equally.
5. It should also maintain system stability.

3.2 Issues of Load Balancing

The issues of load balancing are described below [4]:
1. Load balancing becomes critical because, in the middle of execution, the processes may shift amongst nodes to ensure equal workload on the system [5].
2. For a load balancing scheme to be good it should be scalable, general and stable and should add minimal overhead to the system. These requirements are interdependent [6].
3. One of the critical aspects of the scheduling problem is load balancing [7]. The challenge for a scheduling algorithm is to avoid the conflict between prerequisites: fairness and data locality.
4. Algorithms for load balancing have to be dependent on the hypothesis that the on hand information at each node is accurate to avoid processes from being continuously circulated the system without any progresss[5].
5. How to accomplish a balance in load distribution amongst processors such that the computation can be done in the minimum possible time is one of the important problems to resolve.
6. Load balancing and task scheduling in distributed operating systems is a vital factor in gross system efficiency because the distributed system is not pre-emptive and non-uniform, that is, the processors may be different [7].

3.3Componentsof Load Balancing Algorithms

A load balancing algorithm has five major components [8]
1. Transfer Policy: The portion of the load balancing algorithm that picks a job for moving from a local node to a remote node is stated as Transfer policy or Transfer strategy.
2. Selection Policy: In this policy, it specifies the processors involved in the load exchange (processor matching) so that the overall response time and throughput may be improved.
3. Location Policy: The portion of the load balancing algorithm that is responsible for choosing a destination node for a task to transfer is stated as location policy or Location strategy.
4. Information Policy: The part of the dynamic load balancing algorithm that is in charge of gathering information about the nodes present in the system is started to as Information Policy or Information strategy.

5. Load Estimation Policy: In this policy, it determines the total workload of a node in a system.

IV. CLASSIFICATION OF LOAD BALANCING ALGORITHMS:

Load balancing algorithms have been classified based on current state of system and who initiated the process
1**. Depending on which user initiates the process**:
A. Sender-Initiated: Sender or client initiates the execution of load balancing algorithm on identifying the need for load balancing.
B. Receiver-Initiated: Receiver or server initiates the execution of load balancing algorithm on identifying the need for load balancing.
C. Symmetric: This type of algorithm is a blend of sender-initiated type and receiver-initiated type algorithms.
2**. Depending on current state of system**
Static algorithm: In the static algorithm, there is a uniform distribution of traffic among the servers. This algorithm needs a prior understanding of system resources so that the judgment of shifting of the load does not depend on the current state of the scheme. The static algorithm is perfect in the system which has fewer inequalities in load [3].

4.1 dynamic algorithm

In the dynamic algorithm, for balancing the load the lightest server in the entire system or network is looked upon and preferred. For this real-time communication with the network is needed which can increase the traffic in the system[10]. Here to make decisions for managing the load the current state of the system is used [3].
In a distributed system, dynamic load balancing can be done in two different ways: distributed and non-distributed. In the distributed one, the dynamic load balancing algorithm is executed by all nodes present in the system and the task of load balancing is shared among them. The interaction among nodes to achieve load balancing can take two forms: cooperative and noncooperative [11]. In the first one, the nodes work side-by-side to achieve a common objective, for example, to improve the overall response time, etc. In the second form, each node works independently toward a goal local to it, for example, to improve the response time of a local task. Dynamic load balancing algorithms of distributed nature, usually generate more messages than the non-distributed ones because, each of the nodes in the system needs to interact with every other node. A benefit, of this is that even if one or more nodes in the system fail, it will not cause the total load balancing process to halt, it instead would affect the system performance to some extent. Distributed dynamic load balancing can introduce immense stress on a system in which International Journal of Web & Semantic Technology (IJWesT) Vol.3, No.2, April 2012 41[9] each node needs to interchange status information with every other node in the system. It is more advantageous when mostof the nodes act individually with very few interactions with others.

V. 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. The complex social behaviors of ants have been much studied by science, and computer scientists are now finding that these behavior patterns can provide models for solving difficult combinatorial optimization problems. The attempt to develop algorithms inspired by one aspect of ant behavior, the ability to find what computer scientists would call shortest paths, has become the field of ant colony optimization (ACO), the most successful and widely recognized algorithmic technique based on ant behavior.

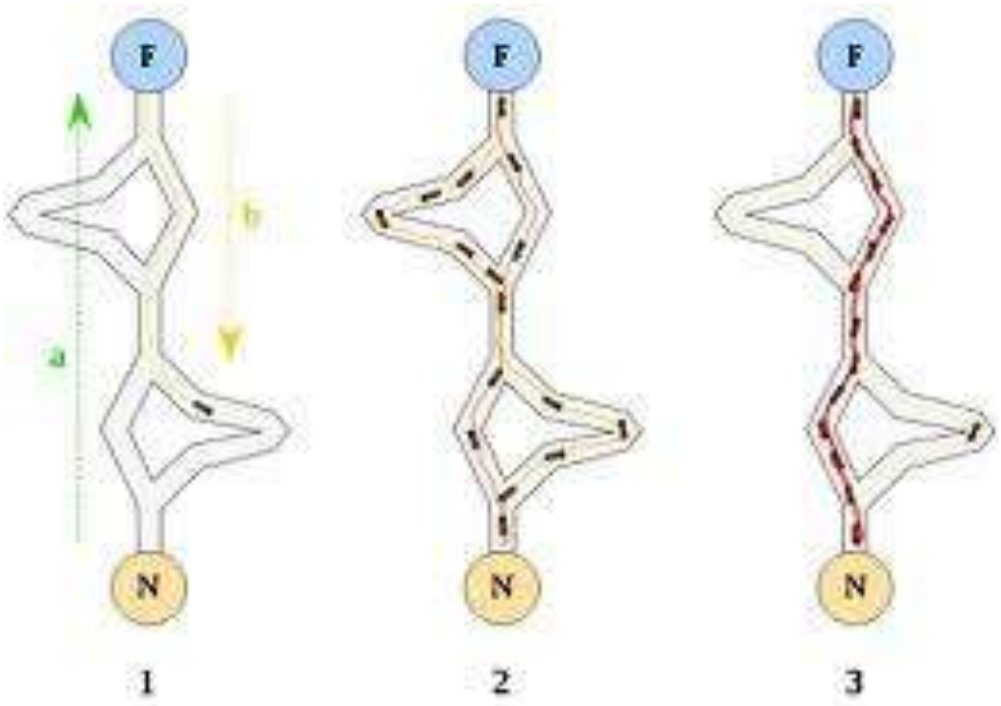

Fig:2 Ant Colony

5.1 Basic principles of trail laying

Depending on the species, ants may lay pheromone trails when travelling from the nest to food, or from food to the nest, or when travelling in either direction. They also follow these trails with a fidelity which is a function of the trail strength, among other variables. Ants drop pheromones as they walk by stopping briefly and touching their gesture, which carries the pheromone secreting gland, on the ground. The strength of the trail they lay is a function of the rate at which they make deposits, and the amount per deposit. Since pheromones evaporate and diffuse away, the strength of the trail when it is encountered by another ant is a function of the original strength, and the time since the trail was laid. Most trails consist of several superimposed trails from many different ants, which may have been laid at different times; it is the composite trail strength which is sensed by the ants. The principles applied by ants in their search for food are best explained by an example as given in [12].

VI. 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.
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 thisapproach for updating pheromone is only appropriate for routing in symmetric networks.
International Journal of Web & Semantic Technology (IJWesT) Vol.3, No.2, April 2012 45 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 *lesslikely* 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.

6.1 Initialization of secretion

In cloud computing, the physical resources allocated to each virtual node are not the same and usually changing dynamically. Due to of this characteristic, we use the physical resources of virtual machines to measure a node's initial pheromone. Five physical resources are involved in pheromone initialization here, that is, CPU (number of cores and MIPS for each core), internal and external storage, I/O interface.
The CPU capability can be calculated by:

$$P_{CPU}=n*p$$

6.2 Pheromone Update

The goal of pheromone update in our advance is to increase the pheromone values for slave nodes related with good conditions and decrease those associated with bad ones. Three factors that impacts the pheromone update are measured in our strategy, namely pheromone evaporation, update by task, and encouragement for successful tasks.

6.3 Pheromone vanishing

The pheromone in the node is decreasing over time due to evaporation. We use the local update approach to modify the pheromone on slave nodes where the pheromone is not zero. The pheromone update by evaporation is defined as:

$$\tau_i\,(t+1)=(1-\rho)\times\tau_i\,(t),0< \rho< 1$$

6.4 Inducement and penalty Rules

We outline 2 rules for task execution in a slave, that is, incentive rule and penalization rule. The previous means the secretion is raised during this node if the tasks are performed with success. The latter denotes that the secretion is shriveled during this node if the tasks are done unsuccessfully.

$$Ti(t+1)=(1+\Theta)\times\tau i(t)$$
$$\text{If (success) } 0 < \Theta < 1$$
$$\text{else} \quad -1 < \Theta <0$$

6.5 Task Execution Prediction

Task execution prediction is to guage the execution rate for a slave node that reflects the aptitude and performance of virtual resources in cloud computing. Usually speaking, the potency of the cloud platform will be improved once allocating the tasks to the slave nodes with higher performance. It tends to style a prediction model that evaluates the execution rate of a slave node for succeeding timeframe by accumulating the previous records. Through this employment and therefore the employment performed last time, we will predict the speed of a slave node for succeeding timeframe by

$$EV_i{}^{a}{}_{k+1}(\,k+1) =a_{k+1} / a_k ((\,1-\omega\,)\,EV_i{}^{a}{}_{k}(\,k\,) +\omega RV_i{}^{a}{}_{k}(\,k\,))$$

VII. CONCLUSION AND FUTURE WORK

Cloud computing is seen to transform the major services and their resources are availed on the internet by users. Techniques of load balancing which exist having been examined mostly focusing on enhancing quality services and offering expected results on time. Hence, there is the requirement to establish the technique of load balancing that may enhance the cloud figuring performance together with maximum resource consumption. The suggested technique of a load balancing basing on ACO maximization provides optimal resource exploitation. This technique does not consider the fault tolerance issues. Researchers can proceed to include the fault tolerance issues in their future researches.

VIII. REFERENCES

[1]D. Saranya et.al, "Load Balancing Algorithms in Cloud Computing: A Review," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 5, Issue 7, July 2015.

[2]S. Sethi et.al, "Efficient Load Balancing in Cloud Computing using Fuzzy Logic," IOSR Journal of Engineering (IOSRJEN) ISSN: 2250-3021 vol. 2, pp. 65-71, July 2012.

[3]T. Desai et.al, "A Survey of Various Load Balancing Techniques and Challenges in Cloud Computing," International Journal of Scientific & Technology Research, vol. 2, Issue 11, November 2013.

[4]S.Rajoriya et.al, "Load Balancing Techniques in Cloud Computing: An Overview," International Journal of Science and Research (IJSR), vol. 3, Issue 7, July 2014

[5]Sharma S. et.al, “Performance Analysis of Load Balancing Algorithms,” World Academy of Science, Engineering and Technology, 38, 2008.

[6]Gross D. et.al, “Noncooperative load balancing in distributed systems”, Elsevier, Journal of Parallel and Distributed Computing, No. 65, pp. 1022-1034, 2005.

[7]Nikravan M. et.al, “A Genetic Algorithm for Process Scheduling in Distributed Operating Systems Considering Load Balancing”, Proceedings 21st European Conference on Modelling and Simulation (ECMS), 2007.

[8]M.Amar et.al, “SLA Driven Load Balancing for Web Applications in Cloud Computing Environment”, Information and Knowledge Management, 1(1), pp. 5-13, 2011.

[9]Ratan Mishra and Anant Jaiswal “Ant colony Optimization: A Solution of Load balancing in Cloud”, Vol.3, No.2, April 2012.

[10]A. Selvakumar, Dr. G. Gunasekaran “A Novel Approach In Load Balancing For Dynamic Cloud Environment Using Aco “,Volume 116 No. 5 2017, 67-72.

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

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