International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869, Volume-3, Issue-4, April 2015

A Study of Load Balancing in Cloud Computing using cloud partitioning and ACO

Ms.S.S.Malani , Dr.S.Y.Amdani

***Abstract*— There are several issues or challenges in the Cloud Computing environment such as Availability, Security, and Resource Allocation etc. Load balancing is an important factor in cloud computing environment. In case of large and complex cloud area cloud partitioning is used. Load balancing techniques generally concentrates on providing the expected output on time, improving the quality of services. Therefore, there is a need to develop load balancing technique that can improve the performance of cloud computing along with maximum resource utilization. In this paper, large cloud is partitioned for ease of management and to improve performance of load balancing algorithm based on Ant Colony Optimization.**

***Index Terms*— cloud computing, load balancing, cloud partitioning, ACO (Ant Colony Optimization)**

I. INTRODUCTION

Cloud Computing is a concept that has many computers interconnected through a real time network like internet. Cloud Computing has widely been adopted by the industry or organization though there are many existing issues like Load Balancing, Virtual Machine Consolidation, Energy Management, etc. which have not been fully implemented. Central to these issues is the issue of load balancing, that is required to distribute the excess dynamic local workload equally to all the nodes in the whole Cloud to achieve a high user satisfaction. The job arrival pattern is not expected and the capabilities of each node in the cloud differ for load balancing problem, workload control is critical to improve system performance and maintain stability.

In case of cloud partitioning, the model has number of partition in the cloud for a public cloud. These partitions make it easy for load balancing when the location is very large and compound. The cloud has a main manager that prefer the appropriate partitions for incoming jobs while for every job that is arriving the balancer decides the finest load balancing policy from Ant colony.

II. LOAD BALANCING ALGORITHMS

A load balancing algorithm aims to increase the utilization of resources with light load or idle resources thereby freeing the resources with heavy load. The algorithm tries to distribute the load among all the available resources. At the same time, it aims to minimize the makespan with the effective utilization of resources.

Several Load Balancing algorithms were proposed. Some of those algorithms are discussed here.

A. Dynamic Round Robin Algorithm

Dynamic Round-Robin [1] method is an extension to the Round-Robin method. It uses two rules that help to consolidate virtual machines. The first rule says that if a virtual machine has finished and stills other virtual machines that are hosted on the same physical machine, then this physical machine cannot accept any new virtual machine. Such physical machines are called "retiring" state physical machines, that means when the all the other virtual machines finish their execution, then we can shut down this physical machine. The second rule says that if there is a "retiring" state physical machine that is used for long period of time, then instead of waiting for that virtual machines to finish, the physical machine is forced to migrate all the other virtual machines to other physical machines, and then shut down the physical machine after the migration finishes. The threshold waiting time is represented by the "retirement threshold". A physical machine will be forced to migrate to all the virtual machines and then shut it down as it is in the retiring state but after the retirement threshold, it could not finish all other virtual machines. These two rules are used by the Dynamic Round- Robin strategy so as to consolidate virtual machines implemented by the Round-Robin method. According to the first rule, adding extra virtual machines to a retiring physical machine is avoided. According to the second rule, the consolidation process become fast and it enables Dynamic Round-Robin to shutdown physical machines, such that the number of physical machine used to run all virtual machines is reduced , hence the power can be saved.

B. Hybrid Algorithm

It is the combination of both Dynamic Round-Robin and First-Fit to form a Hybrid algorithm [2]. The probability distribution (e.g., a normal distribution) is followed and the number of incoming virtual machines is assumed as a function for time. Hybrid algorithm uses virtual machine's incoming rate for the scheduling of virtual machines. The First- Fit is used by the Hybrid method during rush hours to completely utilize the computing power of physical machines, and then it uses the Dynamic Round-Robin for the consolidation of the virtual machines and thus reduce the consumption of the energy in non-rush hours.

C. Min-Min Algorithm

The Min-Min algorithm first finds the minimum execution time of all tasks. Then it chooses the task with the least execution time among all the tasks. The algorithm proceeds by assigning the task to the resource that produces the minimum completion time. The same procedure is repeated by Min-Min until all tasks are scheduled [3].

Manuscript received April 21, 2015.

Ms. Sarika S Malani, Computer Science and Engineering, Sant Gadge Baba Amravati University, Pusad India.

Dr. S. Y. Amdani, Computer Science and Engineering, Sant Gadge Baba Amravati University, Pusad India.

D. *Max-Min Algorithm*

It works as similar to Min-min. Again the minimum completion time for each job is established, but the job with the maximum minimum completion time is assigned to the corresponding processor. The problem is that smaller jobs have to be waiting for long time [4].

E. *Honeybee Foraging Behavior*

It is a decentralized honeybee-based load balancing technique that is a nature-inspired algorithm for self-organization [5]. It achieves global load balancing through local server actions. Performance of the system is enhanced with increased system diversity but throughput is not increased with an increase in system size. It is best suited for the conditions where the diverse population of service types is required.

F. *Ant Colony Optimization*

Ant Colony Optimization (ACO) has been found to be both robust and versatile in handling a wide range of combinatorial optimization problems. The main idea of ACO is to model a problem as the search for a minimum cost path in a graph [6].

TABLE I SHOWS THE COMPARISON OF LB ALGORITHMS DISCUSSED AOVE

Algorithm	Description	Advantages
Dynamic Round Robin Algorithm	1. Uses two rules to save the power consumption 2. Works for consolidation of VM	Reduce the power consumption
Hybrid Algorithm	1. Combination of Dynamic Round Robin and First-Fit Algorithm 2. Applied in non-rush hours and rush hours	1. Improved Resource Utilization 2. Reduced Power Consumption
Min-Min Algorithm	1. Estimates minimum execution time and minimum Completion time 2. Jobs having minimum completion time is executed first	Smaller tasks are executed quickly
Max-Min Algorithm	1. Same as Min-Min 2. Gives more priority to larger tasks than smaller one	Larger tasks are executed quickly and efficiently
Honeybee Foraging Behavior	Achieves global load balancing through local server actions	Improved scalability
Ant Colony Optimization	There are two types for movements: forward and backward	1. Detection of over loaded and under loaded nodes can be done. 2. Path tracing can be done consequently.

Table II shows the comparison between honeybee foraging and ant colony optimization based on various metrics [7].

Techniques / Metrics	Honeybee Foraging Behavior	Ant Colony Optimization
Throughput	Yes	Yes
Overhead	No	No
Response time	No	No
Resource utilization	No	Yes
Performance	Yes	Yes

III. PROPOSED SYSTEM

Existing Load balancing algorithms have some drawbacks in improving overall performance of the cloud environment. Still there is a problem of overloading nodes in the Cloud environment. It is very difficult to manage entire cloud environment. Hence the proposed idea is to divide the entire cloud environment into several partitions based on its geographical locations [8].

Now the Load balancing algorithm can be applied only to the partitions, not to the entire cloud. The load balancing algorithm is applied to each partition in order to avoid overloading of nodes.

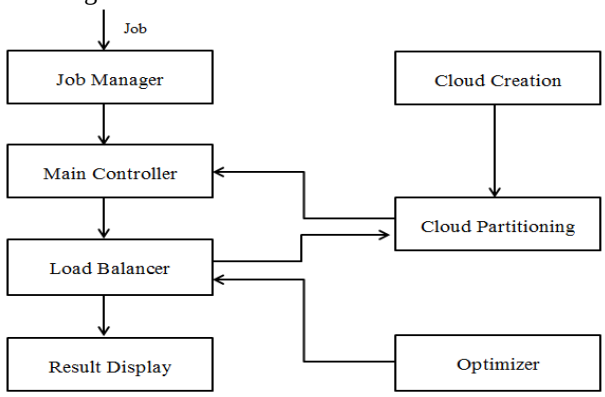

Fig 1:- Block Diagram of proposed load balancing technique

Fig 1. shows the block diagram of the proposed load balancing technique [7]. Cloud environment is created with the help of CloudAnalyst Simulator. User sends request for particular service. Request arrives at JobManager. JobManager maintains a queue data structure for storing the request. If queue is not full then the request is added into the queue else request is rejected. Then, job arrives to the Main Controller.

The function of Main Controller is to find out best cloud partition. It also maintains current status of each cloud partition . Now, job arrives at Load Balancer of the best cloud partition. The function of Load Balancer is to assign job to the particular node according to strategy. Load balancer also maintains the current status of nodes. On the basis of current status of nodes, it calculates the status of cloud partition (i.e. idle/normal/overloaded).

The function of Optimizer is to optimize the resource utilization. Optimization of resource utilization is going on concurrently in each partition. Here, Ant Colony Optimization will be used for resource optimization.

IV. 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 [9].

In ACO algorithm when the request is initiated the ant start its movement. Movement of ant is of two ways:

- Forward Movement means the ant in continuously moving from one overloaded node to another node and check it is overloaded or under loaded, if ant find an overloaded node it will continuously moving in the forward direction and check each nodes.
- Backward Movement- If an ant find an over loaded node the ant will use the backward movement to get to the previous node, in the algorithm if ant finds the target node then ant will commit suicide, this algorithm reduced the unnecessary back ward movement, overcome heterogeneity, is excellent in fault tolerance [10].

In this algorithm, ant would lay down foraging pheromone (FP) after encountering underloaded node for searching overloaded node. Trailing pheromone (TP) is used to find path to the underloaded node after encountering overloaded node.

Following are the steps of ACO for load balancing between cloud nodes [7]

[Start]
Step 1: Initialize the pheromone tables
Step 2: Declare a threshold level for nodes
Step 3: Ants move through nodes.
Step 4: (if else condition)
if(ant-timer > counter)
stop()
Step 5: else
(if else condition)
Check the status of encountered cloud node.
(i.e. whether encountered node is overloaded or underloaded)
if(node is underloaded) then follow steps from 6 to 8
else follow steps from 9 to 11
Step 6: Traverse to one of the neighboring
nodes of encountered node which has maximum TP. This node will be a currentnode.
Step 7: Update pheromone table of both the nodes (i.e. encountered node and currentnode)
Step 8: (if else condition)
Check status of currentnode.
(i.e. whether it is overloaded or underloaded)
if(currentnode is underloaded) then go to Step 6
else reassign resources (between encountered and currentnode) and then go to step 3
Step 9: Traverse to one of the neighboring nodes of encountered node which has minimum FP. This node will be a currentnode.
Step 10: Update pheromone table of both the nodes (i.e. encountered node and currentnode)
Step 11: (if else condition) Check status of currentnode. (i.e. whether it is overloaded or underloaded)
if(currentnode is overloaded)
then go to Step 9
else reassign resources
(between encountered and currentnode) and then go to step 3

V. CONCLUSION

Though there are several issues in cloud environment, it has been widely adopted by many organizations and industries. Researchers are doing many works to resolve those issues. For Load Balancing issue, the solution is to develop suitable algorithms that balance the load across the partitioned cloud environment. Therefore, there is a need to develop load balancing technique that can improve the performance of cloud computing along with maximum resource utilization. In this paper, proposed method of load balancing based on Ant Colony Optimization is used for optimizing resource utilization and Cloud partitioning helps to improve performance of algorithm as well as ease the management of cloud.

REFERENCES

[1] Karanpreet Kaur, Ashima Narang, Kuldeep Kaur, "Load Balancing Techniques of Cloud Computing", *International Journal of Mathematics and Computer Research*, April 2013.
[2] Gaochao Xu, Junjie Pang and Xiaodong Fu, "A Load Balancing Model Based on Cloud Partitioning for the Public Load", *IEEE Transactions on Cloud Computing*, 2013.
[3] T. Kokilavani, Dr. D.I. George Amalarethinam, "Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing" ,*International Journal of Computer Applications (0975 – 8887)Volume 20– No.2*, April 2011.
[4] S.Mohana Priya, B.Subramani, "A New Approach for Load Balancing in Cloud Computing", *International Journal of Engineering and Computer Science*, May 2013.
[5] Nidhi Jain Kansal, Inderveer Chana, "Cloud Load Balancing Techniques: A Step Towards Green Computing", *International Journal of Computer Science*, January 2012.
[6] V.Selvi Dr.R.Umarani **,"** Comparative Analysis of Ant Colony and Particle Swarm Optimization Techniques", *International Journal of Computer Applications* (0975 – 8887) Volume 5– No.4, August 2010.
[7] Ekta Gupta Vidya Deshpande, "A Load Balancing Technique using Cloud Partitioning, Scheduling and Optimization of Resource using Ant Colony Optimization",*International Journal of Engineering Research & Technology* (IJERT) IJERTIJERT ISSN: 2278-0181.
[8] Abhijeet G Purohit1, Md. Abdul Waheed2, Asma Parveen**,**"Load Balancing in Public Cloud by Division of Cloud Based on The Geographical Location", IJRET eISSN: 2319-1163 | pISSN: 2321-7308.
[9] Ratan Mishra1 and Anant Jaiswal,"Ant colony Optimization: A Solution of Load balancing in Cloud", *International Journal of Web & Semantic Technology (IJWesT)* Volume 3, No.2, April 2012.
[10] Preeti Kushwah,"A Survey on Load Balancing Techniques Using ACO Algorithm", *International Journal of Computer Science and Information Technologies*(IJCSIT), Volume 5 (5) , 2014, 6310-6314.

Ms. Sarika S. Malani received the B.E degree in Computer science and Engineering from Babasaheb Naik College of Engineering Pusad And now she is pursuing her M.E degree in Computer science and Engineering under S.G.B.A.U university in Babasaheb Naik College of Engineering, Pusad,Maharashtra.

Dr. S. Y. Amdani, received his Ph.D. degree in Computer science and Engineering from S.G.B.A.U.Amravati. He is an Associate Professor and Head of CSE Department, Babasaheb Naik College of Engineering, Pusad, His major research interest are in the fields of Multimedia signal processing, Real time database scheduling. He has published many research papers in international conferences and Journals.