A Load Balancing Technique using Cloud Partitioning, Scheduling and Optimization of Resource using Ant Colony Optimization

Ekta Gupta
Department of Information Technology
MIT, Pune

Vidya Deshpande
Assistant Professor
Department of Information Technology
MIT, Pune

***Abstract* -- Load balancing is an essential factor in cloud computing. When the cloud environment is very large and complex, the concept of cloud partitioning can be used. Existing Load Balancing techniques mainly focus on improving the quality of services, providing the expected output on time. Therefore, there is a need to develop load balancing technique that can improve the performance of cloud computing along with maximum resource utilization. Good load balancing makes cloud computing more efficient. It also improves user satisfaction and resource utilization ratio. In this paper, large cloud is partitioned for ease of management and to improve performance of load balancing algorithm based on Ant Colony Optimization.**

Keywords* -- *Ant Colony Optimization (ACO), Cloud computing, Cloud partitioning, Load balancing

I. INTRODUCTION

Distributed processing, parallel processing and grid computing collectively emerges as cloud computing. Gartner defines cloud computing as “a style of computing where massively scalable IT-enabled capabilities are delivered ‘as a service’ to external customers using Internet technologies.” There are three keywords in this definition Scalable, Service and Internet. Cloud computing provides services using internet. It provides scalable infrastructure. It helps to adapt changes in demand.

In cloud computing environment, services offered to the customer are based on the concept of “pay as a service”, where each customer pays for the services obtained from provider [8]. So, the system which is incurring a cost for the user should function smoothly even at peak usage hours [8].

A load balancing technique is needed to avoid the condition where some nodes are heavily loaded while others are idle or doing less work. The proper load balancing technique for cloud improves high user satisfaction [8]. There are number of aims of load balancing technique like maximize number of requests handled by cloud and to minimize time required to serve those requests [8]. In this paper, performance of load balancing technique for cloud is improved using efficient scheduling algorithm for assigning jobs to the nodes of cloud and Ant Colony Optimization algorithm is used between cloud nodes for optimizing resource utilization.

II. LITERATURE SURVEY OF DIFFERENT LOAD BALANCING ALGORITHMS FOR CLOUD

A. Minimum Execution Time (MET)

T. Kokilavani et al. [2] have studied the relative performance of Minimum Execution Time (MET). This algorithm assigns job to the resource which has minimum expected execution time without considering the availability of the resource and its current load.

B. Minimum Completion Time (MCT)

Isam Azawi Mohialdeen [3] has studied the performance of the Minimum Completion Time job scheduling algorithm. In this algorithm, selected job will be allocated to the available VM that can offer the minimum completion time taking into account its current load.

C. Min-Min Algorithm

T. Kokilavani et al. [2] have studied the relative performance of static task scheduling algorithm called Min-Min algorithm. The Min-Min algorithm first finds the minimum execution time of all tasks. Then it selects the task with the minimum execution time among all the tasks. Then algorithm assigns the task to the resource that produces the minimum completion time. The same steps are repeated by Min-Min until all tasks are scheduled.

D. Max-Min Algorithm

In this MAX-MIN algorithm, the machine that has the minimum completion time for all jobs is selected. Then the job with the overall maximum completion time is selected and allocated to that resource. The ready time of the resource is updated. This process of algorithm is repeated until all the unmapped tasks are assigned. The aim of this algorithm is to minimize the waiting time of the larger jobs.

E. Opportunistic Load Balancing Algorithm

Isam Azawi Mohialdeen [3] has studied the performance of the Opportunistic load balancing algorithm. This algorithm attempts to dispatch the selected job to the available VMs which has the minimum load compared to the other VMs. The idea is to scale the current loads for each VM before sending the job. Then, the VM that has the minimum load is selected to run the job.

F. Equally Spread Current Execution Algorithm

Jayant Adhikari et al. [4] have discussed an algorithm called Equally Spread Current Execution Algorithm. The cloud manager estimates the job size and checks for the availability and capacity of the virtual machine. When the job size and the available resource size match, the job scheduler immediately allocates the identified resource to the job in queue.

G. Throttled Load Balancing Algorithm

Manoranjan Dash et al. [6] have discussed an algorithm called Throttled algorithm. Throttled algorithm is totally based on virtual machine. In this client first requesting the load balancer to check the right virtual machine which access that load easily and perform the operations which is given by the client or user.

H. Connection Mechanism

Supriya Kinger et al. [1] have discussed a connection mechanism for Load balancing. Load balancing algorithm can also be based on minimum connection mechanism. When a new connection is dispatched to the node, the number of connection increases. When connection finishes or timeout happens, the number of connection decreases.

I Biased Random Sampling

Martin Randles [7] investigated a distributed and scalable load balancing approach. This approach uses random sampling of the system domain to achieve self-organization thus balancing the load across all nodes of the system. This technique fails to provide proper resource utilization.

J. Honeybee Foraging

Martin Randles et al. [7] have discussed honeybee foraging algorithm. This algorithm is derived from the natural behavior of honeybees for finding and reaping food.

K. Ant Colony Optimization

Kumar Nishant et al. [5] proposed an algorithm for load balancing based on Ant colony Optimization (ACO). ACO is inspired from the foraging behavior of ant colonies. In fact the real ants have inspired many researchers for solving the problem in different areas. ACO gives better performance compare to other load balancing algorithms.

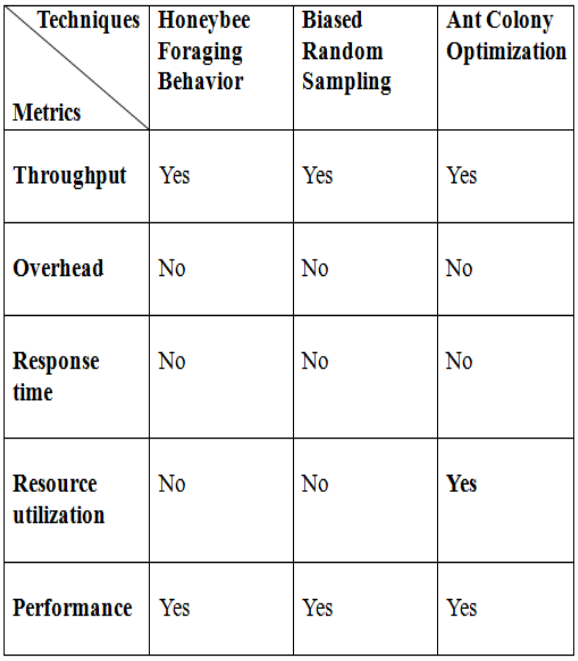
Table I shows the comparison between biased random sampling, honeybee foraging and ant colony optimization based on various metrics [8].

III. PROPOSED LOAD BALANCING TECHNIQUE FOR CLOUD

Figure I. shows the block diagram of the proposed load balancing technique [8]. Cloud environment is created with the help of CloudAnalyst Simulator. Cloud is partitioned based on geographic locations.

User sends request for particular service. Request arrives at JobManager. JobManager maintains a queue data structure for storing the request [8]. If queue is not full then the request is added into the queue else request is rejected. After this, job arrives to the Main Controller.

Table I. Comparison between load balancing algorithms



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

The function of Main Controller is to find out best cloud partition. Main Controller also maintains current status of each cloud partition [8].

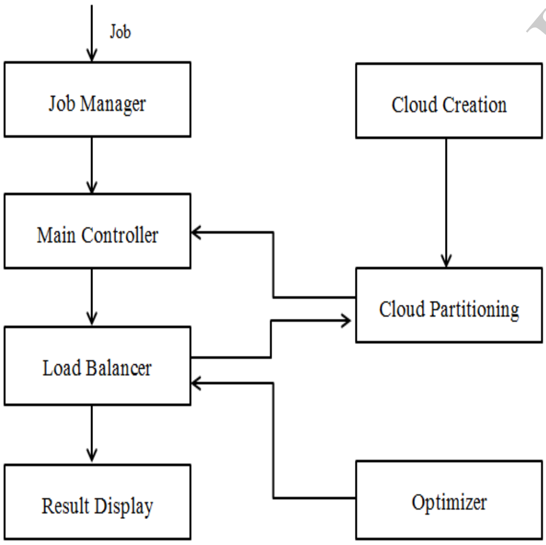


Figure I. Block Diagram of proposed load balancing technique

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 [8]. 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) [8].

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 (ACO)

ACO is a branch of Swarm Intelligence. Inherent parallelism, robustness & scalability along with simplicity of individual agent are the advantages of ant-based systems. The inspiring source of ACO is the foraging behavior of real ants. When searching for food, the ant deposits a chemical called pheromone trail on the ground. The quantity of pheromone deposited, which may depend on the quantity and quality of the food, will guide other ants to find the food source [8].

There are two types of ant movement for load balancing, forward movement and backward movement. In forward movement, ant continuously moves in forward direction and it encounters overloaded or underloaded node of cloud. In backward movement ant moves in backward direction. If in forward movement ant encounters overloaded node first and then underloaded node, then it will go backward to overloaded node to check if node is still overloaded or not and if it finds it still overloaded then it will redistribute the work to underloaded node.

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 [8].

Following are the steps of ACO for load balancing between cloud nodes [8].

```
[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

Load balancing is one of the main challenges in cloud computing. Existing Load Balancing techniques that have been studied mainly focus on improving the quality of services, providing the expected output on time etc. 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. Cloud partitioning helps to improve performance of algorithm as well as ease the management of cloud. Future Work of Proposed System: Sharing of main controller work between finite number of servers or replication

REFERENCES

[1] Amandeep Kaur Sidhu, Supriya Kinger, "Analysis of Load Balancing Techniques in Cloud Computing", *International Journal of Computers & Technology, Volume 4 No. 2, March-April, 2013, ISSN 2277-3061.*

[2] Dr. D.I. George Amalarethinam, and T. Kokilavani, "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*

[3] Isam Azawi Mohialdeen, "Comparative Study Of Scheduling Algorithms In Cloud Computing Environment", *Journal of Computer Science, 9 (2): 252-263, 2013 ISSN 1549-3636 2013.*

[4] Jayant Adhikari, Sulbha Patil, "Load Balancing the Essential Factor in Cloud Computing", *International Journal of Engineering Research & Technology, ISSN: 2278-0181, Vol. 1 Issue 10, December 2012.*

[5] Kumar Nishant, Pratik Sharma, Vishal Krishna, Chhavi Gupta and Kunwar Pratap Singh, "Load Balancing of Nodes in Cloud Using Ant Colony Optimization", *IEEE 14th International Conference on Modelling and Simulation, 2012.*

[6] Manoranjan Dash, Amitav Mahapatra and Narayan Rajan Chakraboty, "Cost Effective Selection of Data Center in Cloud Environment", *Special Issue of International Journal on Advanced Computer Theory and Engineering, ISSN (Print): 2319 – 2526, Volume-2, Issue-1, 2013.*

[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] Ekta Gupta, Vidya Deshpande, "A Load Balancing Technique for Cloud Computing", March 2014 Cyber Times International Journal of Technology and Management.