Load Balancing in Cloud Computing Using Autonomous Agents

Prof. Rahul Bhole, Hoda Jagmeet Singh, Pushkaraj Khamkar, Pranav Joshi, Roshni Bendbhar.
Assistant Professor, Department Of Information Technology, Zeal College OF Engineering And Research, Pune, Maharashtra, India

Abstract: Cloud Computing, a technology based on internet and resources releasing the data center. As cloud computing totally revolves around internet it majorly suffering from overloading of request. Load balancing refers to distribution of resources in such a way that there is no overloading occurs at any machine and there is increase in resource utilization. But this aspect of cloud computing is not taken under consideration. As there are very few algorithms which are been found for cloud computing but there are many which are been developed for load balancing. Although cloud computing is different from all other environments, there should be a specific load balancing algorithm for cloud computing. This work mainly focuses on Autonomous Agent Based Load Balancing algorithm which mainly focuses on dynamic load balancing for the cloud environment.

1. Introduction

The cloud computing is mainly used because it is distributed in nature. It is designed for remote sharing and heterogeneous storage. It also provides some services like computation capabilities and applications etc. Cloud computing comes into toady’s picture because it provides high reliability over large network. But the request comes are dynamic in nature so dynamic resource allocation is required in it. Because of dynamic resource allocation in cloud computing requires efficient load balancing mechanisms. Load neither balancing means distribution of resources among the users or request uniformly so that neither node is overloaded nor sitting idle. If the load balancing is not proper then the efficiency of some overloaded nodes can degrade their performance and hence lead to SLA violation. Basically the load balancing algorithms are of three types which are as static, dynamic or mixed scheduling algorithms based on their nature where.

1) Static load balancing algorithm is suitable for small scale distributed systems with high speed of internet and avoidable delay of communication.

2) Dynamic Load Balancing Algorithm mainly used for reducing communication delays and time needed for execution therefore it is mainly used in large distributed system.

3) Mixed Load Balancing Algorithm is used where symmetrical distribution of assigned computing task and reducing cost required for communication of distributed computing nodes.

As per stated above it is clearly seen that cloud computing falls under the second category. It means balancing load in cloud computing environment requires focusing on dynamic load balancing algorithms. In traditional distributed environments process migration is less expensive because of small process granularity whereas in cc environment, process migration is expensive because of high granularity of data involved. Thus cloud computing environment requires a load balancing algorithm which could serve to dynamic service demands of users while providing optimized load balancing. Following parameters are available in literature for measuring efficiency of a load balancing algorithm in CC environment:

i. Reliability
ii. Adaptability
iii. Fault Tolerance
iv. Throughput
v. Waiting Time

2. Review of Existing Load Balancing Algorithm in Cloud Computing

Mainly load balancing is a problem in heterogeneous computer networks. To overcome this problem many centralized approaches are made but the centralized behavior has raise the scalability tribulation. The previous analysis has reported that the honeybee algorithm has maximum throughput with increased system diversity as compared to the other algorithms. The previous algorithm was the honeybee algorithm which motivated from the behavior of biological bees which move for finding food. In the same way the
load balancing consist of virtual servers which offer various virtual services. In which every server requires for services and then calculates the profit and then post the values on the output. In this the servers which have interest in serving the request also calculate the profit and compare it with the other colony profits. If in case the high colony profit interested server then serves the current virtual server otherwise returns to the scout behavior so that it will be possible for it to choose another server randomly.

The other one proposed a genetic algorithm which is mainly based on scheduling for load balancing in virtual machines. This algorithm designed selects the least virtual machines for the transfer of load and also optimizes the high migration cost. Even though due to huge number of virtual machines and the frequently occurring request in the data centre where there is a chance of insufficient service scheduling.

The third one proposed a system to migrate the network the application by providing the layer of virtualization placed on the top of the operating system and transferring the process group. These achieve lower service downtime but they still use stop and copy method.

From this previous literature one can understand that there are some drawbacks for example the static nature of load balancing algorithms which lack in the scalability and also in the reliability. Further by the analysis it was also observed that the genetic algorithm, honeybee algorithm and intelligent agents had been employed in load balancing in cloud computing which tells that the researchers have found that it is suitable for such applications and there is also a scope of employing them further. Hence there is a strong need for a efficient and reliable load balancing algorithm in cloud computing.

3. Proposed Work

From the literature it is clear that there is limitation for doing work in load balancing in cloud computing and there are mechanisms which have limitations that can be addressed. There is an algorithm which offers maximum resource utilization, maximum throughput, minimum response time, dynamic resource scheduling with its scalability and reliability. Then through this autonomous agent based load balancing algorithm (A2LB) is used to reduce the above issues. Whenever Virtual Machine becomes overloaded, the service provider distributes the resources in such a way that the available resources will be utilized in a proper way and load at all the virtual machines will be balanced. Agent to load balance mechanism is of three agents: Load agent, Channel agent, Migration Agent. Load agent and static agent are static agents whereas migration agent is an ant, which is a category of mobile agents. Reason behind deploying ants is that they have ability to choose shortest/best path to reach to their destination. Ant agents got motivated from biological agents which have a path from there colonies to the food source. While doing this there is a secrete chemical called pheromone on ground which is leaving a trial for other member to follow. This chemical gets evaporate on time. The ants starts searching food source randomly, they can follow different paths to same source that is called shortest path, which takes less time, density of pheromone on shortest path increases and all the ants starts following the path which results in increasing of pheromone density further. The property of ants is that they start moving from source to destination for collecting required information or performing task but they do not come back to their source and destroy themselves at the destination that reduces the unnecessary traffic on network. Since load balance in cloud computing requires searching for loaded servers and resources, ants agents suits for it and fulfill it without leaving any of burden on the network. Description of various agents, deployed in A2LB are:

Load Agent(LA): Controls the information policy and maintains the detail of a data center. The important work of load agent is to calculate the load on every available virtual machine allocating a new job in data center. It is supported by table termed as VM_Load_Fitness table.

VM_Load_Fitness table: Used for maintaining record of virtual machine specification in data center. It contains virtual machine id, status of its memory consumed by CPU utilization, fitness value and load status.

Migration Agent (MA): Mainly Initiated with channel agent. Will move to other data centers to communicate with load agent of that data center to get the status of virtual machines present there, which is looking for desired configuration. After receiving the suitable information it communicates it with the parent channel agent. Now it will stay at its destination location, waiting to get self-destroy message from parent channel agent. Status of migration agent can be live or destroyed based on its applicability.
Load agent acts as an proactive for calculating the load status of various virtual machines available in data center. It determines the workload of virtual machines in terms of available memory, available CPU utilization, and expected response time. Then it calculates fitness value of each virtual machines which is proportional to the memory of a machine.

Whenever a request arrives in a data center, after allocation resources to its load agent, load agent will update VM_Load_Fitness table to reflect present status of all VM. For this the load agents calculates percentage of $\mu$ and $\lambda$ since these both factors affect the processing of incoming requests. Based on value of $\mu$ available, fitness value ($\nu$) for that each node is generated. As long as $\nu$ of a node is greater than a threshold (25%), in this case VM status is normal. As the fitness value of VM becomes less than or equal to threshold value, load balancing needs to performed. Load agent when observes the critical status of a VM it will intimate and send the specification of that VM to channel agent. Then channel agent will interact with the migration agent to other data centers for searching similar specifications. Migration agent being ants will travel one way. On reaching a destination data center, migration agents sends acknowledgement message to its parent channel agent. Afterwards it checks with load agent of that data center for availability of virtual machines which are having similar configuration as desired. If no such virtual machines exists at the data center, migration agent sends a <Not-Applicable> message back to its parent channel agents and waits for <self_destroy>instruction from it. If one or more VMs having desired configuration are found, migration agent will further checks for their $\mu$ and $\nu$ sends it to channel agent.

4. Conclusion

This work focuses on load balancing in cloud computing. Load balancing in cloud computing was being ignored, but the rapidly growth in number of cloud users has raised demand for load balancing. This work has proposed an autonomous agent based load balancing mechanism which provides huge dynamic load balancing for cloud computing. Major contribution of this is load calculation of VM in a DC and when load of VM reaches near the threshold value, then load agent search for a candidate VM from other DCs. Keeping the information of candidate VM, reduces the service time. Result obtained from this algorithm is work satisfactorily.

5. References

[1] Autonomous Agent Based Load Balancing Algorithm in Cloud Computing Aarti Singha, Dimple Junejab, Manisha Malhotraa*
