International Journal of Engineering & Technology, 7 (1.1) (2018) 689-691

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Multi-agent based load balancing in cloud computing

S. Sandeep Kumar [1] *, S. Tarun Kumar [2], G. Sathraja [2], J. Nagatejasri [2]

[1] Assitant professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India
[2] 4th B.tech, Department of CSE, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India
**Corresponding author E-mail: ssandep794@gmail.com*

Abstract

Cloud knowledge centers area unit usually comprised of multiple servers with doubtless completely different specifications and unsteady resource usages. The challenge for these data centers is how to handle and service the millions of Requests such that the Quality of the ser-vice (Qos) is not compromised. Load balancing is an important aspect in cloud computing that involves an even work distribution among the available machines such that no machine is overloaded. This work discusses on numerous Agent primarily based load equalization tech-niques capableof equalization workloads across multiple servers to provide customer Satisfaction and economical Resource Utilization. We propose a Dynamic Multi-Agent based algorithm to address the load balancing issue .In this, we describe about Agents and how they can be used to solve load balancing in cloud computing.

Keywords: *Cloud Data Centers; Load-Balancing; Overloaded; Quality-of-Service (Qos); Underloaded.*

1. Introduction

Cloud computing: It is an on-demand service where resources like on-line office software and operational storage are made available as per the users requirement. The word cloud in cloud computing is used as a metaphor for internet since it offers various computing services like storage, servers and application are provided to devices over the internet.

1.1. Types of cloud services

Most of the services offered by cloud computing are often broadly speaking classified into 3 categories: (IaaS) Infrastructure as a service, (PaaS) Platform as a service and (Saas) Software as a service, can together be called as the cloud computing stack.

a) Infrastructure-as-a-service (IaaS): Provides Users with computing resources as well as servers, networking, storage and information center area on pay-peruse base.
b) Platform as a service (PaaS): Provides a application developing surroundings that has support to the whole lifecycle of building and maintenance of applications.
c) Software as a service (SaaS): Provides Users with software applications that can accessed via the internet without actually installing them.

1.2. Load balancing in cloud computing

To increase performance and resource utilization of the system, it's perpetually needs divide the employment between the accessible processors. The goal of load readjust is distribute the load among nodes in cloud setting, A Load readjust attempt to balance the entire system load between the nodes by transferring the employment between nodes such that system load is optimally distributed among available nodes to ensure good overall system performance.

1.2.1. Taxonomy of load balancing

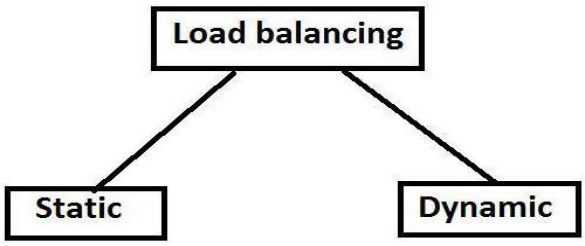

Fig. 1: Taxonomy of Load Balancing.

Static Load-balancing: Static Load-balancing algorithms use solely info concerning the typical behavior of the system. They do not think about the present state of system. Hence, Static algorithms are less complicated and not effective.
Dynamic Load-Balancing: Dynamic algorithms distributes load among the nodes using current state of node. However, they are complex than Static since they must collect and react to system state. Since, Dynamic policies consider current state they tend to be more efficient than Static algorithms.

1.3. Agent

An agent may be a process entity that acts on behalf of another object or objects to perform a task or bring home the bacon a desired goal. Agent systems are complete software system programs that in cooperates with domain information and having ability to behave with a particular degree of independence to hold out actions required to attain such that goals. Agents are implemented during away specified they will operate in a dynamic surroundings.

Copyright © 2018 S. Sandeep Kumar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1.4. Agent-based load balancing

Agent is an autonomous unit, which is capable of performing specified tasks on their own. Agent Based load readjust theme reduces the price communication of servers, since agent-server communication takes less processor time than centralized approach. It additionally improves the speed of load levelling that indirectly enhances the output and latency of the system.

2. Literature survey

In this, we have a tendency to explore the various algorithms of Agent primarily based load equalization mechanisms in cloud environment and also the issues in those areas. Through this we have a tendency to hit the systems that eliminate the mentioned for improvement of performance in cloud environment.

Che-Lun Hung, Hsiao-hsi Wang and Yu-Chen Hu [1] discussed Opportunistic Load Leveling Algorithm which is a Static load balancing algorithm. This system attempts to place the node dealing with randomly allocating tasks to the nodes. It provides load balancing without good results.

We have the work of T. Kokilavani, Dr. D. I. George Amalarethinam [2], proposed a Min-Min Load Balancing Algorithm. This is a static load balancing algorithm. In this, the cloud administrator contracts with the jobs requiring least performance time by conveying them to the processors first. The job with maximum execution time has to wait in the queue. This algorithm is efficient if jobs with minimum execution time are more. The major drawback of this algorithm is that it can lead to Starvation.

Che-Lun Hung, Hsiao-hsi Wang and Yu-Chen Hu [1], proposed an useful load balance algorithm, named Max-Min Load Balance algorithm. It is the similar as Min-Min algorithm excluding next finding the minimum execution time, the cloud manager deals with jobs with maximum execution time. This is a Static load balancing algorithm.

We have work of Ratan Mishra and Anant Jaiswal [3], proposed a solution of Load Balancing in Cloud by Ant Colony Optimization technique. This technique aims to find an find an optimal path between source of food and ant colony. This algorithm efficiently distributes jobs among nodes. Ants records data from every node and store them for future decision making. One algorithm using this approach is Agent-based dynamic Load-levelling in cloud environment that is discussed in the existing system.

From literature review, it has been observed that load readjust issue has been tried to solve mostly using static algorithms (i.e not considering the current state) and dynamic algorithms had not been paid much attention for load readjust in cloud environment and there is scope of research in this area of cloud environment. Next section provides the agent based framework for ensuring load readjust in cloud environment. Optimal path between source of food and ant colony. This algorithm efficiently distribute jobs among nodes.

3. Existing system

Agent-based dynamic load balancing: Below figure describes the Existing system design with many (assume n) sum of clients linked through cloud facility providers through web and facility supplier has simulated machines, controlling system, and m quantity of mutual group of resources that are seeing as clients.

Agent one life cycle comprises of two steps:

- In first step it actions as of Initial server to final server to collect data of all servers that are required for load balancing.
- In second step Agent equilibriums the capacity at servers constructed taking place the average load of the cloud.

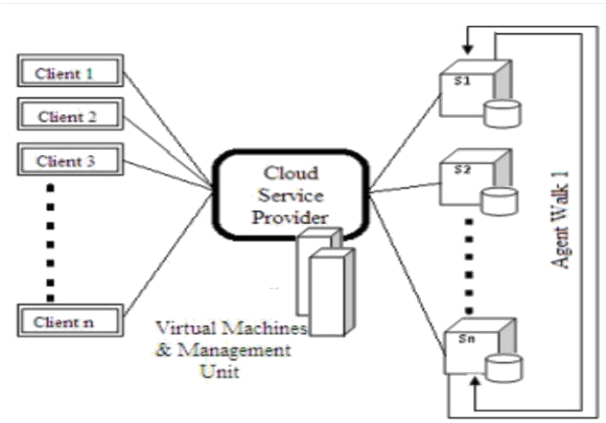

Fig. 2: Block Diagram of Existing Dynamic Load Balancing.

In Agent first step, agent is initiated at unspecific server (generally initial server) and this discovers number of jobs in line of that server. Thereafter it computes average, based on that, it will brains the server's position and it conclude if it is overloaded or under loaded. Agent will repeat this process till it extents last server.

In second step, Agent will jump to the final server from Initial server for corresponding load. At individual server it resolve crosscheck the state, If server is overloaded then jobs will be transmitted to under loaded server and if server is under loaded then it receives job from heavily loaded servers. Agent does this activity up to this one extents at the first server with readjust all servers load containing first server . In this method representative is used to balance the load among the structure nodes.

4. Proposed system

The existing system model Agent Based Dynamic load balancing approach has a disadvantage i.e for every instance of a new job, the agent has to complete a life cycle (i.e calculate average and decide the status of the server).Hence, Agents need to travel twice to allocate the job. Once to find the average work load and second one to balance the loads among nodes. To overcome this we propose a Multi-Agent Architecture for agent-based load balancing. Multi-Agent Architecture comprises of Directory agent and Mobile Agents. The proposed system architecture associates a mobile agent with each node, which contains the information about that node such as jobs allocated to the node, resources available with the node and a status variable to find its status. The Directory agent keeps track of the mobile agents and their status whether overloaded or under loaded. Thus, whenever a service request is received to cloud, the Directory agent checks the database for the status of the servers and assigns jobs accordingly. The proposed system consists of two kinds of agents. Namely, cloud mobile agents and directory agent.

Mobile agent (MA): It is associated with each and every server and is responsible for maintaining resource information of the server as well as their status at any point in time (whether under loaded or overloaded).Every mobile agent report the status of the server to DA periodically.

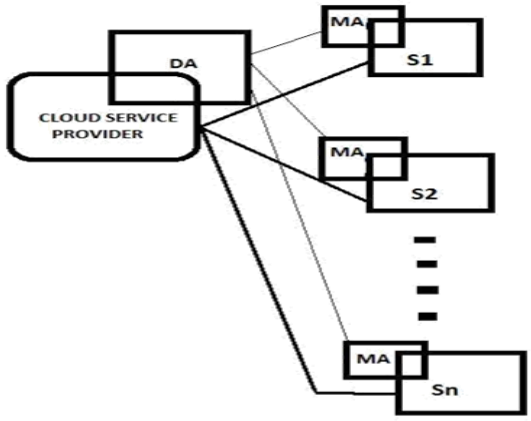

Fig. 3: Block Diagram of Proposed System.

Directory Agent (DA): Directory Agents keeps record of all Mobile agents registered with it, along with Status of the servers. Whenever a server is added its Mobile agent will have to get registered with a DA, which maintains their database necessary for providing Load balancing. Directory agent is responsible for load balancing. Directory agent also acts as request handler i.e whenever a new job arrives directory agent looks for status of all available MA and assigns them to the nodes such that the load is equally balanced among all nodes.
Initially the Mobile agent sends a registration request to Directory agent along with information of the server and status with which it is associated, in response the Directory agent sends back an acknowledgement signal indicating that the Mobile agent has got registered with it. Directory agent receive current status from all the servers and calculate the Average and sends it to all Mobile Agents.

$$\text{Average} = \frac{\sum J_i}{n}$$

Whereas,
J_i= Jobs at i^{th} server
N = Number of servers
MA assign their status to overloaded if number of jobs allocate are greater than average else to under loaded if number of jobs allocated are lesser than average.

For a server S_i
{
If (Ji <Average)
Then Set server's Status as "Underloaded "
Else
Set server's Status as "Overloaded"
}

Now whenever a Mobile agent sets its status to overloaded it sends request to directory agent demanding for the list of other Mobile agent registered with the Directory agent whose status is under loaded. The directory agent on receiving this request from the mobile agent searches its directory and provides the list of Mobile agent with under loaded status. On receiving the list from the Directory agent the Mobile agent sends load request to them and waits for their response. Mobile agent scans all the received responses and then finally assigns the in hand request to the Mobile agent that is most suitable both, in terms of less cost and faster services.

5. Conclusion

Cloud environment is one of the futuristic technologies on which technology giants are counting. In Coming-times, the users of cloud environment will definitely increase as there is good demand for cloud services. In such a scenario, there will be no doubt that load balancing will play a key role in providing efficient and reliable services to users. From research in cloud environment domain, we have identified load balancing problems in cloud technology surroundings. In this, we analyze several load balancing mechanisms using agents and their problems in clouding. We proposed a dynamic agent based algorithm to ensure load balancing in cloud environment environments using cloud mobile agents and directory agents. In Future we implement the proposed algorithm in cloud environment and compare its performance metrics to the previously existing algorithms.

References

[1] Che-LunHung, Hsiao-hsi Wang and Yu-Chen Hu –"Efficient Load Balancing Algorithm for Cloud Computing Network".
[2] T. KokilavaniDr. D.I. George Amalarethinam-"Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing."
[3] O. Elzeki, M. Reshad, M. Elsoud, "Improved max-min algorithm in cloud computing, International Journal of Computer Applications" (2012).
[4] Ratan Mishra and AnantJaiswal –"Ant colony Optimization: A Solution of Load balancing in Cloud".
[5] Jitender Grover and ShivangiKatiyar –"Agent based dynamic load balancing in Cloud Computing".
[6] SumanSangwan-"Load Balancing in Cloud Computing: A Review".
[7] Savitri, S.Sutagatti, S.G.Kulkarni-"Comparitive Analysis and Evaluation of Load Balancing Balancing Algorithms"
[8] Priyank Singh, HadaRanjita Singh and MukulManmohan – "Security Agents: A Mobile Agent Based Trust Model for Cloud Computing".
[9] Pamini Jain, Prof.SapnaChoudhary –"A Review of Load Balancing and its Challenges in Cloud Computing".
[10] Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen– "Live Migration of Virtual Machines".