International Journal of Progressive Research in Science and Engineering
Volume-1, Issue-6, September-2020
www.ijprse.com

Distributed Load Balancing algorithms for Cloud Computing-A Survey

Marakumbi Prakash R[1]
[1]*Assistant Professor, Department of Electronics & Communication, Tontadarya College of Engineering, Gadag, Karnataka, India.*
Corresponding Author: pmarakumbi@gmail.com

Abstract: -The significance of distributed applications is rising due to technological advancements such as increasing internetworking of various computing devices. Cloud computing is a technology paradigm that enables ubiquitous access to shared pools of configurable system resources. Load balancing is a mechanism to distribute the user requests among the virtual machines so that the requests are assigned proportional to the capacity of each virtual machine. Balancing of requests prevents the virtual machines from being either under loaded or over loaded. This paper presents a comparison of distributed load balancing algorithms in cloud computing. The algorithms considered for study are Throttled, Equally Spread Current Execution, Particle swarm optimization and Active Monitoring.

Keywords— ***Distributed systems, load balancing, cloud computing, Active Monitoring, swarm intelligence.***

I. INTRODUCTION

Advancement in computer networking technologies have led to enormous increase in the use of distributed and parallel computing systems. Cloud computing is a technology in the evolution of virtual computing, a field of information technology. Cloud computing is the delivery of computing services: networking, storage, servers, databases, software, analytics and more-over the Internet. It also provides the flexibility, scalability on services and virtualization type of services to the user. Public clouds have been an attraction to the industry as well as to the academia. As the number of users are increasing in cloud environment, challenges like load balancing, data loss, heterogeneity, security, delay have evolved. One among them is the load balancing which has been a issue for cloud service providers. To overcome the issue, different algorithms have been proposed. The main objective of load balancing algorithms is to utilize the software and hardware cloud resources efficiently and to maximize the throughput and resource utilization. With load balancing the response time also reduces and increases the performance of the system. Arise of load balancing issue, gave rise to different load balancing techniques. These techniques are classified as static and dynamic load balancing techniques. In static load balancing technique, the assignments of the requests to virtual machine are made a priori using information that includes arrival time, average execution time and amount of resources needed. The assignment of requests could be done probabilistically [7]. Static load balancing produces good results for stable environments. However, they cannot adapt to dynamic changes in structure of the system. On the other hand, dynamic load balancing technique is flexible and can provide good results in dynamic environment [5]. Dynamic load balancing algorithms evaluate the load conditions at the time of execution in making decisions for request transfer. The workload is not assigned statically to the hosts. However, in dynamic load balancing issues involved are when to invoke a load balancing operation, which host makes load balancing decision, how to collect information and migration of the load between the hosts are to be resolved [2]. The paper is divided into different sections. Section II mentions some of the previous works for load balancing, section III discusses the working of algorithms, Section IV gives the conclusion of the paper.

II. Related Work

There have been constant efforts to address distributed load balancing systems which is one of the significant issues in today's growing heterogeneous networks [10],[12]. In this section we discuss various methods that have been proposed to solve load balancing. Safiriyu Eludiora et al. proposed a solution for improving the quality of web services. The paper addresses unregulated tasks migration among servers. The purpose of the work is to control the movement of jobs to minimize bandwidth consumption. For evaluation of the algorithm Cooperative Adaptive Symmetrical Initiated Diffusion (CASID) the performance metric chosen were average response time and throughput. In paper [6], a different attempt has been made by the authors to solve load balancing issue in distributed systems. The method adopts genetic algorithm and game theory. The load balancing problem has been modelled as a non-cooperative game among users of the system. An algorithm based on the notion of Nash equilibrium is proposed for solving the load

balancing game. The result of the non-cooperative scheme using the genetic algorithm is stable in different states of the system. Athokpam Bikramjit Singh et.al.[11] have described different load balancing algorithms for cloud computing. The Load balance Min-Min Scheduling algorithm is used as its base. This method adopts a hierarchical framework consisting of three levels. The role of first level request manager is to receive the task and assign it to service manager which resides in the second level of the framework. After receiving the request, the request is divided into different subtasks by the service manager to enhance the processing. The request is then assigned to service node for execution considering the parameters such as remaining CPU space, the remaining memory and the transmission rate. This algorithm improves the load imbalance and minimizes the execution time of each node. In paper [14], the work proposes Autonomous Agent Based Load Balancing Algorithm which provides dynamic load balancing for cloud environment. Autonomous Agent Based Load Balancing Algorithm mechanism consists of three agents: Load agent, Channel Agent and Migration Agent. Load agent and channel agent are static agents and migration agent is an ant, which is a special category of mobile agents. The ants are deployed because of their ability to choose shortest path to their destination and are motivated from biological ants which seek a path from their colonies to the food source. The proposed system deploys various agents: Load Agent which maintains all details of a data centre. The major work of a load agent(LA) is to calculate the load on every available virtual machine after allocation of a job in the data centre. The information includes virtual machine unique identity, status of its memory consumed along with central processing unit utilization, fitness value and load status of all virtual machines. An interesting property of ants is that they move from source to destination for collecting desired information but they do not necessarily come back to their source rather they destroy themselves at the destination only thereby reducing unnecessary traffic on the network. Load agent from time to time determines the workload of VMs in terms of available CPU utilization, available memory and response time. Zhang Jiadong et.al.[16] have proposed a framework for cloud platform which includes a threshold window strategy and an auto regressive prediction model to reduce the migration of virtual machines.

III. Algorithms Under Study

The proposed paper describes four algorithms for comparison. The load balancing algorithms considered are Throttled, Equally Spread Current Execution, Particle Swarm Optimization and Active Monitoring algorithms.

A. Throttled Load Balancing Algorithm:

In this method the load balancer in the algorithm maintains an allocation table of virtual machines (VMs) [11]. The allocation table stores unique identification number for each virtual machine and their states. The state of a virtual machine is either BUSY or AVAILABLE. Based on the state of the VM the load balancer decides a suitable machine to perform the required task. Initially all the VMs are in the available state. When a request arrives at the data centre, it forwards it to the load balancer. The load balancer finds the allocation table from the beginning. The request is transferred to the 1st machine found with the state AVAILABLE, and the state is changed to BUSY. After processing the request, the VM notifies the data centre controller about the task completion and also informs the same to the load balancer which makes appropriate changes in the allocation table. If all the VMs are in BUSY state, the user request is queued up at the data centre. Response time to the request is measured as the time taken to search for an available virtual machine, time taken to process the request and transfer the request to the machine.

B. Equally Spread Current Execution Algorithm:

The load balancer assigns load equally to all the VMs connected to the data centre. The virtual machine load balancer maintains an index table of VMs. The allocation table stores the total number of requests currently assigned to each virtual machine. When a request arrives at the data centre for allocation of the new virtual machine, the load balancer looks the index table for least loaded virtual machine and returns the VM unique identification to the data centre controller(DCC). If in case more than one virtual machine is found to be lightly loaded, the virtual machine identified first is selected for processing the client request. After identifying a virtual machine, the DCC transfers the request to the corresponding VM identified by the identity. The load balancer updates the index table by increasing the allocation count of the virtual machine. When virtual machine completes the assigned task, the load balancer revises the index table by decreasing the allocation count by one for the identified virtual machine.

C. Particle Swarm Optimization

Particle Swarm Optimization is used to solve continuous optimization problems. Each particle represents a solution of the problem. Each particle consists of velocity and position vectors. The fitness of the candidate solutions is evaluated by the particles and the locations are remembered where the best success occurred. The individual's best result is called the particle's best. This information is made available by each particle to their neighbour's according to the particle's best result which are updated according to the best position of the particle so far and the best position of the swarm so far. Iterations proceed until the algorithm reaches a stopping criterion.

D. Active Monitoring Algorithm

This load balancing policy attempts to maintain equal workloads on all the available virtual machines. The Active Monitoring algorithm used is quite similar to the throttled case except that in this virtual machine selected for the request processing is the one having the least load.

IV. Conclusion

Load balancing is a thrust issue in the current research field of distributed systems. One of the reasons being the user requests growing exponentially. The proposed work discusses the comparison of the different load balancing algorithms - Throttled, Equally Spread Current Execution algorithm, Particle Swarm Optimization and Active Monitoring Algorithm. All the four algorithms have their own important properties and special characteristics. All these features can be combined to implement a new algorithm that will be best for the implementation of load balancing in all the aspects.

REFERENCES

[1]. Zhang Jiadong Liu Qiongxin Chen Jiayu "An Advanced Load Balancing Strategy for Cloud Environment" 17th International Conference on Parallel and Distributed Computing, Applications and Technologies, 2016, pp 240-243.

[2]. P. Neelakantan and S. Sreekanth "Task allocation in distributed system".

[3]. Michalis Mavrovouniotisa, Changhe Lib, Shengxiang Yang S "A survey of swarm intelligence for dynamic optimization: Algorithms and applications" Swarm and Evolutionary Computation, 33 (2017) 1–17.

[4]. Safiriyu Eludiora, Olatunde Abiona, Ganiyu Aderounmu, Ayodeji Oluwatope, Clement Onime, Lawrence Kehinde "A Load Balancing Policy for Distributed Web Service" Int. J. Communications, Network and System Sciences, 2010, 3, pp 645-654.

[5]. Amit S Hanamakkanavar, Vidya S.Handur "Load Balancing in Distributed Systems: A survey", International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN: 0976-1353 Volume 14 Issue 2 –APRIL 2015.pp 961-966.

[6]. Shweta Rajani, Renu Bagoria "A Semi-Distributed Load Balancing Algorithm Using Clustered Approach", IJCSMC, Vol. 3, Issue. 6, June 2014, pp.843 – 851.

[7]. Hajar Siar, Kourosh Kiani,Anthony T.Chronopoulos " A Combination of Game Theory and Genetic Algorithm for Load Balancing in Distributed Computer Systems", International Journal of Advanced Intelligence Paradigm,s June 2015.

[8]. Haitham Barkallah, Mariem Gzara, Hanene Ben Abdallah "Dynamic and adaptive topology-aware load balancing for Grids", IEEE 17th International Conference on Computational Science and Engineering,2014.

[9]. Deepak Kumar Patel, Devashree Tripathy, C.R.Tripathy "Survey of load balancing techniques for Grid" ,Journal of Network and Computer Applications ,2016,pp 103–119.

[10]. Hamid Shoja Hossein Nahid Reza Azizi "A Comparative Survey On Load Balancing Algorithms in Cloud Computing" 5th ICCCNT July 11- 13, 2014, Hefei, China.

[11]. Athokpam Bikramjit Singh, Sathyendra Bhat J, Ragesh Raju1, Rio D'Souza, "Survey on Various Load Balancing Techniques in Cloud Computing", Advances in Computing 2017, 7(2): 28-34.

[12]. Michalis Mavrovouniotisa, Changhe Lib, Shengxiang Yang "A survey of swarm intelligence for dynamic optimization: Algorithms and Applications",Swarm and Evolutionary Computation 33 (2017).

[13]. Jigna Acharya; Manisha Mehta ; Baljit Saini "Particle swarm optimization based load balancing in cloud computing", International Conference on Communication and Electronics Systems (ICCES),2016.

[14]. Agent Based Load Balancing Algorithm in Cloud Computing "International Conference on Advanced

Computing Technologies and Applications (ICACTA-2015).

[15]. Gema Ramadhan, Tito Waluyo Purboyo, Roswan Latuconsina "Experimental Model for Load Balancing in Cloud Computing Using Throttled Algorithm" International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 2 (2018) pp. 1139-1143.