Dynamic Load Balancing Model for Efficient Work Load Distribution in Cloud Computing

Sarita Simaiya[1], R Agrawal[2]

Institute of Engineering, CSJM,
Kanpur University, UP

***Abstract*: Cloud computing allows multiple users to access computing resources and services in the cloud. However, numerous challenges emerge throughout information sharing, like task scheduling in cloud computing. This article defines load balancing as a technique for evenly distributing workflow across all structure endpoints to maximize resource consumption and user acceptance. It facilitates the distribution and de-allocation of modeling in the task without collapse. This work presents a novel dynamic load balancing technique. The suggested load balancer shortens the completion response period throughout wide data applications run on the public cloud. Workforce planning is just an NP-hard problem that needs addressing. Our suggested methodology offers ways to minimize the search effort, resulting in a load balancing process that is less complicated. An experimental result demonstrates the performance of the proposed method in terms of better response time, less number of work migrations, and less waiting time.**

***Keywords*: Load balancing, Cloud Computing, Workload Distribution, Dynamic load balancing**

1. Introduction

Computing is a form of Internet-based modern computing in which VMware workstations can access standard computer power information and assets on demand. It is a strategy for enabling widespread, on-demand immediate access to a fully customizable pool of computational services, e.g., Web servers, data centers, storage, computer networks, software applications, and any web services. As a result, numerous strategies for adjusting consumers throughout information exchange are used. It is a framework that offers users a pool of configurable computing resources pool of computational and storage resources over the internet. It encompasses many concepts, including load balancing and scheduling [1].

Cloud computing is an emerging virtualization technology that utilizes high-speed Online services to transmit work activities from devices to small subgroups to decrease the waste utilizing services by providers, decrease software time complexity, and maximize revenue increase from application components [2]. Dividing workflows and computing attributes in a cloud technology environment are known as "load balancing." It allows companies to develop workloads and implementation requests by delivering distributed across multiple web servers, connections, and data centers. Cloud task scheduling entails maintaining the flow of task bandwidth and requests that occur well over the World Wide Web. Even as World Wide Web traffic proliferates, it will soon account for about 100% of the current traffic. As a result, the workload on the virtual machine is rapidly increasing, resulting in virtual machine overload, particularly for powerful platforms [3].

This article provides a dynamic load balancing model for cloud computing. This article defines load balancing as a technique for evenly distributing workflow across all structure endpoints to maximize resource consumption and user acceptance. The suggested framework distributes resources to multiple virtual machines first, accompanied by scheduling algorithms, to assist the materials supply at program execution. Our methodology introduces a novel load balancer that dynamically allocates resources to virtual servers and schedules tasks [4]. The following is a summary of the significant contributions:

- In virtualized environments, we present a novel dynamic load balancing algorithm that accomplishes performance and scalability dynamically.
- The proposed dynamic load balancer reduces the time required for cloud-based data processing to execute.

The article is divided into sections, covering section one introduction, two related works, three materials and methods, four discussions, and five covers conclusions and future work.

2. Related work

In order to maximize efficiency in cloud applications, many studies have been done on task scheduling. Most of them enhance connection speed, processing times, time to consider a decision, energy usage, resource optimization, and processing time. However, some overlook significant aspects like accessibility, frequency, and throughput. The significant works related to cloud load balancing are as follows-

Research [5] implements two computational modeling techniques for resource sharing and resource provisioning to virtual servers. The hill-climbing methodology was used to minimize turnaround time in the quality measures. Research [6] recommends an optimum solution approach, and the load balancer accomplishes optimal assessments at an affordable price. Post allotment is used if the load-balancing device fails to identify the optimum judgments. According to research [7], the static algorithm requires information regarding the number of tasks and available resources, but there is no need to supervise the recourses whenever the static method is in use; such algorithms produce more accuracy when the volume of work has been predictable.

According to research [8], the round-robin method distributes equivalent parts to nodes and requires them to wait to implement their activities in a predefined sequence without considering the number of resources available on each server or its time. According to research [9], dynamic load balancing with such a framework of measures and task allocation activities outperforms static methodologies. According to research [10], techniques in the emphasis collective should undertake assigned work and planning and control to use a server that accumulates all data from cloud network statistically or dynamically. Such methods decrease framework overhead but strain the node that handles allotment and planning.

3. Material and methods

The proposed model is characterized as a centralized technique. We think of the cloud as a collection of data centers with several underlying hardware (web servers) and some virtualization software operating the facilities consumers demand. Load impedance refers to the activities completed at the invitation of consumers [11]. The suggested framework accomplishes the distribution of resources to virtual servers first, accompanied by schedulers, to obtain excellent flexibility and efficiency, efficient work overload stabilizing between many resources, and encourage the availability of materials at program execution [12].

The Ant colony optimization methodology is utilized to improve designs and determine the best alternative for resource allocation to virtual servers and work schedule tasks [13]. This methodology, which fulfills the requirement of imperfect methods used to solve multi-objective optimization solutions and is a neighborhood search technique, may not provide relevant results for all input parameters.

3.1 Algorithm Proposed Dynamic load balancing model

// Algorithm: Dynamic load balancing model

1) Setup the Cloud Environments
    a. Create the desired number of virtual machines (VMs), brokers, data centers, cloudlets, and virtual machines (VMs).
    b. All of the servers are partially available for free
    c. Set VM=NULL for all the preliminary load
    d. A balancer keeps track of each VM's id, pace, memory, productive capacity, and processing capabilities in a list.
    e. Use a scheduler to determine the potential of each Virtual machine and label records in the load-balancing list.
    f. Set a preliminary threshold for every virtual machine.
2) The data center controller receives a new request.
    a. Cloud cloudlets accepted a new application.
    b. Determines the MIPS for each assignment
    c. Arrange the jobs in order of priority as well as completion time.
    d. Load Balancer has been appointed to the recommendation by Datacenter
3) apply Task Scheduling Strategies
    a. Create a scout for every virtual machine
    b. Verify if the applied load is equal to or higher than the predefined threshold
    c. MAP the VMs and Tasks if it is less than or equitable to the threshold value. Calculate your fitness value. Assign the task to the virtual machines.
    d. Whenever the load exceeds the VM's capabilities
    e. Delete the overburdened high-priority task from the overburdened virtual machines.
    f. Seek a new virtual machine allotment task (steps 3. a–3.e), calculate the optimal solution, and assign VM tasks to VMs.
4.) Repeat the above procedures 2 and 3 until no jobs remain unassigned.

4. Results and Discussion

The steps, constructive and effective, and evaluations used to verify the proposed methodology are summarized in this section. We briefly go about how to put the proposed methodology into practice and the simulation methods. The algorithm is compared to the Genetic Algorithm and Throttled scheduling methods. The methods were implemented in JAVA Netbeans IDE using the CloudSim structure [14]. We compare different virtual machine instances using response, turnaround, throughput, simulation, and distribution diagrams.

The whole section compares the outcomes of various simulation scenarios. The number of responses has been increased from 1500 to 6,000, and the outcomes for the methodologies are listed below. To verify the proposed method and evaluate, the features of virtual and physical devices inside the cloud are considered for simulation, as shown in Tables 1 and 2.

Table 1: Configuration of physical machines use in simulation

Method Type	MIPS	Storage Capacity in TB	RAM in TB	Bandwidth in GBPS
Type-1	120000	15	0.5 TB	20
Type-2	180000	30	1 TB	150

Table 2: Configuration of virtual machines use in simulation

Method Type	MIPS	Storage Capacity in TB	RAM in TB	Bandwidth in GBPS
Small Machine	250	0.1	0.2	100
Medium Machine	500	0.2	0.4	200
Large Machine	750	0.3	0.6	300
Extra-Large Machine	1000	0.5	1	500

Using the CloudSim simulator, the proposed dynamic load balancing method and existing GA, throttled load balancing methods are implemented, and the following measures are determined:

4.1 **Response Time:** This is the period between sending the request and receiving the first reply. The decrease aids the improved response time of the Virtual machines during the waiting period [15].

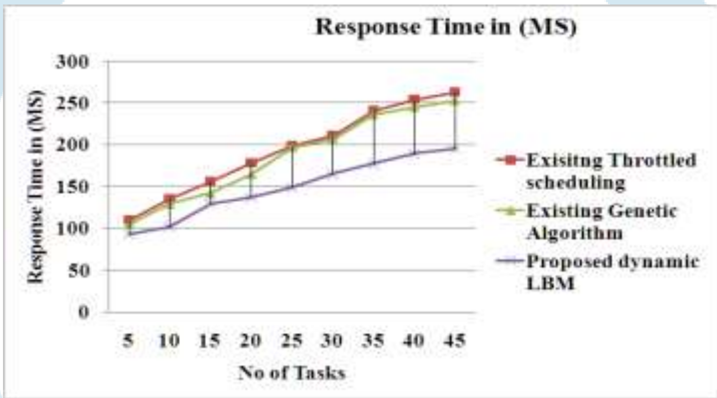

Figure 1: Responses time existing Vs. Proposed method

4.2 Degree of imbalance- Better performance can be seen with a reduced Im-balancing degree [16].

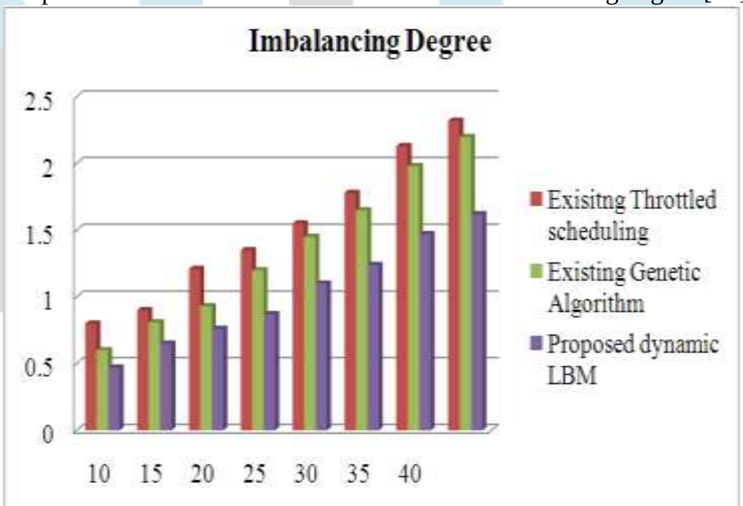

Figure 2: degree imbalance results in existing Vs. Proposed method

4.3 Waiting Time is the amount of time each method must join the queue before receiving its memory location. A shorter waiting period equates to effective improvement [17].

Figure 3: Waiting for time results existing Vs. Proposed method

Table 3 compares existing GA, Throttled, and proposed dynamic LB method simulation outcomes. The result demonstrates that the proposed dynamic LB method performs better in execution time, waiting time, and degree of imbalance.

Table 3: Comparisons of results

Comparison Parameters	Existing Throttled scheduling	Existing Genetic Algorithm	Proposed dynamic LBM	Remark
Make Span	Higher	Higher	Medium	Proposed dynamic LBM Better
Response Time	Higher	Higher	Average	Proposed dynamic LBM Better
Degree of Imbalance	Higher	Higher	Less	Proposed dynamic LBM Better
Waiting Time	Higher	Higher	Average	Proposed dynamic LBM Better
Execution Time	Higher	Higher	Medium	Proposed dynamic LBM Better

5 Conclusion and future work

One of the most fundamental elements of cloud computing is load balancing. It aids in reducing dynamic workload throughout all endpoints, resulting in increased consumer and resource satisfaction. It aids in reducing administrative costs, turnaround time, and flexibility. For processing data in the cloud, this work suggests an effective and flexible optimization-based load balancing framework that is dynamic. The research introduces a novel dynamic load balancing method for cloud computing. Comparing simulation outcomes among existing GA, Throttled and proposed dynamic LB methods demonstrates that the proposed dynamic LB method performs better in execution time, waiting time, and degree of imbalance.

References

[1] Shaw, Subhadra Bose, and A. K. Singh. "A survey on scheduling and load balancing techniques in a cloud computing environment." In 2014 international conference on computer and communication technology (ICCCT), pp. 87-95. IEEE, 2014.
[2] Mishra, Nitin Kumar, and Nishchol Mishra. "Load balancing techniques: need, objectives and major challenges in cloud computing-a systematic review." International Journal of Computer Applications 131, no. 18 (2015): 0975-8887.
[3] Ahmed, Tanveer, and Yogendra Singh. "Analytic study of load balancing techniques using tool cloud analyst." International Journal of Engineering Research and Applications 2, no. 2 (2012): 1027-1030.
[4] Chaczko, Zenon, Venkatesh Mahadevan, Shahrzad Aslanzadeh, and Christopher Mcdermid. "Availability and load balancing in cloud computing." In International Conference on Computer and Software Modeling, Singapore, vol. 14, pp. 134-140. IACSIT Press, 2011.
[5] Singh, Athokpam Bikramjit, Sathyendra Bhat, Ragesh Raju, and Rio D'Souza. "Survey on various load balancing techniques in cloud computing." Adv. Comput 7, no. 2 (2017): 28-34.
[6] Chen, Shang-Liang, Yun-Yao Chen, and Suang-Hong Kuo. "CLB: A novel load balancing architecture and algorithm for cloud services." Computers & Electrical Engineering 58 (2017): 154-160.
[7] Galloway, Jeffrey M., Karl L. Smith, and Susan S. Vrbsky. "Power aware load balancing for cloud computing." In Proceedings of the World Congress on Engineering and Computer Science, vol. 1, pp. 19-21. 2011.

[8] Gabi, Danlami, Abdul Samad Ismail, and Anazida Zainal. "Systematic review on existing load balancing techniques in cloud computing." International Journal of Computer Applications 125, no. 9 (2015).
[9] Katyal, M. and Mishra, A., 2014. A comparative study of load balancing algorithms in cloud computing environment. arXiv preprint arXiv:1403.6918.
[10] Mishra, S.K., Sahoo, B. and Parida, P.P., 2020. Load balancing in cloud computing: a big picture. Journal of King Saud University-Computer and Information Sciences, 32(2), pp.149-158.
[11] Kaur, Jaswinder, and Supriya Kinger. "A survey on load balancing techniques in cloud computing." International Journal of Science and Research (IJSR) 3, no. 6 (2014): 2662-2665.
[12] Sreenivas, Velagapudi, M. Prathap, and Mohammed Kemal. "Load balancing techniques: Major challenge in Cloud Computing-a systematic review." In 2014 International Conference on Electronics and Communication Systems (ICECS), pp. 1-6. IEEE, 2014.
[13] Kashyap, Dharmesh, and Jaydeep Viradiya. "A survey of various load balancing algorithms in cloud computing." Int. J. Sci. Technol. Res 3, no. 11 (2014): 115-119.
[14] Dasgupta, Kousik, Brototi Mandal, Paramartha Dutta, Jyotsna Kumar Mandal, and Santanu Dam. "A genetic algorithm (ga) based load balancing strategy for cloud computing." Procedia Technology 10 (2013): 340-347.
[15] Sajjan, R. S., and Biradar Rekha Yashwantrao. "Load balancing and its algorithms in cloud computing: A survey." International Journal of Computer Sciences and Engineering 5, no. 1 (2017): 95-100.
[16] Mesbahi, Mohammadreza, and Amir Masoud Rahmani. "Load balancing in cloud computing: a state of the art survey." International Journal of Modern Education and Computer Science 8, no. 3 (2016): 64.
[17] Begum, Suriya, and C. S. R. Prashanth. "Review of load balancing in cloud computing." International Journal of Computer Science Issues (IJCSI) 10, no. 1 (2013): 343.