International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Adaptive Load Balancing in Cloud Computing Environment

* Santosh T. Waghmode [1], Bankat M. Patil [2]

Submitted: 23/10/2022 **Revised**: 20/12/2022 **Accepted**: 04/01/2023

Abstract. The cloud computing platform load balancing is one of the challenge. Effective or efficient resource use in the cloud environment also known as load balancing became an important concern with the cloud computing platform's rapid development in users and their need for a variety of services. In order to enhance network capacity and reduce response time, load balancing play is important role in storage system. Our primary goal is to provide a novel load balancing method that, by combining efficient scheduling and cloud -based techniques, can balance incoming requests load from users across the world that are located in various regions to get data from distributed data sources. In addition to effectively utilizing cloud resources and improving task processing times, also worked on develop a dynamic load balancing algorithm that also ensures elasticity in a cloud environment. Divisible weighted round-robin, round-robin, and weighted least connection alorithm are the primary load balancing algorithms discussed in this paper.

Keywords: *Cloud computing, Divisible Weighted round robin, Least connection alorithm, Web server.*

1. Introduction

Every day, the need for computing and storage resources increases along with the IT industry. Large amounts of data are produced and transmitted through networks, demanding an increase in the need for computing resources. Load balancing [2] [3] is a technique that divides the workload among several nodes in the environment so that no nodes is overworked or left idle for any span of time. A load balancing method that is effective will ensure that each node in the system performs approximately equal amounts of work. The Load balancing algorithm objective is to distribute work to online services that are sent to the cloud domain, enhancing the overall response time and ensuring efficient resource use. In order to meet user requests and increase system performance, load balancing in the cloud primarily aims to dynamically balance the load across the nodes[2]. An effective or optimal load balancing algorithm helps to make the best use of the resources that are available, ensuring that no node is overloaded or load balance.

Scalability, the minimization of bottlenecks, and a shorter response time are all benefits of load balancing. It has been determined that effectively distributing the jobs throughout the system is an NP-complete problem [5].

[1]*Dr. Babasaheb Ambedkar Marathwada University, Aurangabad (MS) India*
[1*]*stwaghmode@gmail.com*
[2] *Dr. Babasaheb Ambedkar Marathwada University, Aurangabad (MS) India*
[2]*patilbankat@gmail.com*

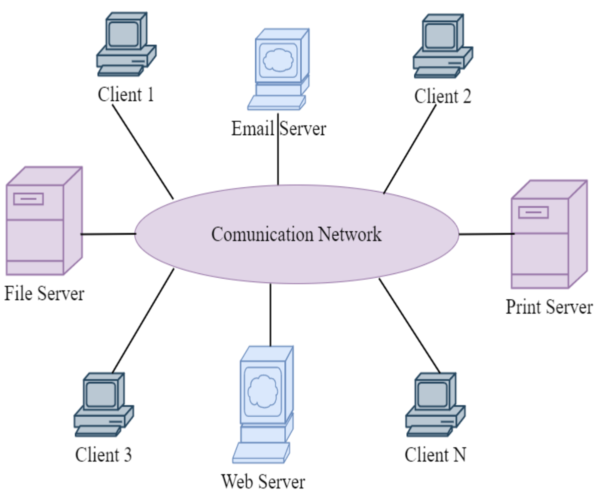

Fig. 1 Diagram for Load balancing [25][31]

The main drawbacks of existing systems include excessive power consumption from overloaded CPUs, increasing costs, and long maximum execution times [12]. The primary goal of this work is to focus on creating a framework for dynamic virtual server allocation implementing a resources webserver in the cloud. In order to achieve this goal, adequate resource provisioning is required, along with dynamic virtual machine placement that reduces total energy use [29] [31].

1.1 Load Balancing in Cloud Computing Envoirment

The dynamic allocation of workload is a key issue in cloud computing. The entire processing time needed by a machine to complete all of the tasks that have been delegated to it is referred to as the machine's workload [1]. By distributing the workload among the processors, load balancing works to increase system performance. Increased resource efficiency leads to improved overall performance and maximum client satisfaction, which are two advantages of workload. By evenly distributing the workload among all virtual machines in a cloud system, throughput is increased and response time is decreased [1] [3].

1.2 Types of approaches to load balancing based on the cloud environment

Static Load Balancing: In static algorithms [3], system characteristics such as processor speed, memory capacity, performance, and information on user needs are already known. These algorithms have important limitations in the event of a sudden failure of a system resource, tasks, and the inability to relocate tasks while they are being executed for load balancing.

Dynamic Load Balancing: Dynamic algorithms make load balancing decisions based on the system's current state, therefore prior system knowledge is not necessary. The limitations of a static method will be overcome by doing this. The dynamic algorithms are challenging, but they can offer higher speed and fault tolerance [1] [25]. The dynamic load balancing method uses some policies. These are best stated as follows:

- Transfer Policy: Transfer policy refers to the decision made by the dynamic load balancing algorithm while transferring a job from a local node to a remote node.
- Information Policy: Information policy or information strategy refers to the gathering of data about the system node for the load balancing algorithm.
- Load estimation Policy: This policy calculates the overall workload on a processor or device.
- Process Transfer Policy: It is used to decide whether a task should be carried out locally or remotely.
- Migration Limiting Policy: It provides a constraint on how frequently a task may move from one machine to another.

1.3 Types of Load Balancing Algorithms

Divisible weighted round robin: Depending on the standards the network administrator specifies, each server is given a weight; the most typical criterion is the server's ability to handle traffic. The server receives more client requests the greater the weight.

Least connections method: In tis algorithm calculates the load distribution for each node based on the quantity of connections. The quantity decreases when a connection is terminated or times out while it increases when a new connection is established.

Dynamic algorithm: The dynamic method seeks for the network's lightest server first and recommends it for load balancing. To boost system traffic, real-time communication with the network is necessary.

Dynamic algorithms are capable of producing load transfer decisions based on the system's current state. In order to achieve this goal, adequate resource provisioning is required, along with dynamic virtual machine placement that reduces total energy use.

1.4 Motivation and Research demands

Generally has the dynamic load balancing problem taken into account the heterogeneity of the computing system in the related study activity [1]. An effective mapping of jobs to the collection of computing nodes is the primary goal of the load balancing problem in distributed computational environments [3] [25].

The following are the main motivations for researching dynamic load balancing solutions in divisible weighted round robin algorithm:

- By creating effective job allocation algorithms that assign each task to the best possible processor node for completion, the processing speed of divisible weighted round robin may be fully realized.
- Distributed systems are facing challenges that must be resolved in order to meet the growing demands for better performance and services for various distributed applications. Distributed systems are continuing to increase in scale, heterogeneity, and different networking technology.
- Jobs execute at different speeds on computer systems nodes as a result of the heterogeneity of those nodes. Therefore, in this article research work focus on arranging i. e. divisible weighted round robin in contexts.

1.5 Objectives of Load Balancing

The objectives of load balancing are:

- To significantly system improve performance
- To protect against system failures and maintain website traffic
- To Handle Sudden Traffic Burst
- To keep the system stable.

2. Literature Review

In the past few years, a large number of static and dynamic load balancing algorithms have been proposed and developed for cloud environments. Static algorithms required the task and resource information at the time of build, i.e., before the execution never started. This kind of algorithm doesn't continuously monitor the state of the virtual machine [1]. However, because the cloud environment is dynamic and the cloud service provider never predicts incoming task requests and the present status of all cloud resources at the time of compilation, static algorithms are not appropriate for it. After a predetermined period of time, dynamic load balancing algorithms continuously monitor the VM [1] [2].

The focus of this publication during analysis is dynamic load balance. Additionally, depending on the migration procedure, dynamic load balancing is known as fixed and iterative load balancing. While iterative load balancing requires multiple steps, direct load balancing only needs one phase to balance the load. Interactive dynamic load balancing is the method for balancing the loads that is further divided into the diffusion and directional exchange methods.

The diffusion method diffuses more lead from fully loaded nodes to lightweight loaded nodes [3] [4][12].

For load balancing in cloud computing, there are various heuristic and meta-heuristic dynamic algorithms available, but each algorithm has its own limitations, meaning that no single algorithm can achieve all the goals at once because some parameters cannot be fully utilized or reduced simultaneously, such as cost and time [5][6].

Execution time reduces as the cost of adding more virtual machines rises because more virtual machines are available to do the task at hand. In the hybrid task scheduling algorithm, the equal distribution of tasks is the main goal in order to lower the communication costs of cloud resources while also achieving balanced scheduling according to the computing capacity of each node. Created a load balancing strategy to reduce task response times in a cloud environment [6][8].

The distributed system has achieved a dynamic load balancing using two different techniques, such as centralized load balancing and distributed load balancing. In the centralize load balance technique, decisions on load balancing are made by a centralized model that manages the distributed system [8]. Every node in a distributed system has metadata information that is kept on one of the architectural control nodes. Metadata is used by a cloud - based data load balancer, and decisions about file allocation or file movement balance the cluster [9] [10].

The author publishes one international journal, Load balancing in multi-agent distributed computer systems[11]. In this article, the load balance is attained with the use of agents. After that, agents are deployed throughout all nodes based on reputation value. According to the credit value, the migration node is selected. A method for distributed file system load balancing called adaptive data loading migration [12] [14] provides a comprehensive explanation of various types of load.

This research presents the network, input and output, and storage loads that are put on the system. Although network load and input and output load increase, the loading of any disc capability stays the same to the current data server Access condition of multiple clients [15]. The solution is provided by networked storage systems with paper users and centralized information processing. In this method, migration decisions are made by each user or application depending on what is best for that application. When network and disc speeds produce variations in device performance when multiple users share storage resources through striping [16] [17] [19].

In this article broker-based architecture for task scheduling and elasticity in cloud environment using cloud analyst as a tool for implementation [19] [25]. The proposed work is restricted by the load balance capacity to keep sessions across various virtual machines because it lacks the session affinity capability [26]. novel method that calculates the computational time of tasks while taking into account the importance of the work and the cost of the project.

3. Methods

Cloud load balancing is a technique for allocating workloads and computing resources in a cloud computing environment. There are two elementary solutions to overcome the problem of overloading on the servers. The first is a single-server approach in which the server is updated to a higher performance server. The new server can also quickly become overloaded, calling for another update. The upgrade process is very expensive and time-consuming. The second approach uses many servers to build a scalable service system on a cluster of servers. Building a server cluster system for network services is therefore more practical from a business viewpoint as well as more scalable.

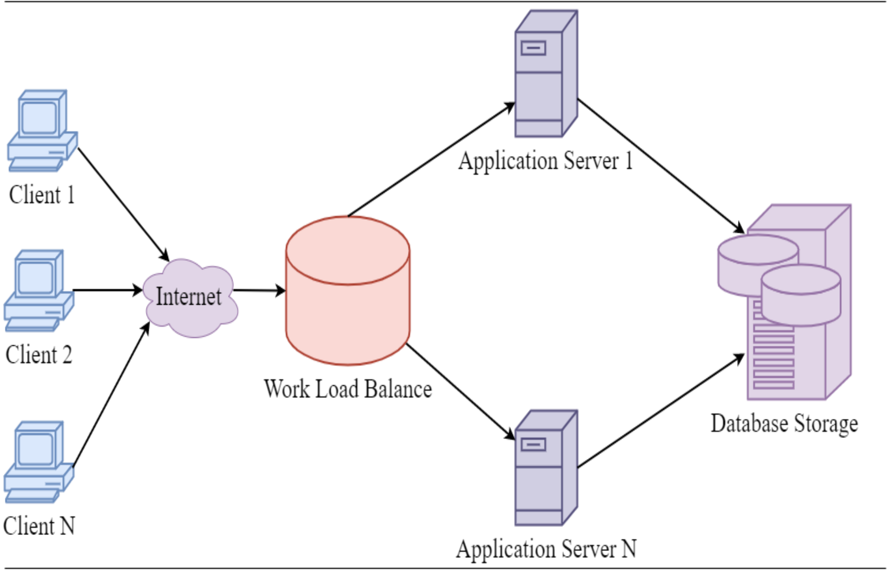

Fig. 2 Work load balancing system [25][30]

The technique was tested using a simulator. Cloud elements including Virtual Machines, Data Centers, Hosts, and Cloud servers are included in the simulator. Cost, disc size, RAM size, storage capacity, and other configuration details for the various components are stated[31]. An abstract cloud environment model is created by the integration of all the parts.

Simulator Modules:

- Virtual machines: Virtual machines make up each data center and are responsible for providing the clients' required resources.
- Service Packs: There are service packs for each virtual machine, e.g. virtual machine-l has a service pack for software, and virtual machine -2 has a service pack for software and network resources.
- Cloud Servers: The tasks or jobs that are now operating in the virtual machine are known as cloud servers. The matching resource pack is processed in response to the client's request, and the requested service is given to the client.
- Load Balancer: A divisible weighted round robin approach is used by default in the load balancing policy. Without taking into account the load on each virtual computer, this rounds-robin distributes all incoming requests to the available virtual machines.

Table 1: Adaptive Workload Balancer

Algorithm 1.1 Adaptive Workload Balancer

```
Input:  Number of Jobs such as J1, J2, …Jn, (J –Jobs) on
                    VM  M1,M2,M3.......Mn

Output: Jobs allotted on available VMs.

1.    Start
2.    Generate n task as jobs  j1 , j2 ……….jn gets at the load balancer
          where i=1,2……n.
3.    Classified based on their size, order the jobs in decreasing order.
4.    Initially, a load balancer would determine whether VMs were available.
5.    Assign the job tasks t1, t2, t3… tn to VM servers as per weight based
server state.
6.    IF (No. Of VM >= 1) then
1.                                                                    C
arry out checks on the jobs, or the tasks.
2.    Keep the virtual machines available (workstations).
3.    The VMs should be arranged in order of their weight-based load.
4.    The job would be allocated to an available VM through a specific data
centre after the load balancer searched the listed VMs' available VM IDs and Data
Center IDs.

7.    Else
      load balancer will look for weight based server
8.    Once a job has finished running in a virtual machine, the related job ID
and the output data will be returned.
9.
ind out how long it took to response to the results.
10.
f the pending jobs are still available, then go to step 6.
11.   End.
```

In out goal in designing this system was to produce a distributed system with at least five nodes that would deliver a collection of web services. Load balancing is a strategy that seeks to enhance network performance as well as the output from database servers. To boost productivity and speed up response times, distributions of visitor load balancing over a multi-course community are implemented [1] [3].

The network protocols respond to the operational request given by the client throughout the duration of the request, maximizing the space that is available and ensuring that no server is overworked while preserving performance.The Divisible weighted round robin load balancer distributes clients to the servers that are still online if a server goes offline [12] [31]. To find out if the underlying server can manage the amount of traffic, they do health checks.

4. Experiment & Result

The user makes a request to the Load Balancer. A Django application sends information to a Master slave server for the Process of Jobs. Data sent across multiple Minions by the Master Slave Distribute. Information about files, blocks, file to block mapping, block to minion mapping, and block storage for Minion will all be kept in the file system namespace of the master. A file will be generated by the master and stored on blocks and a minion. The first minion will get it from the client and pass it along to the next. The client will read the blocks in order from a list of locations that the master sends for reading. If a minion's reading is unsuccessful, it will move on to the another minion

The round robin, weight round robin random & dynamic load balancer technique is used to create load balance modules in Python. Round robin load balancing is one of the simplest techniques for dividing client requests across different servers. Each server in the group receives a client request as the round robin load balancer moves down the list of servers. Performance measures of virtual machines in terms of capacity of virtual machine, execution of time for job, total load for all virtual machine, throughput, scalability and latency as below:

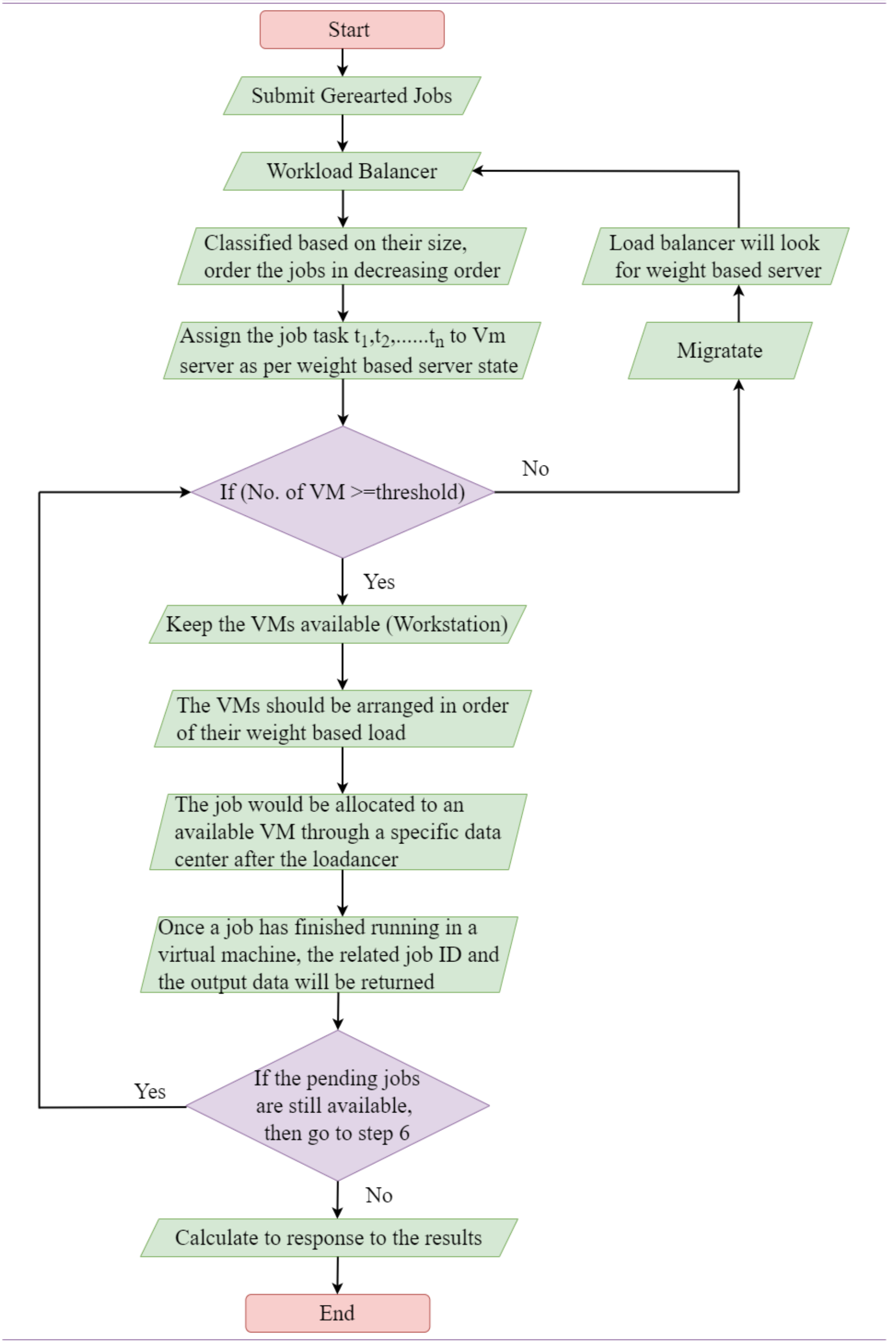
Start
Submit Gerearted Jobs
Workload Balancer
Classified based on their size, order the jobs in decreasing order
Assign the job task $t_1, t_2, ...t_n$ to Vm server as per weight based server state
If (No. of VM >=threshold)
No
Yes
Keep the VMs available (Workstation)
The VMs should be arranged in order of their weight based load
The job would be allocated to an available VM through a specific data center after the loadancer
Once a job has finished running in a virtual machine, the related job ID and the output data will be returned
If the pending jobs are still available, then go to step 6
Yes
No
Calculate to response to the results
End
Migratate
Load balancer will look for weight based server

Fig. 3 adaptive workload balancer flowchart

Using the given equation, we compute each virtual machine's capacity and the total capacity of all virtual machines:

$$Capacity\ of\ VMs\ = P * N + BWij \qquad (1)$$

Where P = Processing of Speed,
N=Number of CPU is used,
Bi-j=bandwidth between VM

Using the given equation, to calculate how long it will take a job to execute on a virtual machine:

$$Execution\ time\ for\ each\ job = Job\ size/P * N \qquad (2)$$

Using the given equation, to calculate total load for all virtual machines i. e datacenter:

$$Total\ Load = \sum_{j=1}^{m} Load\ of\ VMs \qquad (3)$$

Assign load at the virtual machine is:

$$Load\ of\ VMs = Number\ of\ job * Length\ of\ Job \frac{size}{P*N} \qquad (4)$$

The throughput statistic is determined by taking the entire work, translating it to hours from seconds, scaling it by the database volume (SF), and dividing it by the total amount of time (Ts) that has passed between the commencement of the first query and the completion of the last query or update function.

The frequency of successfully sent queries across various servers is known as throughput. The quantity of service delivered by the system in a single time unit is known as throughput. The ratio is calculated by formula such as:

$$y = Number\ of\ Jobs\ /period\ of\ evaluation \quad (5)$$

Scalability is the capacity of a process to grow to handle that growth as well as its ability to convey load balancing from clients to servers in a network. The transmission load ratio (TLR) is determined using a formula such as:

$$TLR = \frac{Assigned\ Load}{Observation\ duration} \quad (6)$$

For distributed web servers, load-balancing (salb), a file-by-file device, enables load-balancing. The mathematical approach for load balance in CC as shown in: Reduce the system's load imbalance and the total transfer costs incurred by the system as a result of load balancing procedures.

$$Ls = fsini$$

$$Lc = No. of\ request\ per\ unit\ time$$

Load considered in SALB is

$$L = f(Lc, Ls)$$

Step 5 in algorithm, determine utilization virtual machine using,

$$U = \sum_{j=1}^{Mi}(Ui \ldots..j) \quad (7)$$

Determine, mean of utilization of VM is

$$M = \frac{1}{P}\sum_{k=1}^{p}(Xk) \quad (8)$$

Where Xk is required total utilization of VMs in time i.e. k
P is denoted total time required

Variance of utilization of VMs is

$$V = \frac{1}{p}\sum_{k=1}^{p}(Xk - M)2 \quad (9)$$

Standard deviation is

$$StdDev = \sqrt{V} \quad (10)$$

Average of jobs at VMs is

$$Avg = U + M * StdDev \quad (11)$$

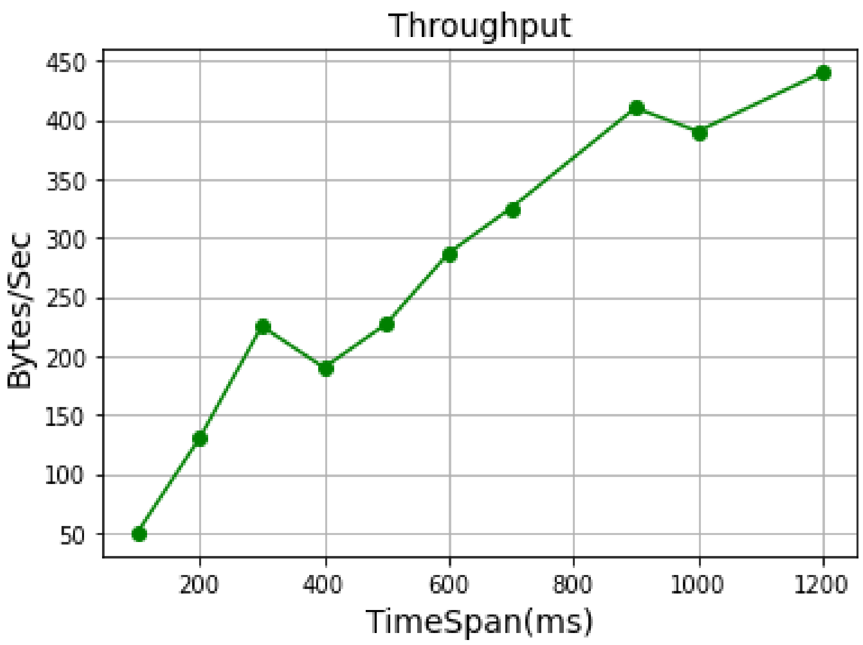

Fig. 4 Throughput over time result

In figure 4 shows Throughput is a metric measure how many data packets are successfully sent each second and is used to evaluate performance. The results indicate that the proposed method provides higher throughput than the conventional algorithms. On the X-axis of this graph, time has passed. The elapsed time in millisecond may be either actual or relative time. Y-axis represents to show how much data was transferred from the server to the client (in bytes/KB/MB) [3] [25].

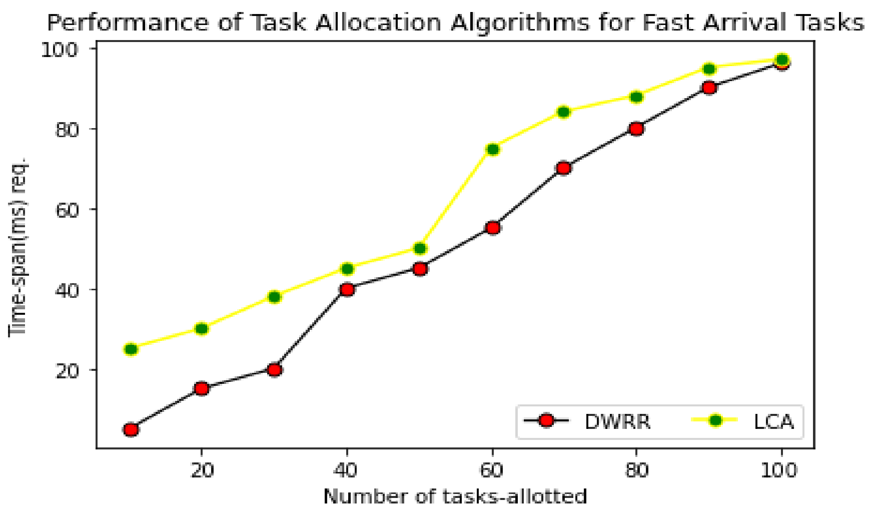

Fig. 5 Latency (ms) over request result of algorithms

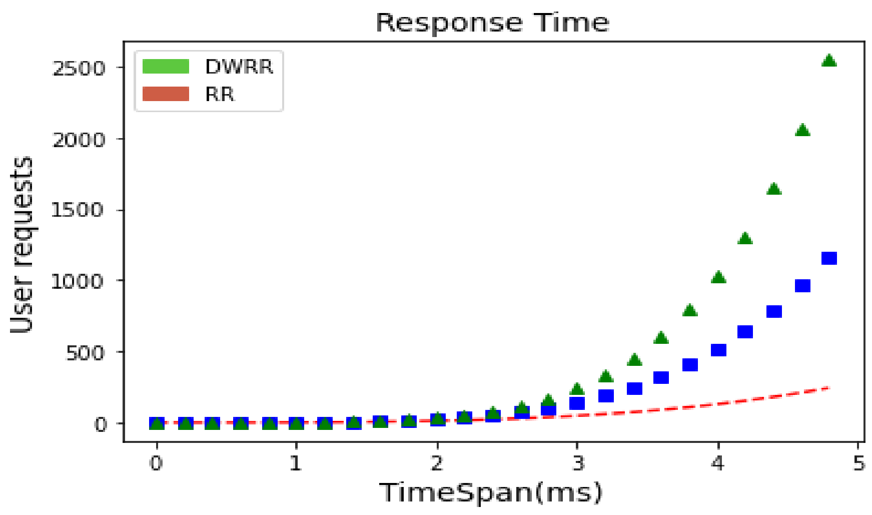

Fig. 6 Comparison of response time using proposed algorithms

In figure 6 compares the response times of the least connection algorithm, load balancing, and the divisible weighted round robin algorithms. Figure 6 provides the execution costs for the three algorithms. The three algorithms' execution response times are shown in fig 6. In comparison to round robin and least connection algorithm, the time required to conduct the load balancing with divisible weighted round robin algorithm is reduced by half [3] [25].

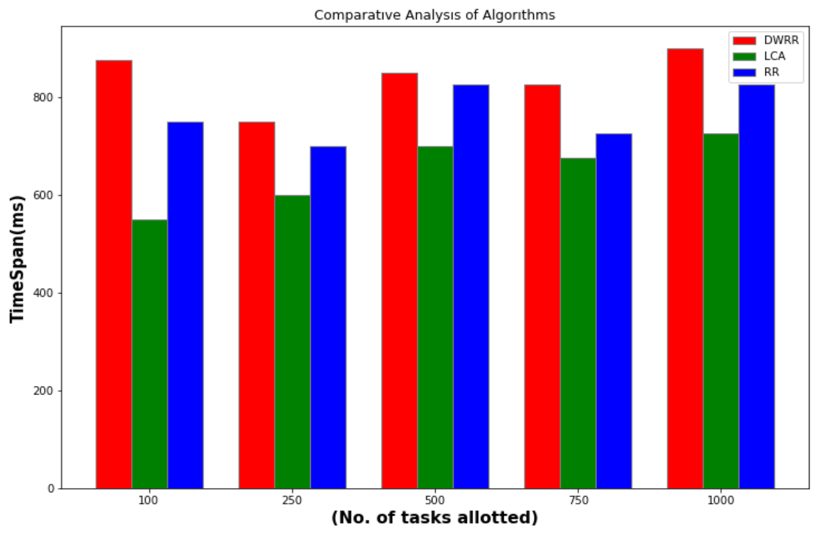

Fig. 7 Comparison of execution cost using proposed different algorithms

In figure 7 show bar plot for adaptive workload balance model with the help of different load balancer algorithms, the proposed algorithm successfully distributes incoming workloads across several virtual machines and processing speeds in various data centers. Comparison of execution cost between proposed algorithms, red color indicate for divisible weighted round robin and green color for least connection algorithm and blue color for eastic load balancer algorithm indicated [3] [25].

Table 2: Comparatıve Analysıs of Algorıthms [25]

Parameters	DWRR	RR	LCA
Scalability	Yes	Yes	Yes
Execution Time	Fast(8 Sec)	Fast	Slow
Turnaround time	Less	Less	More
Throughput	Low	Low	Low
VMs Load Distribution	Fast	Slow	Slow

Table II compared the proposed algorithm with the existing algorithms. Divisible weighted Round Robin, round robin, and least connection method are the algorithms that are come up for comparison.

Experiments are run, data are assessed, and discussions are presented on three metrics: average success rate, resource scheduling efficiency, and response time. This demonstrates the effectiveness of the divisible weighted round robin algorithm.

Impact of Average Success Rate: Average Success Rate (ASR) measures how many requests from users are successfully handled by the virtual machine grid through the resource manager at any given moment in a cloud environment. The following mathematical formula represents the average success rate.

$$ASR = \frac{Virtual\ machine\ grid's\ response\ rate\ to\ user\ request}{\text{Total no.of user requests}} * 100 \quad (12)$$

Impact of Resource Scheduling Efficiency: The ratio of the amount of resources allocated based on user request to the total number of resources in the cloud is referred as the resource scheduling efficiency. Efficiency in resource scheduling can be defined mathematically as follows.

$$\text{Resource Scheduling Efficiency} = \frac{\text{No.of resources are scheduled}}{\text{Total no.of resources}} * 100 \quad (13)$$

Impact of Response Time: Response time is the amount of time needed by a distributed divisible weighted round robin algorithm to respond while scheduling cloud resources. The following is a mathematical formulation of response time.

$$Response\ time = n(users\ request) * time(responed\ required) \quad (14)$$

5. Conclusion

The proposed divisible weighted round robin in this paper successfully handled the incoming jobs for distribution among various virtual machines located in various Data Centers [12]. In order to meet the growing demand for high performance web servers produced by the introduction of improved web environments, cluster-based web servers are used. Through the results of experiments, we evaluate the effectiveness of the proposed load-balancing technique [12]. According to Round-robin scheduling, divisible weight Round-robin scheduling, and weighted least connection alorithm scheduling the approach greatly reduces both the load range and load variance. To reduce response time and complete out load balancing without producing any delays, a divisible weighted round robin load balancer has been implemented.

References

[1] A. Nuaimi, N. Mohamed, A. Jaroodi, "A survey of load balancing in cloud computing: challenges and algorithm", in Proc - *IEEE 2nd Symp. Netw. Cloud Comput. Appl .NCCA*, 2012, pp. 137–142.

[2] Ammar Rayes, Bechir Hamdaoui, Mehiar Dabbagh and Mohsen Guizaniy, "Energy-Efficient Resource Allocation and Provisioning Framework for Cloud Data Centers", IEEE 2015.

[3] Aditya Narhe, Santosh Waghmode, 2019. SQL Query Formation for Database System using NLP. International Journal of Engineering Research and, V8(12).

[4] Babu D, Venkata P. Honey bee behavior inspired load balancing of tasks in cloud computing environments. Appl Soft Comput. 2013; 13(5):2292–2303.

[5] D. Fernández-Baca: Allocating modules to processors in a distributed system, IEEE Transactions on Software Engineering, Vol. 15, No. 11, pp. 1427-1436 (1989).

[6] Elzeki OM, Reshad MZ, Elsoud MA. Improved maxmin algorithm in cloud computing. Int J Comput Appl Technol. 2020; 50(12):22–27.

[7] Ektemal Al-Rayis, Heba Kurdi, "Performance Analysis of Load Balancing Architectures in Cloud Computing" IEEE Modeling Symposium (EMS), 2013 European, 20-22 Nov. 2013, 978-1-4799-2577-3.

[8] Haozheng Ren, Yihua Lan, Chao Yin," The Load Balancing Algorithm in Cloud Computing Environment", 2012 2nd International Conference on Computer Science and Network Technology, IEEE 978-1-4673-2964-4/12.

[9] Kumar, M, Subhash CS. Priority Aware Longest Job First (PA-LJF) algorithm for utilization of the resource in cloud environment. 3rd International Conference on Computing for Sustainable Global Development (INDIACom). New Delhi: IEEE; 2016.

[10] Load Balancing Algorithm Based on Estimating Finish Time of Services in Cloud Computing Nguyen Khac Chien, Nguyen Hong Son, Ho Dac Loc, 2016 18th International Conference on Advance Communication Technology (ICACT).

[11] Maha A Metawei, Salma A. Ghoneim Sahar M Haggag, Salwa M Nassar, Load Balancing in distributed multiagent computing system, Elsevier Aims Shams Engineer-

ing journal 2012.

[12] Mohit Kumar & S. C. Sharma "Dynamic load balancing algorithm to minimize the makespan time and utilize the resources effectively in cloud environment" International Journal of Computers and Applications, 23 Nov 2017.

[13] M. Randles, D. Lamb, T. Bendiab, "A comparative study into distributed load balancing algorithms for cloud computing", presented at Int. Conf. in Adv Inf Netw Appl Work, pp. 551–556.

[14] N. Ajith Singh, M. Hemalatha, "An approach on semi distributed load balancing algorithm for cloud computing systems" International Journal of Computer Applications Vol-56 No.12 2019.

[15] Naha RK, Othman M. Brokering and load-balancing mechanism in the cloud revisited. IETE Tech, Rev., 2014; 31(4):271–276.

[16] O. Kaneria, R. Banyal, "Analysis and improvement of load balancing in cloud computing", presented at Int. Conf. on ICT in Business Industry & Government, 2017.

[17] Onur Destanoglu, F. Erdogan Sevilgen, Randomized Hydrodynamic Load Balancing Approach, IEEE 1530-2016, 2008.

[18] Shanti Swaroop moharana, Rajadeepan d. Ramesh & Digamber Powar ," Analysis of load balancers in cloud computing" International Journal of Computer Science and Engineering (IJCSE)ISSN 2278-9960 Vol. 2, Issue 2, May 2017, 101-108.

[19] Sidra Aslam Munam, Ali Shah "Load Balancing Algorithms in Cloud Computing: A Survey of Modern Techniques" 2015 National Software Engineering Conference (NSEC 2015),IEEE, pp:30-35.

[20] S. Abed, D. S. Shubair, "Enhancement of task scheduling technique of big data cloud computing", presented at Advances in Big Data Comput. and Data Commun. Syst., 2018, pp. 1-6.

[21] S. Wang, K.Q. Yan, W. P. Liao, SS Wang, "Towards a load balancing in a three-level cloud computing network", in Proc. 3rd IEEE Int. Conf. on Comput. Science and Information Technology, 2010, pp. 108–113.

[22] S. Sonkar and M. Kharat, "A Review on Resource Allocation and VM Scheduling Techniques and a Model for Efficient Resource Management in Cloud Computing Environment," in ICTBIG, Indore, India, Nov. 18-19, 2016, pp. 1-7, doi: 10.1109/ICTBIG.2016.7892646.

[23] S. Abed, D. S. Shubair, "Enhancement of task scheduling technique of big data cloud computing", presented at Advances in Big Data Comput. and Data Commun. Syst., 2018, pp. 1-6.

[24] Shridhar G.Damanal and G. Ram Mahana Reddy, "Optimal Load Balancing in Cloud Computing By Efficient Utilization of Virtual Machines", IEEE 978-1-4799-3635-9/14.

[25] Waghmode, S. and Patil, B., 2021. Load Balancing Technique in Distributed Systems: A Review. In: 2021, 2nd Global Conference for Advancement in Technology (GCAT). Bangalore, India: IEEE.

[26] Waghmode, S. and Patil, B., 2022. Optimized and adaptive dynamic load balancing in distributed database server. [Online] Digital-library.theiet.org. Available at: <https://digital-library.theiet.org/content/conferences/10.1049/icp.2022.0607>, 2 October 2022.

[27] X. Shao, M. Jibiki, Y. Teranishi, N. Nishinaga , "Effective load balancing mechanism for heterogeneous range queriable cloud storage", presented at IEEE 7th Int. Conf. Cloud Computing Technology, 2015, pp. 405–412.

[28] X. Shao, M. Jibiki, Y. Teranishi, N. Nishinaga , "Effective load balancing mechanism for heterogeneous range queriable cloud storage", presented at IEEE 7th Int. Conf. Cloud Computing Technology, 2015, pp. 405–41.

[29] Zhipeng Tan, Wei Zhou, Dan Feng, and Wenhua Zhang , ALDM: Adaptive Loading Data Migration in Distributed File Systems, IEEE transactions on magnetics, vol. 49, no. 6, June 2013.

[30] https://www.znetlive.com/multi-cloud/