*International Journal for Modern Trends in Science and Technology*
*Volume 10, Issue 04*, pages 381-385.
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol10issue04.html
DOI: https://doi.org/10.46501/IJMTST1004059

# A Dynamic Load Balancing Strategy for Optimizing Resource Utilization in Cloud Data-Centres

**A.Avinash [1] , Yandamuri Venkata Naga Satya Jaswanth [2], Nookala Divya Durga Veera Sai Avinash [2], Neela Harika [2], Dasari Bharadwaj [2], Malleswarapu Teja Mahindra [2]**

[1]Assistant Professor, Department of Computer Science Engineering, Pragati Engineering College, Surampalem , Andhra Pradesh, India.
[2]Department of Computer Science Engineering, Pragati Engineering College, Surampalem , Andhra Pradesh, India.

**To Cite this Article**
A.Avinash, Yandamuri Venkata Naga Satya Jaswanth, Nookala Divya Durga Veera Sai Avinash, Neela Harika, Dasari Bharadwaj, Malleswarapu Teja Mahindra, A Dynamic Load Balancing Strategy for Optimizing Resource Utilization in Cloud Data-Centres, International Journal for Modern Trends in Science and Technology, 2024, 10(04), pages. 381-385. https://doi.org/10.46501/IJMTST1004059

**Article Info**
Received: 06 April 2024; Accepted: 18 April 2024; Published: 26 April 2024.



## ABSTRACT

***Despite much previous research in the Cloud Computing field, several issues remain related to workload balancing in cloud-based applications, particularly in the infrastructure as a service (IaaS) cloud model. Due to the limited number of resources/virtual machines available in cloud computing, efficient job allocation is critical. IaaS is one of the technological models that manages the backend, which includes servers, data centres, and virtual machines. Cloud service providers should provide good service delivery performance in such models, avoiding circumstances where hosts are overloaded or underloaded, which would result in longer execution times or machine failures, among other things. Task Scheduling significantly adds to load balancing, Task scheduling closely adheres to the standards of the Service Level Agreement (SLA), a contract provided by cloud developers to consumers. The LB algorithm addresses important SLA parameters such as the deadline. The proposed technique aims to optimize resources and enhance Load Balancing while considering Quality of Service (QoS) job characteristics, VM priority, and resource allocation. Based on the findings of the literature, the suggested LB algorithm covers the difficulties raised as well as the existing research gap. When compared to the existing Dynamic LBA algorithm, the new LB algorithm uses 78% more resources on average. It also performs well in terms of execution speed and make span.***

***Keywords: Cloud computing, load balancing, makes pan, optimization, QoS, SLA, task scheduling.***

## 1. INTRODUCTION

As we move more towards online storage and services, Cloud Computing technology becomes an integral component of the company. This technology offers a variety of services, including software via web browsers and platforms for creating and implementing cloud-based applications. Cloud Service Providers (CSPs) oversee the backend of the infrastructure, such as data centers and servers. Although there are many different service delivery methods in this technology, this research focuses on the Infrastructure as a Service

(IaaS) paradigm. It focuses on the server-side of this technique for resource allocation [1].

Virtualization is the foundational and critical component of cloud-based applications [2]. This method can have a major impact on the performance of scalable and on-demand services delivered to customers if the migration process and virtual machine resource allocation are not managed correctly. According to the findings [3], cloud performance is one of the top three issues in cloud computing. This study tries to improve resource allocation in the IaaS paradigm; this idea is crucial since it deals with the balance of resources offered to clients and workload/user demands on servers.

Cloud users access services by sending requests, which are stored in Virtual Machines (VMs) in the cloud environment. CSPs should provide services that benefit businesses and improve customer pleasure. Thus, the suggested Load Balancing algorithm is created primarily concentrating on the IaaS model out of the three service models in the cloud, where authors deal with the Cloud Computing technology's backend, such as server workload.

A basic cloud environment consists of two components: the frontend, which is the user side and may be accessed over the Internet. The backend side manages cloud service models, while the Data Center houses several actual computers (known as servers). Incoming user requests from the program are dynamically scheduled, and clients are provided the required resources via virtualization. The virtualization approach is also responsible for load balancing throughout the system, scheduling, and resource allocation. CSPs and cloud consumers may take use of both virtualization and dynamic job scheduling approaches. Thus, smart scheduling may significantly reduce execution time and boost the resource utilization ratio in cloud-based systems.

## 2. LITERATURE SURVEY

This section contains the literature review for this paper. The notion of load balancing will be discussed, with a focus on the model, measurements, and existing standard techniques. This leads to the contemporary literature on load balancing, in which researchers' suggested algorithms are described and analysed, followed by existing algorithms proposed by researchers in the field of load balancing. An organization chart for section III is shown. First, the subsections cover Task Scheduling and Load Balancing, emphasizing their usefulness in the cloud context. The constraints that are addressed in this proposed study are then highlighted using recent literature on their methodologies.

"Cloud computing: A paradigm shift in the way we compute," Introduction Cloud computing is a new computer paradigm in which resources such as storage, processing power, networking, and applications are supplied as services. Customers can access these services through a subscription-based approach, sometimes known as pay-as-you-go. Customers can access these services on demand regardless of where they are hosted, and they are charged based on their utilization of the services. Cloud computing transforms resources into virtual, limitless entities. Also, the resources may be provided from anywhere, making them always available at any place. So, cloud computing represents a new paradigm in which we may dynamically provide resources, create programs, and access platform-independent services. Cloud computing, the successor of internet computing, is a technology that incorporates the concepts of utility, scalability, and on-demand services. Figure 1 depicts "Internet Computing" against "Cloud Computing". Define Cloud in IT. According to the U.S. National Institute of Standards and Technology (NIST), Cloud is a classical model that enables omnipresent, readily available network access to a publicly accessible pool of adjustable resources like servers, storage, network components, and applications; that can be accessed, manipulated, and released with minimal management effort, Lower costs and fewer involvement with service providers. The following essential qualities characterize cloud computing. Service on demand: Cloud customers may get services whenever and wherever they need them, without having to communicate directly with the cloud service provider. Wide network access: Services can be accessible over the network using a variety of devices (such as laptops, mobile phones, PDAs, tablets, and office computers). Services may be deployed on any platform; hence cloud services are platform agnostic. In cloud computing, resources are pooled together, allowing cloud providers to offer multi-tenant services.

Multi-tenancy allows several users to be serviced at the same time using both real and virtual

resources. These assets can be assigned dynamically and released based on the user's preferences. Increased elasticity: There are no limits to how many resources may be provisioned via the cloud. As a result, services may be scaled up and down fast. For instance, an online shopping site makes advantage of cloud resources in terms of users.

Toward cloud computing: security and performance. Security and performance are fundamental needs for every system. They are regarded as the standard for measuring any advancement in a security system. Security is an indication that influences the degree of performance due to risks that impact the performance of cloud components during service rendering. Both security and performance illustrate the effectiveness of cloud computing, implying that performance and security are indicators of the cloud's level of development. In this study, the link between performance and security will be explored to determine the degree of their influence on the advancement of cloud computing.

## 3. SYSTEM ANALYSIS

### A. EXISTING SYSTEM

This part examines earlier existing methods in the fields of Load Balancing and Task Scheduling. Many modern algorithms have sought to enhance task scheduling and load balancing. However, a few constraints remain owing to the underlying fundamental algorithms utilized, such as Round Robin or First Come, First Serve. These algorithms have the potential to lengthen the wait time or create a gap when scheduling jobs. The authors presented a dynamic load balancing technique to reduce Make span time and optimize resource utilization. It ranks jobs based on length and processing performance using the bubble sort algorithm. Then, jobs are assigned to Virtual Machines in a First-Come, First-Served basis. After allocation is completed, the load is balanced by considering and calculating the load of virtual machines. This method may quickly optimize resources and shorten Make span; however, it does not consider priority or any QoS criteria such as Deadline.

### DISADVANTAGES OF THE EXISTING SYSTEM

**1. Limited Accuracy:** Traditional systems sometimes suffer from low accuracy rates, especially in demanding conditions, resulting in false alarms or delayed replies.

**2. Lack of Adaptability:** Conventional systems may have difficulty adapting to changing circumstances and shifting firing patterns, making them less effective in dynamic settings.

**3. Manual Intervention:** Many present systems depend on manual surveillance and human interaction, which can cause delays in identifying and responding to fire occurrences.

**4. Limited Remote Monitoring:** Monitoring measures in distant places are frequently insufficient, making crucial locations subject to fire-related dangers.

### B. PROPOSED SYSTEM

In this paragraph, we outline the study aim in an example diagram to clarify the Load Balancing problem and the function of the proposed LB algorithm, as shown in Figure 3 below. The primary purpose of this suggested approach is to enable optimal resource allocation in a cloud environment while avoiding imbalanced workloads in cloud computing applications. This paradigm addresses concerns with workload transfer and task rejection in the cloud. The suggested structure contains two levels:

- Top Layer: handles requests from numerous clients (application users) on both mobile and desktop. Clients can connect to the Internet via various devices and submit request to the cloud. In this layer, the model employs the Cloudlet Scheduler Time Shared method to send in tasks in a random sequence (Arrival Time) and allocate them to Virtual Machines using two important parameters: Deadline and Completion Time. A data center (DC) is a huge storage facility that houses cloud servers and data. DC takes requests and sends them to the active load balancer. In this layer of the architecture, the proposed approach is implemented as a load the balanced system, among others, which functions as the principal balancing in the cloud environment to complete migration. To the author's knowledge, this was not addressed in previous work.
- Bottom Layer: handles the allocation of user requests to virtual machines (VMs). As shown in the picture, we have a major batch of VMs; VM2's status is assigned to high priorities since it violates the SLA prerequisite, implying that its Completion Time exceeds the Deadline. Thus, the suggested LBA should use a migration mechanism to move the burden to another accessible Virtual Machine by altering the MIPS of both VMs both

before and after assigning resources to them. The resource allocation table is then updated anytime a Virtual Machine is violated or not, along with the quantity of requests it has been assigned. There is one situation in which there is no SLA violation. Assume the Time to Complete (TTC) is shorter than the SLA (Deadline) for jobs running on virtual machines. Then no SLA violations occur. Overall, the proposed structure provides continuous planning and balanced load to fully use both the CPU and cloud resources.

## ADVANTAGES OF THE PROPOSED SYSTEM

1. **High Accuracy:** The system uses powerful Convolutional Neural Networks (CNNs) and transfer learning to achieve an amazing accuracy rate of 99.5% in fire and smoke detection, reducing false alarms and assuring trustworthy notifications.
2. **Real-time Detection:** It allows real-time monitoring of CCTV camera feeds, allowing for early identification of fire or smoke occurrences and prompt action, minimizing the risk of fire-related tragedies.
3. **Adaptability:** The system's use of transfer learning enhances its adaptability to evolving fire patterns and changing environmental conditions, ensuring continued effectiveness.
4. **Remote Monitoring:** It extends surveillance capabilities to remote and challenging environments, where traditional systems often fall short, enhancing safety in critical areas.

## 4. SYSTEM DESIGN

### SYSTEM ARCHITECTURE

Below diagram depicts the whole system architecture.

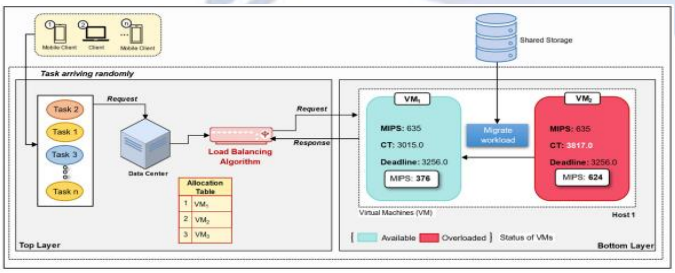


**Fig 1. Methodology followed for proposed model**

## 5. SYSTEM IMPLEMENTATION

### MODULES

The suggested technique improves cloud performance by addressing both task scheduling and load balancing. It makes use of all available CPUs in computers and arranges jobs effectively to decrease Make span and execution time while maximizing resource consumption. The following are the assumptions made in the suggested algorithm:

1.Each Virtual Machine (VM) can have one or more cloudlets (also known as tasks or user requests).

2. Cloudlets come in random order (Arrival Time). Each Cloudlet has a length, a time to finish known as the deadline (contained in the Service Level Agreement document), a completion time, and, most importantly, the arrival time.

3.The suggested approach compares the completion time for each task (a total of cloudlets) to the deadline.

4. If a VM's completion time exceeds the deadline, the suggested method adjusts its priority based on CPU. If successful, the cloudlets are scheduled, otherwise the VM's workload will be migrated.

5. To determine expected completion time, divide the cloudlet length (MIPS) by the Virtual Machine MIPS (CPU).

6. Initially, all VMs share an equal amount of available CPU, which is later reconjured based on violation status. The CPU is set to full usage in the suggested approach.

## 6. RESULTS AND DISCUSSION

The goal of this experiment is to demonstrate the decrease of Make span, execution time, and increased resource usage in a dynamic cloud environment. During algorithm testing, we evaluated pre-emptive task scheduling. This implies that if the workload violates the SLA, the job can be paused and transferred to another resource to continue the execution, as demonstrated in. During the scheduling process, numerous QoS performance criteria of cloudlets are examined, such as:

========== OUTPUT ==========

| Cloudlet ID | STATUS | DC ID | VM ID | Time | Start Time | Finish Time |
|---|---|---|---|---|---|---|
| 8 | SUCCESS | 2 | 3 | 90.01 | 0.1 | 90.11 |
| 24 | SUCCESS | 2 | 6 | 126.05 | 0.1 | 126.15 |
| 16 | SUCCESS | 2 | 5 | 163.14 | 0.1 | 163.24 |
| 14 | SUCCESS | 2 | 4 | 183.28 | 0.1 | 183.38 |
| 15 | SUCCESS | 2 | 4 | 195.09 | 0.1 | 195.19 |
| 21 | SUCCESS | 2 | 6 | 226.25 | 0.1 | 226.35 |
| 3 | SUCCESS | 2 | 1 | 255.81 | 0.1 | 255.91 |
| 11 | SUCCESS | 2 | 3 | 312.27 | 0.1 | 312.37 |
| 10 | SUCCESS | 2 | 3 | 369.06 | 0.1 | 369.16 |

========== OUTPUT ==========

| Cloudlet ID | STATUS | DC ID | VM ID | Time | Start Time | Finish Time |
|---|---|---|---|---|---|---|
| 13 | SUCCESS | 2 | 2 | 97.87 | 0.1 | 97.97 |
| 2 | SUCCESS | 2 | 3 | 124.52 | 2.1 | 126.62 |
| 5 | SUCCESS | 2 | 2 | 173.31 | 10.1 | 183.41 |
| 0 | SUCCESS | 2 | 1 | 246.24 | 0.1 | 246.34 |
| 7 | SUCCESS | 2 | 4 | 254.8 | 11.1 | 265.9 |
| 8 | SUCCESS | 2 | 1 | 416.64 | 3.1 | 419.74 |
| 9 | SUCCESS | 2 | 2 | 457.08 | 1.1 | 458.18 |

*Fig 2.* Same arrival time & random arrival time.

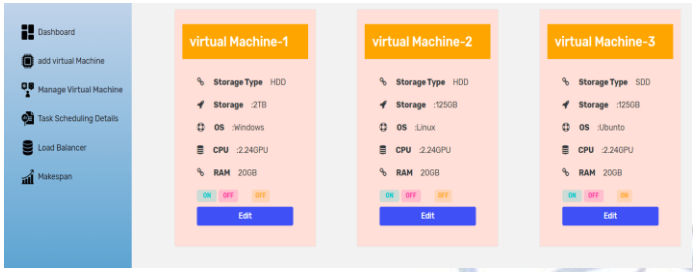


fig 2. Load balancing

## 7. CONCLUSION AD FUTURE WORK

This part wraps up the article by discussing the outcomes and discoveries obtained using the suggested LB algorithm. As evidenced by the literature, work scheduling makes a substantial contribution to load balancing in the cloud. Using Task Scheduling to improve the Load Balancing process can result in more effective use of cloud resources. This work attempted to develop a more advanced load balancing approach. The results showed that, compared to standard Dynamic LBA, our solution uses resources 78% more efficiently while minimizing Make span. It also demonstrates that the suggested technique may be used in a dynamic cloud environment, where user requests arrive in a random order and regularly change in duration. Furthermore, the technology can handle bigger queries than the prior methods. The approach solves SLA breaches in virtual machines by reallocating resources to guarantee efficient task execution. In the future, the authors want to better optimize cloud resources and increase cloud-based application performance by taking into account more SLA criteria. For example, the approach will be evaluated depending on the amount of violations and migrations in order to increase performance. Furthermore, the approach will be carefully compared with other current algorithms in the literature.

### Conflict of interest statement

Authors declare that they do not have any conflict of interest.

## REFERENCES


[1] H. Shukur, S. Zeebaree, R. Zebari, D. Zeebaree, O. Ahmed, and A. Salih. "Cloud computing virtualisation of resource allocation for distributed systems." Journal of Applied Science and Technology Trends, vol. 1, no. 3, pp. 98-105, June 2020, doi: 10.38094/jastt1331.

[2] M. Agarwal and G. M. Saran Srivastava, "Cloud computing: A paradigm shift in the way of computing," International Journal of Modular Education and Computer Science, vol. 9, no. 12, pp. 38-48, December 2017, doi: 10.5815/ijmecs.2017.12.05.

[3] N. Zanoon, "Toward cloud computing: Security and performance," International Journal of Cloud Computing: Services Architecture, vol. 5, no. 5, nos. 5-6, pp. 17-26, December 2015, doi: 10.5121/ijccsa.2015.5602.

[4] C. T. S. Xue and F. T. W. Xin, "Benefits and obstacles of cloud computing adoption in business, International Journal of Cloud Computing: Services Architecture, vol. 6, no. 6, pp. 1–15, December 2016, doi: 10.5121/ijccsa.2016.6601.

[5] D. A. Shafiq, N. Jhanjhi, and A. Abdullah, "Proposing a load balancing algorithm for the optimization of cloud computing applications," in Proc. 13th International Conf. Math., Actuarial Sci., Comput. Sci. Statist. (MACS), December 2019, pp. 1-6, doi: 10.1109/MACS48846.2019.9024785.

[6] S. K. Mishra, B. Sahoo, and P. P. Parida, "Load balancing in cloud computing: A broad picture J. King Saud Univ.-Comput. Inf. Sci., vol. 32, no. 2, pp. 149-158, 2020; doi: 10.1016/j.jksuci.2018.01.003.

[7] I. Odun-Ayo, M. Ananya, F. Agono, and R. Goddy-Worlu, "Cloud-based computing architecture: A critical analysis," in Proc. 18th International Conference on Computer Science and Applications (ICCSA), July 2018, pp. 1-7, doi: 10.1109/ICCSA.2018.8439638.

[8] A. Jyoti, M. Shrimali, and R. Mishra, "Cloud computation and load balancing in cloud computing—survey," in Proc. 9th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence), Jan. 2019, pp. 51-55, doi: 10.1109/confluence.2019.8776948.

[9] S. H. H. Madni, M. S. Abd Latiff, M. Abdullahi, S. M. Abdulhamid, and M. J. Usman, "Performance comparison of heuristic methods for work scheduling in IaaS cloud computing environment, PLoS ONE, vol. 12, no. 5, May 2017, Article number. e0176321, doi: 10.1371/journal.pone.0176321.

[10] Adhikari and Amgoth, "Heuristic-based load-balancing method for IaaS cloud," Future Generation Computer Systems, vol. 81, pp. 156-165, April 2018, doi: 10.1016/j.future.2017.10.035.

[11] B. Singh and G. Singh, "A research on virtualization and hypervisor in cloud computing," International Journal of Computer Science and Mobile Applications, vol. 6, no. 1, pp. 17-22, 2018.

[12] M. Kumar, S. C. Sharma, A. Goel, and S. P. Singh, "A comprehensive survey of scheduling techniques in cloud computing." Journal of Network and Computer Applications, vol. 143, pp. 1-33, Oct. 2019, doi: 10.1016/j.jnca.2019.06.006.

[13] F. Zabini, A. Bazzi, B. M. Masini, and R. Verdone, "Optimal performance versus fairness tradeoff for resource allocation in wireless systems," IEEE Trans. Wireless Commun., vol. 16, no. 4, pp. 2587-2600, April 2017, doi: 10.1109/TWC.2017.2667644.