International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

EFFECTIVE LOAD BALANCED OPTIMIZED TASK ALLOCATION FOR MOBILE CLOUD COMPUTING WITH MACHINE LEARNING SCHEME

***Puneet Kaur[1], Sudha Narang[2], Anindita Khade[3], Bollampally Anupama[4], Ketki Kshirsagar[5], Manju Dhillon[6], Nisha Rani. S [7], Muthuchelvi. P[8]**

[1]Associate Professor, Department of Electrical and Electronics Engineering, University Institute of Engineering and Technology (UIET), Panjab University, Chandigarh, India, puneetee@pu.ac.in

[2]Associate Professor, Department of Computer Science and Engineering, Maharaja Agrasen Institute of Technology, Delhi, India, sudhanarang@mait.ac.in

[3]Assistant Professor, Department of Computer Engineering, School of Technology Management and Engineering, SVKM'S NMIMS Deemed to be University, Navi Mumbai, Maharashtra, India, aninditaac1987@gmail.com

[4]Associate Professor, Department of Electronics and Communication Engineering, B.V. Raju Institute of Technology, Narsapur, Telangana, India, anupama.bollampally@bvrit.ac.in

[5]Associate Professor, Department of Electronics and Telecommunication, Vishwakarma Institute of Technology, Pune, Maharashtra, India, ketki.kshirsagar@vit.edu

[6]Assistant Professor, Department of Computer Applications, Maharaja Surajmal Institute, New Delhi, India, manjudalal@msijanakpuri.com

[7]Assistant Professor, Department of Electronics and Communication Engineering, Kamaraj College of Engineering and Technology, Virudhunagar, Tamilnadu, India, nisharani.84@gmail.com

[8]Professor, Department of Artificial Intelligence and Data Science, Jeppiaar Institute of Technology, Sriperumbudur, Chennai, Tamilnadu, India, muthuchelvi69@gmail.com

Abstract

Effective management of workloads remains a major challenge in cloud computing, especially within the IAAS model. Proper distribution of tasks and load compensation are essential to preventing or receiving the resources charged, which can lead to delays and system errors. This study introduces the EDLB algorithm (enhanced dynamic final balance) aimed at optimizing task planning and resource management in Cloud Systems. In contrast to traditional methods that rely on fixed VM selection or lazy cloudlet reallocation, EDLB determines the optimal cloudlet

Received: August 03 ,2025

International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

placement in real time. Assign a VMS proactive cloudlet based on real-time system conditions and SLA schedule (depends on service level) that helps to avoid SLA violations. The results of the CloudSim simulation show that the EDLB algorithm surpasses standard and advanced algorithms, exceeding the overall make-up reduction by 59.46%, average make-up 12.70%, shorter running times, and reduced resource utilization by 3.10%. Additionally, load compensation is improved by 47.48%. These results demonstrate strong potential of the EDLB algorithm dissolving critical load compensation challenges and improving overall performance of cloud computing. This study offers considerable advances by presenting new strategies that significantly improve system efficiency and performance metrics.

Keywords: Cloud computing, Task scheduling, Load balancing, Machine learning, Optimization techniques.

1. Introduction:

CSP play a main role in managing infrastructure by handling backend operations, such as server and data center maintenance. In recent years, infrastructure adoption has increased significantly as a cloud computing service model (IAAS). IAAS provides scalable and flexible virtual computer resources that allow you to create and manage infrastructure without investing in physical hardware. The increasing relevance of this model can be seen in the growing forecasts of the global market. In response to this trend, many service providers have set up large data centers to provide important cloud resources such as servers [2]. Cloud-based applications rely heavily on virtualization, a fundamental aspect of your company [3]. Unmanaged VM migration and resource allocation can strongly impact services and make cloud performance an important challenge [4]. In this environment, users access the service by sending queries processed by the virtual machine (VM) [5]. Cloud Service Providers (CSPs) are responsible for providing services that support business requirements and ensuring a high level of user satisfaction [6]. As a result, the purpose of this study is to introduce load compensation algorithms specifically tailored for the IAAS model and aimed at the following challenges for the core backend of cloud computing: B. Server workload. Figure 1 shows how tasks are distributed through several physical machines to achieve effective load compensation.

A standard cloud environment consists of two main components: front-ends where users interact with over the Internet, and back-ends where the cloud service model is manipulated and served [7]. The backend includes many physical servers housed in a data center. User registration is dynamically planned, and virtualization is used to allocate appropriate resources to customers. Virtualization plays a critical role in compensation for system-wide workloads by promoting efficient planning and resource distribution [8]. Both Cloud Service Providers (CSPs) and users benefit from virtualization and dynamic task planning. Because user inquiries are directed

Received: August 03 ,2025

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

through a cloud broker, task planning is closely linked to workload balancing. This highlights the need for effective algorithms to assign tasks to the appropriate virtual machines (VMs), while also taking into account important factors such as bookings, which are very important for the delivery for increasing quality services. These enquiries travel over the Internet and are stored on a VM. Efficiency of planning strategies plays a critical role in compensation for workloads through VMS and servers. Therefore, implementations of DLB (dynamic load balancer) can provide effective planning and optimal use of resources. In a cloud environment, hypervisors such as host-level Virtual Machine Monitor (VMM) or VMware are responsible for managing several VM on a single physical machine [12].

The virtualization of cloud systems, which is a particularly important role in meeting the corresponding VM or resource [13] [14]. Such inefficiencies can lead to uneven server workloads. Therefore, within cloud computing, there is considerable potential to improve task allocation to resources by improving planning strategies. To maximize resource use without violating SLAs, it is important to consider the most important quality quality (QoS) factors, including deadlines and task priorities [15].

Contribution of this work

The most important contributions of this work include:

- Includes suggestions for the EDLB algorithm (enhanced dynamic final balance) to optimize task planning and task distribution in a cloud environment.
- This study presents an innovative strategy for dynamic load compensation strategies that combine prevention planning with adaptive resource allocation. This is directed according to SLA requirements.
- The effectiveness of the proposed algorithm was evaluated using CloudSim simulations to significantly decreases both overall make-up time and runtime, while ensuring high resource utilization and balanced workloads for various VMs and Cloudlet setups.
- The most important performance indicators are analyzed to measure the efficiency, scalability and load compensation performance of the EDLB algorithm. In other words, the benefits of existing benchmark algorithms have been documented under many workload conditions.

Section III discusses the proposed EDLB algorithm covering system architecture, flow diagrams, pseudocode, implementation approaches, simulation structures, and evaluation metrics. Section IV contains experimental results and discussion, including comparisons with existing methods. Finally, Section V concludes the research with a summary of the most important contributions.

Related Work:

Received: August 03 ,2025

International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

In task planning and load compensation within computer systems have been greatly promoted. This is due to the ongoing development of hardware technology due to the increasing complexity of computational workloads. Effective management of resources and distribution of tasks across various computer components have become essential to improving system performance and maximizing resource utilization. Planning core load compensation goals becomes increasingly important as cloud users increase, as inadequate planning can occur without the right mechanism. Special algorithms are required to address these challenges [13] [17].

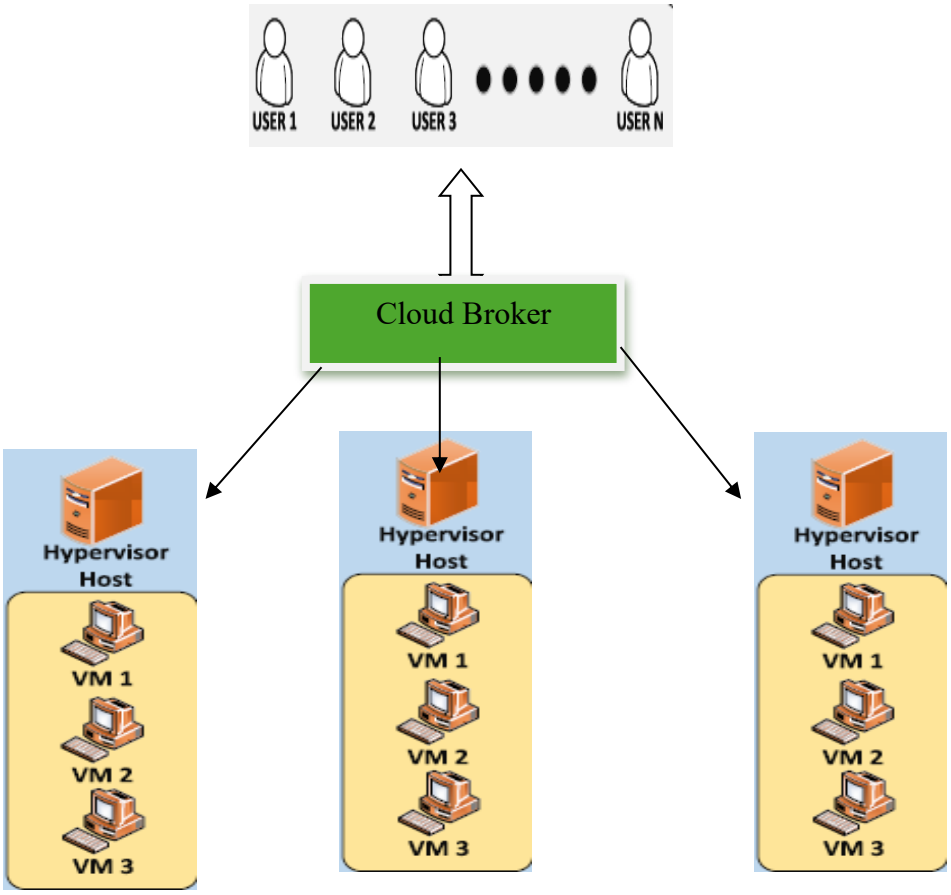


Fig 1: Load balancing model in IaaS

The number of users increases, algorithms supporting these services can face challenges. Load compensation is extremely important in cloud computing to prevent task latency and reduce user response times. Issues such as long hours can affect system performance and risk violations against service level agreements (SLAs). Such violations can cause system overload, which lead to tasks in which incoming inquiries are not handled properly and may be rejected. Several factors contribute to the weight that is loaded in an IAAS-Cloud environment, including

Received: August 03 ,2025

International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

inappropriate tasks, inefficient planning, different task requirements for heterogeneous users, and uneven workload distribution across the VM. This algorithm takes important task requirements such as deadlines and final times, the most important parameters of service quality. By implementing effective planning and preventing VM overload, the algorithm aims to maintain a balanced workload across the cloud system.

Improved in [20] by the ELBMM approach, Min-Min-Min-Algorithm aims to optimize resource allocation by selecting the task with the shortest running time and assigning VMs at the final final time. Load compensation with minimum classification of load compensation (LBMM). This hierarchical network structure improves OLB task planning, but several levels of additional complexity can slow down processing. Another hybrid method, Resource-Based Load Balance Minimin (RBLMM) [21], focuses on minimizing MASKPAN and compensation for workloads via VMS by calculating make-up swit values according to resource allocation. It improves resource optimization, but does not have preventive planning and real-time SLA monitoring. In contrast, the EDLB algorithm dynamically reduces resources based on SLA requirements, prevents SLA violations, and actively transfers tasks between data centers.

This algorithm surpasses the original HTV balancer in resource performance and usage, but some data centers do not have real-time dynamic resource allocation. In contrast to this approach, our approach dynamically adapts to workload and traffic changes, optimizes SLA compliance, and optimizes more efficient management of resources in a distributed cloud environment. For improved quality of service (QOS), the authors of [24] propose a QoS-based algorithm that assigns an extended balancing method to cloudlets.

The GTS algorithm (grouped task planning) in [25] classifies tasks according to QOS parameters to improve the delays of emergency tasks, but is not suitable for tasks that require specific sequences or planning restrictions. [26] introduces an improved version of the Shortest Programming Algorithm (SJF) PEDUL algorithm, and introduces starvation, where high-response VMs are assigned to long tasks. However, this method ignores VM availability and current load. This makes the planning task based solely on priority-based length. Compared to other approaches, such as honeybee and dynamic load compensation, fewer movements improve performance. [28] combines the new hybrid load compensation algorithm Krill -Herds, Whale Optimization and Deep Conviction Neuronal Networks to optimize and surpass existing methods. However, it focuses on hosts for all VMs and ignores cloudlets.

VM execution times to manage dynamic workload compensation in cloud computing. Finally, resource-related resource compensation monkey gas (HO-CB-Larb-SA) proposed in [30] uses hybrids of surgical optimization algorithms (WOA) and lyrebird optimization algorithms (LOA) to optimize resource usage between load distributions and VMs. Correct your workload by

Received: August 03 ,2025

International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

assessing VM processing power and redistributed tasks accordingly. While this method claims good resource management and load compensation, VM processing performance and dependency on loan systems limits adaptability to highly lead to inefficiency in the event of rapid workload changes.

Proposed System:

This section Plan your tasks strategically to reduce overall and average make span, minimize runtime and optimize resource utilization by using all available CPUs in your machine. SLAs are key indicators for cloud service providers (CSPs) and reflect efforts to minimize SLA violations in relation to deadlines. The balanced work division between the VMs of a cloud system. This means that task redistribution occurs in SLA violations. This assignment is dynamically compliant as it ensures that the cloudlet completes on time and meets the SLA conditions. Figure 2 shows a flow diagram of the EDLB algorithm, explaining the process of assigning Cloud to a VM in a way that optimizes processing efficiency and prevents over-control of a single VM. Factors such as the last time of the VM, available processors, and how long the cloudlet runs are all taken into account. This measures the usable processing power in MIP. The important symbols used in the related equations are shown in Table 2.

$$CTVM_i = \in CTCij \quad (1)$$

$$CTCj = L_j / MIPS_i \quad (2)$$

This will check if the completion time of the Cloudlet (CTC) sticks to the deadline. If the appropriate VM is not determined, the algorithm begins transferring the cloudlet to the secondary data center where processes will halt within a single data center and overall efficiency is improved. This includes available MIPS (AM) representing excess MIP through all VMs above the total -NM -NM -MIPS. Furthermore, CG indicates the number of MIPs that a VMS can release without violating the limits, while N measures the measured MIP to complete the cloud within the period, as shown in the equation.

$$CG_k = MIPS_k - NM_k \quad (3)$$

$$N_j = L_j / D \quad (4)$$

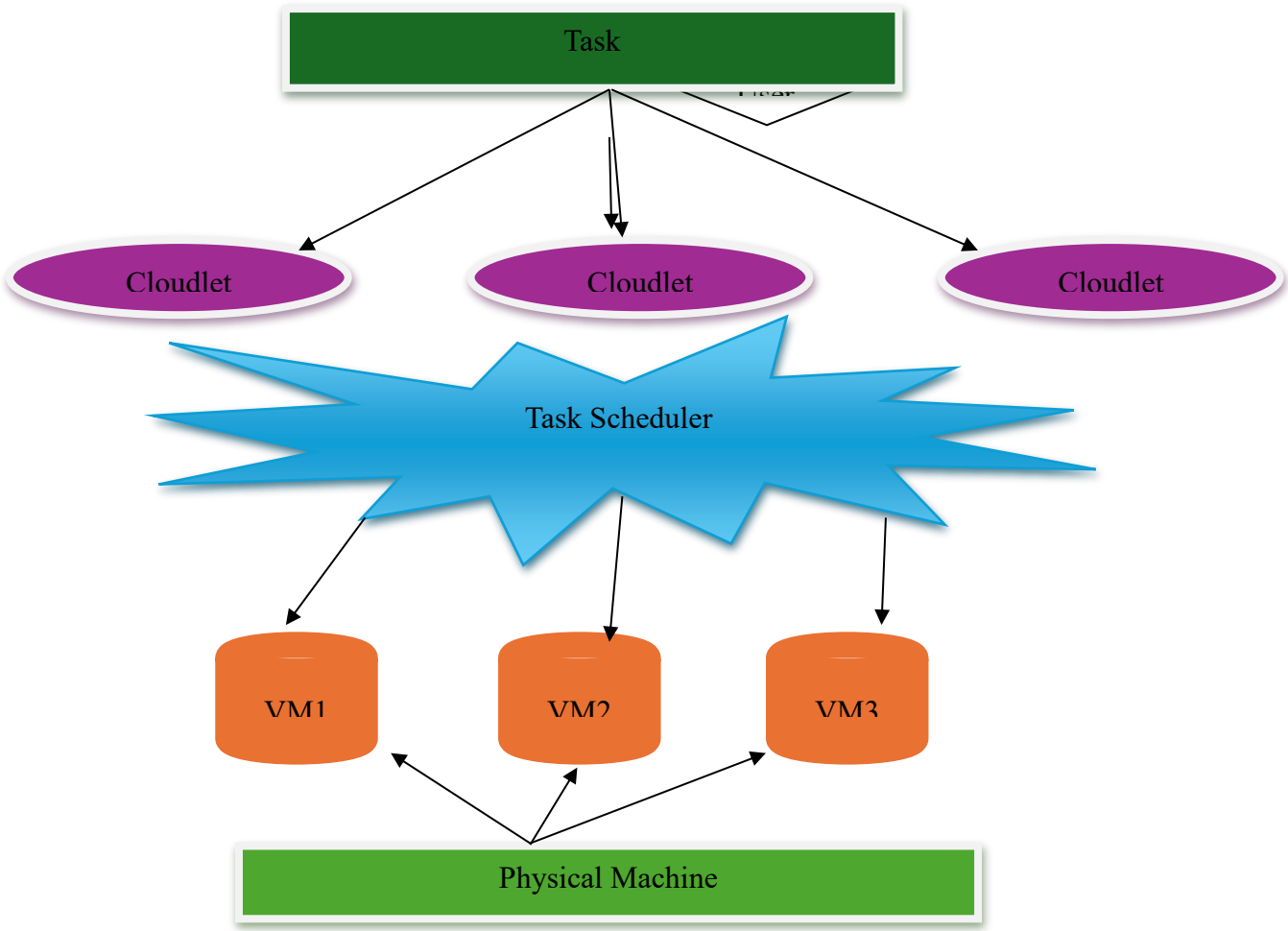
$$TG = N - MIPS_{min} \quad (5)$$

Received: August 03 ,2025

International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)



If the VM cannot fully pass the assigned parts, the S value will be adjusted accordingly. During the reconfiguration phase, a VMS that cannot meet the allocated MIP contributes all the resources possible. The remaining surplus is rather distributed based on the values of other VMs, as explained in Equation (6).

$$P = S / \alpha - 1 - \beta \qquad (6)$$

The variable β refers to the number of virtual machines that cannot contribute to the assigned part. As soon as the reconfiguration phase is completed, the CTVM will be changed and the updated MIPS distribution will be reflected in all VMs. This process ends by recalculating the CTVM to maintain newly planned cloudlet execution times and NM updates to maintain efficient resource allocation while meeting SLA requirements.

Received: August 03 ,2025

International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

This illustrates the algorithm 1, related expressions and line mechanisms. The algorithm is a dynamic and adaptive approach to resource distribution. In contrast to existing benchmark algorithms, we assign cloudlets according to random principles and assign only SLA violations according to the assignment, but EDLB makes real-time decisions. There is a current final period for both VMS (CTVM) and CloudLets (CTC), and Task deadlines are observed to ensure that system resources are used optimally from the start.

Algorithm 1: Improved Dynamic Load Compensation Algorithm

Input: A List of clouds with random length and deadlines.

Output: Cloudlet assignment to the corresponding VM

Adjust NM_i to all VMs with value 1000

for j in J **do**

find VM with least CTVM (VM_{min})

calculate N with Equation (5)

if VM_{min} has enough MIPS to execute cloudlet before deadline

($MIPS_{min} >= N$)

then

allocate cloudlet to VM_{min}

end

else

set min to α

if VM_{min} has enough MIPS to execute cloudlet before

deadline ($MIPS_{min} >= N$)

then

Allocate cloudlet to VM_{min}

break

end

Calculate TG using equations (3), (6), and CG for each VM in equations (3), (6), and (4)

Received: August 03 ,2025

International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Determine *S*

else if

AM is enough to execute cloudlet before deadline (AM >= N)

then

Adjust N MIPS to VM_{min}

Initialize β **for** k in K **do**

if VM_k do not have its portion to give ($CG_k < TG$)

then

Set HE_k to False Update S with Equation (7)

Update β to +1

end

else

Set HE_k to True

end

end

Calculate P with Equation (8)

for k in K **do**

if VM_k have enough MIPS to give (HE_k is True)

then

Give their portion to VM_{min}

end

else

Give CG_k to VM_{min}

end

end

Received: August 03 ,2025

International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

```
            for k in K do
              while u <= j do
                 if cloudlet allocated to VM k
                  then
                     Calculate CTV Mk by using Equation (1)
                 end
              end
            end
            Allocate cloudlet to VM min
        end
        else
           Allocate cloudlet to VM min
        end
     end
    Update CTVM min and NM min
end
```

The algorithm has two important progress:

1. In contrast to traditional benchmark methods that randomly assign tasks, EDLB selects a virtual machine based on the shortest estimated final time. This targeted approach allows cloudlets to be assigned to the most efficient VMs, reducing the overall make-span and preventing uneven workload distribution.
2. Real-time tuning: In contrast to benchmark algorithms that add MIPS to low-performance VMs after detection of SLA violations, our method is dynamically distributed from VMs with excessive capacity to more people. This new allocation in real time increases resource efficiency and ensures that the VM does not remain idle .
3. CTC calculates final time for each cloud and the processing service (MIP) explained in equation (2).
4. Available MIPS (AM) refers to the excess processing functionality available to all virtual machines as soon as the minimum MIPS (NM) required to run each VM is met.

Received: August 03 ,2025

International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

5. As soon as it is confirmed that the VM is sufficient to edit assigned tasks, all additional MIPs can be made available for redistribution to other VM.

3.1 Evaluation Performance

Performance Metrics

The algorithms, including overall mugserpan, average make span, runtime, resource utilization, and load compensation share. These five indicators were specifically chosen to allow for a rounded assessment of the total performance of the algorithm. Overall and average MASMSPAN reflects how efficiently you complete all your cloudlets on average and on average. Runtime provides a detailed overview of how quickly an individual task is processed. This indicates the responsiveness of the algorithm. Resource usage measures how effectively the available resources are allocated and used to ensure minimal waste. Finally, the load compensation ratio percentage contributes to the fact that even the workload is distributed across all VMs, which is why a single VM is overloaded. Overall, algorithms to optimize task planning, resource usage, and workload management.

1) Total Make span (TMT): This refers to the final final time recorded under all virtual machines (VMs). This serves as an indicator of how effectively the algorithm manages system-wide tasks. The TMT is determined by identifying the maximum cloud end time of the VM.

$$TMT=\text{Max}(MT) \tag{7}$$

2) Make span (MT): To plan single cloud and plays an important role in assessing the time efficiency of algorithm planning [35]. Reducing MakePan is advantageous as tasks can accelerate and resources may become available for subsequent tasks. MASMEPAN is calculated using the cloudlet completion time (CT) and the total number of VMS (M) as defined in Equations (8) and (9) [36].

$$MT=\text{Max}(CT) \tag{8}$$

$$MTavg=\frac{\in Max(CT)}{m} \tag{9}$$

3) Runtime (ext): This parameter evaluates the duration required for a virtual machine (VM) to perform a particular task [26]. Shorter execution times improve the efficiency of the algorithm. The calculation of this metric includes the actual CPU time (ACT) of the cloudlet.

$$ExT = AcT \tag{10}$$

$$ExT_{avg} = \in AcT / n \tag{11}$$

Received: August 03 ,2025

International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4) Resource Utilization (RU): This metric statement quantitatively evaluates how resources are used in a cloud environment and closely links to other important performance metrics [33]. Calculated using total execution time (ext) and total make-up man (MT). The average RU value effectively reflects the use of CPU resources of the EDLB algorithm. Metrics range from 0 to 1, where 1 means the use of optimal (100%), and 0 does not refer to occupancy, as defined in equations (12) and (13) [37].

$$RU = ExT / MT \quad (12)$$

$$RU_{avg} = ExT_{avg} / MT_{avg} \times 100 \quad (13)$$

5) Balancing Percentage (BP): This metric evaluates how even workloads are distributed across virtual machines by calculating the proportion of recipients of the average make span. It serves as an indicator of the effectiveness of the algorithm in achieving balanced use of resources, as defined in Equation (14).

$$BP = MT_{avg} / TMT \times 100 \quad (14)$$

The five selected performance indicators of total mixer, average make-up, execution time, resource utilization, and compensation share provide a thorough assessment of the proposed algorithm efficiency. Evaluate how effective tasks are completed, resources are managed, and workloads are distributed to virtual machines. This gives you a comprehensive understanding of the algorithmic ability to improve performance and optimize resource allocation.

Result

The make-span offers a time-efficient perspective for all tasks. This is very important in environments with tasks of different complexity. This metric helps assess the capabilities of the algorithm, manage individual tasks effectively, and ensure strong performance for various workloads. An average make span at 1200 cloudlets with 1200 cloudlets compared with 90.32 seconds for HO-CB-Ralb-SA. This demonstrates the ability of the EDLB algorithm to optimize task execution time with a total improvement of 10% on average make-up compared to HO-CB-Ralb-SA.

Received: August 03 ,2025

International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

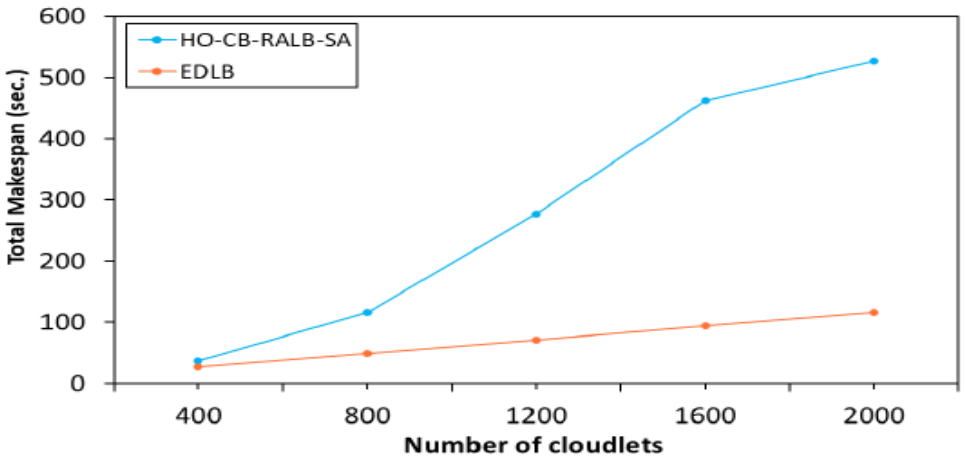


Fig 3: Total make span

Runtime is an important metric for evaluating how efficiently an algorithm uses processing resources. Shorter execution times reduce operational costs and increase throughput. This is especially important. For example, EDLB reaches a runtime of 81.29 seconds on 2,000 cloudlets, while HO-CB-Larb-SA lasts 172.32 seconds, highlighting the benefits of EDLB when minimizing runtime. On average, EDLB achieved a 15.6% improvement in runtime compared to HO-CB-Ralb-SA.

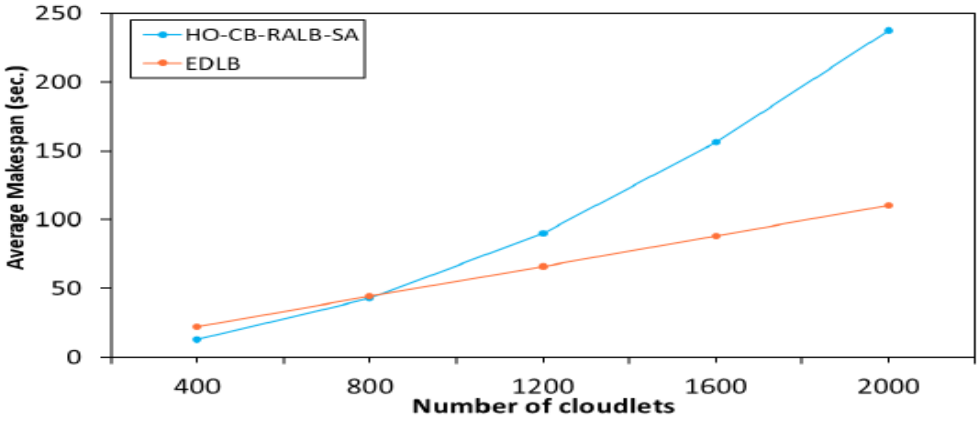


Fig 4: Average make span

Resource usage indicates how efficiently the algorithm uses available resources. This makes higher values more efficient. For example, EDLB reaches 75.44% utilization on 800 cloudlets, while HO-CB-Larb-SA reaches 80.61%. This consistent performance across different cloud sizes highlights the reliability of EDLB in effective management and use of system resources.

Received: August 03 ,2025

International Journal of Applied Mathematics

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

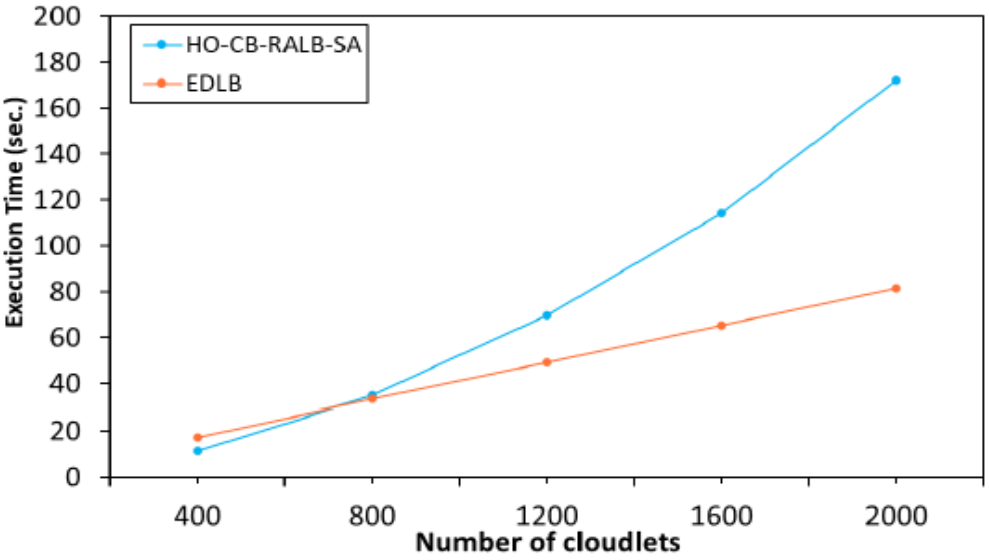


Fig 5: Execution time

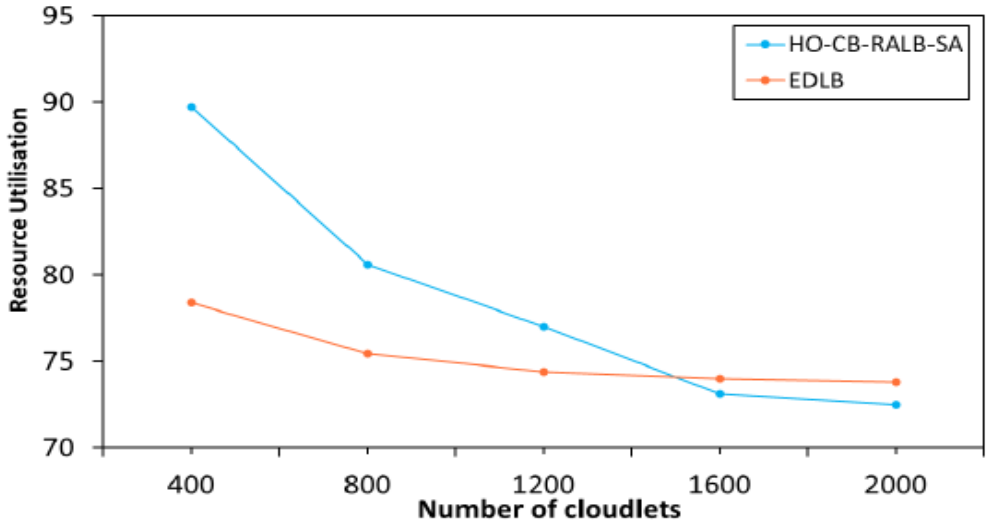


Fig 6: Resource utilization

Despite the growing number of cloudlets, EDLB consistently receives stable resource usage, while HO-CB-Ralb-SA-Algorithm is experiencing a decline. With EDLB achieved a total improvement of 3.8%.

The reward process assesses the extent to which effective workloads are distributed across available resources. The higher the value, the more uniform the distribution. As shown in Figure

Received: August 03 ,2025

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

7, the EDLB algorithm significantly improves load compensation performance, especially for many cloudlets. For example, EDLB achieves 95% compensation efficiency with 2,000 cloudlets, compared to only 45.2% of the HO-CB-Ralb-SA algorithm. This improved compensation capability is consistent across all cloud sizes tested. Overall, EDLB shows an improvement of 59.2% in compensation efficiency for cloud counts between 400 and 2,000.

The EDLB algorithm uses an advanced task allocation approach that takes into account the final time of individual cloudlets before allocating virtual machines. This strategy ensures optimal distribution of workloads and prevents imbalances that can affect system performance. If an SLA violation occurs, use EDLB Proaktiv Cloudlets to adapt your CPU configuration to improve resource shortages. Through intelligent management of tasks based on the specific characteristics of each cloudlet, EDLB achieves consistently effective load compensation, minimizes make-up, and is subject to high resource usage.

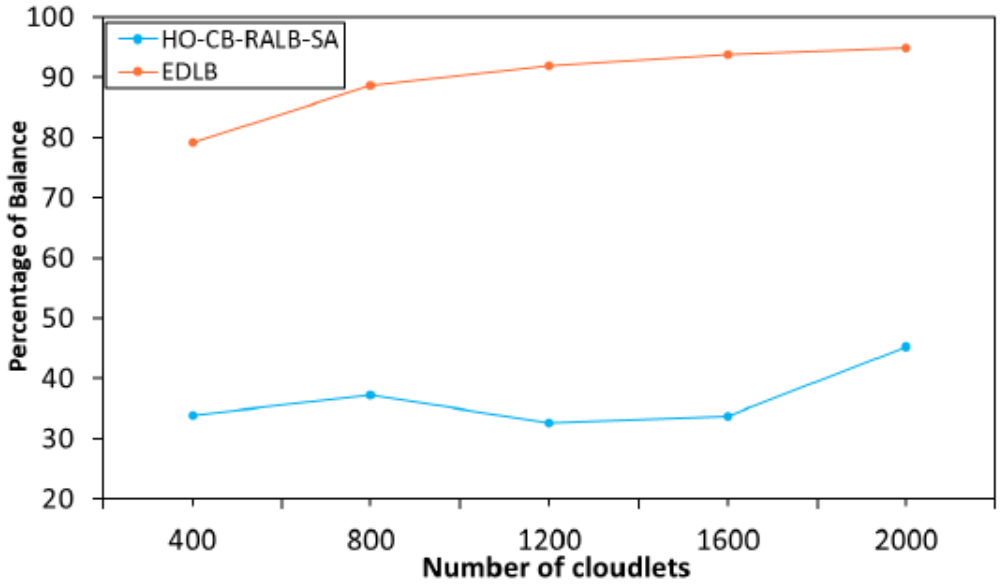


Fig 7: Balancing percentage

Conclusion

This study highlights the key role of cloud computing particularly as a service model and highlights the need for sophisticated load compensation sales to improve resource allocation. Highlights the planning of preventive tasks and takes into account Cloudlet attributes such as arrival times, task length, and deadlines. This approach consistently improves the most important performance metrics and effectively deals with SLA conformance. Benchmarking methods often lead to workload inspiration and inefficiency, and usually have long running times, HO-CB-

Received: August 03 ,2025

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

RALB-SA, but EDLB is characterized by evenly distributing workloads, reducing make-up and execution times, and ultimately improving resource usage.

Furthermore, evaluations to a wider set of factors, including response times, throughput, & fault tolerance, provide a more comprehensive understanding of performance. It is also important to include additional resources such as RAM in your allocation strategy. This is because it can have a significant impact on the total efficiency and effectiveness of the cloud.

References

[1] Statista. (Jul. 2024). Infrastructure as a Service: Market Data & Analysis. Accessed: Sep. 8,2024.[Online].Available:https://www.statista.com/study/84972/infrastructure-as-a-service report/

[2] F. Xia, Y. Chen, and J. Huang, ‘‘Privacy-preserving task offloading in mobile edge computing: A deep reinforcement learning approach,’’ Softw., Pract. Exper., vol. 54, no. 9, pp. 1774–1792, Sep. 2024.

[3] T. Kumar, P. Sharma, J. Tanwar, H. Alsghier, S. Bhushan, H. Alhumyani, V. Sharma, and A. I. Alutaibi, ‘‘Cloud-based video streaming services: Trends, challenges, and opportunities,’’ CAAI Trans. Intell. Technol., vol. 9, no. 2, pp. 265–285, Apr. 2024.

[4] S. M. Rozehkhani, F. Mahan, andW. Pedrycz, ‘‘Efficient cloud data center: An adaptive framework for dynamic virtual machine consolidation,’’ J. Netw. Comput. Appl., vol. 226, Jun. 2024, Art. no. 103885.

[5] J. Sung, S.-J. Han, and J.-W. Kim, ‘‘Cloning-based virtual machine pre-provisioning for resource-constrained edge cloud server,’’ Cluster Comput., vol. 27, no. 2, pp. 1831–1847.

[6] S. K. Mishra, B. Sahoo, and P. P. Parida, ‘‘Load balancing in cloud computing: A big picture,’’ J. King Saud Univ.-Comput. Inf. Sci., vol. 32, no. 2, pp. 149–158, Feb. 2020.

[7] A. Rajagopalan, D. Swaminathan, M. Bajaj, I. Damaj, R. S. Rathore, A. R. Singh, V. Blazek, and L. Prokop, ‘‘Empowering power distribution: Unleashing the synergy of IoT and cloud computing for sustainable and efficient energy systems,’’ Results Eng., vol. 21, Mar. 2024.

[8] M. Zakarya, A. A. Khan, M. R. C. Qazani, H. Ali, M. Al-Bahri, A. U. R. Khan, A. Ali, and R. Khan, ‘‘Sustainable computing across datacenters: A review of enabling models and techniques,’’ Comput. Sci. Rev., vol. 52, May 2024, Art. no. 100620.

[9] F. Qazi, D. Kwak, F. G. Khan, F. Ali, and S. U. Khan, ‘‘Service level agreement in cloud computing: Taxonomy, prospects, and challenges,’’ Internet Things, vol. 25, Apr. 2024.

Received: August 03 ,2025

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[10] M. A. Ala'anzy and M. Othman, ''Mapping and consolidation of VMs using locust-inspired algorithms for green cloud computing,'' Neural Process. Lett., vol. 54, no. 1, pp. 405–421, Feb. 2022.

[11] M. A. Ala'anzy, M. Othman, S. Hasan, S. M. Ghaleb, and R. Latip,''Optimising cloud servers utilisation based on locust-inspired algorithm,''in Proc. 7th Int. Conf. Soft Comput. Mach. Intell. (ISCMI), Nov. 2020, pp. 23–27.

[12] S. Sha, C. Li, X. Wang, Z. Wang, and Y. Luo, ''Hardware-software collaborative tiered-memory management framework for virtualization,'' ACM Trans. Comput. Syst., vol. 42, nos. 1–2, pp. 1–32, May 2024.

[13] M. Ala'anzy and M. Othman, ''Load balancing and server consolidation in cloud computing environments: A meta-study,'' IEEE Access, vol. 7, pp. 141868–141887, 2019.

[14] Z. Ahmad, A. I. Jehangiri, M. A. Ala'anzy, M. Othman, and A. I. Umar, ''Fault-tolerant and data-intensive resource scheduling and management for scientific applications in cloud computing,'' Sensors, vol. 21, no. 21, p. 7238, Oct. 2021.

[15] M. Kumar, S. C. Sharma, A. Goel, and S. P. Singh, ''A comprehensive survey for scheduling techniques in cloud computing,'' J. Netw. Comput. Appl., vol. 143, pp. 1–33, Oct. 2019.

[16] J. Zhou, U. K. Lilhore, P. M. T. Hai, S. Simaiya, D. N. A. Jawawi, D. Alsekait, S. Ahuja, C. Biamba, and M. Hamdi, ''Comparative analysis of metaheuristic load balancing algorithms for efficient load balancing in cloud computing,'' J. Cloud Comput., vol. 12, no. 1, p. 85, Jun. 2023.

[17] A. Arunarani, D. Manjula, andV. Sugumaran, ''Task scheduling techniques in cloud computing: A literature survey,'' Future Gener. Comput. Syst., vol. 91, pp. 407–415, Feb. 2019.

[18] T. Bu, Z. Huang, K. Zhang, Y.Wang, H. Song, J. Zhou, Z. Ren, and S. Liu, ''Task scheduling in the Internet of Things: Challenges, solutions, and future trends,'' Cluster Comput., vol. 27, no. 1, pp. 1017–1046, Feb. 2024.

[19] M. Kumar and S. C. Sharma, ''Dynamic load balancing algorithm to minimize the makespan time and utilize the resources effectively in cloud environment,'' Int. J. Comput. Appl., vol. 42, no. 1, pp. 108–117, Jan. 2020.

[20] G. Patel, R. Mehta, and U. Bhoi, ''Enhanced load balanced min-min algorithm for static meta task scheduling in cloud computing,'' Procedia Comput. Sci., vol. 57, pp. 545–553, Jan.

Received: August 03 ,2025

Volume 38 No. 5 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

2015,doi:10.1016/j.procs.2015.07.385.[Online].Available:https://www.sciencedirect.com/science/article/pii/S1877050915019146

[21] B. H. Shanthan and L. Arockiam, ‘‘Resource based load balanced min min algorithm (RBLMM) for static meta task scheduling in cloud,’’ in Proc. Int. Conf. Adv. Comput. Sci. Technol., 2018, pp. 1–8.

[22] M. Adhikari and T. Amgoth, ‘‘An enhanced dynamic load balancing mechanism for task deployment in IaaS cloud,’’ in Proc. Int. Conf. Comput., Power Commun. Technol. (GUCON), Sep. 2018, pp. 451–456.

[23] S. Acharya and D. A. D’Mello, ‘‘Enhanced dynamic load balancing algorithm for resource provisioning in cloud,’’ in Proc. Int. Conf. Inventive Comput. Technol. (ICICT), vol. 2, Aug. 2016, pp. 1–5.

[24] S. Banerjee, M. Adhikari, S. Kar, and U. Biswas, ‘‘Development and analysis of a new cloudlet allocation strategy for QoS improvement in cloud,’’ Arabian J. Sci. Eng., vol. 40, no. 5, pp. 1409–1425, May 2015.

[25] H. Gamal El Din Hassan Ali, I. A. Saroit, and A. M. Kotb, ‘‘Grouped tasks scheduling algorithm based on QoS in cloud computing network,’’ Egyptian Informat. J., vol. 18, no. 1, pp. 11–19, Mar. 2017.

Received: August 03 ,2025