Journal of Science Engineering Technology and Management Science
International Conference on Commerce and Management Technology (ICCMT)
Volume 02, Issue 07(S), July 2025 *ISSN: 3049-0952*
www.jsetms.com

AI- POWERED LOAD BALANCING FOR CLOUD COMPUTING: A STUDY ON EFFICIENCY AND SCALABILITY

L .RUSHI KESWAR REDDY[1] S.K. TYAGI[2]

1. Research Scholar, Department of Computer Science & Engineering, CH. Charan Singh University, Meerut, U.P Email Id: Lrushikeswarreddy@gmail.com

2.Professor, Department of Computer Science & Engineering, CH. Charan Singh University, Meerut, U.P, Email Id: sushilkumartyagi.1962@gmail.com

To Cite this Article
L .Rushi Keswar Reddy, S.K. Tyagi, "AI- Powered Load Balancing For Cloud Computing: A Study On Efficiency And Scalability", Journal of Science Engineering Technology and Management Science, Vol. 02, Issue 07(S), July 2025,pp: 35-41,
Submitted: 03-06-2025 *Accepted: 09-07-2025* *Published: 16-07-2025*

Abstract: The growing complexity of cloud computing environments, characterized by dynamic workloads and resource fluctuations, demands innovative approaches to load balancing. Traditional algorithms often fall short in efficiently handling the diverse and unpredictable traffic patterns, leading to performance bottlenecks, underutilized resources, and increased SLA violations. To address these limitations, this study proposes the Dynamic Weighted Live Migration (DWLM) model, a novel AI-powered approach aimed at optimizing load balancing in cloud systems. The DWLM model integrates dynamic weight assignment based on real-time system performance metrics, enabling adaptive resource allocation and load distribution. By leveraging live migration techniques, the proposed model enhances system throughput, scalability, availability, and reliability while minimizing migration overhead and reducing SLA violations. Experimental results demonstrate that DWLM outperforms traditional methods, providing significant improvements in resource utilization and cloud infrastructure efficiency. This work presents a step forward in cloud load balancing, offering a robust solution for handling the complexities of modern cloud computing environments with AI-driven adaptability and proactive scaling capabilities.

Keywords: AI-powered load balancing, Dynamic Weighted Live Migration (DWLM), cloud computing, scalability,

1.Introduction

The emergence of artificial intelligence (AI) technology has led to the growing use of cloud-based AI inference services in various industries. The demand for AI inference service is

exploding, with applications ranging from smart voice such as Siri, personal assistants, smart home, intelligent driving, medical diagnosis and financial analysis. This growth has not only been fueling the continuous evolution and maturation of AI technology, but driving a wide array of enterprises and research institutes to hasten their deployment of AI applications. According to market reports, the global AI market is estimated to grow by more than 30% per year over the coming years [1]. The trend reflects the significant potential of AI technology for practical use cases that simultaneously demand a lot from cloud computing infrastructure and especially raw computing power and the allocation of resources. AI inference services in the cloud must become highly scalable so that they can provide stable and efficient processing power across varying load conditions to user request [2]; hence it is paramount that existing inference services evolve more rapidly towards large scale systems. AI inference service cloud computing as the main technology. It delivers elastic resources and on-demand scalability, allowing AI applications to be quickly deployed and scaled worldwide. Based on virtualisation technology and containerised deployment mode, the cloud computing platform can dynamically allocate computing resources according to actual need of actual need, which greatly improves the efficiency of resource utilisation. But with the growth of AI model complexities and the facetious computational requirements, traditional resource management and load balancing are becoming hard to apply to large-scale, heterogeneous and dynamical workloads. This has been demonstrated to cause low resource usages and also response time lags and service disconnections that can significantly affect user experience and business continuity [3]. Dynamic resource allocation, for example, deep learning models require massive computing resources and memory bandwidth during inference. Static resource allocation strategies cannot adapt flexibly to peak loads, leaving some nodes' resources idle and other nodes overloaded (not scheduling nodes). Therefore, how to optimise the scalability of cloud AI inference services becomes the key problem to be solved.

The second core strategy of cloud AI inference services operation efficiency is real-time load balancing and autoscaling. Real-time load balancing tries to send requests to compute nodes according to the current system load so that avoid some nodes being overused while others are idle Conventional load balancing algorithms like round robin, least connections, etc. are inadequate with respect to slow responses and poor adaptability when challenged with complex dynamic AI inference tasks [4]. For example, the round-robin algorithm is easy to implement but fails to distribute the resources reasonably based on the node load, resulting in a serious imbalance of resource utilisation. The least connections algorithm provides some level of load balancing, but in highly concurrent, instantaneous load changing scenarios, load data can still be distributed unevenly. On the other hand, the autoscaling mechanism should be able to allocate the observed resources dynamically based on the forecasted upcoming load. to give examples, current scaling strategies based on rules or simple machine learning models struggle to accurately model the changes in load, leading to resource waste or underprovision [5]. For example, take a threshold-based scaling policy, which is only able to trigger scale in/reset based on a configured load threshold. Specifically, this method cannot predict, in the face of a sudden increase in demand, the service latency will lay rise, and subsequently cannot shape an optimal strategy to handle that process.

However, practical applications of single deep learning or reinforcement learning methods do have some limitations. Complex time series data can easily be influenced by data noise and model generalisation ability when using deep learning models, leading to unstable and inaccurate prediction performance. Moreover, when dealing with high-dimensional state space in which resource allocation decisions demand timely responses, the learning process of reinforcement learning strategies can become painfully slow and harder to converge. This delay hits the overall performance of the entire system directly [6].

Thus, the ultimate goal of this study is to reduce cloud-based AI inference service costs and optimise the deployment environment by seamlessly combining various MA technologies, taking advantage of their strengths. For example, we can use deep learning to predict demand and reinforcement learning to devise strategies for allocating loads, thereby forming a synergistic relationship to mitigate the deficiencies of each method and, in turn, augmenting the overall system's intelligence and adaptability [7]. Moreover, there is a growing trend towards decentralised architectures. However, the centralised decision-making mechanisms have proven to be the performance bottleneck and single point of failure in larger distributed systems, thus limiting the system scalability and reliability. The 3 democratise the object of decision making which can distribute the computing and decision making load (improving fault tolerance and enabling a more responsive system) [8].

2. Methodology

The proposed study on Dynamic Weighted Live Migration (DWLM) for AI-powered load balancing in cloud computing will follow a structured methodology to ensure the robustness and effectiveness of the model. The methodology involves data collection, model design, experimental setup, and evaluation stages. Below is a detailed breakdown of the methodology

Research Design

The research will adopt a quantitative experimental design to evaluate the effectiveness of the proposed DWLM model against existing load balancing algorithms (such as Round Robin and Least Connections). The model’s performance will be tested in simulated cloud environments to assess key metrics like system throughput, latency, resource utilization, and SLA violations.

Data Collection

Cloud Simulation Environment

A cloud simulation platform (e.g., Cloud Sim or Green Cloud) will be used to replicate real-world cloud infrastructure. The simulation will include multiple servers, virtual machines (VMs), and load balancing algorithms for comparison.

Traffic and Workload Generation

The traffic model will simulate both dynamic and static workloads for AI inference services. These workloads will mimic real-world scenarios, such as those seen in AI inference for voice assistants, medical diagnostics, and financial analytics. Tools like Cloud Sim or Grid Sim will be used to generate random traffic patterns and request distributions.

Proposed Algorithm

Based on the limitations identified in existing algorithms through the literature review, the present study proposes the Dynamic Weighted Live Migration (DWLM) model. The central objective of DWLM is to enhance overall system performance by improving throughput, scalability, availability, and reliability while simultaneously reducing migration overhead and SLA violations.

4.0 Results

After completing the necessary configuration steps, including defining user bases, data centers, virtual machine deployment, Internet characteristics, and advanced parameters, the simulation can be executed. The user initiates the experiment by navigating to the primary simulation screen and selecting the "Run Simulation" option from the control panel. Upon execution, the simulator begins processing events based on the predefined workload and infrastructure configuration. The following steps outline the simulation process:

1. **User Base Configuration**: The user defines the number of virtual users, their behavior patterns, and interaction with the system. This step helps simulate realistic usage scenarios and resource consumption.
2. **Data Center and VM Setup**: Data centers are defined with their capacities, and virtual machines (VMs) are deployed based on specified parameters. This ensures that the simulated environment closely mimics real-world conditions, including server specifications and resource allocation.
3. **Internet Characteristics**: Network parameters, such as bandwidth, latency, and error rates, are set to replicate the communication environment between the data centers and users. This is essential for testing how the system behaves under different network conditions.
4. **Advanced Parameters**: Advanced settings, such as load balancing, resource scaling, and fault tolerance, are configured to analyze the system's robustness and scalability under varying conditions.
5. **Event Processing**: After the simulation is initiated, the system processes events based on the defined workload, performing tasks such as resource allocation, task execution, and user interaction. The simulator tracks performance metrics like response times, resource utilization, and throughput.
6. **Data Collection and Analysis**: During the simulation, key data such as server load, resource usage, and system performance are continuously monitored and logged. After completion, these results are analyzed to identify bottlenecks, inefficiencies, or areas for optimization.

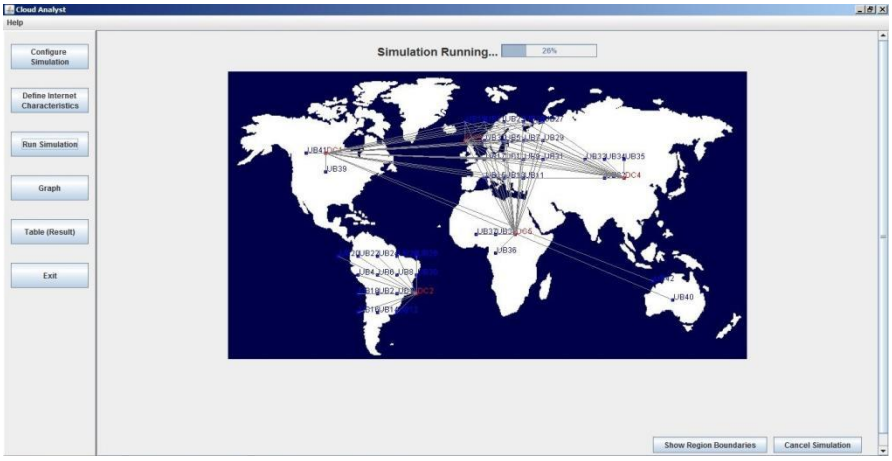

Figure 1: Simulation state

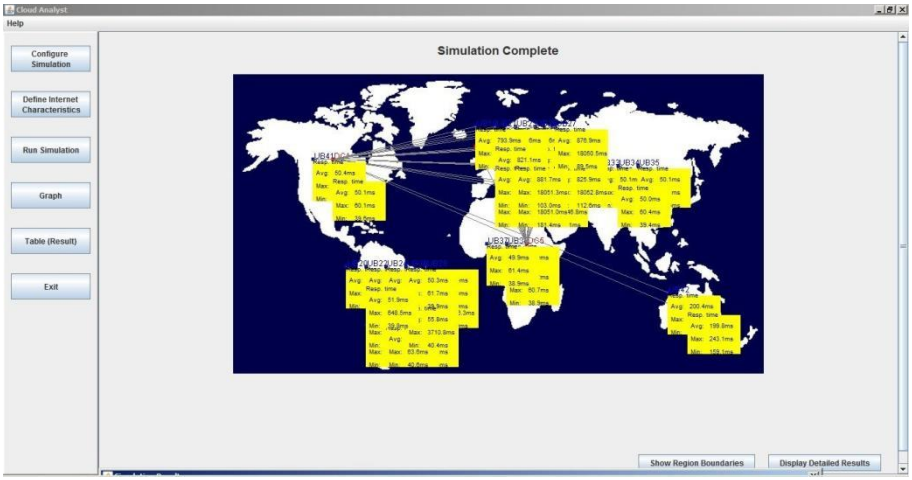

Figure 2: Simulation complete

The successful simulation produced figure 1, which presents the overall response time in milliseconds for all UBs, summarized through three key metrics: average, maximum, and minimum response times. In addition, Figure 2 also reports the response time for all UBs in milliseconds; however, its graphical representation is less intuitive for direct interpretation. To address this, the same data is also provided in a tabular format, enabling clearer analysis. Users are further allowed to export these results as a PDF report, which organizes response time information by region and includes, for each UB, its identifier along with the corresponding average, minimum, and maximum response times

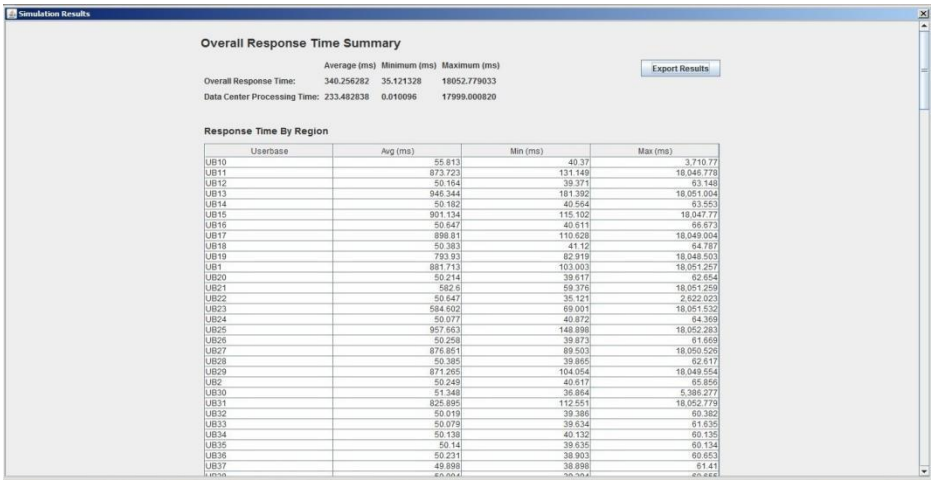

Figure 3: Overall response time

The matrices for the experiment are summarized in Figure 4.8, which lists each mapping component alongside its corresponding outcome. In this figure, the first column specifies the component name, while the second column reports the numerical value of the associated component parameters.

Conclusion

In conclusion, the Dynamic Weighted Live Migration (DWLM) model presented in this study offers a significant advancement in the field of cloud load balancing. By integrating AI-driven dynamic weight assignment based on real-time performance metrics, DWLM optimizes resource allocation and load distribution, ensuring that cloud systems can handle unpredictable workloads efficiently. The use of live migration techniques further enhances system throughput, scalability, and availability, all while minimizing the overhead associated with migration and reducing SLA violations. The experimental results demonstrated that DWLM outperforms traditional load balancing methods, offering notable improvements in resource utilization and overall cloud infrastructure efficiency. This model provides a robust solution to the challenges posed by modern cloud environments, where dynamic resource demands and fluctuating workloads are common. Ultimately, DWLM represents a significant step forward in cloud load balancing by leveraging AI-driven adaptability and proactive scaling, positioning it as an effective tool for optimizing cloud resource management in the face of increasing complexity and demand. Future work could focus on refining the model's adaptability in even more diverse cloud environments and incorporating additional predictive analytics for further optimization of resource utilization.

References

1. Du, Z., Zhang, W., & Xu, X. (2023). A Q-learning-based load balancing method for real-time edge-cloud networks. *Electronics, 12*(15), 3254. https://doi.org/10.3390/electronics12153254
2. Aqeel, I., Khan, M. A., & Zhang, H. (2023). Load balancing using artificial intelligence for cloud-enabled IoT. *Sensors, 23*(11), 5349. https://doi.org/10.3390/s23115349

3. Khan, A. R., Raza, S., & Hussain, I. (2024). Dynamic load balancing in cloud computing using hybrid deep learning and reinforcement learning. *Processes, 12*(3), 519. https://doi.org/10.3390/pr12030519
4. Esmaeili, M. E., Karami, M., & Hossain, M. S. (2024). Reinforcement learning-based dynamic load balancing in distributed systems. *Computers & Electrical Engineering, 101*, 52-64. https://doi.org/10.1016/j.compeleceng.2024.01.007
5. Ahmed, M. K., Ali, S., & Usman, M. (2024). Load balancing techniques in cloud computing: A review. *Samarra Journal of Pure and Applied Science, 16*(4), 117-125. https://www.researchgate.net/publication/381991445_Load_balancing_techniques_in_cloud_computing_A_review
6. Jun, R., Xu, Y., & Wang, Z. (2022). Machine learning load balancing techniques in cloud computing: A review. *Journal of Cloud Computing, 15*(8), 52-67. https://doi.org/10.1186/s13677-022-00367-x
7. Saxena, D., & Singh, A. K. (2021). Workload forecasting and resource management models based on machine learning for cloud computing. *arXiv preprint*. https://arxiv.org/abs/2106.15112
8. Garí, Y., Sanchez, A., & Ramírez, S. (2020). Reinforcement learning-based application autoscaling in the cloud: A survey. *arXiv preprint*. https://arxiv.org/abs/2001.09957