International Journal of Engineering Applied Sciences and Technology, 2025
Vol. 10, Issue 06, ISSN No. 2455-2143, Pages 31-33
Published Online October 2025 in IJEAST (http://www.ijeast.com)

A COST-EFFICIENT LOAD BALANCING ALGORITHM FOR MULTI-CLOUD DEPLOYMENTS

E. Deepika
PG Student
Department of Computer Applications
Jaya College of Arts and Science, Chennai

S. Amsa
Assistant Professor
Department of Computer Applications
Jaya College of Arts and Science, Chennai

Abstract— **The growing importance of cost-aware resource management in multi-cloud computing environments. As cloud technologies rapidly expand, organizations are increasingly adopting multi-cloud strategies to achieve greater flexibility, fault tolerance, and performance optimization by combining services from multiple providers such as AWS, Microsoft Azure, and Google Cloud Platform. However, this diversity introduces complexity in managing workloads efficiently, as traditional load balancing algorithms mainly prioritize performance metrics—like response time, bandwidth utilization, or processing delay—while overlooking varying pricing models and cost structures across providers. Consequently, organizations may experience rising operational expenses even when performance levels are satisfactory. To address this, the paper proposes a novel Cost-Efficient Load Balancing Algorithm (CELBA) that dynamically distributes workloads across multiple clouds based on real-time cost, latency, and resource utilization metrics. CELBA continuously monitors each provider's performance and pricing to determine the most cost-effective allocation for incoming tasks. Unlike conventional algorithms such as Round Robin or Least Connection that rely on static or performance-only metrics, CELBA uses a weighted composite scoring function that evaluates both economic and technical factors in real time, ensuring workloads are assigned not only to the most responsive nodes but also to those offering the best cost-to-performance ratio. Experimental evaluation shows that CELBA can reduce operational costs by up to 25% compared to traditional load balancing strategies while maintaining or improving system throughput and latency, validating its efficiency and practicality for modern cloud infrastructures. In summary, the abstract highlights the motivation, methodological innovation, and key findings, underlining that CELBA offers a balanced solution to performance optimization and cost reduction. The algorithm represents a significant step toward intelligent, adaptive, and economically sustainable load balancing for future multi-cloud ecosystems, where both cost efficiency and high performance are essential for competitive cloud operations. To address this challenge, the paper proposes a novel Cost-Efficient Load Balancing Algorithm (CELBA) that dynamically distributes workloads across multiple cloud environments based on a combination of real-time cost, latency, and resource utilization metrics. CELBA continuously monitors each provider's performance and pricing parameters to identify the most cost-effective allocation for incoming tasks. Unlike conventional algorithms, such as Round Robin or Least Connection, which use static or performance-only metrics, CELBA employs a weighted composite scoring function that evaluates both economic and technical factors in real-time. This ensures that workloads are assigned not only to the most responsive nodes but also to those that offer the best cost-to-performance ratio.**

The experimental evaluation presented in the abstract demonstrates that CELBA can achieve a reduction of up to 25% in operational cost compared to traditional load balancing strategies, while maintaining comparable or even improved system throughput and latency. This validates the efficiency and practicality of the proposed approach for modern cloud infrastructures.

Keywords— **Multi-cloud, Load Balancing, Cost Optimization, Resource Management, Cloud Computing, Scheduling Algorithm**

I. INTRODUCTION

Cloud computing enables organizations to leverage scalable and flexible infrastructure resources. However, relying on a single cloud provider introduces limitations such as vendor lock-in, cost volatility, and limited geographical redundancy. To overcome these constraints, many enterprises adopt multi-cloud environments, integrating resources from multiple providers such as AWS, Microsoft Azure, and Google Cloud.

While multi-cloud deployments increase resilience and flexibility, they also introduce complex challenges in workload distribution, performance monitoring, and cost optimization. Current load balancing algorithms (e.g., Round Robin, Least Connection, or Weighted Least Response Time) do not account for dynamic cost variations and resource heterogeneity across cloud providers.

This paper introduces CELBA – a Cost-Efficient Load Balancing Algorithm, which optimizes workload placement based on three core parameters:

1.Service Cost
2.Resource Utilization
3.Performance Latency

The proposed model dynamically adjusts allocation decisions to minimize total cost while sustaining service quality.

II. LITERATURE REVIEW

Existing research primarily focuses on performance-based load balancing or cost-aware resource scheduling, but few integrate both effectively.

- ➢ **Performance-Oriented Algorithms:** Traditional approaches such as Least Response Time or Weighted Round Robin prioritize latency reduction but disregard cost differentials between cloud vendors.
- ➢ **Cost-Based Models:** Some studies employ linear programming or auction-based resource selection to minimize cost; however, they often fail to maintain optimal load distribution under varying workloads.
- ➢ **Hybrid Methods:** Few hybrid approaches exist that adaptively balance both cost and performance, yet most lack real-time adaptability in multi-cloud contexts.

CELBA bridges this gap by integrating real-time pricing, resource metrics, and QoS parameters into a unified decision framework.

III. METHODOLOGIES

A. Existing System

In the existing multi-cloud load balancing systems, algorithms like Round Robin and Least Connection distribute workloads evenly among servers or based on the number of active connections [4]. These algorithms primarily target performance parameters such as response time or bandwidth utilization but ignore cost fluctuations between providers. Consequently, organizations experience unnecessary cost escalation, as workloads may be directed to higher-priced cloud instances even when cheaper alternatives are available [3]. Moreover, static algorithms lack adaptability to real-time changes in resource availability, utilization, or pricing.

B. Proposed Algorithm: Celba

Algorithm Overview

CELBA dynamically evaluates each cloud node Ci based on the following composite score:

Score(Ci) = $\alpha \times 1/\text{Latency}(Ci) + \beta \times 1/\text{Utilization}(Ci) + \gamma \times 1/\text{Cost}(Ci)$

Where:

- α,β,γ are adjustable weight factors based on policy priorities.
- Latency(Ci) represents average response time.
- Utilization(Ci) represents CPU or memory load.
- Cost(Ci) represents per-unit operational cost.

The algorithm selects the node with the maximum score for new workload allocation.

Algorithm Steps

- **Input**: Set of available cloud nodes N={C1,C2,...,Cn}
- **Monitor**: Collect real-time metrics for latency, utilization, and cost.
- **Normalize**: Scale all metrics between 0 and 1.
- **Compute**: Evaluate the composite score for each node.
- **Select**: Assign the workload to node with highest Score(Ci).
- **Update**: Recompute scores periodically to adapt to workload changes.

C. Modules

The proposed CELBA framework consists of four major components:

- **Resource Monitor**: Collects live metrics such as CPU usage, bandwidth, and latency from all connected cloud instances.
- **Cost Analyzer**: Continuously retrieves cloud provider pricing data (on-demand, spot, or reserved instances).
- **Decision Engine**: Executes the CELBA algorithm to determine optimal task-to-cloud assignments.
- **Load Dispatcher**: Redirects incoming requests according to the Decision Engine’s output.

IV. EXPERIMENTAL SETUP AND RESULTS

A. Test Environment

- **Cloud Providers**: AWS, Azure, Google Cloud
- **Instances**: 12 total (4 per provider)
- **Workload**: Synthetic HTTP requests (1000–10,000 per cycle)

- **Metrics Evaluated**: Response time, cost per request, throughput.

B. Results Summary

Algorithm	**Average Cost ($/hour)**	**Avg Latency (ms)**	**Throughput (req/sec)**
Round Robin	12.4	110	820
Least Connection	11.8	100	840
CELBA (Proposed)	**9.1**	**95**	**855**

The results demonstrate that CELBA achieves up to 25% cost reduction without compromising performance.

V. DISCUSSION

CELBA's adaptability allows it to maintain efficiency in environments with fluctuating prices and heterogeneous instance performance. The algorithm also supports hybrid and edge-cloud scenarios. However, limitations include:

- Dependence on accurate, real-time pricing APIs
- Potential overhead for frequent metric collection

Future enhancements could include integrating machine learning prediction models to forecast workload spikes and pricing trends.

VI. CONCLUSION

This paper presents CELBA, a cost-efficient load balancing algorithm for multi-cloud deployments. By considering both cost and performance metrics in real time, CELBA provides a balanced approach to workload distribution. Experimental results confirm its effectiveness in reducing operational costs while maintaining service quality, making it a promising solution for dynamic multi-cloud environments.

VII. FUTURE ENHANCEMENT

Future developments of CELBA could involve integrating machine learning-based prediction models to anticipate workload spikes and price fluctuations. This predictive capability would enable proactive decision-making rather than reactive balancing. Additionally, extending CELBA for hybrid and edge-cloud environments could further enhance scalability and responsiveness. Implementing energy-aware scheduling and carbon footprint optimization may also contribute to sustainable cloud computing strategies in the future

VIII. REFERENCES

[1]. M. Armbrust et al., "A View of Cloud Computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.

[2]. K. Hwang and J. Dongarra, Cloud Computing for Machine Learning and Cognitive Applications, MIT Press, 2017, 624 pp.

[3]. N. Grozev and R. Buyya, "Multi-Cloud Provisioning and Load Distribution," Future Generation Computer Systems, vol. 29, no. 6, 2014.

[4]. C. Li and L. Li, "Cost-Aware Resource Scheduling in Multi-Cloud Systems," IEEE Trans. Cloud Comput., 2022.

[5]. T. Erl, Cloud Computing: Concepts, Technology & Architecture, Prentice Hall, 2013, 528 pp.

[6]. J. Mulder, Mastering Multi-Cloud: Architecting Solutions Across AWS, Azure, and GCP, Packt, 2023, 470 pp.

[7]. K. Hwang, G. Fox, and J. Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Morgan Kaufmann, 2012, 672 pp.

[8]. J. Weinman, Cloudonomics: The Business Value of Cloud Computing, Wiley, 2012, 416 pp.

[9]. A. Chandaka and K. Kant, Multi-Cloud Architecture and Governance, Microsoft Press, 2021, 304 pp.

[10]. P. Raj and G. Sam, Handbook of Research on Cloud Computing and Big Data Applications in IoT, IGI Global, 2018, 609 pp.

[11]. R. Weber, D. Mateos, and R. Buyya, "Cost-aware Multi-Cloud Resource Allocation: A Survey," J. Cloud Comput., DOI: 10.1186/s13677-019-0148-9, 2019.

[12]. S. Tuli et al., "HUNTER: AI-based Holistic Resource Management for Sustainable Cloud Computing," arXiv preprint, arXiv:2110.05529, 2021.

[13]. A. Aghdashi and S. L. Mirtaheri, "Novel Dynamic Load Balancing Algorithm for Cloud-Based Big Data Analytics," arXiv preprint, arXiv:2101.10209, 2021.

[14]. F. Li, J. Wen, W. Tan, and W. Cai, "Multi-objective Optimization of Clustering-based Scheduling for Multi-workflow on Clouds," arXiv preprint, arXiv:2205.11173, 2022.

[15]. K. Matrouk (Ed.), "Scheduling Algorithms in Fog and Cloud: Survey and Perspectives," Int. J. Networked Distributed Computing, DOI: 10.2991/ijndc.k.210111.001, 2021.