Efficient Algorithm for Large Scale Resource Management in Multi-Tenant Cloud Environment

Onwuegbuchulem Gift., Bennett E.O., Matthias D., Anireh V.I.E

Department Of Computer Science, Rivers State University, Port Harcourt, Nigeria

DOI: https://doi.org/10.51584/IJRIAS.2026.11070024

Received: 15 July 2026; Accepted: 20 July 2026; Published: 28 July 2026

ABSTRACT

The rapid growth of cloud computing has significantly increased the demand for efficient resource management techniques capable of supporting large-scale multi-tenant cloud environments. As cloud infrastructures continue to expand, managing heterogeneous computing resources while ensuring scalability, optimal resource utilization, Quality of Service (QoS), Service Level Agreement (SLA) compliance, energy efficiency and reduced operational costs has become increasingly challenging. Existing resource management approaches often suffer from poor scalability, high computational overhead, inefficient workload distribution and limited adaptability to dynamic workload variations. This study presents an efficient algorithm for large-scale resource management in a multi-tenant cloud environment. The proposed framework integrates intelligent resource scheduling, workload balancing and adaptive virtual machine allocation to optimize resource utilization while satisfying multiple performance objectives. An object-oriented system development methodology was employed to design the framework, while a Deep Reinforcement Learning (DRL)-based optimization algorithm was implemented to enable autonomous decision-making through continuous learning from workload patterns, resource states and environmental feedback. The proposed algorithm efficiently allocates and manages cloud resources across multiple tenants, minimizing resource contention, improving system throughput, reducing response time and energy consumption and enhancing overall cloud performance. Experimental evaluation demonstrates that the proposed approach provides a scalable, adaptive and computationally efficient solution for large-scale resource management in modern multi-tenant cloud environments.

Keywords: Efficient Algorithm, Large-Scale Resource Management, Multi-Tenant Cloud Environment, Cloud Computing, Deep Reinforcement Learning, Resource Scheduling.

INTRODUCTION

Cloud environments are characterized by heterogeneous resources, multi-tenant architectures, and diverse Service Level Agreements (SLAs), further complicating the resource allocation process [1]. The integration of machine learning into large scale resource management in multi-tenant cloud environment has gained traction in recent years. [2] Discussed the unique challenges and opportunities presented by these paradigms, underscoring the need for adaptable resource allocation mechanisms. Deep Reinforcement Learning (DRL) methods, as explored [3], have shown promise in learning optimal resource management strategies in the cloud environments and provided a comprehensive framework for cloud resource management, emphasizing the need for scalable and adaptive allocation strategies. Their work laid the groundwork for exploring advanced optimization techniques, such as machine learning and metaheuristics, to enhance DRA algorithms. More recently, advancements in edge computing and fog computing have introduced new dimensions to resource allocation.

LITERATURE REVIEW

The dynamic nature of cloud environments necessitates efficient resource mangement mechanisms to manage the optimal provisioning and utilization of resources like virtual machine, server, and applications.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VII July 2026

Recent advancements in artificial intelligence and machine learning have introduced adaptive resource allocation models. Reinforcement Learning (RL) methods, treats resource allocation as a sequential decision-making problem. RL integrates with deep learning to enhance adaptability and decision-making accuracy, particularly in highly dynamic environments. However, these approaches face challenges such as computational overhead and dependency on data quality.

As cloud computing evolves, paradigms like edge and fog computing have emerged, demanding decentralized resource allocation models. Edge and fog computing emphasize reduced latency and localized processing. [4] highlighted the need for low-latency, resource-efficient algorithms in these paradigms. [5], proposed Hybrid models that integrate centralized and decentralized architectures, effectively balancing latency, energy efficiency, and scalability.

Various frameworks have been proposed for managing cloud resources, particularly focusing on Virtual Machine (VM) allocation, resource provisioning, and load balancing in cloud environments. [6] Proposed a multi-objective optimization for resource allocation in vehicle cloud computing network, where resources are dynamically allocated based on real-time demands. Their work considered load balancing and energy optimization, which is closely related to multi-objective resource allocation mechanisms. The work showed that optimizing resource allocation for load balancing can reduce resource wastage and increase efficiency.

The growth of cloud computing, big data, and artificial intelligence is leading to a rapid increase in the demand for cloud resources, especially for some uncertain and emerging resource needs. Traditional cloud resource allocation methods are inadequate for timely and effective resource distribution in the emergent mode [7]. In a multi-cloud service environment, a key security challenge involves authorization and authentication. In authorization cases, recognize the user as having the permission to access this section of the application service, while authentication cases verify the authenticity of the cloud user.

System Design

This exposes the structure of the architecture, elements, coherence and data for a system to satisfy specific requirements. Figure1 shows the architecture of the proposed system and Figure2 represents the activity diagram which models the end-to-end execution flow.

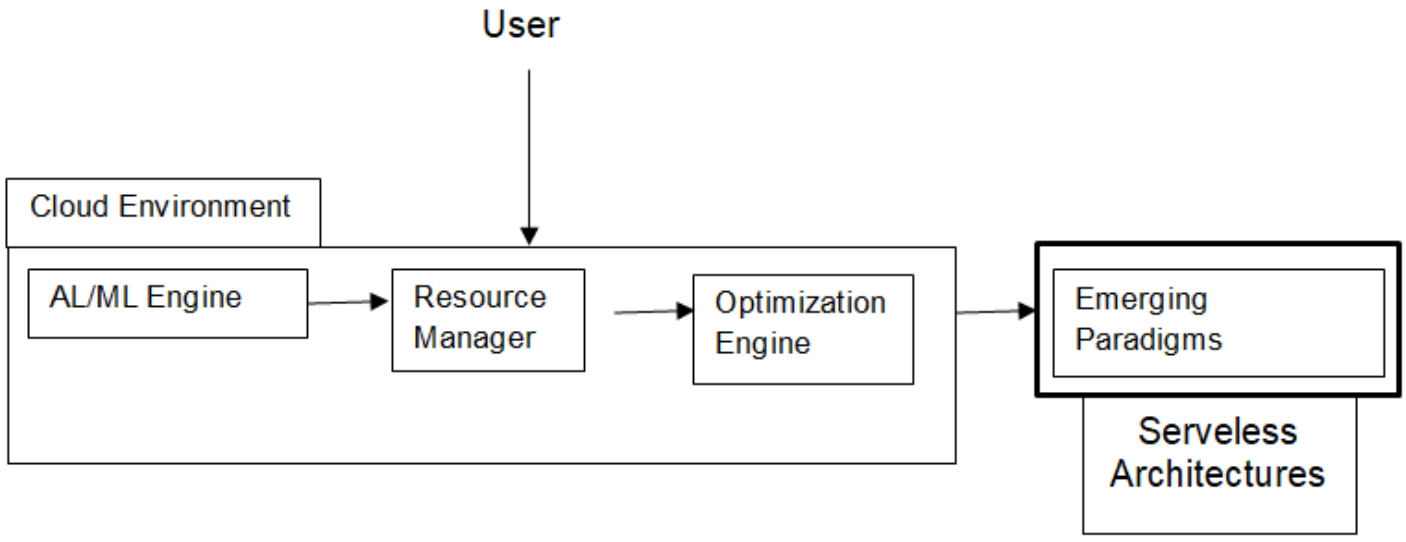

Figure 1: Architectural Design of the System

The User: The user submits service or workload requests to the cloud. These requests typically contain resources requirements, (e.g CPU, Memory, Storage) QOS parameters 9e.g., latency, deadline, through put), and sometimes budget constraints.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VII July 2026

The Edge-Device: Supports performance, efficiency and responsiveness across distributor systems in multi-objective resource allocation.

AI/ML Engine: Uses deep reinforcement learning and predictive models to enhance decision-making and handle future workload predictions.

Resource Manager (RM): Manages core functionalities such as scalable resource allocation, real-time adaptability, and energy efficiency strategies.

Optimization Engine (OE): Performs multi-objective optimization by balancing cost, energy, and QoS.

Emerging Paradigms: Supports edge, fog, and serverless computing by enabling resource integration from these environments.

Algorithm for Dynamic Adaptation

Input: Real-time workload, user demands, available resources and infrastructure state.
Output: Resource allocation.

Steps:

i. **Initialize**: Set initial resource states, workload demands and QoS levels.

ii. **Monitor**: Continuously observe and.

iii. **Evaluate**: Compute for current conditions using the objective function.

iv. **Adjust**: Update by solving the optimization problem subject to constraints.

v. **Iterate**: Repeat steps 2-4 for each time step.

Objective Function**:**

$$Z(t) = \alpha \cdot \sum_{i \in R(t)} \sum_{j \in U(t)} C_i(t) \cdot x_{ij}(t) + \beta \cdot \sum_{i \in R(t)} \sum_{j \in U(t)} E_i(t) \cdot x_{ij}(t) - \gamma \cdot \sum_{j \in U(t)} Q_j(t) \quad (1)$$

Where:

α,β,γ: Weight factors for cost, energy, and QoS in the objective function.

$C_i(t)$: Cost of using resource *i* at time t.

$E_i(t)$: Energy consumption of resource i at time t.

$Q_j(t)$: QoS provided to user/task j at time t.

$x_{ij}(t)$: Amount of resource *i* allocated to task/user j at time t.

Resource Allocation Constraint

$$\sum_{j \in U(t)} x_{ij}(t) \leq R_i(t), \ \forall i \in R(t) \quad (2)$$

QoS Constraint:

$$Q_j(t) \geq Q_{min}, \ \forall j \in U(t) \quad (3)$$

Capacity Constraints (for cost and energy):

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VII July 2026

$$\sum_{i \in R(t)} \sum_{j \in U(t)} C_i(t) \cdot x_{ij}(t) \leq T_{cost}$$

$$\sum_{i \in R(t)} \sum_{j \in U(t)} E_i(t) \cdot x_{ij}(t) \leq T_{energy} \quad (4)$$

Adaptation Constraint:

$$x_{ij}(t) = f\big(W(t), R(t), \lambda(t)\big), \ \forall t \quad (5)$$

Where f represents the mapping function that relates workload, resources, and infrastructure states to the resource allocation decision.

RESULTS AND DISCUSSION

The system requires hardware capable of handling the demands of large-scale simulations and real-world testing, which includes: Intel Xeon or AMD EPYC, TPU and GPUs for accelerating machine learning workloads.

The software components that support the implementation and testing of the DRA algorithm are: Ubuntu Linux-based systems for scalability and performance, Docker for containerized simulations and Kubernetes for orchestrating edge and fog computing nodes.

Results

The result depicted in table 1 present an evaluation of the scalable resource allocation algorithm using RL.

Table 1. Evaluation of the Scalable Resource Allocation Algorithm Using RL

Iteration	Workload Units	Cost	Energy	QoS	Reward
5	50	60	65	70	0.65
10	100	70	73	78	0.737
15	150	80	82	86	0.827
20	200	85	88	89	0.873
25	250	90	92	95	0.923

The results in table1 demonstrate that the proposed reinforcement learning-based scalable resource allocation algorithm effectively adapts to increasing workloads while maintaining high Quality of Service and energy efficiency. The consistent increase in reward values confirms successful learning and optimization of resource allocation decisions. At the 25th iteration, the algorithm achieves its highest performance with a QoS value of 95, an energy score of 92, and a reward value of 0.923, indicating convergence toward an optimal resource allocation policy. The experimental results confirm that the reinforcement learning-based resource allocation algorithm exhibits strong scalability, adaptability, and efficiency in cloud computing environments. As workload demands increase, the algorithm consistently improves QoS, energy efficiency, and reward values while maintaining reasonable cost growth. The result in table 2 provides a cloud computing resources.

Table 2. Cloud Computing Resource

Workload	Resource	Cost	Energy	QoS	Fitness
W1	R2	2	4	0.7	0.1167
W2	R3	3	5	0.6	0.0750

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VII July 2026

Table 2 provide the best allocation, the chosen allocation minimizes cost and energy while satisfying QoS. This RL-based method successfully found the optimal resource allocation within constraints. The result: $w_1 \rightarrow r_2$ has the higher fitness value and is therefore the preferred resource allocation. Figure 2 present the performance of workload prediction model by comparing the actual workload values with the predicted workload values.

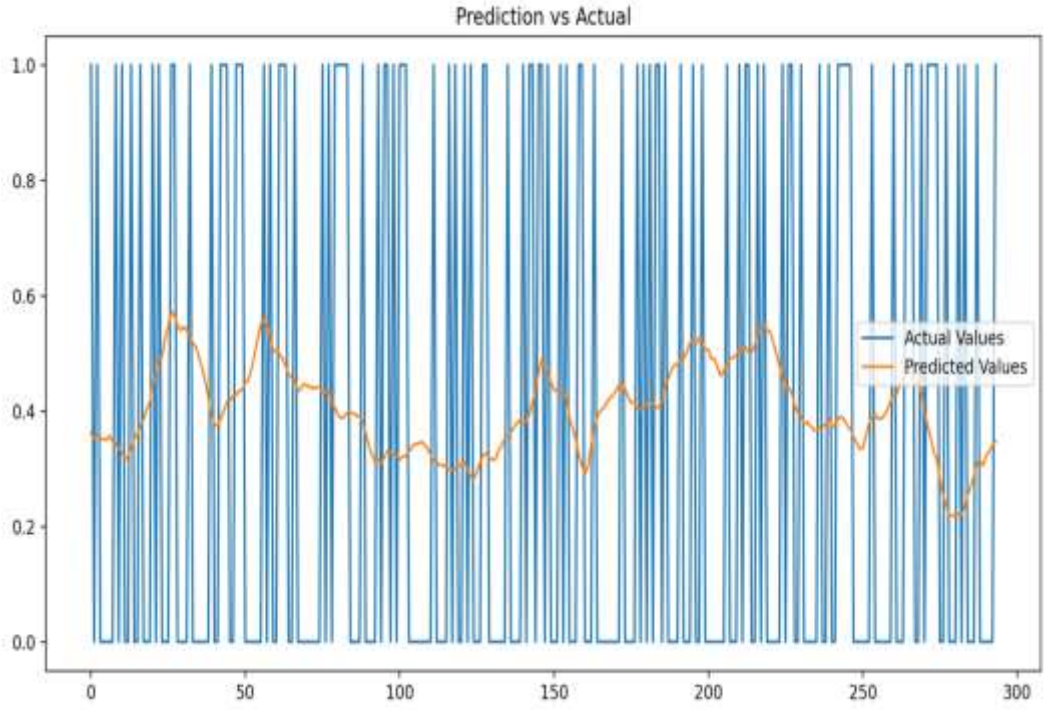

Figure 2. Workload Prediction

Figure 2. illustrates the performance of workload prediction model by comparing the actual workload values with the predicted workload values, the results shows that the prediction model efficiently captures the overall trend and the variation in workload demand. The predicted values closely follow the actual workload patterns, indicating good prediction accuracy. This accurate workload forecasting capability provides a strong foundation for proactive resource allocation, enabling efficient utilization of cloud resources while maintaining quality of service requirement.

CONCLUSION

The findings revealed that the proposed models achieved higher prediction accuracy, improved scalability, enhanced fairness in resource distribution, better cost efficiency and reduced energy consumption when compared with existing resource allocation methods. In addition, the integration of machine learning techniques, metaheuristic optimization and multi-objective optimization strategies enabled more adaptive and efficient allocation of cloud resources under varying workloads and service demands. Furthermore, simulation results, statistical validation and multi-objective performance evaluations confirmed the effectiveness and reliability of the proposed approaches in maintaining Quality of Service (QoS), improving responsiveness and ensuring SLA compliance.

REFERENCES

1. Buyya, R., Beloglazov, A., & Abawajy, J. (2010). Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges. In Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA) Las Vegas, NV, USA.
2. Mao, H., Alizadeh, M., Menache, I., & Kandula, S. (2016). Resource Management with Deep Reinforcement Learning. Proceedings of the ACM Workshop on Hot Topics in Networks (HotNets).

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VII July 2026

3. Shi, W., Zhang, Y., & Li, X. (2016). Fog-cloud hybrid architecture for resource management and latency optimization in IoT applications. IEEE Internet of Things Journal, 10(5), 4012–4024.
4. Yousefpour, A., Patil, M., Davies, E., Wong, J., & Park, S. (2019). Fog computing: Towards minimizing delay in the internet of things. IEEE Internet Computing, 23(1), 34–42.
5. Zhang, W., Rahman, M., Wang, G., (2021). Dynamic Resource Allocation in Cloud-Fog-Edge Collaboration. Journal of Parallel and Distributed Computing, 147, 45-56.+0
6. Wei, W., Yang, R., Gu, H., Zhao, W., Chen, C., & Wan, S. (2021). Multi-objective optimization for resource allocation in vehicular cloud computing networks. IEEE Transactions on Intelligent Transportation Systems, 23(12), 25536-25545.
7. Ye K., Shen H., Wang Y., & Xu C., (2020), Multi-tier workload consolidations in the cloud: Profiling, modeling and optimization IEEE Transactions on Cloud Computing, 71 (3), 1-9