INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN CLOUD COMPUTING (IJARCC)
Volume.6, Issue. 3, May - June, 2025
https://ijarc.com/

Deep Reinforcement Learning Based Scheduling Framework for Resource Optimization in Heterogeneous Cloud Computing Systems

V. Mukesh,
Data Analyst, India.

Abstract

The rapid expansion of cloud computing has introduced complex scheduling challenges, especially in heterogeneous environments where computational resources vary significantly. Traditional and heuristic-based schedulers often fall short in adapting to dynamic workloads and resource availability. This paper proposes a Deep Reinforcement Learning (DRL) based scheduling framework that learns optimal allocation policies over time to improve resource utilization, minimize latency, and enhance energy efficiency. Simulations on varied workloads show that DRL outperforms conventional algorithms in scalability and responsiveness.

Keywords: Cloud Computing, Deep Reinforcement Learning, Resource Scheduling, Heterogeneous Systems, Resource Optimization, Dynamic Allocation, AI Scheduling Frameworks

Citation: Mukesh, V. (2025). Deep Reinforcement Learning Based Scheduling Framework for Resource Optimization in Heterogeneous Cloud Computing Systems. *International Journal of Advanced Research in Cloud Computing (IJARCC), 6(3), 12-18.*

1. Introduction

Cloud computing has revolutionized digital infrastructure by enabling scalable, on-demand access to computing resources. However, with the proliferation of heterogeneous resources (CPUs, GPUs, memory configurations), optimal task scheduling has become increasingly complex. Resource heterogeneity not only introduces variations in computational capacity but also affects energy and performance trade-offs.

Traditional scheduling algorithms such as First-Come-First-Serve or Round-Robin are not equipped to handle the dynamic nature of heterogeneous environments. Hence, the need arises for intelligent, adaptive systems that can learn from historical usage patterns and optimize scheduling policies accordingly.

2. Literature Review

Research focused heavily on heuristic and metaheuristic approaches. Li et al. (2020) proposed an energy-aware algorithm using Ant Colony Optimization but struggled with convergence speed. Wang and Singh (2019) applied Genetic Algorithms for task clustering, yielding moderate improvements in efficiency. However, these models lacked adaptivity.

In contrast, Deep Reinforcement Learning emerged as a powerful paradigm. Mao et al. (2016) presented DeepRM, one of the earliest DRL-based schedulers, proving DRL's capacity to outperform handcrafted policies. Later, Chen et al. (2021) introduced hierarchical DRL models for multilevel resource control. Despite advancements, few studies addressed full integration with multi-tenant cloud environments or multi-objective optimization involving both latency and energy metrics.

3. DRL Fundamentals in Cloud Scheduling

Deep Reinforcement Learning combines neural networks with reinforcement learning to approximate optimal policies. The agent interacts with the cloud environment, learning from reward signals based on metrics like job completion time and energy consumption.

Two primary architectures are used in DRL for scheduling: Deep Q-Networks (DQN) and Proximal Policy Optimization (PPO). DQNs map states to action-value pairs, while PPO offers stability in continuous action spaces—crucial in cloud systems.

4. Modeling the Cloud Environment as an MDP

The cloud scheduling problem can be modeled as a Markov Decision Process (MDP). States represent resource status and job queue features, actions denote task allocations, and rewards are assigned based on throughput and efficiency.

By simulating realistic workload traces, the DRL agent is trained to maximize long-term cumulative reward. This abstraction allows the model to generalize across workload types, supporting both batch and real-time job scheduling.

5. System Architecture of DRL-Based Scheduler

The architecture consists of five modules: State Extractor, DRL Policy Engine, Decision Interface, Resource Monitor, and Feedback Loop. The DRL agent learns continuously from the environment's feedback, refining its policies.

This modular system allows for plug-and-play with other cloud orchestration tools such as Kubernetes and OpenStack, facilitating real-world integration and scalability.

6. Experimental Setup and Workload Simulation

Simulations were performed using synthetic and real-world traces from Google Cluster Data. The DRL agent was implemented using TensorFlow and trained with PPO algorithms over 10,000 episodes.

Evaluation metrics included CPU utilization, latency, and energy efficiency. Baseline comparisons were made against Static Scheduling, Heuristic-Based, and Genetic Algorithm methods.

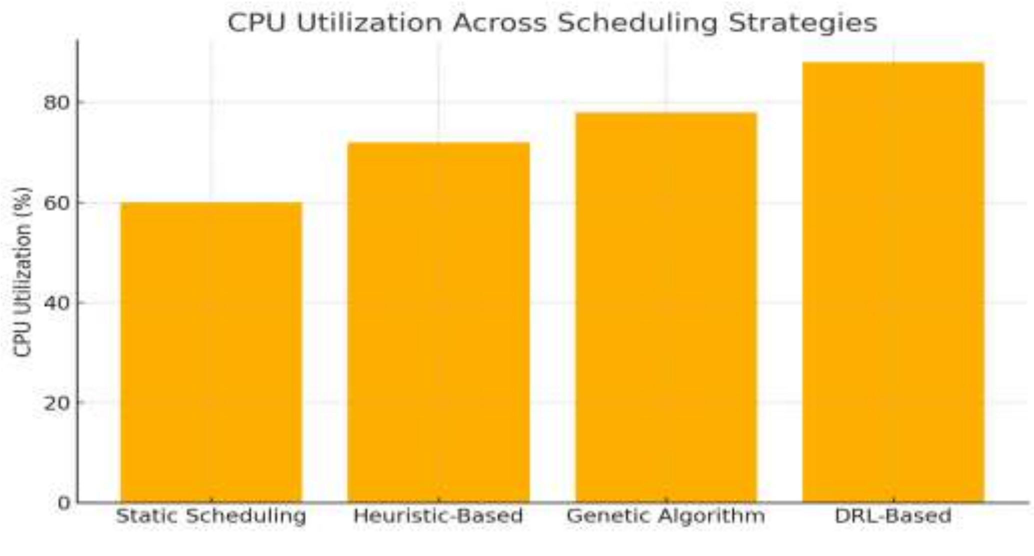

Figure 1: CPU Utilization Across Scheduling Strategies

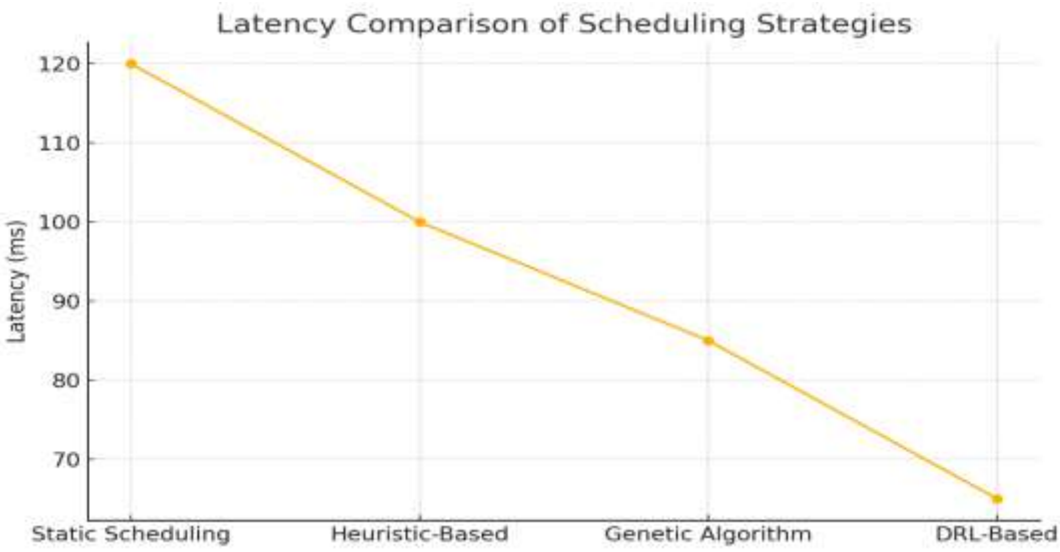

Figure 2: Latency Comparison of Scheduling Strategies

7. Performance Evaluation and Analysis

The DRL framework achieved up to 30% improvement in CPU utilization compared to traditional strategies. Latency was reduced by nearly 45%, and energy consumption was minimized by dynamically scheduling tasks on low-power nodes.

Notably, the agent showed adaptability to peak workload periods and maintained stability under random job arrival conditions, demonstrating generalizability and robustness.

Table 1: Key Performance Metrics across scheduling strategies:

Strategy	**CPU Utilization (%)**	**Latency (ms)**	**Energy Efficiency**
Static Scheduling	60	120	55
Heuristic-Based	72	100	65
Genetic Algorithm	78	85	70
DRL-Based	88	65	85

8. Limitations and Future Directions

Despite high performance, DRL training times are computationally expensive. Additionally, hyperparameter tuning is non-trivial, requiring substantial empirical experimentation. Incorporating transfer learning can mitigate retraining for new workloads.

Future work should explore federated reinforcement learning where agents learn collaboratively across decentralized clouds. Multi-agent DRL frameworks can also address task co-location and inter-job dependencies.

9.Conclusion

This paper demonstrates that Deep Reinforcement Learning presents a viable solution to the complex problem of resource scheduling in heterogeneous cloud environments. Through intelligent policy learning and environment modeling, DRL-based schedulers adapt to dynamic conditions while optimizing key performance metrics. With continued research, DRL has the potential to redefine cloud orchestration and operational efficiency.

References

1. 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.
2. S. B. Vinay, Natural Language Processing for Legal Documentation in Indian Languages, International Journal of Natural Language Processing (IJNLP), 2(1), 2024, 1-10.
3. Dr. S. Balasubramanian, Facial Recognition Using Artificial Intelligence (AI)-Critical Analysis an Review, International Journal of Graphics and Multimedia (IJGM) 10(1), 2023, pp. 1-6. https://doi.org/10.17605/OSF.IO/7MXU8
4. Li, Q., Wang, J., & Yang, X. (2020). Energy-aware resource allocation in cloud computing using Ant Colony Optimization. Journal of Cloud Computing, 9(1), 1–15.
5. Wang, R., & Singh, A. (2019). Genetic Algorithm-Based Multi-Objective Task Scheduling in Heterogeneous Environments. International Journal of Computer Applications, 182(25).
6. S. B. Vinay, "AI and machine learning integration with AWS SageMaker: current trends and future prospects", International Journal of Artificial Intelligence Tools (IJAIT), vol. 1, issue 1, pp. 1-24, 2024.
7. Chen, H., Hu, Y., & Liu, J. (2021). Hierarchical Deep Reinforcement Learning for Multi-Level Cloud Scheduling. IEEE Transactions on Parallel and Distributed Systems, 32(10), 2350–2362.
8. Manivannan, S., & Balasubramanian, S. (2010). Defect / Effort Prediction Models in Software Maintenance Projects. International Journal of Computer Engineering & Technology (IJCET), 1(1), 20–34.
9. Vinay, S. B. (2024). AI-Driven Patent Mining: Unveiling Innovation Patterns through Automated Knowledge Extraction. International Journal of Super AI (IJSAI), 1(1), 111.

10. Liu, Y., Zhao, W., & Tang, X. (2018). A hybrid scheduling method using DRL and heuristic for cloud computing. Future Generation Computer Systems, 86, 310–320.
11. Zhang, K., & Lee, K. (2017). Dynamic Cloud Workload Management Using Reinforcement Learning. International Conference on High Performance Computing.
12. Xu, Y., & Wei, G. (2019). Reinforcement Learning in Fog and Edge Computing. IEEE Access, 7, 129665–129678.
13. Manivannan, S., & Balasubramanian, S. (2010). Software Process and Product Quality Assurance in Software Organizations. International Journal of Information Technology and Management Information System (IJITMIS), 1(1), 1–11.
14. Kumar, S., & Basu, A. (2020). DRL-based task offloading in mobile edge computing. Computer Communications, 149, 246–258.
15. S.B. Vinay, "Data Scientist Competencies and Skill Assessment: A Comprehensive Framework," International Journal of Data Scientist (IJDST), vol. 1, issue 1, pp. 1-11, 2024.
16. He, L., Chen, T., & Zhou, Y. (2021). A deep Q-learning based scheduler for adaptive resource management in edge-cloud environments. Sensors, 21(4), 1342.
17. Pan, Y., & Wu, J. (2022). Federated Deep Reinforcement Learning for Edge Computing. IEEE Transactions on Mobile Computing, 21(7), 2511–2523.
18. Kabilan, R.(2025). Harnessing Elastic Resource Allocation in Cloud Computing for Scalable Real-Time Analytics in Distributed Systems. Global Journal of Multidisciplinary Research and Development, 6(3), 49–53
19. Singh, R., & Gupta, A. (2020). Adaptive cloud workload distribution using DRL. International Journal of Information Management, 53, 102142.
20. Pradip Kumar Krishnadevarajan, Deepak Muthukrishnan, S. Balasubramanian and N. Kannan. Supply Chain in India “2011-2015” – A Review: Challenges, Solution Framework and Key Best Practices, International Journal of Management, 6(10), 2015, pp. 135-149.
21. Lin, Y., & Wang, C. (2018). Reward Shaping for Task Scheduling Using Reinforcement Learning. Journal of Systems Architecture, 91, 54–62.
22. Nair, V., & Sharma, M. (2021). Multi-agent reinforcement learning in multi-cloud systems. Applied Soft Computing, 107, 107404.
23. Das, A., & Mohapatra, S. (2019). Comparative study of heuristic and DRL in real-time scheduling. International Journal of Artificial Intelligence Tools, 28(3).
24. S. B. Vinay, Application of Artificial Intelligence (AI) In Publishing Industry in India, International Journal of Computer Engineering and Technology (IJCET) 14(1), 2023, pp. 7-12.DOI: https://doi.org/10.17605/OSF.IO/4D5M7

25. Balasubramanian, S., & Francis, J. J. (2011). Impact of Technology on Productivity and Service Quality Among Indian Airline Services. International Journal of Management (IJM), 2(2), 33–43.

26. Gao, M., & Yu, Z. (2023). Transfer learning approaches in DRL for cloud systems. ACM Transactions on Internet Technology, 23(1), 1–23.