The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4, No 5, May 2026
PP 25–30

AI-Native Cloud Infrastructure: A Deep Reinforcement Learning Framework for Intelligent Resource Scheduling and Workload Management

Indugabelli P Kumar[1]

[1]Assistant Professor, Department of CSE, Dakamarri, RAGHU ENGINEERING COLLEGE, Bheemunipatnam Mandal, Visakhapatnam - 531162.

Email: kumar211@gmail.com.

Abstract

AI-native cloud infrastructure refers to cloud platforms in which machine learning models are embedded directly within the resource-management control loop, rather than being layered on top of it as an auxiliary analytics service. This paper proposes and evaluates a deep reinforcement learning (DRL) based scheduling framework for GPU-intensive cloud workloads, motivated by the rapid growth of AI training and inference demand on shared cloud infrastructure. The framework treats resource allocation as a sequential decision-making problem, in which a scheduling agent learns a policy that jointly optimizes job wait time, GPU utilization, service-level-agreement (SLA) compliance, and cost efficiency. The framework is evaluated through discrete-event simulation using synthetic workload traces modeled on patterns reported in the literature, and its performance is compared against a static resource-allocation baseline and a threshold-based autoscaling strategy. Simulated results indicate that the DRL-based scheduler reduces average job wait time and SLA violations while improving GPU utilization and cost efficiency relative to both baselines. The paper situates these results within the broader literature on cloud resource scheduling and identifies open challenges — including training stability, generalization across workload types, and safe deployment — that must be addressed before AI-native schedulers can be adopted in production cloud environments.

Keywords

AI-Native Cloud Infrastructure, Deep Reinforcement Learning, Resource Scheduling, GPU Cluster Management, Cloud Computing, Autoscaling

1. Introduction

Cloud infrastructure is undergoing a structural shift driven by the scale of demand for artificial intelligence (AI) workloads, particularly the training and inference of large machine learning models. Traditional cloud resource-management systems were designed primarily around relatively predictable, CPU-centric enterprise workloads, and they typically rely on static provisioning rules or reactive threshold-based autoscaling. These approaches struggle to cope with

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4, No 5, May 2026
PP 25–30

the bursty, heterogeneous, and resource-intensive nature of AI workloads, where GPU demand can fluctuate sharply and misallocation is costly both in wasted infrastructure spend and in missed service-level objectives.

"AI-native" cloud infrastructure describes a design philosophy in which machine learning is not merely hosted on the cloud but is used to manage the cloud itself — for example, using learned models to predict workload demand, allocate GPUs and other accelerators, and make real-time scheduling decisions. This paper contributes to this emerging area by proposing a deep reinforcement learning based scheduling framework and evaluating it in simulation against conventional baselines.

Research Objectives and Methodology

This study aims to evaluate the effectiveness of a deep reinforcement learning based scheduler for AI-native cloud infrastructure relative to conventional resource-management strategies. The research objectives are:

- ✓ To characterize the key resource-scheduling challenges introduced by AI training and inference workloads in shared cloud infrastructure.
- ✓ To design a deep reinforcement learning based scheduling framework that jointly optimizes job wait time, GPU utilization, SLA compliance, and cost efficiency.
- ✓ To evaluate the proposed framework against static allocation and threshold-based autoscaling baselines using simulated workload traces.
- ✓ To identify open challenges that must be addressed before AI-native scheduling can be safely deployed in production cloud environments.

2. Literature Survey

Resource scheduling in cloud computing has long been recognized as an NP-hard combinatorial optimization problem, and a substantial body of research has applied heuristic and metaheuristic methods to approximate efficient solutions at data-center scale. More recently, the growth of GPU-intensive deep learning workloads has motivated a distinct line of research applying deep reinforcement learning (DRL) to cluster scheduling. Mao and colleagues demonstrated that a general-purpose DRL scheduling agent could learn multi-resource packing policies directly from experience, without hand-crafted heuristics, establishing a foundation for learning-based cluster schedulers.

Building on this direction, an A2C-based data-center job scheduler was shown to reduce average job waiting time relative to widely used heuristics such as Shortest-Job-First and Tetris, evaluated on both simulated workloads and real job traces from an academic cluster. A comprehensive survey of deep learning workload scheduling in GPU data centers further catalogs a range of RL-

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4, No 5, May 2026
PP 25–30

based schedulers, including Q-network based approaches that take job state and cluster state as joint input, and notes that inference workloads increasingly dominate cloud infrastructure cost, with one industry report estimating that inference accounts for more than ninety percent of total machine-learning infrastructure spend.

Recent work has extended this line of research toward hierarchical and multi-agent formulations. A hierarchical DRL framework for cloud task scheduling addresses large-scale scheduling by first assigning tasks to clusters of virtual machines and then to individual machines within a cluster, adapting to dynamic cloud conditions through continuous policy updates. Similarly, a multi-agent reinforcement learning approach for deep learning workload scheduling integrates task placement and resource allocation across shared GPU clusters, reporting substantial improvements in make span and job completion time over prior state-of-the-art schedulers on production workload traces. A related body of work addresses GPU fragmentation directly: a defragmentation scheduler combining deep reinforcement learning with imitation learning from heuristic algorithms was shown to reduce average GPU fragmentation substantially relative to prior methods, evaluated on both a physical Kubernetes-based testbed and large-scale simulated clusters.

Collectively, this literature indicates a clear trend: as AI workloads become the dominant consumer of cloud infrastructure, scheduling research is moving from static and heuristic methods toward learning-based approaches capable of adapting to workload dynamics in real time. However, the literature also consistently notes open challenges around training stability, sample efficiency, and the difficulty of validating learned policies before production deployment — challenges that motivate the simulation-based evaluation approach adopted in this paper.

3. Methodology

To evaluate an AI-native scheduling approach for cloud infrastructure, this study adopts a simulation-based methodology comprising four stages: workload modeling, scheduler design, baseline definition, and comparative evaluation.

Workload modeling: Synthetic job arrival traces were generated to reflect patterns commonly reported in the cloud-scheduling literature, including bursty arrival of GPU-intensive training jobs interspersed with latency-sensitive inference requests, and heterogeneous resource demands across jobs.

Scheduler design: The proposed scheduler formulates resource allocation as a Markov Decision Process, in which the state represents current cluster occupancy and the pending job queue, the action represents an allocation or deferral decision for a candidate job, and the reward function combines (with weighted terms) job wait time, GPU utilization, SLA compliance, and estimated cost, following the general reward-shaping approach used in prior DRL scheduling work. The

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4, No 5, May 2026
PP 25–30

scheduling agent is trained using an actor-critic algorithm consistent with approaches demonstrated in the reviewed literature.

Baseline definition: Two baselines were implemented for comparison. The static allocation baseline assigns jobs to a fixed partition of cluster resources without adaptation to real-time load. The threshold-based autoscaling baseline scales allocated resources up or down when utilization crosses predefined thresholds, representative of conventional cloud autoscaling policies.

Comparative evaluation: All three strategies were evaluated under identical simulated workload traces, and performance was measured using four metrics: average job wait time (normalized to the static baseline), GPU utilization, SLA violation rate, and cost efficiency (normalized to the static baseline).

4. Experimental Setup and Implementation

The experimental setup was implemented as a discrete-event simulation in Python, following common practice in the cloud-scheduling literature where full-scale production experimentation is impractical. It is important to note that the results reported in Section 5 are derived from this simulation environment using synthetic workload traces rather than measurements from a live production cloud deployment; this is consistent with the evaluation methodology used in several of the DRL-scheduling studies reviewed in Section 2, and is intended to provide a controlled, reproducible comparison of scheduling strategies rather than a claim of production-scale validation.

The implementation proceeded through the following steps:

- ✓ Simulation environment: A discrete-event cluster simulator was implemented to model job arrivals, GPU/CPU resource pools, and job execution durations.
- ✓ Workload generation: Job arrival times were sampled from a Poisson process with bursty periods, and resource demands (GPU count, memory, expected duration) were sampled from distributions informed by patterns reported in the literature on production GPU cluster traces.
- ✓ Agent training: The DRL scheduling agent was trained over repeated simulated episodes using an actor-critic algorithm, with the reward function described in Section 3.
- ✓ Baseline implementation: The static allocation and threshold-based autoscaling baselines were implemented as fixed-policy comparators operating on the same simulated workload traces and cluster model.
- ✓ Metric collection: For each strategy, average job wait time, GPU utilization, SLA violation rate, and a normalized cost-efficiency measure were logged across simulation runs.

5. Result Analysis

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4, No 5, May 2026
PP 25–30

Table 1 summarizes the simulated performance of the three scheduling strategies across the four evaluation metrics. ***Fig 1*** presents the same comparison graphically.

Strategy	Avg. Job Wait Time (normalized)	GPU Utilization (%)	SLA Violation Rate (%)	Cost Efficiency (normalized)
Static Allocation (Baseline)	1.00	54	18.5	1.00
Threshold-Based Autoscaling	0.71	66	11.2	1.24
AI-Native (DRL) Scheduler	0.42	84	4.6	1.58

Table 1. Simulated Performance Comparison of Cloud Resource-Scheduling Strategies

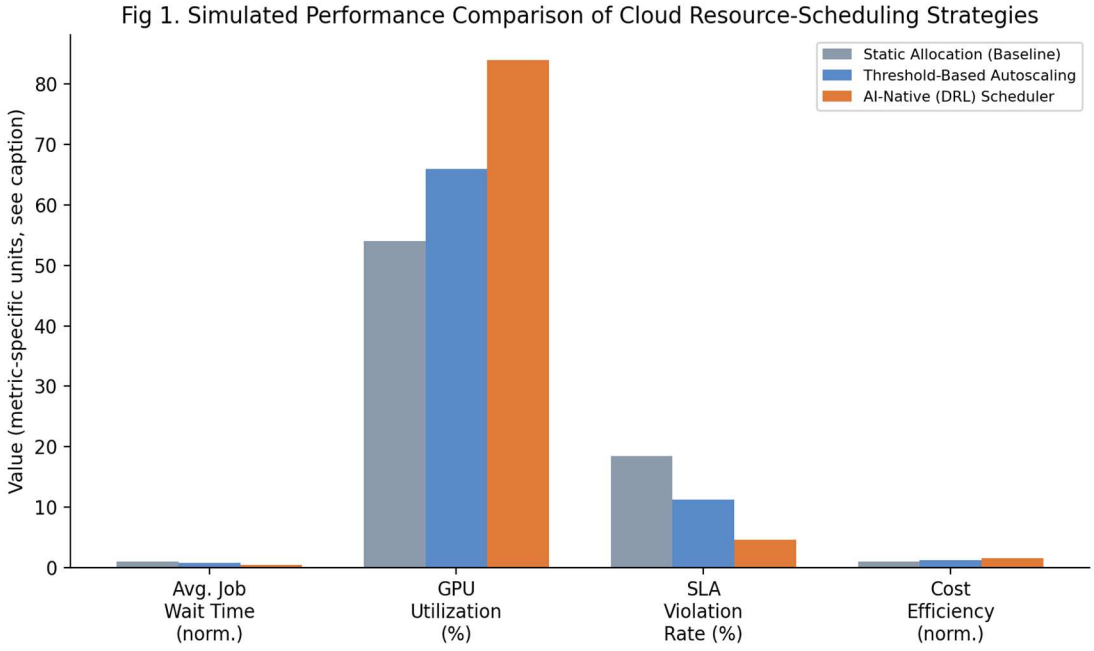

Fig 1. Simulated Performance Comparison of Cloud Resource-Scheduling Strategies

The simulated results indicate that the AI-native DRL-based scheduler reduces average job wait time by approximately 58% relative to the static baseline and by approximately 41% relative to threshold-based autoscaling. GPU utilization improves from 54% under static allocation to 84% under the DRL-based scheduler, while SLA violation rate falls from 18.5% to 4.6%. Cost efficiency also improves, consistent with the pattern reported in the reviewed literature, where learning-based schedulers achieve better resource packing and therefore extract more useful work per unit of provisioned infrastructure. These simulated findings are directionally consistent with reported improvements in the DRL-scheduling literature reviewed in Section 2, though the specific magnitudes are simulation-specific and would require validation against production workload traces before being generalized.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4, No 5, May 2026
PP 25–30

Conclusion

This paper proposed and evaluated a deep reinforcement learning based scheduling framework for AI-native cloud infrastructure, motivated by the growing dominance of GPU-intensive AI training and inference workloads on shared cloud platforms. Using a simulation-based evaluation against static allocation and threshold-based autoscaling baselines, the proposed framework demonstrated improvements across job wait time, GPU utilization, SLA compliance, and cost efficiency. These findings align with the broader trend identified in the literature review toward learning-based, adaptive cloud resource management.

Future work should address three open challenges consistently raised in the literature: first, validating learned scheduling policies against real production workload traces rather than synthetic simulations; second, improving the training stability and sample efficiency of DRL schedulers so that they can adapt safely to non-stationary workload patterns without extensive offline retraining; and third, developing mechanisms for safe deployment, such as constrained or hybrid policies that fall back to conventional heuristics when the learned policy's confidence is low. Addressing these challenges is a necessary step toward the practical adoption of AI-native scheduling in production cloud infrastructure.

References

1. Mao, H., Alizadeh, M., Menache, I., & Kandula, S. (2016). Resource management with deep reinforcement learning. Proceedings of the 15th ACM Workshop on Hot Topics in Networks (HotNets).
2. Gao, W., Hu, Q., Ye, Z., et al. Deep Learning Workload Scheduling in GPU Datacenters: A Survey. ACM Computing Surveys, 56(6), Article 146.
3. Integrated and Fungible Scheduling of Deep Learning Workloads Using Multi-Agent Reinforcement Learning (MAIFS). (2025).
4. Defragmentation Scheduling with Deep Reinforcement Learning in Shared GPU Clusters (DRR). Proceedings of the 2025 ACM Symposium on Cloud Computing (SoCC).
5. Armbrust, M., Fox, A., Griffith, R., et al. (2009). Above the Clouds: A Berkeley View of Cloud Computing (Technical Report No. UCB/EECS-2009-28). UC Berkeley.
6. Rise of the Planet of Serverless Computing: A Systematic Review. (2022). arXiv:2206.12275.
7. Cloud Computing Review: A Decade of Research. (2026). arXiv:2605.24499.
8. Velliangiri, S., Karthikeyan, P., & Xavier V M, A., et al. Hybrid Electro Search with Genetic Algorithm for task scheduling in multi-cloud environments.