Artificial Intelligence and Computing Innovations

Intelligence Science Publishing

https://www.ispublishing.com

ISSN: 3070-4944

Vo.4, No.2, 2024

Research on Hierarchical Multi-Agent Reinforcement Learning Resource Orchestration for Large-Scale Heterogeneous Distributed Clusters with Dynamic Load Awareness

Yuqi Tang

New York University, New York, USA

yuqit2861@gmail.com

Abstract: To address the challenges of complex resource states, random task arrivals, significant differences in node capabilities, and multiple couplings of scheduling objectives in large-scale heterogeneous distributed clusters, a hierarchical multi-agent reinforcement learning resource orchestration method integrating dynamic load awareness is constructed. This method first extracts time-varying load information from machine resource occupancy, task queuing status, service deployment density, and node capacity constraints, and constructs a state representation with global context awareness capabilities by combining cluster topology relationships to enhance the method's ability to characterize complex operating environments. Based on this, a hierarchical collaborative decision-making mechanism is adopted, dividing the resource orchestration process into global coordination by the upper-level manager and local control by the lower-level executors, achieving effective connection between global planning and node-level execution through sub-objective propagation. Furthermore, to address the matching problem between heterogeneous nodes and task requirements, a compatibility-aware scoring mechanism is introduced to improve the rationality of resource allocation, execution stability, and orchestration accuracy. This method can balance resource utilization, task waiting control, load balancing, and scheduling success capability within a unified framework, thereby improving the overall resource organization quality in complex cluster environments. Comparative experimental results show that the proposed method exhibits good overall performance across multiple key evaluation metrics, indicating that the method can effectively adapt to large-scale heterogeneous distributed cluster resource orchestration scenarios driven by dynamic loads, and has strong method effectiveness and application value.

Keywords: Heterogeneous computing power orchestration; time-varying load characterization; hierarchical collaborative decision-making; cluster scheduling optimization

1. Introduction

With the continuous evolution of cloud computing, edge computing, and high-performance computing, large-scale heterogeneous distributed clusters have become the core infrastructure for supporting artificial intelligence training, data-intensive analysis, online service deployment, and complex industrial computing tasks. These clusters typically consist of multiple types of nodes with varying computing capabilities, storage tiers, network characteristics, and energy consumption attributes[1,2]. Task loads exhibit characteristics such as strong arrival randomness, multidimensional resource requirements, large differences in execution time, and complex service constraints. In actual operation, the heterogeneity of resource supply and the dynamism of task demand are coupled, making resource allocation, task scheduling, and system collaborative control increasingly complex. Traditional resource management methods relying on static rules or local heuristics struggle to maintain efficient, stable, and globally adaptable orchestration capabilities in complex environments. Therefore, there is an urgent need to develop new resource orchestration methods with stronger environmental awareness and autonomous decision-making capabilities for dynamic scenarios[3].

In large-scale heterogeneous distributed clusters, resource orchestration not only affects computing resource utilization and task completion efficiency but also directly impacts system throughput, service quality assurance, energy consumption control, and platform operating costs. As business types expand, clusters often simultaneously host batch processing tasks, streaming tasks, latency-sensitive tasks, and high-priority service requests. These tasks create complex competition in terms of resources such as computation, communication, caching, and bandwidth[4]. When system load fluctuates rapidly, hotspot nodes frequently appear, or local resource bottlenecks accumulate, a lack of timely awareness and coordinated response to dynamic load conditions can easily lead to increased resource fragmentation, prolonged task queuing, node load imbalance, and global performance degradation. Therefore, how to achieve refined awareness of heterogeneous resources, cross-level collaborative decision-making, and adaptive orchestration under dynamic load conditions has become a key issue in distributed intelligent computing infrastructure research, possessing significant theoretical value and practical necessity.

Reinforcement learning provides a new research path for sequential decision-making problems in complex dynamic systems. It can learn resource allocation strategies through continuous interaction with the environment, demonstrating strong potential in high-dimensional state spaces and long-term reward optimization tasks. However, resource orchestration scenarios for large-scale heterogeneous distributed clusters typically feature high state dimensions, vast action spaces, tight decision-making coupling, and the coexistence of local and global objectives[5]. Monolithic decision-making frameworks often struggle to balance decision efficiency, policy generalization ability, and system scalability. In contrast, hierarchical multi-agent reinforcement learning can decompose complex resource orchestration tasks into cross-level, cross-regional, and cross-functional collaborative decision-making processes[6]. This enables agents at different levels to effectively coordinate between global control and local execution, providing a more structured modeling approach to address issues such as excessive decision complexity, coordination difficulty, and insufficient real-time response in large-scale clusters. Furthermore, introducing a dynamic load awareness mechanism can enhance the policy's sensitivity to resource state changes, task distribution drift, and local congestion, making the orchestration process more consistent with real-world system operation.

Research on hierarchical multi-agent reinforcement learning resource orchestration methods incorporating dynamic load awareness is of significant importance in promoting the intelligent, autonomous, and refined development of distributed cluster management. From a theoretical perspective, this approach helps promote the integrated development of multi-level collaborative decision-making modeling, dynamic state representation learning, and long-term benefit optimization mechanisms in complex heterogeneous environments, enriching the research system in the field of intelligent resource management[7]. From an application perspective, this method is expected to improve the resource utilization efficiency and scheduling robustness of large-scale clusters under complex task-mixed deployment conditions, enhance the system's adaptability to sudden loads, heterogeneous node differences, and multi-objective constraints, and provide methodological support for the high-quality operation of cloud platforms, computing networks, industrial internet, and new intelligent computing centers. As computing infrastructure continues to evolve towards large-scale, heterogeneous, and intelligent directions, researching hierarchical multi-agent reinforcement learning resource orchestration methods for dynamic load environments not only has significant academic frontiers but also important engineering application value.

2. Datasets And Dataset Preprocessing

2.1 Dataset

This paper selects Alibaba Cluster Trace V2018 as the research dataset. This dataset consists of large-scale cluster operation trajectories from a real production environment, originating from the Alibaba Open Cluster Trace Program, and is characterized by its public availability, realistic scenarios, and complete structure. Compared to scheduling data that typically only contains a single task type or resource dimension, this dataset simultaneously covers the mixed operation of online long-duration services and offline batch processing tasks, realistically reflecting the resource competition, task co-location, and orchestration coupling characteristics of heterogeneous distributed clusters under dynamic load fluctuations. Official documentation indicates that Cluster Trace V2018 records the operation information of approximately 4000 machines over eight consecutive days, organizing machine metadata, machine resource usage, container metadata, container resource usage, batch processing instance information, and batch processing task information into six core data tables. The batch processing tasks also include descriptions of task dependencies, providing strong data support for resource orchestration modeling for hierarchical decision-making.

2.2 Data Preprocessing

In the data preprocessing stage, the machine metadata, machine resource usage records, container metadata, container resource usage records, batch processing task information, and batch processing instance information closely related to resource orchestration in Alibaba Cluster Trace V2018 are first uniformly organized. Multi-table joins are then performed based on machine identifiers, task identifiers, instance identifiers, and timestamps to construct a unified time-series sample across the machine, service, and task layers. Subsequently, data items with severe missing values, abnormal fields, duplicate records, and those clearly not conforming to resource constraints are removed. Core resource indicators such as CPU and memory are normalized, and continuous monitoring sequences are aggregated and statistically analyzed according to fixed time windows to alleviate the problems of inconsistent sampling granularity and excessive instantaneous fluctuations in the original trajectory. Based on this, key features characterizing dynamic load changes, such as node load intensity, resource utilization, task queuing status, service deployment density, and cross-node resource competition, are further extracted. Simultaneously, a state representation is constructed by combining machine resource capacity and task resource requirements to characterize the global operating state and local load distribution in a large-scale heterogeneous distributed cluster. Finally, to adapt to the hierarchical multi-agent reinforcement learning modeling process, the preprocessed data is divided into continuous decision segments according to time sequence, and cluster-level statistical information, regional load characteristics, and node-level resource observations are organized into hierarchical inputs, thereby providing a data foundation with clear structure, consistent time sequence, and dynamic load awareness for subsequent resource orchestration strategy learning.

Figure 1 shows that data preprocessing effectively improves the stability and usability of the original cluster trajectory data. Before preprocessing, various indicators generally exhibited strong random fluctuations and local abnormal peaks, indicating that the original records contained a lot of noise interference and irregular sampling features. After preprocessing, the curves of CPU utilization, memory utilization, average batch task waiting time, and the standard deviation of CPU load between machines were significantly smoother, and the overall fluctuation amplitude was significantly reduced. Among them, the decrease in waiting time and load dispersion was particularly significant, indicating that after outlier cleaning, time window aggregation, and feature normalization, the data were improved in terms of temporal consistency, load representation ability, and statistical interpretability, thus providing a more reliable data foundation for subsequent dynamic load perception and hierarchical resource orchestration modeling.

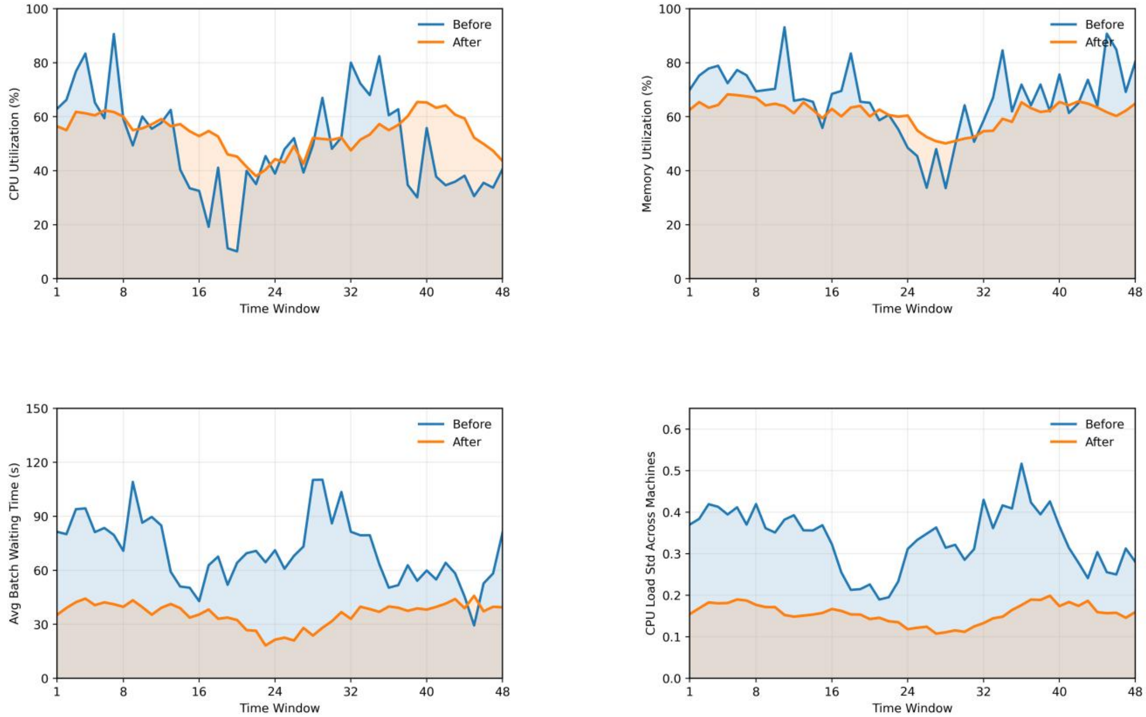

Figure 1. Comparison of data before and after preprocessing

3. Method

In large-scale heterogeneous distributed clusters, effective resource orchestration must jointly address dynamic load fluctuation, heterogeneous node capacity, task dependency, and cross-level decision coupling, and therefore, the proposed method formulates the cluster as a hierarchical interactive system in which global coordination and local execution are optimized in a unified reinforcement learning framework. Rather than treating scheduling as a flat action selection problem, the cluster is represented as a time-evolving resource graph where node states preserve machine-level observability and edge relations describe communication affinity, co-location interference, and service coupling, so that orchestration decisions can remain aware of both structural heterogeneity and temporal congestion. To preserve the global operating context before high-level planning is performed, the cluster state at time step t is defined as:

$$G_t = (\mathcal{V}_t, \mathcal{E}_t, X_t, A_t) \tag{1}$$

where $\mathcal{V}_t$ denotes the set of active machines and running entities, $\mathcal{E}_t$ captures their time-varying interactions, X_t contains heterogeneous resource observations such as CPU utilization, memory occupancy, queue length, and bandwidth pressure, and A_t is the adjacency structure induced by placement relations and communication links. Since short-term resource spikes often obscure persistent system pressure, a dynamic load perception module is introduced to extract a more stable load descriptor by jointly encoding instantaneous observations and recent historical trends, and the corresponding representation is written as:

$$h_t^L = \phi_L(X_t, X_{t-w:t-1}, M) \tag{2}$$

with $\phi_L(\cdot)$ denoting the load encoder, w representing the temporal window size, and M denoting machine capacity metadata, including processor type, memory size, accelerator availability, and network class. Such a design is important because resource orchestration in heterogeneous clusters should not react only to current utilization values, but should also distinguish transient noise from sustained imbalance, thereby enabling subsequent policies to make decisions that better reflect true system risk and latent contention. Its overall model architecture is shown in Figure 2.

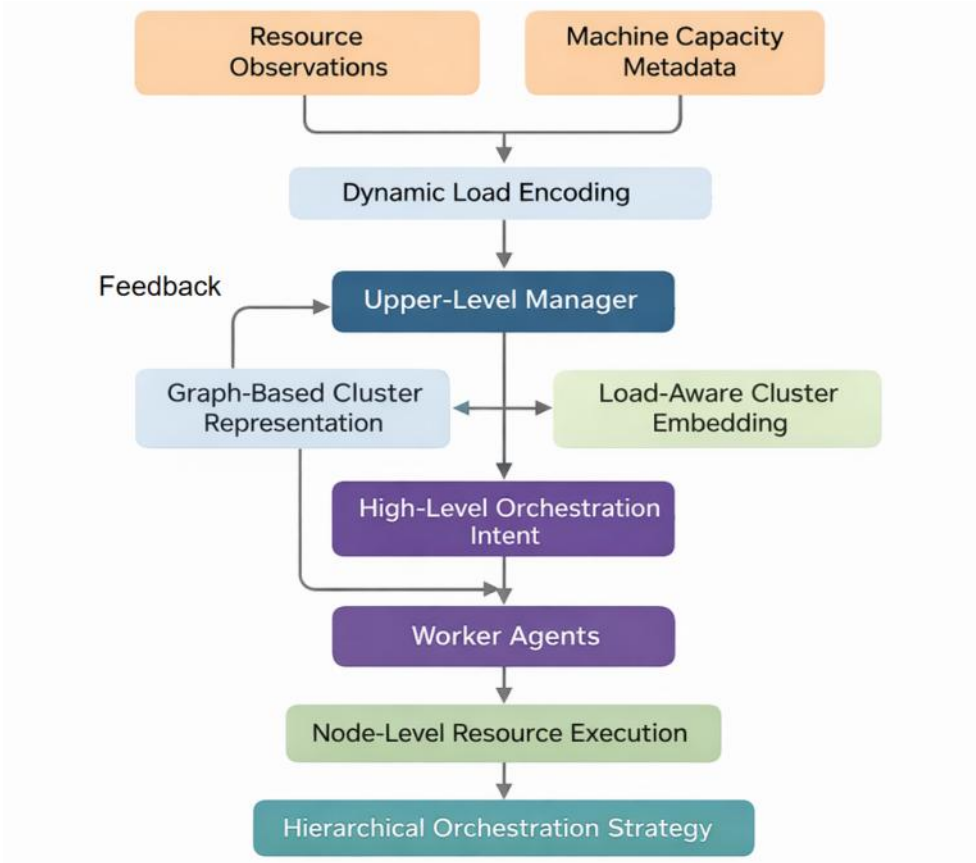

Figure 2. Overall model architecture

Building on the load-aware cluster representation, the upper-level manager is designed to generate coarse-grained orchestration intents over cluster regions, resource pools, or task groups, which reduces the effective decision horizon and improves coordination scalability under large action spaces. Instead of directly assigning every task to a concrete node, the manager first selects a high-level subgoal that specifies the preferred orchestration direction, such as relieving hot spots, consolidating fragmented resources, or prioritizing latency-sensitive workloads, and this process is modeled as:

$$g_t \sim \pi_\theta^M (g|z_t^G) \tag{3}$$

Here π_θ^M is the manager's policy and z_t^G is a global embedding derived from the graph state and the dynamic load representation. To retain sensitivity to structural imbalance across machines, the global embedding is aggregated as:

$$z_t^G = Readout(GNN(G_t, h_t^L)) \tag{4}$$

with the graph encoder preserving both topological interactions and heterogeneous resource semantics. This hierarchical decomposition has clear methodological significance because the upper layer no longer solves the entire combinatorial assignment problem in one step, while the subgoal variable g_t provides an explicit coordination signal that aligns later node-level actions with long-term cluster-wide objectives.

At the lower level, each worker agent focuses on local resource execution within a region or machine subset, thereby transforming abstract orchestration intent into fine-grained placement and allocation decisions that can respond to local queue pressure and instantaneous contention. Given the manager-issued subgoal and local observations, the worker policy produces an action that determines task admission, node selection, or resource quota adjustment according to:

$$a_t^i \sim \pi_\psi^W (a_t^i | o_t^i, g_t) \tag{5}$$

In Formula (5), o_t^i contains the local state of worker i, including machine availability, pending task demand, local interference level, and neighborhood load density. Because direct execution in heterogeneous environments must account for mismatches between task demand and node capability, a compatibility-aware scoring mechanism is further introduced to stabilize local choices, and the placement preference for task k on node j is computed by:

$$s_{k,j}^t = \frac{d_k^t c_j^t}{\| d_k^t \| \ \| c_j^t \|} - \lambda_1 \rho_j^t - \lambda_2 \eta_{k,j}^t \tag{6}$$

where d_k^t is the demand vector of task k, c_j^t denotes the residual capacity of node j, ρ_j^t measures current load pressure, and $\eta_{k,j}^t$ quantifies expected interference or migration overhead. Such a formulation is necessary because local scheduling should prefer nodes that are both resource-compatible and congestion-aware, which improves the semantic consistency between hierarchical intent and executable placement behavior.

Finally, policy learning is driven by a multi-objective reward that explicitly balances throughput, delay, load balancing, and resource efficiency, so that the resulting orchestration strategy does not overfit to a single operational criterion while ignoring system-wide service quality. From a long-horizon perspective, the reward at time step t is defined as:

$$r_t = \alpha_1 U_t - \alpha_2 D_t - \alpha_3 B_t - \alpha_4 C_t \tag{7}$$

where U_t denotes effective resource utilization, D_t is the aggregated task waiting and completion delay, B_t measures cross-node load imbalance, and C_t reflects orchestration cost such as migration, communication, or resource fragmentation. To strengthen coordination between global planning and local execution, the overall objective combines discounted return maximization with hierarchical consistency regularization and is expressed as:

$$\mathcal{L} = -\mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_t\right] + \beta \mathcal{L}_{align} \tag{8}$$

Through this design, dynamic load perception, hierarchical task decomposition, heterogeneous compatibility modeling, and coordinated reinforcement learning are integrated into a single resource orchestration framework, which allows the method to capture cluster-scale dependencies while preserving local responsiveness under complex and continuously changing distributed environments.

4. Experimental Results and Analysis

To further demonstrate the comprehensive advantages of the proposed method in large-scale heterogeneous distributed cluster resource orchestration scenarios, Table 1 selects representative related studies on reinforcement learning-driven cloud task scheduling, virtual machine placement, batch job scheduling, and multi-agent resource allocation for comparative analysis. The comparison dimensions are uniformly defined as four core indicators, namely resource utilization level, task waiting overhead, load dispersion degree, and task completion reliability, denoted by UTI, AWT, LDI, and TSR, respectively. These indicators enable a more focused reflection of the differences among various methods in terms of dynamic load awareness, global coordination capability, and local execution stability.

Table 1. Experimental results compared with other models

Method	UTI	AWT	LDI	TSR
Mao et al.[8]	72.84	43.21	0.214	91.36
Rjoub et al.[9]	75.62	38.47	0.197	92.84
Islam et al.[10]	77.15	34.92	0.185	93.41
Cheng et al.[11]	78.03	33.58	0.179	93.96
Zhang et al.[12]	79.44	31.76	0.171	94.52
Belgacem et al.[13]	80.18	30.94	0.166	95.07
Ghasemi et al.[14]	81.36	29.87	0.158	95.64
Ours	84.92	24.36	0.121	97.48

The proposed algorithm exhibits superior overall performance in resource orchestration tasks, achieving satisfactory results in resource utilization efficiency, task scheduling timeliness, load distribution balance, and overall scheduling success rate. This indicates that the constructed dynamic load awareness mechanism can more accurately characterize the resource change state in complex cluster environments, and the hierarchical multi-agent decision-making structure can effectively enhance the synergy between global coordination and local execution, thereby improving the overall orchestration quality while ensuring system stability. Related results demonstrate that the proposed method can adapt well to complex scenarios involving multidimensional resource constraints and dynamic task arrival in large-scale heterogeneous distributed clusters, possessing strong practical application value and methodological effectiveness.

To further analyze the contribution of each key component module to the overall resource orchestration performance, this paper designs targeted ablation settings based on the complete model and comprehensively examines four aspects: resource utilization efficiency, task waiting latency, load balancing degree, and scheduling success capability. Table 2 shows the performance changes under different module combinations, which can more clearly illustrate the roles of dynamic load modeling, hierarchical management mechanisms, and heterogeneous resource matching strategies in the methodology.

Table 2. Ablation test results

Ablation Setting	UTI	AWT	LDI	TSR
w/o Dynamic Load Encoding	81.47	28.91	0.149	95.86

w/o Upper-Level Manager	80.93	30.24	0.156	95.31
w/o Compatibility-Aware Scoring	82.18	27.63	0.142	96.22
Ours	84.92	24.36	0.121	97.48

Under ablation settings, the proposed method still demonstrates the crucial supporting role of each key module in overall resource orchestration performance, and the complete model achieves superior overall performance. This indicates that dynamic load coding enhances the effective perception of complex load states, the upper-level management mechanism improves global coordination capabilities, and compatibility-aware scoring helps improve the matching quality between task requirements and node capabilities. The results also show that the components of the proposed method are not simply superimposed, but rather work collaboratively in a layered manner, thereby improving resource utilization, task waiting control, load balancing, and scheduling stability, thus ensuring the effectiveness and integrity of the proposed algorithm in large-scale heterogeneous distributed cluster scenarios.

There is a significant dynamic coupling between task queuing status and resource release rhythm, and their changes can intuitively reflect the coordination capability of resource orchestration strategies under complex load environments. For hierarchical multi-agent resource orchestration methods for large-scale heterogeneous distributed clusters, focusing solely on instantaneous resource occupancy status often fails to reveal the intrinsic relationship between task backlog, resource reclamation, and subsequent scheduling responses. Therefore, it is necessary to jointly visualize task queuing intensity and resource release rhythm from the perspective of continuous temporal evolution to characterize the operational behavior of the proposed method under dynamic load perception and hierarchical collaborative control. The experimental results are shown in Figure 3.

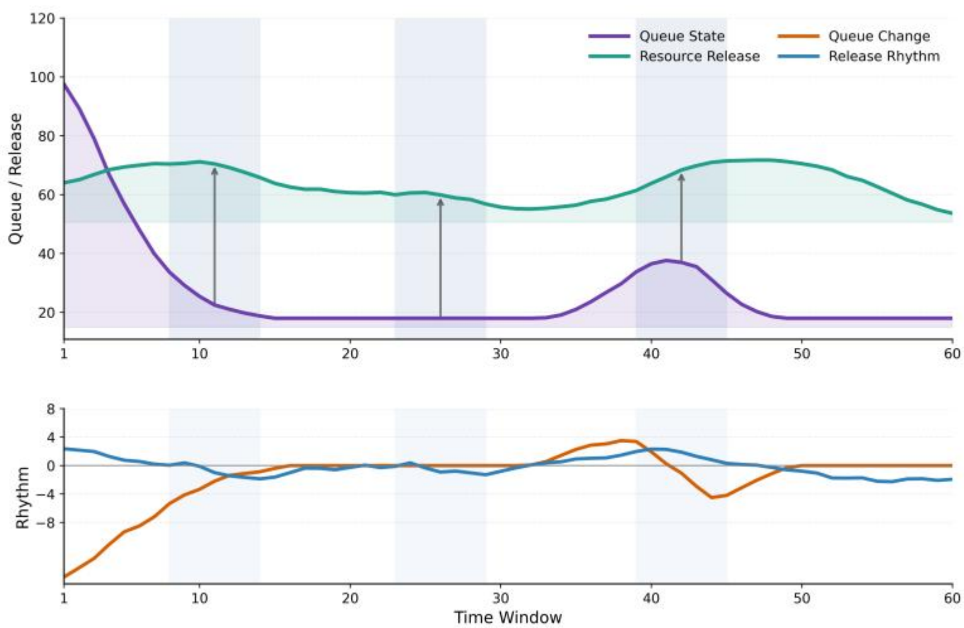

Figure 3. Visualization Experiment of the Coupling Relationship between Task Queuing Status and Resource Release Rhythm

The proposed algorithm effectively coordinates the relationship between task queuing status and resource release rhythm, making resource supply response more closely aligned with queue changes, thus maintaining relatively stable orchestration behavior under complex load environments. Changes in queuing pressure are promptly transmitted to the resource release stage, indicating a strong linkage between dynamic load perception and hierarchical collaborative control in actual operation. This also demonstrates that the constructed method effectively alleviates the problem of task backlog and resource response mismatch. Overall, the visualization results further illustrate that the proposed method achieves high-quality resource adjustment in large-scale heterogeneous distributed clusters, balancing local execution efficiency with global operational stability.

5. Conclusion

This paper addresses the resource orchestration problem in large-scale heterogeneous distributed clusters, proposing a hierarchical multi-agent reinforcement learning method that integrates dynamic load awareness. This aims to solve the problems of insufficient global coordination, lagging local response, and limited heterogeneous resource adaptability inherent in traditional resource management strategies in complex dynamic environments. Considering the realities of cluster operation, such as the high randomness of task arrival, multidimensional coupling of resource demands, significant differences in node capabilities, and continuous evolution of system states, a relatively complete methodological framework is constructed, encompassing dynamic state modeling, hierarchical decision-making organization, and task-resource matching optimization. By organically combining a dynamic load encoding mechanism, upper-level management decisions, and lower-level execution control, this method enhances the resource orchestration process's ability to perceive and coordinate complex operational states, providing a structured and scalable research approach for efficient resource scheduling in large-scale intelligent computing infrastructures.

From a methodological perspective, the significance of this work lies not only in introducing reinforcement learning to solve the resource orchestration problem but also in unifying the hierarchical modeling concept with the dynamic load awareness process in heterogeneous environments. This allows resource orchestration to move beyond single-layer static allocation and achieve adaptive adjustments to local execution under global objective constraints. This design effectively enhances the method's adaptability to complex scenarios involving mixed task deployment, resource contention, and localized congestion. It also fosters a more coordinated optimization relationship between resource utilization, task waiting control, load balancing, and system stability. Related research indicates that for resource management in heterogeneous distributed systems, only by simultaneously considering cross-level collaboration, dynamic state representation, and task node compatibility can the practicality and robustness of resource orchestration strategies be truly improved in real-world environments. This paper systematically explores this direction.

From an application perspective, the proposed method has strong applicability to scenarios such as cloud computing platforms, intelligent data centers, edge cloud collaborative systems, computing power networks, and high-performance computing resource pools. With the continuous growth of AI training tasks, the deep integration of online and offline operations, and the increasing number of heterogeneous computing devices, resource orchestration has become a critical factor affecting platform throughput, service quality assurance, and operational cost control. The proposed method provides a more autonomous resource organization mechanism for new intelligent computing infrastructures, helping to improve computing power utilization efficiency and scheduling stability in complex business environments, and providing methodological support for achieving future-oriented elastic computing, green computing, and intelligent operation and maintenance. In a broader industry context, this research has significant practical implications for driving the transformation of distributed cluster management from experience-driven to intelligence-driven approaches.

Future research can be further expanded in several directions. First, more constraints specific to real-world systems can be incorporated into the unified modeling framework, such as energy consumption constraints, network latency constraints, task priority constraints, and service-level protocol constraints, thereby enhancing the method's adaptability to real-world production environments. Second, resource orchestration issues in cross-cluster, cross-regional, and even cloud-edge-device collaborative scenarios can be further considered to adapt to more complex distributed computing power organization forms. Third, online learning, adaptive migration, and continuous optimization mechanisms can be combined to improve the model's self-evolution capabilities and cross-scenario generalization capabilities during long-term operation. Fourth, deep integration with digital twin platforms, intelligent monitoring systems, and automated operation and maintenance frameworks can be explored to promote the transition of resource orchestration methods from theoretical research to engineering implementation. As intelligent computing infrastructure develops towards larger scale, higher heterogeneity, and stronger autonomy, resource orchestration methods that integrate dynamic load awareness and hierarchical multi-agent collaborative decision-making are expected to play an increasingly important role in future related application areas.

References

[1] T. Sukprasert, A. Souza, N. Bashir, D. Irwin and P. Shenoy, "On the Limitations of Carbon-Aware Temporal and Spatial Workload Shifting in the Cloud," Proceedings of the Nineteenth European Conference on Computer Systems, pp. 92-941, 2024.

[2] J. Cai, W. Liu, Z. Huang, et al., "Task decomposition and hierarchical scheduling for collaborative cloud-edge-end computing," IEEE Transactions on Services Computing, vol. 17, no. 6, pp. 4368-4382, 2024.

[3] L. Cheng, Y. Wang, F. Cheng, C. Liu, Z. Zhao, and Y. Wang, “A deep reinforcement learning-based preemptive approach for cost-aware cloud job scheduling,” IEEE Transactions on Sustainable Computing, vol. 9, no. 3, pp. 422-432, 2023.

[4] D. Qi, X. Xi, Y. Tang, Y. Zheng, and Z. Guo, “Real-time scheduling of power grid digital twin tasks in cloud via deep reinforcement learning,” Journal of Cloud Computing, vol. 13, no. 1, p. 121, 2024.

[5] J. Wang, S. Li, X. Zhang, F. Wu, and C. Xie, “Deep reinforcement learning task scheduling method based on server real-time performance,” PeerJ Computer Science, vol. 10, p. e2120, 2024.

[6] M. Mounesan, M. Lemus, H. Yeddulapalli, P. Calyam, and S. Debroy, “Reinforcement learning-driven data-intensive workflow scheduling for volunteer edge-cloud,” in Proc. IEEE 8th Int. Conf. Fog and Edge Computing (ICFEC), May 2024, pp. 79-88.

[7] E. Russo, F. G. Blanco, M. Palesi, G. Ascia, D. Patti, and V. Catania, “Towards fair and firm real-time scheduling in DNN multi-tenant multi-accelerator systems via reinforcement learning,” in Proc. IEEE Int. Symp. Circuits and Systems (ISCAS), May 2024, pp. 1-5.

[8] H. Liu, "Structural Regularization and Bias Mitigation in Low-Rank Fine-Tuning of LLMs," 2023.

[9] C. Hua, "A Semantic-Prior-Guided AI Framework for Collaborative Environment Understanding and Robust Agent Decision Making," 2024.

[10]M. T. Islam, S. Karunasekera, and R. Buyya, "Performance and cost-efficient spark job scheduling based on deep reinforcement learning in cloud computing environments," IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 7, pp. 1695-1710, 2021.

[11]F. Cheng, Y. Huang, B. Tanpure, et al., "Cost-aware job scheduling for cloud instances using deep reinforcement learning," Cluster Computing, vol. 25, no. 1, pp. 619-631, 2022.

[12]Y. Deng, "Transfer Methods for Large Language Models in Low-Resource Text Generation Tasks," 2024.

[13]Q. Gan, "Large Language Model Framework for Multi-Document Financial Anomaly Detection in Intelligent Auditing via Semantic Mapping and Risk Reasoning," 2024.

[14]Y. Li, "Task-aware Differential Privacy and Modular Structural Perturbation for Secure Fine-Tuning of Large Language Models," 2024.