Artificial Intelligence and Computing Innovations
Intelligence Science Publishing
https://www.ispublishing.com
ISSN: 3070-4944
Vo.4, No.1, 2024

Reinforcement Learning-Based Intelligent Decision Modeling for Large-Scale Distributed Systems

Ruoxuan Wei[1]*, Leo Tan[2]

[1]Trine University, Virginia Reston, USA

[2]University of Nebraska–Lincoln, Lincoln, USA

**Corresponding author: ruoxuanwei.roxanne@gmail.com*

Abstract: To address challenges in large-scale distributed systems, including high-dimensional state spaces, dynamic operating environments, and strongly coupled multi-node decisions, this study proposes a reinforcement learning based approach for intelligent decision modeling that provides a unified sequential formulation of the scheduling process. System states are abstracted into structured representations, while policy networks and value evaluation mechanisms jointly model the long-term impact of scheduling actions on system objectives. This enables integrated optimization of task completion efficiency, resource utilization, and load distribution. The method explicitly accounts for parallel operation across multiple nodes and the requirement of global objective consistency, allowing stable and effective scheduling behavior under complex dependencies. Comparative evaluations conducted in a public distributed system data environment demonstrate clear advantages across multiple system-level metrics, confirming the effectiveness and practical value of reinforcement learning driven intelligent decision modeling for improving the overall operational quality of large-scale distributed systems.

Keywords: Distributed systems; reinforcement learning; intelligent scheduling; resource management

1. Introduction

With the rapid development of cloud computing, edge computing, and next-generation networking technologies, large-scale distributed systems have become core infrastructures supporting modern information societies. Such systems are widely deployed in data center scheduling, distributed storage, microservice architectures, intelligent network control, and complex business process management. They are characterized by large node populations, heterogeneous structures, high-dimensional system states, and highly dynamic operating environments. Components within these systems cooperate through complex dependency relations [1]. Overall performance and stability depend not only on local node behavior but also on global resource allocation, temporal decision making, and interaction strategies. Under these conditions, achieving efficient, stable, and adaptive decision-making in uncertain environments has become a critical challenge for distributed system intelligence [2].

Conventional decision mechanisms in distributed systems mainly rely on manual rules, static strategies, or optimization models built on simplified assumptions. These approaches can be effective when the system scale is limited and operational patterns are relatively stable. In large-scale scenarios with frequent workload fluctuations, highly uneven resource demands, and unpredictable events, rule-based methods struggle to balance efficiency and robustness. Rule design requires substantial domain expertise and incurs high maintenance costs, while offering limited flexibility. Static or semi-static models fail to capture the evolving state space and long-term effects of decisions. This often leads to local optima or even system performance degradation. As a result, intelligent approaches that can learn decision patterns from system dynamics and continuously adapt to environmental changes are of significant theoretical and practical value [3].

Reinforcement learning, as a decision-oriented modeling paradigm, provides a promising direction for addressing these challenges. It enables agents to learn state-action mappings through continuous interaction with the environment. Learning is guided by feedback signals and focuses on long-term return accumulation and global objective optimization. These properties align well with the fundamental requirements of resource scheduling, task allocation, and control policies in distributed systems. Compared with model-dependent approaches, reinforcement learning shows stronger adaptability in complex and dynamic environments. It can gradually learn effective policies without complete prior knowledge, thus enabling data-driven and self-evolving decision capabilities in distributed systems [4, 5].

Despite its potential, directly applying reinforcement learning to large-scale distributed systems remains difficult. The state space often grows exponentially, and interactions among nodes introduce strong non-independence and temporal coupling in decision making [6]. In addition, system operation imposes strict requirements on stability and safety, where improper actions may trigger cascading failures. These factors pose serious challenges to modeling efficiency, convergence stability, and scalability in conventional reinforcement learning frameworks. Targeted studies that consider distributed structures, hierarchical decision mechanisms, and agent cooperation are therefore essential for improving practical applicability.

From a broader perspective, intelligent decision modeling for large-scale distributed systems represents both a key direction for artificial intelligence deployment and a foundation for enhancing the efficiency and resilience of digital infrastructures [7]. By incorporating reinforcement learning principles, systems can shift from passive response to proactive regulation, and from local optimization to global coordination. This transition can improve resource utilization, reduce operational costs, and strengthen adaptability to uncertainty. Continued research in this area will provide important theoretical foundations and methodological support for next-generation distributed systems and will contribute to the sustainable development of cloud services, intelligent networks, and complex information systems.

2. Proposed Framework

2.1 Problem Modeling and Overall Framework

Intelligent decision-making problems in large-scale distributed systems can be uniformly modeled as a sequential decision-making process, in which the system evolves continuously within consecutive time steps and needs to select appropriate control actions based on the current operating state. This paper formalizes this process as a Markov decision process, whose core elements include the system state space, action space, state transition mechanism, and long-term optimization objective. Let the global state of the system at time step t be represented by a vector $s_t \in S$, which comprehensively characterizes the load, resource consumption, and key operating indicators of each node. The intelligent decision-making module generates action $\alpha_t \in A$ based on the current state and applies it to the system environment, causing it to transition to the next state s_{t+1}. This transition relationship can be expressed as:

$$s_{t+1}=f(s_t,a_t) \tag{1}$$

Here, $f(\cdot)$ represents the implicit dynamic evolution function of the system. To characterize the long-term impact of decisions, an immediate reward function $r_t=R(s_t,a_t)$ is introduced to measure the contribution of the current decision to the system's operational goals. The overall modeling framework aims to optimize the long-term cumulative reward, laying a unified foundation for subsequent reinforcement learning strategy design. This paper also presents the overall model architecture, as shown in Figure 1.

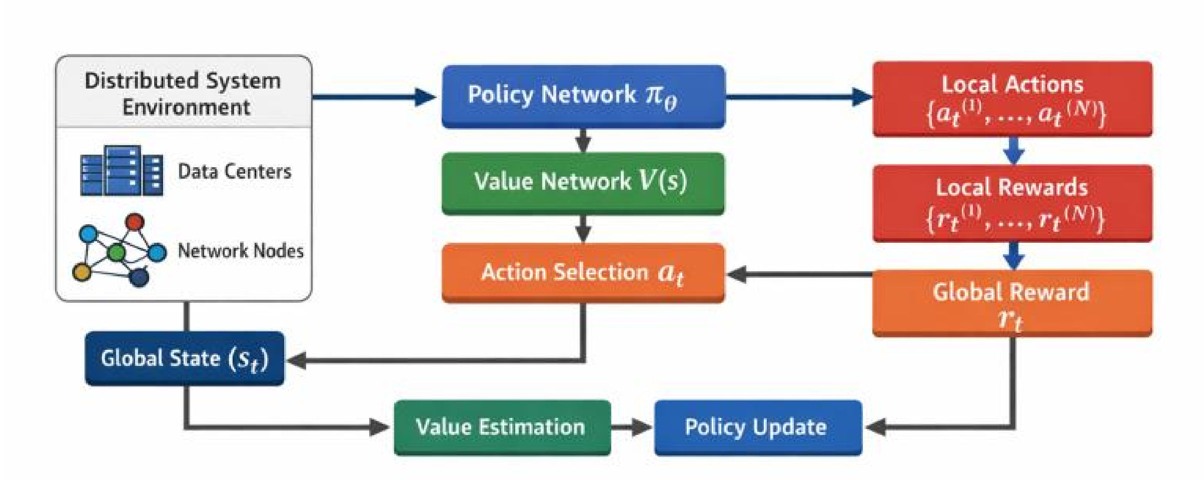

Figure 1. Overall model architecture

2.2 Reinforcement Learning Decision Strategy Modeling

Building upon the aforementioned modeling, reinforcement learning achieves the mapping from states to actions through a policy function. This paper employs a parameterized policy representation, where $\pi_\theta(a|s)$ describes the probability distribution of choosing action a in state s, and θ represents learnable parameters. The optimization objective of the policy is to maximize the expected cumulative reward, defined as follows:

$$J(\theta)=E_{\pi_\theta}[\sum_{t=0}^{T} r^t r_t] \tag{2}$$

Where $r\in(0,1)$ is the discount factor, used to balance immediate and long-term returns. To evaluate the decision quality of the current strategy, a state-value function is introduced:

$$V^\pi(s)=E_\pi[\sum_{t=0}^{T} r^t r_t |s_0=s] \tag{3}$$

This function characterizes the expected long-term return that the system can obtain under policy π starting from a given state, providing a theoretical basis for policy improvement.

2.3 Value Function and Policy Update Mechanism

To more precisely measure the impact of specific actions on system evolution, we further define the action value function:

$$Q^\pi(s,a)=E_\pi[r_t+\gamma V^\pi(s_{t+1})|s_t=s,a_t=a] \tag{4}$$

This function directly reflects the long-term effect of acting a in state s. During policy optimization, policy parameters are adjusted through gradient updates, and its basic form can be written as:

$$\theta\leftarrow\theta+\alpha\nabla_\theta J(\theta) \tag{5}$$

Where α is the learning rate. To reduce estimation variance and improve learning stability, an advantage function is introduced:

$$A^\pi(s,a)=Q^\pi(s,a)-V^\pi(s) \tag{6}$$

This function measures the relative merits of an action compared to the average level of decision-making, thereby guiding the strategy to be more inclined to choose the behavior that is more beneficial to the overall goal of the system.

2.4 Collaborative Decision-Making Modeling in Distributed Systems

To address the characteristics of parallel decision-making across multiple nodes in large-scale distributed systems, this paper further decomposes the global decision-making process into collaborative modeling of multiple local decision-making units. Let the system consist of N subsystems, with the local state of the i-th subsystem represented by $s_t^{(i)}$ and its corresponding action by $a_t^{(i)}$. The global state and action can be represented as follows:

$$s_t=\{s_t^{(1)},\ldots,s_t^{(N)}\},\quad a_t=\{a_t^{(1)},\ldots,a_t^{(N)}\} \tag{7}$$

Based on this, the global return is modeled as a weighted combination of the local returns:

$$r_t=\sum_{i=1}^{N} w_i r_t^{(i)} \tag{8}$$

Here, w_i represents the importance weight of different subsystems in the overall objective. This collaborative modeling approach, while maintaining flexibility in local decision-making, introduces global consistency constraints, providing a unified mathematical framework for stable decision-making and coordinated control of reinforcement learning in distributed systems.

3. Experimental Analysis

3.1 Dataset

This study adopts the open source production cluster trace released by the Alibaba Cluster Trace Program as the experimental data foundation. The specific version used is cluster trace v2018. The dataset is collected from real production clusters in Internet data centers. It reflects resource supply and demand relationships, workload arrivals, and scheduling execution in large-scale distributed systems under practical operation. This makes it suitable for research on intelligent decision modeling in resource scheduling, task allocation, and load management.

The cluster trace v2018 dataset is organized in multiple files and tables. It covers two core aspects, namely machine-side resource supply and task-side resource requests. On the one hand, it provides resource capacity and operational status at the machine or node level. On the other hand, it records job and task submission, scheduling, and execution evolution, together with time-varying resource usage statistics. This structure naturally aligns with the design of states, actions, and rewards in reinforcement learning. States can be defined by node load and resource utilization. Actions can correspond to scheduling, migration, or rate limiting decisions. Rewards can be abstracted from metrics such as throughput, latency, violation risk, or resource waste. As a result, the dataset supports sequential decision modeling.

In terms of scale and duration, this trace version covers approximately eight days of continuous operation. It involves around four thousand servers. It includes both online application containers and offline batch computing tasks. The data captures heterogeneous workload coexistence, resource contention, and temporal fluctuations that are typical in large-scale distributed systems. Compared with small-scale synthetic datasets, such production-level traces are more suitable for validating decision effectiveness and engineering transferability in complex dynamic environments. They also provide reliable data support for building reinforcement learning environments and policy learning objectives that are closer to real systems.

3.2 Experimental Results

This article first reports the outcomes obtained from the comparative experiments, which are summarized in Table 1, in order to provide an initial and direct basis for evaluating the proposed approach under a consistent experimental protocol. Specifically, the methods considered for comparison are organized within a unified tabular format so that the key performance indicators can be inspected in an aligned manner, facilitating a clear and structured presentation of the experimental setup and measured metrics. By presenting these comparative results at the beginning of the experimental analysis, the paper establishes a coherent starting point for subsequent discussion, ensuring that the relative behavior of different methods can be examined on the same footing with respect to the selected evaluation criteria and reporting conventions.

Table 1. Comparative experimental results

Method	JCT	Resource Utilization	Makespan	Hotspot Ratio
Rlhf deciphered [8]	1.00	0.71	1.00	0.085
Gymnasium [9]	0.94	0.73	0.96	0.078
Dapo [10]	0.89	0.75	0.92	0.072
Ttrl [11]	0.87	0.76	0.90	0.069
Gepa [12]	0.84	0.78	0.88	0.065
Ours	0.78	0.82	0.83	0.057

Overall performance trends show consistent improvement across multiple system-level metrics for different methods. This indicates that learning based decision mechanisms can effectively mitigate performance bottlenecks caused by static strategies in traditional distributed systems. As the ability to model system states and long-term feedback improves, task completion efficiency, resource utilization, and operational stability are enhanced simultaneously. These results highlight the overall advantage of reinforcement learning in complex scheduling decisions.

For metrics related to task completion time and global execution efficiency, the proposed approach demonstrates more pronounced benefits. This suggests improved coordination of resource allocation and task execution order across nodes. Unnecessary waiting and conflicts are reduced. A more efficient system-wide scheduling rhythm is achieved. The improvement does not rely on fine-tuning local rules. It results from holistic modeling of long-term decision impacts. This aligns with the need for cross-temporal and cross-node coordination in large-scale distributed systems.

From the perspective of resource utilization and load distribution, baseline methods increase resource usage but still exhibit load concentration to some extent. The proposed method achieves a better balance between utilization and distribution. This indicates that the strategy considers not only whether resources are used but also actively avoids the persistent accumulation of local hotspots during decision making. Load distribution becomes more even across the system. This property is especially important for distributed systems with large node populations and strong resource heterogeneity. It helps reduce performance fluctuations and potential cascading risks.

Considering all evaluation metrics together, the proposed method shows stronger system-level decision capability in terms of efficiency, stability, and resource coordination. These findings suggest that unified modeling of distributed system dynamics through reinforcement learning can effectively capture complex temporal evolution. It can also produce more globally consistent policies under multiple constraints. The results confirm the practical value of intelligent decision modeling for improving the overall operational quality of large-scale distributed systems.

The discount factor is used to balance the relative importance of immediate and long-term returns in policy learning, and its value changes the agent's focus on future rewards. For decision modeling of large-scale distributed systems, changes in the discount factor often affect how the policy responds to dynamic processes such as load fluctuations, resource competition, and hotspot formation. The experimental results are shown in Figure 2.

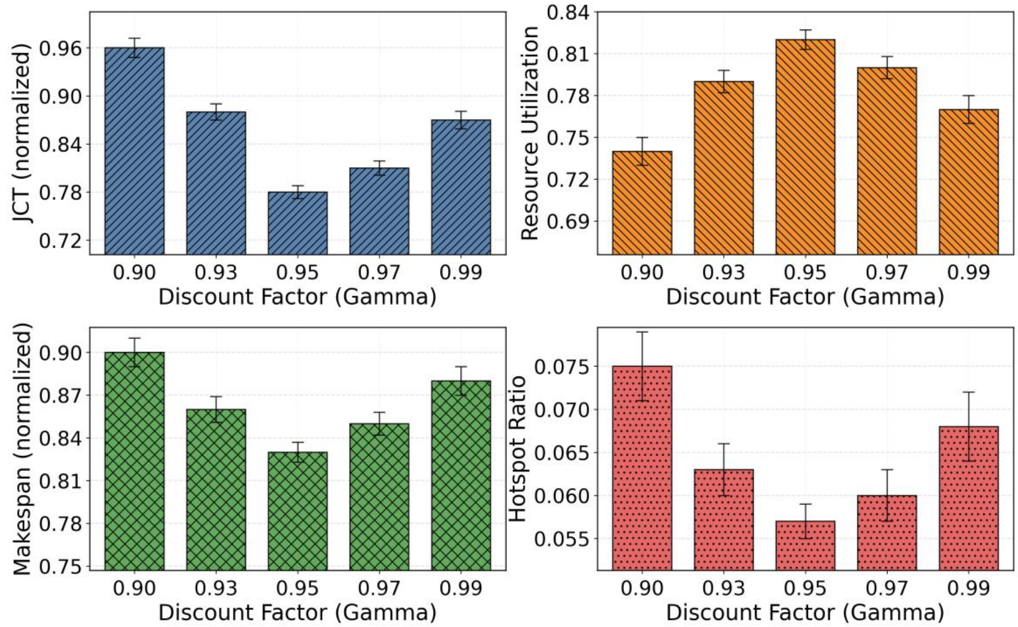

Figure 2. The effect of the discount factor on experimental results

Observed from the overall variation patterns, the discount factor has a significant influence on decision behavior in distributed systems. Under different discount settings, system-level metrics exhibit clear nonlinear responses. This indicates that the trade-off between immediate rewards and long-term returns directly shapes how scheduling decisions attend to system evolution. The results suggest that proper modeling of long-term effects is essential for achieving stable and efficient decision-making in large-scale distributed environments.

For metrics related to task completion efficiency, changes in the discount factor guide the policy to adjust between short-term reactions and long-term planning. When the discount factor takes a moderate value, the policy better accounts for task dependencies and persistent resource occupation. A more coordinated scheduling rhythm is formed. Overemphasis on immediate rewards or excessive focus on distant returns both weaken control over overall system efficiency. This observation shows that reinforcement learning for distributed scheduling must consider multi-time scale characteristics to realize its advantages.

From the perspective of resource utilization and system load, the discount factor directly shapes the global view of the policy. Appropriate discount settings encourage the model to avoid excessive local resource concentration during decision making. Resource allocation becomes more balanced. In contrast, improper discount choices tend to drive load skew under short-term reward incentives. This demonstrates the importance of long-term return modeling for suppressing hotspot formation and maintaining system stability.

Considering all metrics together, the discount factor is not merely a training hyperparameter but a fundamental determinant of policy preference and behavioral inclination in sequential decision making. By controlling how future returns are weighted relative to immediate feedback, it effectively shapes what the policy treats as important over time, thereby governing whether decisions emphasize short-term responsiveness or longer-horizon coordination. In large-scale distributed systems where actions can have delayed and cascading effects, this temporal weighting directly affects the depth of perception of long-term system states, including how the policy internalizes evolving load patterns, resource contention, and dependency-driven dynamics. At the same time, the discount factor influences the consistency of decisions by modulating how strongly the policy commits to stable long-horizon objectives versus adapting to transient fluctuations, which in turn impacts the reliability of the learned control behavior under uncertainty.

As a consequence, the discount factor further influences overall performance in complex and dynamic distributed environments, not as a secondary tuning knob but as a central element that governs the policy's temporal priorities. When the environment exhibits multi-timescale variation, such as bursty arrivals, heterogeneous resources, and shifting bottlenecks, the discount factor determines how effectively the policy can integrate near-term signals with longer-term operational goals, maintaining coherent scheduling intent across time. These observations indirectly confirm that careful design of temporal trade-off mechanisms is a key element for improving scheduling quality and operational reliability in large-scale distributed systems, because such mechanisms dictate how the policy balances immediate operational pressures against the sustained system-level objectives that ultimately define robust and dependable scheduling behavior.

Entropy regularization coefficients are used to adjust the strength of the trade-off between exploration and convergence in a policy, thereby affecting the diversity of decision-making behavior in uncertain distributed environments. For intelligent scheduling of large-scale distributed systems, appropriate entropy constraints help prevent policies from falling into local patterns prematurely and improve adaptability to load fluctuations and resource contention. The experimental results are shown in Figure 3.

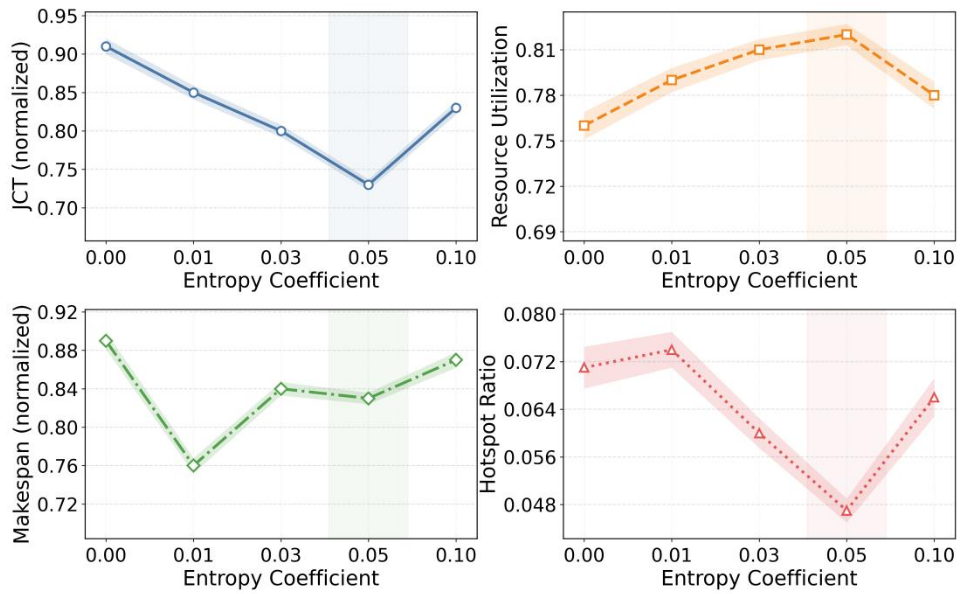

Figure 3. The effect of the entropy regularity coefficient on experimental results

The overall variation patterns indicate that the entropy regularization coefficient plays a significant role in regulating the decision behavior of reinforcement learning policies in distributed systems. As the strength of entropy constraints changes, the model continuously rebalances exploration diversity and policy stability. The decision style is adjusted accordingly. This suggests that entropy regularization affects not only the convergence behavior during training, but also the operational characteristics of the policy in complex system environments.

For metrics related to task completion efficiency, changes in the entropy coefficient reveal the direct influence of exploration mechanisms on scheduling behavior. When entropy constraints are weak, the policy tends to converge quickly to deterministic decisions. Its ability to adapt to system state changes is limited. Moderate increases in entropy constraints encourage more thorough evaluation of alternative decision paths. This leads to more flexible task assignment and execution ordering. Such behavior is consistent with the highly dynamic loads and complex decision spaces in large-scale distributed systems.

From the perspective of resource utilization and load structure, entropy regularization contributes to mitigating excessive resource concentration. Proper entropy constraints guide the policy to retain a certain level of randomness while pursuing efficiency. Resource scheduling becomes more dispersed. Risks of local congestion and hotspot accumulation are reduced. This observation shows that introducing controlled uncertainty into distributed systems can improve resource coordination and operational stability.

Considering all metrics together, the entropy regularization coefficient is a critical control factor shaping the global behavior of reinforcement learning policies. Its value directly determines whether the policy can balance exploration and decision consistency in complex dynamic environments. It has a long-term impact on system performance. The experiment further confirms that well-designed entropy constraints can effectively enhance the adaptability and robustness of intelligent decision models in large-scale distributed systems.

The length of the state observation window determines the range of historical context that the policy can utilize when making scheduling decisions, thus affecting its ability to characterize short-term fluctuations and long-term trends. For large-scale distributed systems, a window that is too short may miss key temporal dependencies, while a window that is too long may introduce redundant noise and increase decision uncertainty. The experimental results are shown in Figure 4.

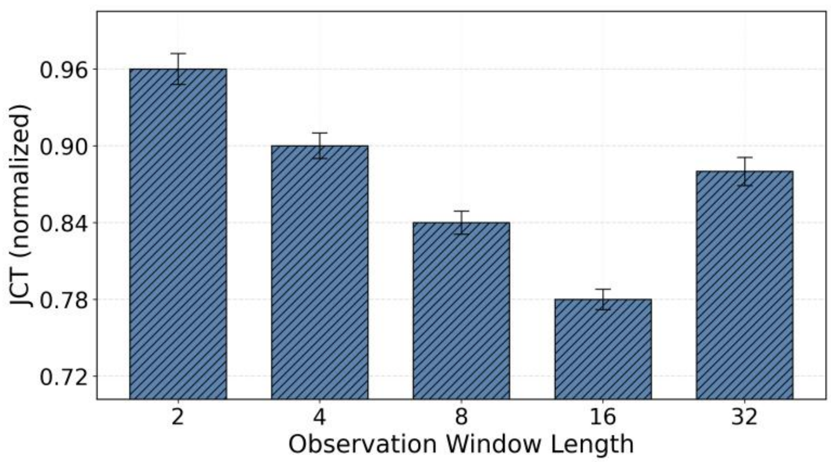

Figure 4. Sensitivity experiment of the state observation window length to JCT

The overall trends show that the length of the state observation window has a clear impact on the decision quality of reinforcement learning based scheduling policies. Performance differences across window lengths indicate that how historical state information is utilized directly affects the depth of system dynamic understanding. This further influences scheduling effectiveness in complex distributed environments. The results demonstrate that temporal information modeling is a critical factor for intelligent decision-making in large-scale distributed systems.

When the observation window is short, the policy mainly relies on limited recent information for decision-making. Long-term evolution of system load and resource contention is insufficiently captured. In this case, the model is more sensitive to transient fluctuations and struggles to form stable and consistent scheduling strategies. This reflects limited capability in modeling cross-temporal dependencies. For distributed systems with strong temporal correlations, such restricted state representations are inadequate for high-quality global decisions.

As the observation window length increases, the model can integrate richer historical context. It gains a more comprehensive understanding of task arrival patterns and resource usage trends. Stronger global coordination emerges during scheduling. This indicates that moderate historical information helps reinforcement learning policies balance short-term responsiveness and long-term planning. Such behavior aligns with the coexistence of multiple time scales in distributed system operations.

However, further increasing the observation window leads to a certain performance decline. This suggests that overly long historical sequences may introduce redundancy or noise. Decision uncertainty and complexity increase. The results indicate that longer observation windows are not always better. The window length should match system dynamics and decision complexity. This experiment confirms that well-designed state representations are essential for improving stability and efficiency of reinforcement learning based scheduling in large-scale distributed systems.

The load burst intensity is used to characterize the concentration of request arrivals and resource demands within a short period of time. Its changes can alter the instantaneous congestion pattern of a distributed system and the timing pressure faced by scheduling decisions. The experimental results are shown in Figure 5.

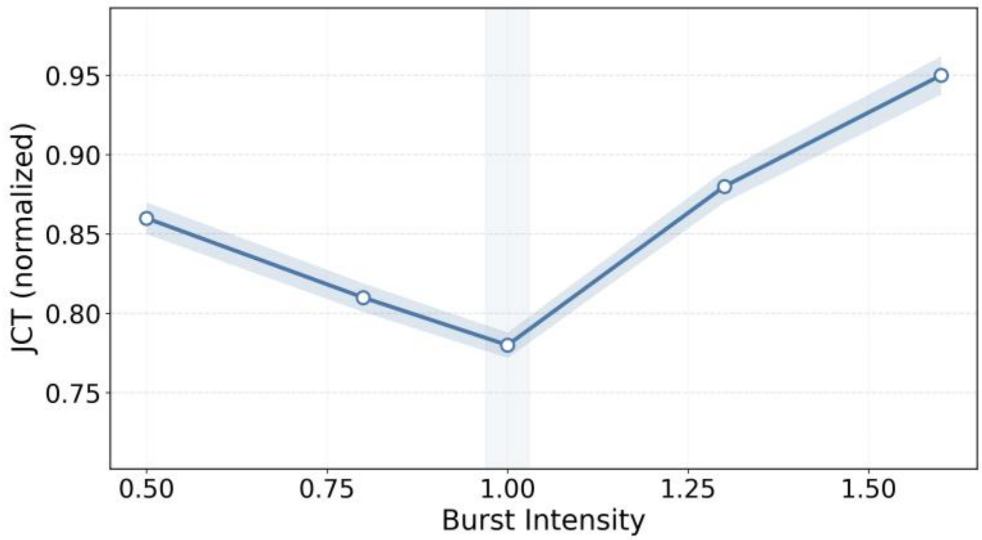

Figure 5. Environmental sensitivity test of JCT to sudden load intensity changes

The overall patterns indicate that changes in burst intensity substantially alter the operating environment for scheduling decisions, and thus directly affect task completion efficiency. As the concentration of requests within a short time period varies, the scheduling policy must respond under higher uncertainty and stronger timeliness pressure. This imposes stricter requirements on the environmental adaptability of reinforcement learning models. It also highlights the necessity of environmental sensitivity analysis in distributed system studies.

Under low burst intensity, system load is relatively smooth, and contention pressure is limited. The policy can coordinate resource allocation and task execution ordering in a more manageable manner. This stage suggests that the model can effectively use current state information in normal operating conditions. It can maintain stable scheduling behavior and provide a basis for handling more complex environments.

When burst intensity approaches the typical operating load level, scheduling decisions must address the trade-off between short-term congestion and long-term efficiency. Accurate characterization of system dynamics becomes particularly important. The experimental trends indicate that the model has a strong capability to perceive and adapt to environmental changes under this condition. It can adjust decisions promptly when burst loads occur and prevent excessive performance fluctuations. This reflects the adaptive properties of reinforcement learning in complex distributed scenarios.

As burst intensity further increases, the system enters a high-pressure operating state, and resource contention becomes substantially stronger. The scheduling policy faces more stringent challenges. Under such conditions, changes in task completion efficiency reflect the model's responsiveness to extreme load levels. They also indirectly reveal the performance boundary of distributed systems under high-intensity bursts. The experiment suggests that the proposed intelligent decision model can maintain reasonable scheduling behavior across different burst intensity levels. This provides strong support for its deployment in real and complex environments.

4. Conclusion

This work addresses core challenges in large-scale distributed systems, including complex decision processes, highly dynamic environments, and difficulties in achieving global coordination. A reinforcement learning based intelligent decision modeling framework is developed to provide a unified representation of system states, scheduling actions, and long-term objectives. By abstracting system operation as a sequential decision process, the model learns globally consistent scheduling policies in environments with multiple nodes and concurrent tasks. This approach overcomes limitations of rule-driven and static optimization methods in adaptability and scalability. The study demonstrates that data-driven decision modeling offers a feasible and general technical pathway for intelligent management of complex systems.

From an application perspective, the proposed modeling framework provides valuable insights for resource scheduling in cloud platforms, data center operations, and performance optimization of large-scale service systems. By introducing decision mechanisms that perceive system dynamics and account for long-term operational goals, distributed systems can achieve more stable and efficient performance under high load and complex workloads. Coordinating local behaviors from a global perspective improves resource utilization and service quality. It also provides theoretical support for building adaptive intelligent infrastructures.

At the methodological level, this research emphasizes integrating system structural characteristics with the decision process. This enables reinforcement learning policies to operate effectively in environments with complex dependencies and multiple time scales. The generality of the framework allows strong transferability across different distributed decision scenarios. These include computing resource management, network scheduling, and service orchestration. A unified modeling perspective enables the analysis and solution of diverse decision problems within a single theoretical framework. This reduces the complexity of system design and optimization.

Looking ahead, as distributed systems continue to grow in scale, become more open, and increase in heterogeneity, intelligent decision models will play an increasingly critical role in system management and optimization. Further integration of system structure, operational semantics, and cross-layer coordination mechanisms may drive distributed systems toward proactive control and self-evolution. This study lays a foundation for this direction. It also provides useful references for the broad application of intelligent distributed systems in next-generation digital infrastructures.

References

[1] S. S. Mondal, N. Sheoran and S. Mitra, "Scheduling of Time-Varying Workloads Using Reinforcement Learning," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 10, pp. 9000-9008, 2021.

[2] H. Qiu, W. Mao, C. Wang et al., "AWARE: Automate Workload Autoscaling with Reinforcement Learning in Production Cloud Systems," Proceedings of the 2023 USENIX Annual Technical Conference (USENIX ATC), pp. 387-402, 2023.

[3] J. Pan and Y. Wei, "A Deep Reinforcement Learning-Based Scheduling Framework for Real-Time Workflows in the Cloud Environment," Expert Systems with Applications, vol. 255, p. 124845, 2024.

[4] A. Jayanetti, S. Halgamuge and R. Buyya, "Multi-Agent Deep Reinforcement Learning Framework for Renewable Energy-Aware Workflow Scheduling on Distributed Cloud Data Centers," IEEE Transactions on Parallel and Distributed Systems, vol. 35, no. 4, pp. 604-615, 2024.

[5] Z. Li, J. Yu, X. Liu et al., "Load Balancing for Task Scheduling Based on Multi-Agent Reinforcement Learning in Cloud-Edge-End Collaborative Environments," Proceedings of the 2024 8th International Conference on Machine Learning and Soft Computing, pp. 94-100, 2024.

[6] Y. Tang, "Research on Hierarchical Multi-Agent Reinforcement Learning Resource Orchestration for Large-Scale Heterogeneous Distributed Clusters with Dynamic Load Awareness," 2024.

[7] C. Zhang, "Adaptive Multi-Tenant Resource Scheduling in Cloud Computing via Reinforcement Learning," 2024.

[8] A. Verma, G. Das, T. K. Nayak et al., "Server Workload Analysis for Power Minimization Using Consolidation," Proceedings of the 2009 USENIX Annual Technical Conference, pp. 28-28, 2009.

[9] M. Towers, A. Kwiatkowski, J. Terry et al., "Gymnasium: A Standard Interface for Reinforcement Learning Environments," arXiv preprint arXiv:2407.17032, 2024.

[10] X. Dutreilh, H. Morel, J. Malenfant, et al., "From Data Center Resource Allocation to Control Theory and Back," Proceedings of the IEEE International Conference on Cloud Computing, pp. 410-417, 2010.

[11] M. Mao and M. Humphrey, "A Performance Study on the VM Startup Time in the Cloud," Proceedings of IEEE CLOUD, pp. 423-430, 2012.

[12] J. Xu, M. Zhao, J. Fortes, et al., Yousif, "Autonomic Resource Management in Virtualized Data Centers Using Fuzzy Logic-Based Approaches," Cluster Computing, vol. 11, no. 3, pp. 213-227, 2008.