Artificial Intelligence and Computing Innovations

Intelligence Science Publishing

https://www.ispublishing.com

ISSN: 3070-4944

Vo.4, No.3, 2024

Reinforcement Learning–Based Dynamic Decision Framework for Server Backend Resource Management

Junjie Jiang[1]*

[1] *Illinois Institute of Technology, Chicago, USA*

**Corresponding author: Junjie. junjiejiang1112@gmail.com*

Abstract: With the continuous development of cloud computing platforms, microservice architectures, and online service systems, server backend resource management faces challenges such as frequent load fluctuations, significant resource contention, complex service relationships, and diverse control objectives. Traditional management methods relying on static rules or local optimization are no longer adequate for the real-time control needs of highly dynamic environments. To address this issue, this paper studies reinforcement learning modeling and dynamic decision-making methods for server backend resource management, constructing a unified framework covering state representation, context awareness, policy learning, value assessment, and multi-objective reward constraints. At the state modeling level, operational information such as resource occupancy, task queuing, service latency, and dependencies is jointly represented to enhance the ability to perceive complex backend environments. At the decision modeling level, a context-aware mechanism and policy network are introduced to adaptively control resource scheduling behavior during continuous operation. At the optimization level, the stability of policy updates is improved by combining value functions and advantage estimation, and a comprehensive reward design is used to coordinate the relationship between throughput, latency, resource utilization efficiency, and operating costs. An experimental environment was built based on open-source cluster trajectory data, and compared with other methods in the same direction. The results show that the proposed method can achieve good overall performance in complex backend resource management scenarios, demonstrating strong decision-making effectiveness and system adaptability. The research results indicate that reinforcement learning can provide a more intelligent, systematic, and long-term optimization-capable modeling path for server backend resource allocation and dynamic scheduling.

Keywords: Server backend resource management; reinforcement learning; dynamic decision-making; resource scheduling optimization

1. Introduction

Against the backdrop of the continuous expansion of digital services and the ongoing evolution of computing infrastructure, server backend systems have become the core foundation supporting online business operations, data processing, scheduling, and continuous service provision[1,2]. With the widespread application of cloud computing, virtualization deployment, microservice collaboration, and hybrid load operation modes, backend resource management faces greater dynamism and complexity. Resource demands fluctuate significantly over time, different services have close dependencies, and multiple computing tasks run in parallel in a shared environment, creating a highly coupled interactive process between resource allocation, scheduling control, and service assurance. Under these conditions, resource management not only affects system throughput and service response efficiency but also directly impacts infrastructure utilization, operating cost control, and overall stability. Therefore, research on more intelligent, adaptive, and long-term optimization-capable decision-making methods for server backend resource management has significant theoretical and practical value[3].

However, existing backend resource management methods still face many limitations in complex and dynamic environments[4]. On the one hand, traditional management mechanisms relying on fixed thresholds, empirical rules, or static configurations struggle to accurately respond to real-time load changes, often only making passive adjustments to local conditions and failing to consider overall operational quality. On the other hand, resource consumption, queuing pressure, service latency, and task scheduling behavior in backend systems are not independent but exhibit significant temporal correlations and interactive effects. This makes it difficult for single-step optimization or single-objective control methods to characterize the long-term feedback relationships in real-world systems. Simultaneously, server backend resource management typically needs to simultaneously satisfy multiple objectives such as performance guarantees, resource utilization, and overhead constraints. Without a unified modeling mechanism, this can easily lead to effective local optimization but overall regulatory imbalance. Therefore, establishing a dynamic decision-making framework oriented towards long-term benefits in complex backend environments has become a crucial problem that urgently needs to be addressed in related research[5].

To address these shortcomings, this paper focuses on the dynamic regulation requirements in server backend resource management and studies a modeling and decision-making method based on reinforcement learning. This method starts from the operational characteristics of the backend system, incorporating resource status, task queuing, service response, and dependency information into a unified state space. It enhances the ability to express complex environmental changes through a context-aware mechanism and leverages policy learning and value assessment to collaboratively optimize the decision-making process[6]. During the modeling process, reinforcement learning is used to characterize the continuous interaction relationships in resource allocation and scheduling control, enabling the model to learn more adaptive control strategies as it continuously receives environmental feedback. Meanwhile, to improve decision-making quality and system practicality, this paper further constructs a multi-objective reward constraint, unifying throughput, latency control, resource utilization efficiency, and operating costs into the optimization objective. This transforms the resource management process from a passive response based on local rules to an active decision-making process oriented towards global benefits.

The main contributions of this paper are as follows:

(1) Focusing on the server backend resource management problem, a reinforcement learning modeling framework oriented towards long-term benefit optimization is constructed, unifying state

perception, resource regulation, and dynamic decision-making in complex backend environments into a single research paradigm.

(2) Addressing the problem of strong coupling and rapid dynamic changes of multi-source operational information in backend systems, a state representation method integrating resource occupancy, queuing behavior, service latency, and dependencies is designed, enhancing the model's ability to structurally characterize complex backend environments.

(3) Addressing the multi-objective collaborative needs in resource management, a reward constraint mechanism that balances system performance, resource efficiency, and operating costs is constructed, and a more stable decision-making learning process is achieved by combining policy networks and value assessment.

(4) Based on the actual needs of intelligent management of server backend, a dynamic decision-making method with adaptive regulation capability is proposed, which provides a valuable research idea for cloud platform resource scheduling, online service governance, and intelligent operation and maintenance scenarios.

2. Background

With the continuous development of cloud computing, edge computing, and microservice architectures, server backend systems are increasingly characterized by diverse task types, frequent fluctuations in resource requests, complex service dependencies, and significant dynamic changes in the operating environment. Especially in high-concurrency access, hybrid load deployments, and multi-tenant shared scenarios, the coupling effect between various resources such as CPU, memory, bandwidth, cache, and storage is constantly increasing. Traditional resource management methods relying on static thresholds, empirical rules, or single-step optimization are no longer sufficient to meet the comprehensive requirements of complex backend systems for real-time performance, stability, and resource utilization efficiency. In actual operation, resource allocation not only affects the processing efficiency of individual service instances but also further impacts task queuing latency, service response quality, system throughput, and overall energy consumption. This transforms backend resource management into a complex sequential decision-making problem with a high-dimensional state space, long-term benefit constraints, and continuous feedback characteristics[7].

Against this backdrop, how to achieve adaptive dynamic decision-making oriented towards long-term goals based on system operating states has become an important direction in server backend resource management research. Reinforcement learning methods can learn resource scheduling strategies through continuous interaction with the environment. Without explicitly constructing an accurate analytical model of the system, they can gradually characterize the intrinsic relationships between load changes, resource contention, and performance feedback, providing a new technical path for decision optimization in complex dynamic environments. Compared to traditional optimization methods that primarily focus on local configuration adjustments or short-term performance improvements, reinforcement learning is better suited to handling common resource management issues such as delayed rewards, policy transfer, online adjustments, and multi-objective collaboration. Therefore, research on reinforcement learning modeling and dynamic decision-making methods for server backend resource management scenarios not only has significant theoretical research value but also practical implications for improving the intelligent operation and maintenance capabilities, service assurance levels, and overall resource allocation efficiency of modern computing infrastructure.

3. Methodology

3.1 Overall Reinforcement Learning Formulation

Server-side backend resource management is formulated as a constrained sequential decision problem in which the controller continuously observes workload evolution, service interaction status, and infrastructure pressure, and then issues adaptive control actions for resource allocation and scheduling adjustment. Rather than optimizing each decision in isolation, the proposed framework emphasizes long-horizon utility so that transient performance gains do not induce cumulative instability, excessive migration overhead, or delayed service degradation under bursty demand. At each decision epoch, the environment state integrates resource occupancy, request backlog, latency signals, and service dependency statistics, allowing the policy to capture both instantaneous operating conditions and latent coupling across backend components. Such a design is essential because CPU contention, memory pressure, queue accumulation, and cross-service invocation delay rarely evolve independently in real deployments, and effective control therefore requires an explicit mechanism for representing multi-factor interactions within a unified optimization process. From a formal perspective, the backend control loop is modeled as a Markov decision process:

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma) \tag{1}$$

where $\mathcal{S}$ denotes the state space composed of multi-source runtime observations, $\mathcal{A}$ denotes the action space for scaling and scheduling operations, $\mathcal{P}$ characterizes the environment transition induced by workload and control feedback, $\mathcal{R}$ defines the reward function, and γ is the discount factor used to balance short-term responsiveness and long-term stability. To better describe heterogeneous backend dynamics, the state at time step t is organized as:

$$s_t = [u_t, q_t, l_t, d_t, m_t] \tag{2}$$

in which u_t aggregates utilization signals such as CPU, memory, and bandwidth, q_t captures queue-related load intensity, l_t reflects latency and response-time behaviors, d_t represents service dependency and invocation information, and m_t contains auxiliary monitoring statistics that improve robustness under nonstationary traffic patterns. By embedding these elements into a single state representation, the decision process can evaluate whether local resource scarcity originates from genuine demand growth, upstream bottlenecks, temporary burst accumulation, or inefficient service coordination, thereby making subsequent policy learning more structurally informed. Furthermore, this paper also presents the overall model architecture diagram, as shown in Figure 1.

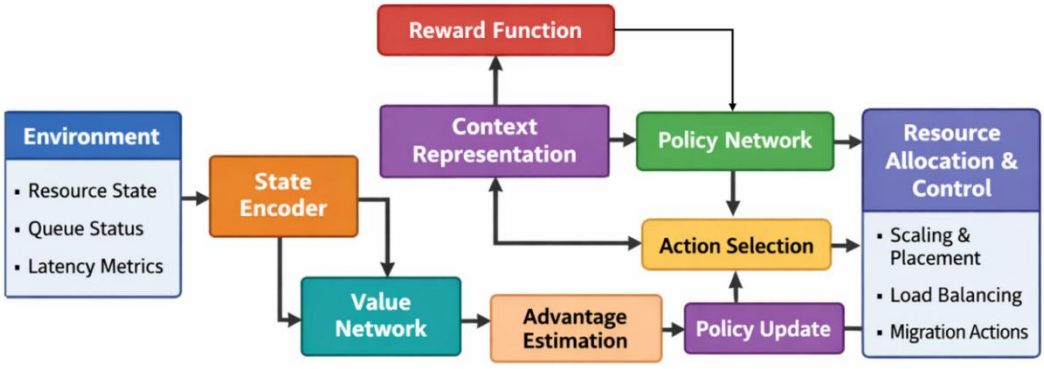

Figure 1. Overall model architecture diagram

3.2 State Representation and Context-Aware Policy Modeling

A direct use of raw monitoring vectors is often inadequate because backend signals are noisy, asynchronous, and highly correlated across time, so a compact context-aware encoder is introduced to transform the original observation into a control-oriented latent representation. Instead of treating all runtime dimensions equally, the encoder highlights dominant pressure sources and suppresses redundant fluctuations, which helps the policy focus on decision-relevant patterns such as persistent queue growth, coordinated latency increase among dependent services, and early-stage resource saturation. Given state s_t, a nonlinear embedding is produced by:

$$h_t = \phi(W_s s_t + b_s) \tag{3}$$

where W_s and b_s are learnable parameters and $\phi(\cdot)$ is a nonlinear activation that improves the expressive capacity of the latent state. Since sudden workload bursts and delayed performance feedback can distort one-step observations, temporal context is further integrated through a gated update mechanism:

$$c_t = \sigma(W_c[h_t; c_{t-1}] + b_c) \odot h_t + (1 - \sigma(W_c[h_t; c_{t-1}] + b_c)) \odot c_{t-1} \tag{4}$$

where c_{t-1} preserves historical control context, $\sigma(\cdot)$ generates adaptive gates, and $\odot$ denotes element-wise interaction that balances new evidence and prior system memory. On this basis, the policy network outputs an action distribution conditioned on the context-enhanced representation:

$$\pi_\theta(a_t|s_t) = softmax(W_p c_t + b_p) \tag{5}$$

so that the controller can choose among scaling, throttling, load redistribution, or placement adjustment operations according to the estimated system condition. This contextual policy formulation is meaningful because backend control quality depends not only on the current utilization snapshot but also on whether the observed pressure is transient, recurrent, or structurally propagated through service dependencies, and such distinctions are more effectively captured in the latent control state than in the original metric space.

3.3 Multi-Objective Reward Design for Dynamic Backend Control

Resource management in server backends is inherently multi-objective, as higher utilization is desirable only when service quality remains within acceptable bounds and operational overhead does not become excessive. For this reason, the reward is designed to reflect a coordinated trade-off among latency reduction, throughput preservation, efficient resource usage, and control smoothness, preventing the learned policy from converging to degenerate behaviors such as aggressive over-provisioning or unstable oscillatory scaling. At time step t, the immediate reward is defined as:

$$r_t = \alpha \widehat{T}_t - \beta \widehat{L}_t - \chi \widehat{C}_t - \delta \widehat{O}_t \tag{6}$$

where $\widehat{T}_t$ denotes normalized throughput benefit, $\widehat{L}_t$ represents the normalized latency penalty, $\widehat{C}_t$ measures resource consumption cost, $\widehat{O}_t$ quantifies control overhead such as migration or reconfiguration frequency, and $\alpha, \beta, \chi, \delta$ are balance coefficients that regulate the relative importance of these objectives. To avoid reward domination by any single metric and to ensure stable training across different workload scales, the long-term return is accumulated as:

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \tag{7}$$

which encourages the policy to consider downstream consequences of current actions, including delayed queue release, future congestion propagation, and repeated adjustment costs. This reward construction is particularly important in backend environments because an action that temporarily lowers latency may later trigger resource fragmentation or service imbalance, whereas a slightly conservative action can yield superior long-horizon efficiency by maintaining smoother system evolution. Through this formulation, the decision agent is guided to learn a control strategy that jointly respects performance, efficiency, and stability rather than pursuing any single operational indicator in an isolated manner.

3.4 Value Optimization and Stable Policy Update

Policy learning is carried out through an actor-critic optimization mechanism so that action selection and value estimation can be improved in a mutually reinforcing manner under dynamic workload transitions. A separate value network is introduced to evaluate the expected cumulative return of the current control context, which reduces variance in policy optimization and improves convergence reliability in complex backend environments with delayed and noisy feedback. The state value is estimated by:

$$V_\omega(s_t) = f_\omega(c_t) \tag{8}$$

where $f_\omega(\cdot)$ is a learnable approximator parameterized by ω and driven by the context representation c_t produced by the encoder. Based on the estimated value, the advantage function is computed as:

$$A_t = r_t + \gamma V_\omega(s_{t+1}) - V_\omega(s_t) \tag{9}$$

so that the policy update can emphasize whether the selected action performs better or worse than the current expectation under the same system condition. The training objective for the actor then takes the form:

$$\mathcal{L}_{actor} = -\mathbb{E}_t[\log \pi_\theta(a_t|s_t) A_t] \tag{10}$$

while the critic minimizes temporal-difference error to refine return estimation and preserve learning stability throughout nonstationary traffic fluctuations. Such a joint optimization process is meaningful because backend resource management requires both immediate adaptability and consistent evaluation of future impact, and the coupling of policy and value learning allows the controller to improve decision quality without relying on handcrafted control rules or static threshold heuristics. As training progresses, the learned strategy gradually acquires the ability to allocate resources in a demand-aware and system-aware manner, thereby providing a principled solution for dynamic backend decision making under complex operational constraints. Finally, the overall procedure of the proposed method is summarized in Figure 2.

Algorithm 1 Overall Reinforcement Learning Procedure for Backend Resource Management

Require: Backend environment $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, policy network π_θ, value network V_ω, maximum episode number N, horizon T

Ensure: Learned policy parameters θ and value parameters ω

Initialize policy parameters θ and value parameters ω
Initialize backend environment and monitoring streams
for *episode* $= 1$ to N **do**
  Reset environment and obtain initial state s_0
  for $t = 0$ to $T - 1$ **do**
    Encode the current observation and construct context representation c_t
    Sample action $a_t \sim \pi_\theta(a_t \mid s_t)$
    Execute a_t for backend resource allocation and control
    Observe reward r_t and next state s_{t+1}
    Estimate $V_\omega(s_t)$ and $V_\omega(s_{t+1})$
    Compute advantage $A_t = r_t + \gamma V_\omega(s_{t+1}) - V_\omega(s_t)$
    Update the actor by minimizing $\mathcal{L}_{\text{actor}} = -\log \pi_\theta(a_t \mid s_t) A_t$
    Update the critic with temporal-difference regression
    Set $s_t \leftarrow s_{t+1}$
  end for
end for
return θ, ω

Figure 2. Overall Procedure of the Proposed Method

4. Experimental Results and Analysis

4.1 Dataset

This paper selects Alibaba Cluster Trace v2018 as the data foundation for research on server backend resource management. This dataset originates from the operational tracking information of a real production cluster and is a publicly released open-source cluster trajectory data resource, possessing strong engineering realism and research reproducibility. The dataset simultaneously includes mixed deployment information of online long-running services and offline batch processing tasks, reflecting the actual operational characteristics of modern server backends under conditions of shared resources, dynamic load, and multi-task parallelism. Compared to simple monitoring data containing only a single performance indicator, this dataset covers multi-dimensional information such as task scheduling, resource consumption, instance operation, and machine status, making it more suitable as a foundational data source for server backend resource management, dynamic scheduling optimization, and reinforcement learning decision modeling.

From the perspective of relevance to the paper's theme, this dataset can provide key observational evidence for reinforcement learning modeling, including state construction, action design, and reward feedback. On the one hand, the data contains information on CPU and memory resource usage, as well as operational behavior at the task and instance levels, which can be used to characterize the resource pressure and load fluctuations of the server backend at different times. On the other hand, the resource competition, scheduling coupling, and performance changes brought about by the mixed deployment of online services and batch tasks are highly consistent with the dynamic decision-making problem in backend resource management. Therefore, based on this dataset, reinforcement learning tasks such as

resource allocation, load regulation, and service stability optimization can be constructed relatively naturally, ensuring that the proposed method maintains good consistency between research objectives, data semantics, and application scenarios.

4.2 Experimental setup

To verify the effectiveness of the proposed reinforcement learning method in server backend resource management scenarios, the experiment constructed a resource management environment based on the Alibaba Cluster Trace v2018 dataset. Task execution records, instance state information, and machine resource usage information were organized and segmented chronologically. The first 70% of the time-series data was used for training, the last 15% for validation, and the last 15% for testing, ensuring that the policy learning process conforms to the temporal evolution characteristics of dynamic decision-making problems. The state space consists of CPU utilization, memory utilization, task queue length, instance load intensity, and latency-related statistics. The action space includes discrete control operations such as resource scaling, load redistribution, and scheduling adjustments. The reward function comprehensively considers throughput gains, latency penalties, resource overhead, and control costs. During training, an actor-critic optimization framework was used. Both the policy network and value network used a two-layer fully connected structure. The optimizer was Adam, and a discount factor was used to control long-term reward modeling. To ensure a clear experimental setup and ease of reproduction, the specific parameter configurations are shown in Table 1.

Table 1. Experimental Setup Parameters

Parameter Category	Parameter Name	Value
Dataset	Dataset	Alibaba Cluster Trace
Data Split	Training Set	70%
Data Split	Validation Set	15%
Data Split	Test Set	15%
State Space	State Dimension	5
Action Space	Action Type	Scaling, Redistribution, Scheduling
Policy Network	Hidden Layers	2
Policy Network	Hidden Units	128, 64
Value Network	Hidden Layers	2
Value Network	Hidden Units	128, 64
Optimizer	Optimizer	Adam
Learning Rate	Learning Rate	0.001
Discount Factor	Discount Factor	0.99
Batch Size	Batch Size	64
Training Process	Episodes	200
Training Process	Decision Horizon	100

4.3 Experimental Results and Analysis

To enhance the completeness of the comparative analysis, reinforcement learning methods closely related to server backend resource management, dynamic scheduling in cloud environments, Kubernetes auto-scaling, and microservice resource control were selected as reference methods. The selected methods all revolve around resource allocation, load balancing, quality of service assurance, or operational overhead control, demonstrating high consistency with the reinforcement learning modeling and dynamic decision-making theme of this paper on server backend resource management. Considering both the backend system's operational characteristics and resource control objectives, four metrics: TH, LAT, RU, and COST, are presented in a unified manner in the table. TH represents throughput, LAT represents latency, RU represents resource utilization, and COST represents overall resource overhead. The experimental results are shown in Table 2.

Table 2. Experimental results compared with other models

Method	TH	LAT	RU	COST
Method[8]	91.36	28.74	83.52	31.48
Method[9]	89.92	31.08	81.67	33.15
Method[10]	90.47	30.21	82.34	32.06
Method[11]	88.75	33.46	79.88	35.27
Method[12]	87.94	34.12	78.95	36.03
Method[13]	90.83	29.67	82.91	31.74
Method[14]	89.58	31.56	80.73	33.82
Ours	93.24	26.35	86.47	29.16

The algorithm proposed in this paper demonstrates stronger comprehensive optimization capabilities in server backend resource management tasks, indicating that it can effectively coordinate the relationship between throughput, service latency, resource utilization efficiency, and operating costs. Its advantage lies not in the local improvement of a single indicator, but in the unified modeling of the overall system performance. This shows that the reinforcement learning decision-making process can effectively capture load fluctuations, resource contention, and scheduling feedback information in the backend environment and gradually form a more reasonable control strategy through continuous interaction. For highly dynamic and tightly coupled management scenarios like server backends, this decision-making ability that balances performance and overhead has high practical value, further demonstrating the good adaptability and stability of the constructed method in complex resource regulation problems.

Furthermore, the good results achieved by the algorithm reflect the well-coordinated synergy between the designed state representation, context modeling, reward constraints, and policy update mechanism. By incorporating resource status, queuing pressure, latency behavior, and service-related information into a unified decision-making framework, the model not only enhances its perception of the backend operating status but also improves the matching degree between policy output and real system requirements. Building upon this foundation, the joint optimization approach combining value assessment and advantage-guided optimization further improves the decision-making quality during the policy learning

process, making resource allocation and scheduling control more aligned with the actual requirements of stable server backend operation and efficient service. Therefore, the method presented in this paper not only enhances the intelligence level of backend resource management but also provides a valuable modeling approach for subsequent research on dynamic decision-making for complex computing infrastructures.

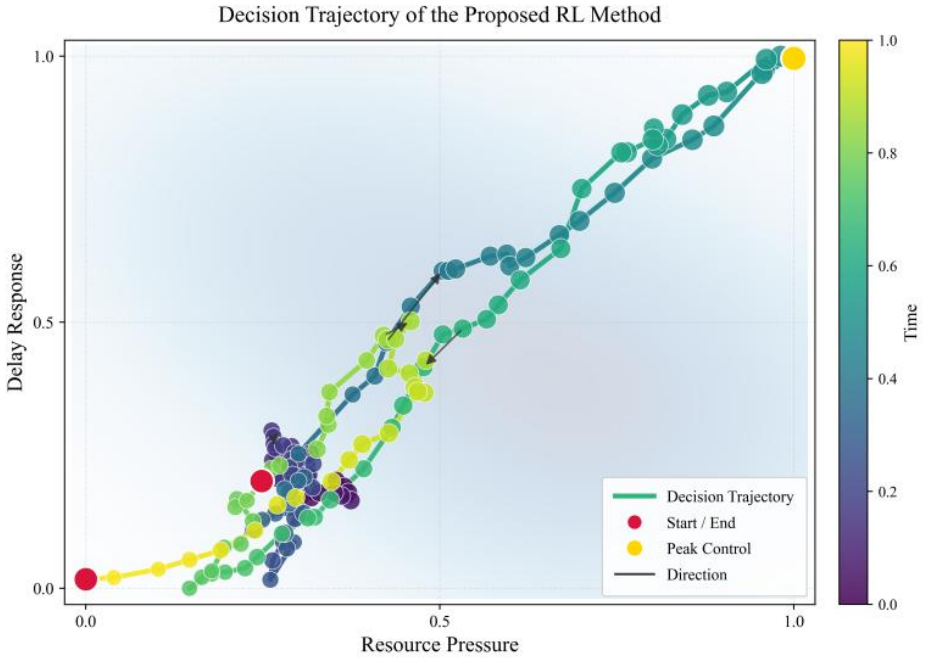

Figure 3. The proposed reinforcement learning method is used to visualize the dynamic decision trajectory in a backend resource management scenario

As shown in Figure 3, the proposed algorithm demonstrates good decision-making and state control capabilities in backend resource management scenarios, indicating that the constructed reinforcement learning framework can form a relatively clear and stable control path in complex operating environments. The trajectory distribution reflects that the model can effectively adjust according to changes in resource pressure and service response status, making the decision-making process highly coherent and targeted. It also shows good synergy between state representation, context modeling, and policy updates. Overall, this method not only effectively portrays the dynamic interaction relationships in the backend system but also provides more structured decision support for resource allocation and scheduling control, thus demonstrating high application value.

The algorithm proposed in this paper demonstrates strong action segmentation and state identification capabilities in backend resource management scenarios, forming a relatively clear decision region based on the joint changes in resource pressure and load intensity. As shown in Figure 4, the distribution of various actions in the state space exhibits strong structure, indicating that the constructed policy network can effectively learn the control requirements under different operating states and maintain good decision consistency in complex environments. Overall, this method can effectively map multi-dimensional operational information in the backend system into control behaviors with practical management significance, providing a more stable and reasonable decision-making basis for resource scheduling and dynamic management.

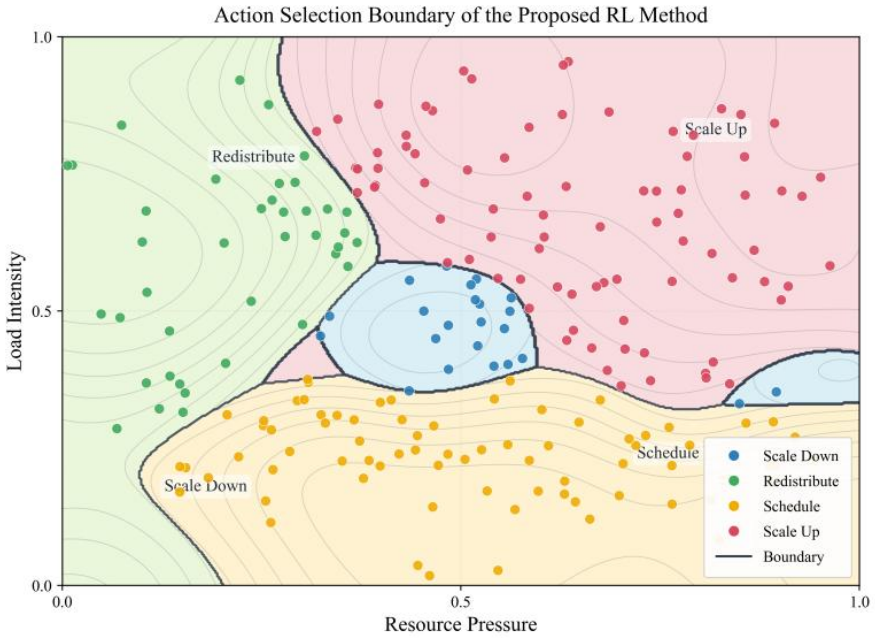

Figure 4. The proposed reinforcement learning method presents visualization results of action selection boundaries under different load intensities

5. Conclusion

This paper addresses the challenges of dynamic fluctuations in resource demand, high-dimensional coupling of system states, and multiple constraints on control objectives in server backend resource management. It proposes a reinforcement learning model and a dynamic decision-making method focused on long-term benefit optimization. To overcome the limitations of traditional resource management methods in complex backend environments, which struggle to balance throughput, service latency, resource utilization efficiency, and operating costs, a unified framework is constructed, encompassing state representation, context modeling, policy learning, value assessment, and multi-objective reward constraints. This framework enables more adaptive dynamic optimization of resource allocation and scheduling control during continuous interaction. By uniformly modeling resource pressure, load changes, service response behavior, and scheduling feedback relationships in server backend operation scenarios, the proposed method theoretically enhances reinforcement learning's ability to characterize complex infrastructure environments and provides a more systematic and intelligent solution to backend resource management problems.

From an overall research perspective, the significance of this work lies not only in the methodological decision modeling and control mechanism design but also in its promotion of intelligent operation and maintenance and dynamic resource regulation practices. With the continuous development of cloud computing platforms, microservice systems, edge service architectures, and hybrid task deployment environments, server backend resource management has become a crucial factor affecting system performance stability, service quality assurance capabilities, and infrastructure utilization efficiency. The reinforcement learning method proposed in this paper enables more targeted resource control under complex operating conditions, providing a valuable technical path for cloud data center scheduling, elastic scaling of online services, collaborative management of backend tasks, and resource governance in multi-tenant environments. Simultaneously, this research helps to promote the transformation of backend resource management from static configuration-driven to data feedback-driven and intelligent policy-driven approaches, which have significant practical implications for improving the automation level of

modern computing infrastructure, reducing operational costs, and enhancing the continuous service capabilities of systems.

Future research can further expand the application boundaries of the proposed method in larger-scale, more complex, and more uncertain backend environments. On the one hand, by combining richer heterogeneous monitoring information, cross-level operational characteristics, and service dependencies, the fine-grained expressive power of state modeling can be further improved, enabling the decision-making process to more fully reflect the complex behavior of real backend systems. On the other hand, it can continuously enhance the generalization, transferability, and engineering applicability of policy learning to meet practical needs such as multi-node collaborative control, cross-cluster resource orchestration, online deployment adaptation, and dynamic scheduling under security constraints. With the continuous evolution of intelligent computing infrastructure, cloud-native systems, and autonomous operation and maintenance technologies, reinforcement learning methods for server backend resource management are expected to play a more profound role in related application areas such as cloud platform scheduling optimization, distributed service governance, green computing resource allocation, and high-reliability business support, and provide a more extensible theoretical foundation and methodological support for intelligent infrastructure management research.

References

[1] M. Xu, et al., "CoScal: Multifaceted scaling of microservices with reinforcement learning," IEEE Trans. Netw. Service Manag., vol. 19, no. 4, pp. 3995-4009, 2022.

[2] H. Qiu, et al., "Reinforcement learning for resource management in multi-tenant serverless platforms," in Proc. 2nd Eur. Workshop Mach. Learn. Syst., 2022, pp. 1-6.

[3] A. Zafeiropoulos, et al., "Reinforcement learning-assisted autoscaling mechanisms for serverless computing platforms," Simul. Model. Pract. Theory, vol. 116, p. 102461, 2022.

[4] A. Abdel Khaleq and I. Ra, "Intelligent microservices autoscaling module using reinforcement learning," Cluster Comput., vol. 26, no. 5, pp. 2789-2800, 2023.

[5] A. Mampage, S. Karunasekera, and R. Buyya, "Deep reinforcement learning for application scheduling in resource-constrained, multi-tenant serverless computing environments," Future Gener. Comput. Syst., vol. 143, pp. 277-292, 2023.

[6] S. Agarwal, M. A. Rodriguez, and R. Buyya, "A deep recurrent-reinforcement learning method for intelligent autoscaling of serverless functions," IEEE Trans. Serv. Comput., vol. 17, no. 5, pp. 1899-1910, 2024.

[7] C. Shao, "Multi-Scale Temporal Deep Learning with Transformers for Microservice Backend Anomaly Detection", 2024.

[8] J. Zhao, M. A. Rodriguez, and R. Buyya, "A deep reinforcement learning approach to resource management in hybrid clouds harnessing renewable energy and task scheduling," in Proc. IEEE Int. Conf. Cloud Comput. (CLOUD), 2021, pp. 1-8.

[9] Z. Li, "Log Event Graph Modeling for Backend Anomaly Detection with Multi-Relational Representation Learning", 2024.

[10] C. Wen, "Modeling Evolving Service Dependencies: Dynamic Graph Learning for Microservice Anomaly Detection", 2024.

[11] K. Hu, M. Xu, K. Ye, and C. Xu, "Adaptive SLO-aware Resource Scaling for Dynamic Microservice Systems," arXiv preprint arXiv:2409.14953, 2024.

[12] L. Schuler, S. Jamil, and N. Kühl, "AI-based resource allocation: Reinforcement learning for adaptive auto-scaling in serverless environments," in Proc. IEEE/ACM Int. Symp. Cluster, Cloud Internet Comput. (CCGrid), 2021, pp. 804-811.

[13] Y. Ni, "Learning Multi-Scale Generative Representations for Cloud Performance Anomaly Detection via Self-Distillation", 2024.

[14] X. Yang, "Trend-Fluctuation Decomposition with Deep Residual Networks for System Forecasting", 2024.