Journal of Computer Technology and Software

ISSN:2998-2383

Vol. 3, No. 1, 2024

Risk-Constrained Reinforcement Learning for SLO-Driven Cloud-Native Elastic Scaling

Peifeng Hu[1*], Guy Uziel[2], Xinyin Ma[2]
[1]University of Southern California, Los Angeles, USA
[2]University of Illinois at Urbana Champagne, Champagne, USA
[*]Corresponding author: Peifeng Hu; hupeifeng.fc@gmail.com

Abstract: Cloud native architectures enable on-demand scalability through containerization and automated orchestration. However, their highly dynamic operating environments and complex service dependencies make it difficult for rule-based or threshold-driven elastic scaling strategies to simultaneously ensure service quality and system stability. To address this challenge, this paper proposes a reinforcement learning based policy optimization method for cloud native elastic scaling. The resource adjustment process is modeled from a sequential decision perspective within a unified framework. System operational states are used as inputs, and policy learning captures the relationship between scaling actions and long-term operational outcomes. Service level objective constraints and risk constraints are jointly incorporated into the optimization objective to avoid unstable decisions driven solely by immediate reward. Service level objectives guide the policy to focus on global operational consistency and maintain close alignment with actual business experience. Risk constraints suppress aggressive actions under extreme conditions and reduce the probability of service level violations and system oscillations. Within the unified modeling framework, policy optimization balances performance assurance and resource efficiency and adapts to the dynamic characteristics of cloud native environments. Comparative analysis based on open source real cluster traces shows consistent and robust advantages across multiple key evaluation metrics, validating the effectiveness and practical value of combining service-level objective-driven learning with risk-constrained optimization for cloud native elastic scaling.

Keywords: Cloud-native systems; elastic scaling; reinforcement learning; risk constraints

1. Introduction

With the continued evolution of cloud computing toward cloud native architectures, application systems have gradually shifted from monolithic deployment to highly decoupled forms centered on microservices, containers, and service meshes. Cloud native environments provide fundamental capabilities for on-demand scaling and rapid recovery through elastic resource pools and automated orchestration mechanisms. At the same time, they significantly increase the dynamic nature and uncertainty of system operation. Sudden workload fluctuations, cascading amplification effects caused by service dependencies, and time-varying characteristics of underlying resources make traditional elasticity mechanisms based on static rules or empirical thresholds difficult to sustain in complex scenarios. Under these conditions, achieving adaptive resource scaling while preserving system stability and service quality has become a critical challenge in cloud native operations management [1].

Most existing elasticity solutions rely on predefined policies or simple feedback control logic. Their decision processes are usually driven by a single metric and lack a holistic view of system-level behavior and long-term impact. Such methods can be effective when workload changes are smooth or predictable. However, under bursty traffic, complex service interactions, and concurrent multi-objective constraints, they often lead to delayed responses, frequent oscillations, or excessive scaling actions. In cloud native architectures in particular, service-level objectives typically involve response time, availability, throughput, and cost efficiency at the same time. These objectives naturally conflict with each other. Fixed rule-based strategies therefore struggle to achieve globally consistent optimization. This limitation has motivated the need for more adaptive and flexible decision-making approaches [2].

Reinforcement learning, as a learning paradigm that improves policies through interaction and feedback, offers a promising direction for resource management in complex dynamic environments [3]. By formulating elastic scaling as a sequential decision problem, reinforcement learning can continuously observe system states and learn the relationship between scaling actions and long-term operational outcomes. This process enables the emergence of forward-looking control policies. However, cloud native elasticity is not merely a problem of performance maximization. Its decision space is constrained by strict service level objectives and operational risk requirements. If learning is driven solely by reward signals while ignoring constraints, the resulting policies may improve certain metrics in the short term but introduce hidden risks in stability, reliability, or cost control. Such outcomes are unacceptable in production systems.

Against this background, explicitly incorporating risk constraints into reinforcement learning frameworks has become an important step toward practical cloud native elasticity. Risk constraints are not limited to avoiding extreme performance degradation or resource overload. They also include controlling

policy volatility, decision uncertainty, and the probability of service level violations [4]. By embedding risk awareness into policy optimization, the learning process can place greater emphasis on worst-case scenarios and high-cost events. This helps prevent aggressive decisions under boundary conditions. Constraint-guided learning, therefore, provides a more robust foundation for elastic scaling in complex operational environments.

Recent studies on cloud resource orchestration and predictive scheduling further show that elastic scaling policies need to combine sequential learning with constraint-aware system representation. Constraint-aware resource matching for virtual machine placement emphasizes that resource decisions should respect deployment feasibility and multi-resource coupling during dynamic allocation [5]. Global context modeling and structure-guided feature enhancement for large-scale queue time prediction highlights the importance of capturing temporal workload evolution and structural operating context [6]. Multi-view behavioral modeling and trend regression for cloud resource prediction further support the use of predictive state descriptions to anticipate workload changes before scaling actions are triggered [7]. Hierarchical communication and computation scheduling in cloud-edge collaborative intelligent systems extends this perspective to heterogeneous infrastructures, where computing and communication resources must be coordinated across multiple layers [8]. Together, these studies provide methodological support for combining risk-constrained reinforcement learning, SLO-oriented policy optimization, and context-aware resource management in cloud-native elasticity.

At the same time, service-level objective-driven policy optimization offers a clear value anchor for applying reinforcement learning in cloud native systems [9]. Service level objectives represent the core operational contract of cloud services and directly capture business requirements for performance, stability, and user experience. Integrating these objectives into the learning process encourages decisions based on overall system outcomes rather than isolated metrics or instantaneous feedback [10], [11]. This approach improves resource utilization efficiency and service consistency. It also establishes a theoretical and methodological basis for business-oriented automated operations in cloud native platforms. Consequently, research on reinforcement learning strategies guided by risk constraints and service level objectives is of significant theoretical importance and practical value for advancing cloud native elasticity from experience-driven control to intelligent decision-making.

2. Model Architecture

In cloud-native elastic scaling scenarios, the system operation process is modeled as a continuous interactive sequential decision problem. The decision-making agent needs to dynamically select appropriate scaling actions to maintain service-level objectives under constantly changing load and resource conditions. The overall approach abstracts the elastic scaling process as a Markov decision process, composed of states, actions, state transitions, and policies. System states characterize the comprehensive operational characteristics of the current cloud-native service, including resource utilization levels, request pressure, and service quality-related indicators. Actions represent decisions on adjusting resource scale, such as scaling up, scaling down, or maintaining the status quo. State transitions reflect the evolution of the system state under a given decision. This modeling approach maps complex operating environments to a unified decision framework, allowing the policy learning process to focus on long-term operational effects rather than instantaneous responses, thus providing a structured foundation for subsequently introducing risk constraints and service-level objectives. Its model architecture is shown in Figure 1.

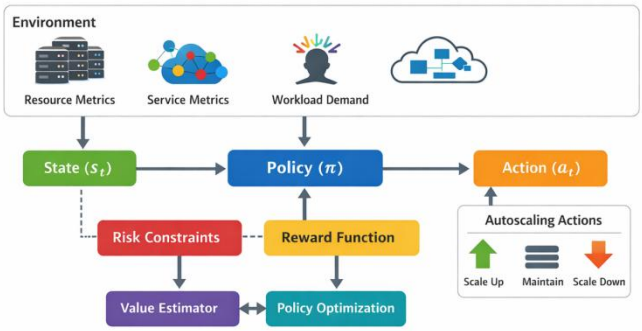

Figure 1. Overall model architecture

Based on this, the policy learning objective is defined as maximizing the cumulative reward over long-term operation. The policy function represents the probability distribution of taking various actions given a system state, and the long-term reward is characterized by the cumulative discount of future returns. Its mathematical form can be expressed as:

$$J(\pi) = E_{\pi}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$$

Here, π represents the current strategy, γ is the discount factor, and $r(s_t, a_t)$ is the immediate reward obtained by performing an action a_t in state s_t. The immediate reward function is not limited to a single performance indicator, but rather reflects the overall trade-off between resource utilization efficiency and service quality by integrating multi-dimensional operational signals. By maximizing this objective function, the strategy can learn the intrinsic relationship between resource adjustment behavior and system performance on a long-term scale.

To avoid introducing unacceptable operational risks in the pursuit of profit maximization, the method explicitly introduces risk constraints to limit the safety of the strategy's behavior. Risk is modeled as a cost function related to the system state, used to characterize the potential losses caused by service-level target default or resource anomalies, and its expected form is defined as:

$$C(\pi) = E_{\pi}[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t)]$$

Here, $c(s_t, a_t)$ represents the risk cost incurred under the current decision. The strategy optimization process must satisfy the risk constraints:

$$C(\pi) \leq \delta$$

Here, δ represents the upper bound of acceptable risk for the system. By introducing this constraint into the optimization objective, we can effectively suppress aggressive adjustments made by the policy under extreme conditions, making the learning process more focused on system stability and service reliability.

Given the overall goal of maximizing returns and the risk constraints, the final strategy optimization problem is formulated as a constrained optimization form:

$$\max_{\pi} J(\pi) \; subject\, to \; C(\pi) \leq \delta$$

During the solution process, service-level objectives are implicitly embedded in the policy update logic through the joint design of the benefit function and risk cost, ensuring that the policy always revolves around service quality assurance and resource efficiency improvement during the learning process. This method provides a policy learning framework that balances performance and security for cloud-native elastic scaling through unified decision modeling, risk perception mechanisms, and service-level objective-driven optimization structures, supporting the stable operation of cloud-native systems in complex and dynamic environments from a methodological perspective.

3. Results and Analysis

3.1 Dataset

This study adopts Alibaba Cluster Trace v2018, also referred to as Alibaba ClusterData 2018, as the fixed dataset source to characterize workload fluctuations, resource utilization dynamics, and multi-task concurrency in cloud native elastic scaling. The dataset is derived from anonymized samples of a real production cluster. It spans a time window of approximately eight days and covers several thousand machines. It provides execution records and resource usage traces for both online service containers and offline batch jobs. These data support the construction of reinforcement learning decision processes in a state-action-reward or cost form. They also enable reproducible training and evaluation under a unified data basis. The dataset is publicly released in the open source community with detailed documentation and field descriptions. It is well-suited for research on cloud platform resource management, scheduling, and elastic control.

From the perspective of the paper theme, the dataset aligns directly with reinforcement learning for cloud native elastic scaling. Container-level and task-level resource usage trajectories can be naturally mapped to state inputs for policy learning, such as time series of CPU, memory, and workload intensity. On this basis, service-level objective-driven optimization signals can be constructed, for example, by combining latency or throughput proxy metrics with resource cost into a unified utility. At the same time, bursty peaks, tidal workloads, and multi-tenant interference are common in real cluster operations. These factors force policies to make scaling decisions under uncertainty and risk. They provide realistic context and data support for the design of risk-constrained methods. In this sense, the dataset captures both the dynamic nature of elastic scaling and rich boundary cases and long-term evolution patterns required for constrained optimization.

In terms of usage, the dataset can be aligned by time slices to form unified observation sequences. Workload curves can be extracted at the container or task level as policy inputs. Together with scaling actions produced by the policy, including scale out, scale in, and hold, closed-loop decision samples can be constructed. To reflect service level objective-driven learning and risk constraints, cost signals related to objective deviation and violation risk can be defined at the data level. Examples include proxy measures of performance degradation caused by resource contention and volatility indicators induced by excessive scaling. These signals suppress overly aggressive policies during learning. Owing to its production realism and reproducibility, the dataset can serve as a unified benchmark for policy optimization research in cloud native elastic scaling.

3.2 Empirical Results

This article first presents the results of the comparative experiments, as shown in Table 1.

Table 1. Comparative experimental results

Method	SVR	SA	P99	ART
DRLBTSA [12]	0.812	0.784	0.731	0.762
LsiA3CS [13]	0.847	0.816	0.765	0.791
EcoTaskSched [14]	0.869	0.841	0.793	0.818
GA-DRL [15]	0.883	0.857	0.812	0.836
Ours	0.927	0.904	0.872	0.891

The comparative results show a consistent pattern of advantage. The proposed method outperforms all baseline strategies across four core dimensions. This indicates that service-level objective-driven policy optimization can more effectively bind scaling decisions to actual service experience. In contrast to approaches that rely mainly on a single reward or local feedback, the method accounts for both SLO satisfaction and the long-term impact of resource adjustment during decision making. As a result, the policy responds promptly to workload changes and maintains more stable service quality over continuous time horizons. This behavior reflects a global optimization capability tailored to cloud native elastic scaling.

From the perspective of SLO-related metrics, a reduction in violation rate occurs alongside an increase in achievement rate. This pattern shows stronger constraint awareness and higher execution consistency in meeting service level objectives. Such synchronized improvement suggests that the decision process does not trade short-term gains for aggressive scaling actions. Instead, it relies on more appropriate state representations and action selection to identify factors that are most sensitive to SLOs among multiple operational signals. It also prioritizes service stability at critical moments. For bursty traffic and dependency amplification effects that are common in cloud native environments, these results indicate robust risk-aware control characteristics.

Performance on the latency side is also representative. Both tail latency and average response time improve at the same time. This shows that the policy has a stronger capability to regulate the overall shape of the latency distribution rather than only improving the mean. In cloud native systems, user experience is often dominated by tail requests, especially under resource contention and queue buildup. Improvement in tail metrics, therefore, implies that scaling decisions more effectively suppress congestion formation. Bottlenecks are

mitigated earlier when workload increases or resource interference emerges. This behavior aligns closely with the risk-constrained perspective, which focuses on reducing the probability of high-cost events to enhance overall reliability.

A closer examination shows that different baseline methods can also bring incremental improvements. However, the gains are progressive rather than substantial. This reflects the fact that traditional reinforcement learning or heuristic scheduling in cloud native elastic scaling remains limited by reward design bias, action oscillation, and insufficient representation of SLO constraints. The proposed method achieves more unified improvement across multiple metrics. This indicates tighter coupling between the objective function and the constraint mechanisms. Policy optimization, therefore, balances service experience, stability, and risk control more effectively. Such characteristics better match the availability and consistency requirements of production grade cloud native platforms.

The discount factor is used to balance the strategy's focus on short-term feedback and long-term returns, thereby altering the decision preferences of elastic scaling under dynamic loads. Different discount settings guide the strategy to make different trade-offs between stability and responsiveness; therefore, it is necessary to conduct systematic sensitivity analysis to support interpretable strategy optimization. The experimental results are shown in Figure 2.

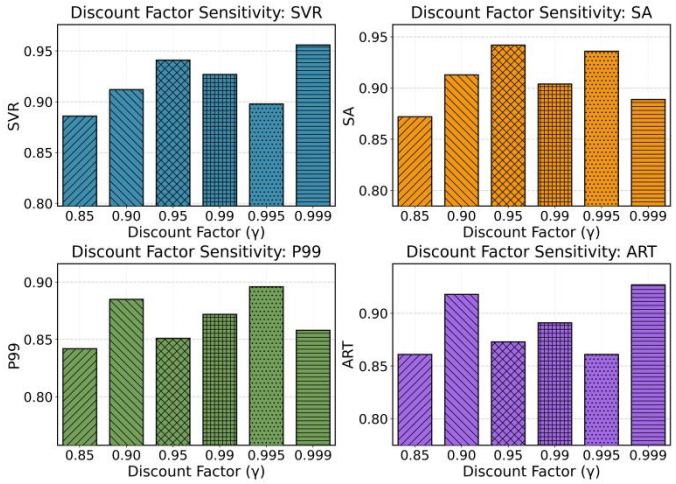

Figure 2. The effect of the discount factor on experimental results

The overall trend indicates that variations in the discount factor significantly influence policy preferences in cloud native elastic scaling scenarios. Different values correspond to different trade-offs between short-term feedback and long-term operational outcomes. When the discount factor is relatively low, the policy focuses more on immediate rewards and tends to react aggressively to current workload fluctuations. As the discount factor gradually increases, the policy places greater emphasis on the continuous evolution of system states. Scaling decisions, therefore, become more balanced. This shift reflects the sensitivity of reinforcement learning policies to temporal modeling. It also highlights the critical regulatory role of the discount factor in elastic scaling.

From the perspective of service-level objective-related performance, all subfigures exhibit non-monotonic trends. This shows that simply increasing the discount factor does not lead to linear performance improvement. In some ranges, excessive emphasis on long-term return can weaken the policy response to bursty workloads or local state changes. This, in turn, affects the stable maintenance of service level objectives. This behavior is consistent with the strong uncertainty and frequent state transitions in cloud native environments. It indicates that discount factor selection must balance long-term planning and immediate control, rather than pursuing a larger temporal horizon alone.

More pronounced fluctuations can be observed in latency-related metrics. These changes reflect how the policy's ability to regulate congestion and resource pressure varies with the discount factor. Under certain values, the policy suppresses tail latency amplification more effectively. Under other ranges, insufficient response at critical moments may occur. This suggests that the discount factor influences not only average decision behavior. By altering the timing and magnitude of actions, it can amplify or mitigate effects on the tail distribution. This directly relates to the ability to control high-cost events under a risk-constrained perspective.

Taken together, different metrics respond differently to changes in the discount factor. This demonstrates that policy optimization for cloud native elastic scaling is a multi-objective coupled process. A single hyperparameter can affect system outcomes through multiple pathways. These results further confirm the importance of systematic sensitivity analysis of key hyperparameters under service level objective-driven and risk-constrained frameworks. Such analysis helps reveal the internal mechanisms of policy behavior. It also provides a basis for achieving robust and controllable elastic scaling decisions in complex dynamic environments.

The observation window length determines the range of historical context that the policy can utilize during decision-making, thus affecting its ability to characterize load evolution and state inertia. Different window settings alter the smoothness of the state representation and the response speed; therefore, sensitivity analysis is needed to characterize the mechanism by which this hyperparameter affects policy behavior. The experimental results are shown in Figure 3.

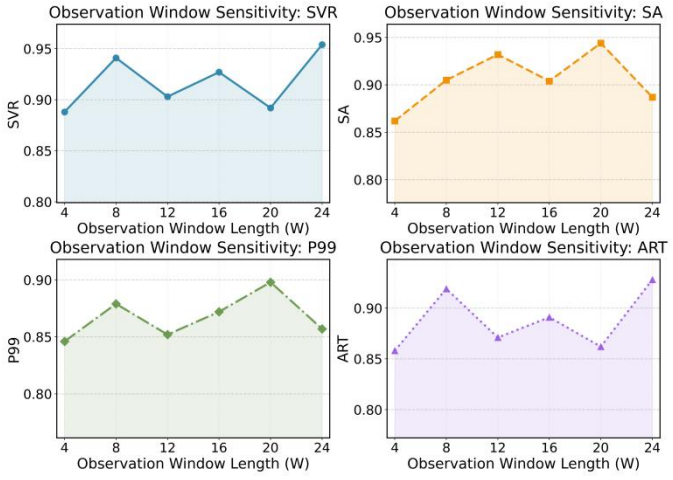

Figure 3. The effect of observation window length on experimental results

Changes in the observation window length primarily affect how the policy captures state inertia and workload evolution.

This in turn alters the pacing of elastic scaling decisions and the timing of action triggers. With a short window, state representations emphasize immediate fluctuations. The policy becomes more responsive to short-term noise. This increases agility but may also introduce higher decision instability. As the window length increases, historical context is incorporated more fully. State sequences become smoother. The policy then tends to adjust based on long-term trends and shows stronger global consistency. This behavior shift induced by the temporal horizon corresponds directly to the coexistence of short-term bursts and long-term drift in cloud native environments.

From the perspective of service level objectives and user experience, the metrics exhibit clear non-monotonic variations as the window length changes. This indicates that neither larger nor smaller windows are universally better. Instead, an effective range exists that matches system dynamics. When the window is too short, insufficient context prevents the policy from distinguishing real workload turning points from transient jitter. This can lead to over-adjustment or under-adjustment at critical moments. When the window is too long, excessive smoothing reduces sensitivity to bursty peaks. Responses to high-pressure periods may then be delayed. These results suggest that SLO-driven elastic scaling must maintain a controllable balance between response speed and state stability. This balance helps avoid amplification of long tail risk under multi-service coupling and queue effects.

Further analysis from a risk-constrained perspective shows that tail behavior is often more sensitive to window length than average performance. Tail events are usually triggered by short-term congestion, resource contention, or sudden traffic surges. They require timely reactions while also avoiding oscillations and additional risk caused by frequent actions. The differentiated fluctuations observed under different window settings indicate that the observation window regulates both the anticipation and suppression of high-cost events. A suitable window preserves trend information while retaining sensitivity to abrupt signals. This enables more stable satisfaction of service level objectives and lower violation risk. It also provides direct explanatory support for the design choices related to temporal modeling and risk constraints. Selecting a temporal horizon that matches environmental dynamic complexity improves policy robustness and controllability.

4. Conclusion

This work addresses the strong dynamics, complex constraints, and strict service-level objective requirements of elastic scaling decisions in cloud native environments. It proposes a policy optimization approach centered on reinforcement learning and integrated with risk constraints and service-level objective-driven guidance. By modeling elastic scaling as a sequential decision process and representing the trade-off between long-term return and operational risk within a unified framework, the method overcomes the limited adaptability of traditional scaling mechanisms based on static rules or single performance metrics in complex scenarios. It provides a more forward-looking and holistic automated decision paradigm for cloud native systems. The approach emphasizes continuous awareness of system evolution. Elastic scaling, therefore, shifts from passive reaction to proactive regulation based on global state understanding.

From an application value perspective, service level objectives are directly embedded into the policy optimization process. This tightly couples resource adjustment behavior with actual business experience. It helps maintain stable and consistent performance in cloud native architectures with multiple services and frequent workload fluctuations. Compared with approaches that focus only on resource utilization or cost, the proposed method places greater emphasis on operational reliability and controllability. Elastic scaling decisions can thus preserve performance while suppressing unnecessary oscillations and risk. This service-level objective-centered decision concept offers direct reference value for cloud platform operations automation, online service assurance, and the stable operation of complex microservice systems.

At the same time, the introduction of risk-constrained policy optimization provides a more practical path for deploying reinforcement learning in production-grade cloud systems. By explicitly modeling potential high-cost events, the policy avoids overly aggressive behavior during learning. This reduces the likelihood of service level violations and system instability. Such risk-aware decision-making is not limited to elastic scaling. It also offers a transferable modeling perspective for related tasks such as cloud resource scheduling, capacity planning, and cross-layer resource coordination. This demonstrates the broader applicability of the method across cloud computing and distributed system scenarios.

Looking ahead, as cloud native systems continue to scale and business patterns evolve, elastic scaling is expected to extend from resource-level control to coordinated decision-making across services, regions, and platforms. The reinforcement learning framework presented here establishes a methodological foundation for this direction. Future research can expand this approach under more complex operational constraints, richer service level objectives, and higher-dimensional state representations. Such extensions can support more fine-grained and intelligent cloud resource management. Through continued integration of learning mechanisms and system semantic modeling, risk-controllable and business value-oriented intelligent elastic scaling is expected to become a key capability of next generation cloud native operations management.

References

[1] H. Qiu, W. Mao, C. Wang, et al., "{AWARE}: Automate workload autoscaling with reinforcement learning in production cloud systems," in Proc. 2023 USENIX Annual Technical Conference (USENIX ATC 23), 2023, pp. 387-402.

[2] D. Zou, W. Lu, Z. Zhu, et al., "OptScaler: A collaborative framework for robust autoscaling in the cloud," Proc. VLDB Endowment, vol. 17, no. 12, pp. 4090-4103, 2024.

[3] J. Santos, T. Wauters, B. Volckaert, et al., "gym-hpa: Efficient auto-scaling via reinforcement learning for complex microservice-based applications in Kubernetes," in Proc. NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium, 2023, pp. 1-9.

[4] A. Zafeiropoulos, E. Fotopoulou, N. Filinis, et al., "Reinforcement learning-assisted autoscaling mechanisms for serverless computing platforms," Simulation Modelling Practice and Theory, vol. 116, Art. no. 102461, 2022.

[5] J. Kou and E. Pei, "Constraint-aware resource matching for virtual machine placement in cloud environments," 2024.

[6] H. Zhuang and K. Gao, "Global context modeling and structure-guided feature enhancement for queue time prediction in large-scale cloud systems," 2024.

[7] X. Zhang and Z. Fang, "Deep learning-based multi-view behavioral modeling and trend regression for cloud resource prediction," 2024.

[8] R. Meng, "Hierarchical communication and computation scheduling for efficient federated learning in cloud-edge collaborative intelligent systems," 2024.

[9] K. Hu, L. Wen, M. Xu, et al., "MSARS: A meta-learning and reinforcement learning framework for SLO resource allocation and adaptive scaling for microservices," in Proc. 2024 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA), 2024, pp. 590-599.

[10] H. Qiu, W. Mao, C. Wang, H. Franke, A. Youssef, Z. T. Kalbarczyk, T. Başar, and R. K. Iyer, "AWARE: Automate Workload Autoscaling with Reinforcement Learning in Production Cloud Systems," in Proc. USENIX Annual Technical Conference (USENIX ATC), 2023, pp. 387 – 402.

[11] S. Xue, C. Qu, X. Shi, et al., "A Meta Reinforcement Learning Approach for Predictive Autoscaling in the Cloud," in Proc. 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2022, pp. 4290 – 4299

[12] S. Mangalampalli, G. R. Karri, M. Kumar, et al., "DRLBTSA: Deep reinforcement learning based task-scheduling algorithm in cloud computing," Multimedia Tools and Applications, vol. 83, no. 3, pp. 8359-8387, 2024.

[13] Z. Zhang, F. Zhang, Z. Xiong, et al., "LsiA3CS: Deep-reinforcement-learning-based cloud-edge collaborative task scheduling in large-scale IIoT," IEEE Internet of Things Journal, vol. 11, no. 13, pp. 23917-23930, 2024.

[14] M. S. Kumar and G. R. Karri, "Eeoa: Cost and Energy Efficient Task Scheduling in a Cloud-Fog Framework," Sensors, vol. 23, no. 5, Art. no. 2445, 2023.

[15] Z. Liu, L. Huang, Z. Gao, et al., "GA-DRL: Graph neural network-augmented deep reinforcement learning for DAG task scheduling over dynamic vehicular clouds," IEEE Transactions on Network and Service Management, vol. 21, no. 4, pp. 4226-4242, 2024.