Journal of Computer Technology and Software
ISSN:2998-2383
Vol. 4, No. 8, 2025

Research on a Deep Reinforcement Learning-Based Adaptive Replica Allocation Method for Data Replica Placement Optimization in Distributed Storage Systems

Bo Zhang[1*], Ilyar Abdurahman[2]
[1]The University of Chicago, Chicago, USA
[2]Stevens Institute of Technology, Hoboken, USA
*Corresponding author: Bo Zhang; gs.res.zhang@gmail.com

Abstract: To address the challenges of data replica placement in distributed storage systems, such as significant fluctuations in access load, dynamic changes in node resource status, insufficient adaptability of traditional rule-based methods, and difficulties in coordinating multi-objective optimization, this paper proposes a deep reinforcement learning-based adaptive replica allocation method for replica placement optimization. This method models replica placement as a continuous decision-making process, comprehensively considering node storage utilization, request intensity, network operating status, fault risk information, and historical hotspot changes within a unified framework. It constructs a state representation for complex system environments and achieves coordinated optimization among replica quantity adjustment, placement location selection, and migration control through a policy learning mechanism. In terms of method design, this paper enhances the model's ability to perceive dynamic environments and optimize long-term returns through key steps such as state encoding, latent feature extraction, policy network decision-making, reward function construction, and parameter iterative updates. This enables the replica allocation process to balance access latency, system reliability, load balancing, and resource overhead control. To verify the effectiveness of the proposed method in distributed storage replica management scenarios, this paper selects publicly available cluster operation trajectory data to construct an experimental environment and compares it with related methods in the same direction. The results show that the proposed method outperforms the traditional method in terms of overall performance, achieving a more reasonable replica layout and more stable policy decisions under complex load conditions, demonstrating strong global coordination and environmental adaptability. This research indicates that introducing deep reinforcement learning into the replica placement optimization task of distributed storage systems can provide a new technical approach for ensuring data availability, optimizing resource scheduling, and improving service quality.

Keywords: Distributed storage system; data copy placement; deep reinforcement learning; adaptive resource allocation

1. Introduction

With the continuous development of cloud computing, big data processing, edge computing services, and artificial intelligence applications, distributed storage systems have become a crucial infrastructure supporting massive data management and high-concurrency access. In actual operation, data replication mechanisms not only undertake critical tasks such as fault tolerance and recovery, service continuity assurance, and access acceleration, but also directly impact system storage overhead, network communication load, and overall resource utilization efficiency[1,2]. Especially in complex environments with multi-tenancy, heterogeneous nodes, and dynamic loads, data access hotspots, node states, and network conditions all exhibit significant time-varying characteristics, making traditional static or semi-static replication strategies increasingly inadequate for adapting to continuously changing system requirements. Therefore, research on the optimization of data replication placement in distributed storage systems has significant theoretical and practical value[3].

The data replication placement problem essentially involves the coordination and unification of multiple objectives, including reliability, access latency, load balancing, storage cost, and network bandwidth consumption. Its optimization results directly relate to the performance boundaries and service quality of the distributed storage system. Inappropriate replication allocation can easily lead to problems such as excessive load on hotspot nodes, local link congestion, wasted storage resources, and decreased fault recovery efficiency, thereby weakening the system's stable support capability under complex business scenarios. Table 1 summarizes the main driving factors and their corresponding impacts in optimizing data replica placement in distributed storage systems. These factors collectively indicate that replica placement is no longer a simple capacity allocation problem, but a core optimization problem oriented towards global resource coordination and dynamic service assurance.

From a methodological evolution perspective, replica management in distributed storage systems is gradually shifting from an approach based on empirical rules and local heuristic design to an intelligent approach relying on environmental awareness, state modeling, and continuous decision optimization. Because the replica placement process has significant temporal decision-making characteristics, and the system state often adjusts continuously with changes in node load, fluctuations in access requests, and the evolution of failure risks, deep reinforcement learning provides a new research perspective for this problem[4]. On the one hand, this method can learn replica allocation strategies in a complex state space, enhancing its adaptability to dynamic environments. On the other hand, this method helps overcome the limitations of fixed rules in covering diverse scenarios, enabling replica placement decisions to shift from passive response to proactive optimization, thereby improving the adaptive control level of distributed storage systems under complex operating conditions.

Table 1. Main Driving Factors and Impacts of Data Replica Placement Optimization in Distributed Storage Systems

Driving Factor	Specific Manifestation	Main Impact on System Operation
Continuous growth in data scale	Massive files, objects, and block data continue to accumulate	Increases the complexity of replica management and the pressure on storage resources
Dynamic changes in access patterns	Rapid migration of hot data and uneven request distribution	Leads to local node overload and fluctuations in access latency
Increasing heterogeneity of nodes and networks	Significant differences in node computing power, capacity, and bandwidth	Reduces the applicability of a unified static strategy
Higher requirements for service reliability	Growing demand for fault recovery, continuous availability, and disaster tolerance	Raises the difficulty of replica redundancy design and placement decision making
Rising demand for resource utilization optimization	Constraints on storage, bandwidth, and energy consumption have become more prominent.	Requires fine-grained balancing among multiple objectives

Therefore, researching deep reinforcement learning-based adaptive replica allocation methods for the data replica placement optimization problem in distributed storage systems not only responds to the urgent need for refined resource allocation and stable service quality in massive data scenarios but also aligns with the overall trend of intelligent infrastructure development towards autonomous perception, autonomous decision-making, and autonomous optimization. This research helps to drive the transformation of distributed storage systems from the traditional static replica management model to a dynamic intelligent replica collaborative model, providing new theoretical support and methodological foundation for improving system reliability, reducing access latency, alleviating resource contention, and enhancing overall operational resilience[5]. At the same time, this type of research also plays an important practical role in building a new type of data infrastructure that is highly available, highly elastic, and highly efficient.

2. Background

Distributed storage systems, as a crucial component of modern digital infrastructure, are widely used in various scenarios such as cloud platforms, industrial internet, smart cities, financial data centers, and large-scale content distribution. With the continuous expansion of business scale, the number of data objects, the intensity of access concurrency, and the demand for cross-regional collaboration are constantly increasing. The system operating environment has shifted from a relatively stable centralized management model to a highly dynamic, highly coupled, and highly complex collaborative management model. In this context, data storage no longer merely serves the function of information preservation, but has gradually evolved into a critical foundational capability that simultaneously supports high-availability services, rapid access response, fault recovery assurance, and elastic resource scheduling[6]. As a vital component of distributed storage systems, replication mechanisms play a fundamental role in ensuring data reliability and improving service continuity; therefore, their configuration methods and distribution patterns have become significant factors affecting the overall operational quality of the system.

Meanwhile, the operational constraints faced by distributed storage systems are constantly increasing. Node failures, load fluctuations, network congestion, resource contention, and access hotspot migrations occur frequently, making replication management issues more dynamic and complex. Traditional methods typically rely on preset rules, fixed thresholds, or local experience for replica allocation[7]. While simple to implement, these approaches often struggle to balance global efficiency and real-time adaptability in complex scenarios. Especially when multiple objectives coexist, the number of replicas, their placement, and migration timing are tightly coupled; optimizing any single dimension can lead to a decline in other performance metrics. Therefore, research into more intelligent and adaptive replica allocation in distributed storage systems has become a crucial foundation for improving system service capabilities, enhancing resource scheduling flexibility, and supporting new data-intensive applications.

3. Method

To address the replica placement optimization problem in distributed storage systems, a deep reinforcement learning based adaptive replica allocation framework is constructed to couple global resource coordination with sequential decision making. Unlike rule-driven strategies that react to isolated indicators, the proposed method models replica placement as a long-horizon control process in which storage load, access concentration, communication cost, and failure risk are evaluated jointly. This paper presents the overall model architecture, as shown in Figure 1.

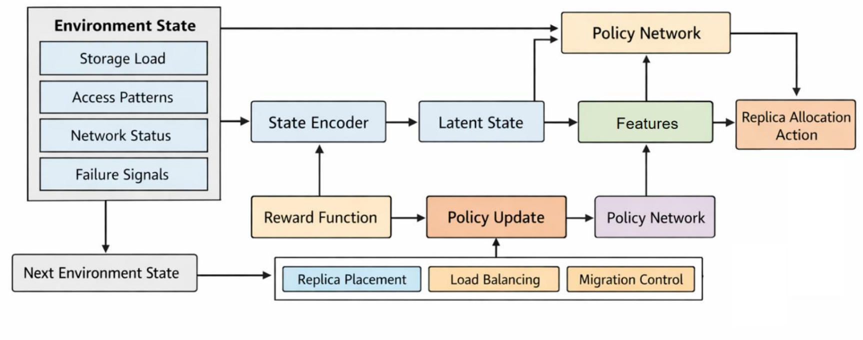

Figure 1. Overall model architecture

At each decision step, the storage cluster is represented by a system state vector that aggregates node capacity utilization, request intensity, network condition, and reliability-related signals, so that the policy can infer both instantaneous pressure and future adjustment demand from a unified observation space:

$$s_t = [\boldsymbol{u}_t, \boldsymbol{q}_t, \boldsymbol{b}_t, \boldsymbol{f}_t, \boldsymbol{h}_t]$$

Here, denotes node storage utilization, describes access request intensity, characterizes available bandwidth, reflects node failure-related risk, and summarizes historical hotspot evolution, which together enable the agent to perceive the operational context of the cluster beyond static resource snapshots. Rather than assigning replicas through fixed thresholds, the decision action is defined as a structured allocation operation that determines both the target nodes and the adjustment scale for each data block under the current state: $\boldsymbol{u}_t \boldsymbol{q}_t \boldsymbol{b}_t \boldsymbol{f}_t \boldsymbol{h}_t$

Variational-autoencoder-based query-plan anomaly detection and cost-deviation warning illustrates how reconstruction-sensitive latent representations can expose operational irregularities before they become costly system-level effects[8]. This principle supports the state encoder by enriching replica placement decisions with anomaly-aware signals derived from the relationship between current resource behavior and expected operating patterns.

$$a_t = \pi_\theta(s_t)$$

Such a design is meaningful because replica placement is not only a combinatorial selection problem but also a dynamic balancing process in which the policy must trade short-term migration overhead for long-term service stability. Once an action is executed, the environment transitions to a new state according to the system dynamics, allowing the agent to continuously refine replica placement behavior through iterative interaction:

$$s_{t+1} = \mathcal{T}(s_t, a_t)$$

Given that practical replica placement must reconcile multiple system objectives simultaneously, the reward function is formulated as a weighted integration of latency reduction, load balancing improvement, reliability enhancement, and migration cost suppression, so that the learning target remains aligned with the operational requirements of distributed storage services:

$$r_t = \alpha_1 \Delta L_t + \alpha_2 \Delta B_t + \alpha_3 \Delta R_t - \alpha_4 C_t$$

In this expression, measures the improvement in access latency, evaluates the gain in load balance, reflects the increase in reliability, and denotes the overhead induced by replica migration, while to control the relative emphasis of different optimization targets. Beyond simple performance aggregation, this reward mechanism explicitly embeds the practical significance of replica placement by encouraging the policy to avoid myopic decisions that reduce one metric at the expense of the overall system condition. A long-term optimization objective is then introduced to guide the agent toward sustainable allocation behavior across evolving workloads: $\Delta L_t \Delta B_t \Delta R_t C_t \alpha_1 \alpha_4$

Reinforcement-learning-driven dynamic cache management with hotspot-aware replacement provides a related control perspective in which access concentration and replacement cost are balanced through feedback from shifting demand[9]. Reinforcement-learning-based elastic scaling policy optimization for microservices likewise motivates treating resource expansion, contraction, and placement adjustment as coordinated decisions under changing service pressure[10]. These principles reinforce the multi-objective reward design by linking latency reduction to sustained resource efficiency rather than one-step performance gains.

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$$

Meanwhile, stable policy learning requires an effective state representation mechanism because the raw system observation often contains heterogeneous and correlated signals with different temporal sensitivities. For this reason, an encoder network is employed to project the original state into a compact latent space, thereby preserving the dominant structural information required for decision making while reducing interference from redundant fluctuations:

$$z_t = \phi_\omega(s_t)$$

By operating on the latent representation parameterized by encoder weights , the policy network can better capture the interaction between hotspot evolution and node-level resource constraints, which is essential for identifying where additional replicas should be placed and where redundant replicas should be removed. Subsequent policy improvement is performed through gradient-based optimization so that the expected cumulative return can be maximized under the learned state abstraction:$z_t \omega$

$$\theta \leftarrow \theta + \eta \nabla_\theta J(\theta)$$

This update rule, with learning rate , gives the framework the capacity to continuously adapt to nonstationary environments instead of relying on a handcrafted response pattern that becomes ineffective when access distributions shift across time.η

From a system perspective, the proposed method is meaningful because replica placement is inherently shaped by the coupling between storage heterogeneity and workload dynamics rather than by a single static constraint. Conventional allocation logic usually treats reliability protection and access efficiency as loosely connected goals, whereas the present formulation places both aspects inside one sequential optimization process and lets the policy discover their coordinated balance through interaction with the environment. Another important implication is that adaptive replica allocation can transform the role of replica management from passive redundancy maintenance into active resource orchestration, which improves the capability of distributed storage systems to respond to changing service demands. Ultimately, the method provides a principled path for integrating intelligent control into distributed storage infrastructure, making replica placement decisions more context aware, more globally consistent, and more compatible with the long-term evolution of complex data-intensive platforms.

Learning-based intelligent agents for backend resource scheduling and operational decision making further supports coupling system observations with adaptive control policies that can revise allocation choices as operational evidence evolves[11]. In the replica setting, this perspective strengthens the use of long-horizon feedback when coordinating node-level actions, migration cost, and service reliability.

4. Experimental Results and Analysis

4.1 Dataset

This paper selects Alibaba Cluster Trace 2018 as its data foundation. Released by the publicly available Alibaba Cluster Trace Program, this dataset originates from a real production cluster and represents readily available open-source cluster runtime trajectory data. The dataset includes co-location runtime information for online services and batch processing tasks, providing machine-level resource status, job instance status, and task scheduling records. It comprehensively reflects load fluctuations, resource consumption changes, and request pressure evolution in a large-scale distributed environment. For research on data replica placement optimization in distributed storage systems, while this type of real-world cluster trajectory is not specifically designed for replica management tasks, the node resource utilization, task activity, load distribution, and temporal changes it contains provide reliable data support for hotspot identification, node status awareness, and dynamic scheduling modeling in replica allocation decisions.

From a data composition perspective, Alibaba Cluster Trace 2018 records the runtime status of a large-scale production cluster within a continuous time window, describing machine events, machine resource usage, container instance deployment, and batch task execution information in multiple tables. Therefore, it is suitable for further mapping to the environmental state, action constraints, and system feedback in the replica placement problem. Using this dataset, node capacity utilization, approximate access pressure metrics, network-related load changes, and potential hotspot migration trends can be organized as reinforcement learning state inputs, thereby constructing a data representation that matches the distributed storage replica allocation process. Compared to purely simulated data, this publicly available dataset has a real business context and strong dynamic characteristics, which are more conducive to supporting the research settings of adaptive replica allocation methods and also ensure that the data source of the paper is consistent with the theme of distributed resource collaborative optimization.

4.2 Experimental Setup

To ensure the stability and reproducibility of the training process for optimizing data replica placement in distributed storage systems, the experimental section employs a unified training configuration for parameter setting of the deep reinforcement learning model, and controls key aspects such as state encoding, policy update, reward discounting, and batch learning. Considering that the replica allocation problem simultaneously involves high-dimensional state awareness, sequential decision optimization, and multi-objective tradeoffs, hyperparameter settings need to balance convergence efficiency, policy exploration capability, and training stability. Table 2 presents the main hyperparameter settings used in this experiment. These parameters collectively constitute the basic configuration for model training and policy optimization, and provide consistent experimental conditions for subsequent method comparisons and performance analysis.

Table 2. Core Hyperparameter Settings

Hyperparameter	Value
Learning rate	0.0003
Discount factor	0.99
Batch size	128
Replay buffer capacity	100000
State encoding dimension	128
Hidden layer dimension	256
Policy update interval	10

Target network update coefficient	0.005
Maximum training epochs	200
Maximum decision steps	500
Initial exploration coefficient	1.0
Final exploration coefficient	0.05

4.3 Experimental Results and Analysis

To demonstrate the differences and advantages of our proposed method compared to existing research in the area of distributed storage data replica placement, we selected representative papers closely related to replica placement, data placement, adaptive replica management, and reinforcement learning optimization for a comparative analysis. The comparison focuses on four dimensions: access latency, system reliability, load balancing, and overall overhead, denoted as Lat., Rel., L.B., and Cost., respectively. Lower values for Lat. and Cost. indicate better method performance, while higher values for Rel. and L.B. indicate better method performance. The experimental results are shown in Table 3.

Table 3. Experimental results compared with other models

Method	Lat.	Rel.	L.B.	Cost.
MacCormick et al.[12]	31.42	95.13	88.24	20.67
Liu et al.[13]	28.95	96.04	89.76	18.93
Costa Filho et al.[14]	27.84	96.38	90.42	18.11
Zheng et al.[15]	24.63	97.21	92.35	16.84
Lu et al.[16]	23.95	97.48	92.91	16.37
Shi et al.[17]	29.74	96.57	88.95	17.76
Chun et al.[18]	30.86	95.86	87.64	19.58
Ours	21.47	98.26	94.38	14.92

In the table, Lat. represents access latency, measuring the time required for a replica placement strategy to complete data access after a request arrives. A lower Lat. indicates more efficient data location and access paths, and stronger system responsiveness. Rel. represents reliability, reflecting the replica allocation scheme's ability to maintain data availability and service continuity in the event of node failures, link fluctuations, or local anomalies. A higher Rel. indicates stronger system stability in complex operating environments. L.B. represents load balancing, depicting the reasonable distribution of request traffic, storage usage, and processing pressure across different nodes. A higher L.B. indicates more balanced resource utilization, helping to alleviate hotspot aggregation and localized overload issues. Cost. represents replica management cost, primarily reflecting the combined overhead of replica creation, migration, maintenance, and resource consumption. A lower Cost. indicates that the strategy effectively controls additional resource consumption while ensuring service availability.

Overall, the proposed algorithm demonstrates strong comprehensive advantages in access efficiency, reliability assurance, load coordination, and cost control, indicating that it can effectively adapt to complex scenarios in distributed storage systems characterized by dynamic state changes, heterogeneous resource distribution, and fluctuating access demands. The results show that the constructed deep reinforcement learning adaptive replica allocation mechanism not only improves service responsiveness while ensuring data availability and system stability, but also more rationally adjusts the distribution of resource pressure among nodes, reducing unnecessary replica management overhead. This demonstrates that the proposed method possesses stronger global coordination and environmental awareness capabilities under multi-objective constraints, providing more effective decision support for replica placement optimization in distributed storage systems.

In distributed storage systems, changes in request intensity directly affect the decision frequency of replica placement strategies, resource scheduling pressure, and state feedback complexity. Therefore, it is necessary to quantitatively examine the dynamic adaptability of the proposed method under different load levels. To this end, the evolution characteristics of the strategy during the training process are observed under a uniform maximum number of training rounds. This setting can be used to characterize the stable adjustment capability and continuous optimization capability of the proposed method when facing increased load. The experimental results are shown in Figure 2.

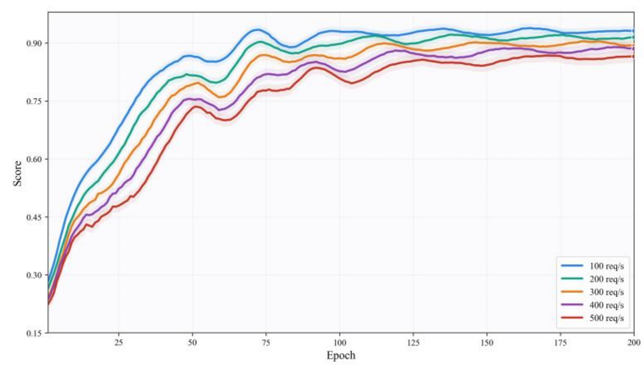

Figure 2. Visualization experiment of the dynamic convergence process of the adaptive replica allocation strategy under different replica request intensities

The proposed algorithm maintains good convergence quality and strong stability under different replica request intensities, indicating that the method has good environmental adaptability and continuous optimization capabilities. Even under a continuously increasing request load, the constructed adaptive replica allocation mechanism can still effectively coordinate the relationship between replica layout, resource consumption, and decision updates, ensuring that the policy learning process maintains high consistency and controllability. This demonstrates that the proposed method can accurately perceive state changes in the distributed storage system and form more reasonable replica allocation decisions accordingly, thus providing effective support for data availability assurance, resource adjustment, and service quality maintenance under complex load scenarios.

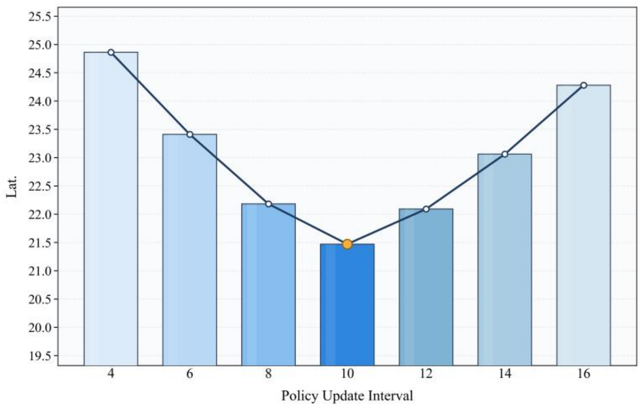

Figure 3. The impact of policy update step size changes on Lat.

As shown in Figure 3, the proposed algorithm maintains good access latency control under these parameter settings, indicating a high degree of consistency between the policy update mechanism and replica allocation decisions. A reasonable policy update step size helps enhance the model's ability to perceive changes in system state, making replica placement behavior more aligned with resource fluctuations and request changes in a distributed storage environment, thereby ensuring the stability and effectiveness of the decision-making process. Overall, the proposed method effectively coordinates the relationship between policy learning frequency and latency optimization objectives, providing reliable support for improving data access efficiency under complex operating conditions.

5. Conclusion

This paper addresses the data replica placement optimization problem in distributed storage systems. To overcome the shortcomings of traditional replica allocation methods, such as insufficient adaptability to dynamic environments, limited global coordination capabilities, and difficulties in balancing multiple objectives, a deep reinforcement learning-based adaptive replica allocation method for complex system state awareness is proposed. This method models the replica placement process as a continuous decision-making problem, unifying node load, access requests, resource status, and system reliability requirements into a unified policy learning framework. This allows replica allocation to move beyond static rules or local heuristic adjustments and dynamically adjust according to environmental changes. The results show that the proposed method better coordinates the relationship between access efficiency, data reliability, load balancing, and resource overhead, providing a systematic and intelligent solution to the replica management problem in distributed storage systems.

From a methodological perspective, the value of this work lies not only in the improvement of the replica placement strategy but also in the effective embedding of an intelligent decision-making mechanism into the distributed storage resource management process. By introducing deep reinforcement learning, replica allocation is transformed from a traditional passive response approach to a proactive optimization approach focused on long-term gains, thereby enhancing the policy's adaptability to complex states, dynamic loads, and heterogeneous resource conditions. This research approach helps drive the transformation of distributed storage systems from experience-driven management to data-driven management and provides a new theoretical foundation for resource collaborative scheduling in high-concurrency, high-availability, and high-elasticity scenarios. For cloud computing platforms, big data processing centers, edge storage networks, industrial internet data infrastructure, and smart service systems, this research has strong application support significance, helping to improve the stability, resource utilization efficiency, and continuous operation capability of underlying data services.

Meanwhile, this research also has a wide-ranging practical impact on related application areas. As the scale of digital business continues to expand and data access behavior becomes increasingly complex, the fixed configuration methods relied upon by traditional replication mechanisms are increasingly unable to meet the operational requirements of multiple scenarios, strong fluctuations, and high reliability. The method proposed in this paper provides a feasible path for building more intelligent storage systems, enabling replication management to more closely serve core needs such as business continuity assurance, enhanced system resilience, and improved service quality. In key scenarios such as financial trading platforms, medical information systems, video content distribution, smart manufacturing platforms, and city-level data service networks, the optimization of replication placement strategies not only relates to data availability and access efficiency but also to the overall security boundary and service stability level of the system. Therefore, this work has certain theoretical value and engineering significance for promoting the intelligent upgrading of critical data infrastructure.

Future research can be further deepened in several directions. First, by combining more refined network topology modeling and fault propagation characteristic analysis, the ability of replica allocation strategies to identify risks in complex systems can be further improved. Second, for scenarios such as multi-datacenter, cross-regional collaboration, and edge-cloud integration, the generalization and collaborative optimization capabilities of the method can be extended to larger-scale environments, enabling replica placement decisions to simultaneously consider local performance and global resource coordination. Third, energy consumption constraints, data security requirements, and service level agreements can be further incorporated into a unified decision-making framework, thereby enhancing the applicability of the method in green computing and trusted storage scenarios. Overall, intelligent research on replica management in distributed storage systems still has broad development prospects, and the continuous improvement of related methods will help support the construction and evolution of next-generation highly reliable, highly resilient, and highly efficient data infrastructure.

References

[1] J. Liu, M. Xie, S. Chen, et al., "TS-REPLICA: A novel replica placement algorithm based on the entropy weight TOPSIS method in spark for multimedia data analysis", Information Sciences, vol. 626, pp. 133-148, 2023.

[2] D. W. Sun, G. R. Chang, S. Gao, et al., "Modeling a dynamic data replication strategy to increase system availability in cloud computing environments", Journal of Computer Science and Technology, vol. 27, no. 2, pp. 256-272, 2012.

[3] Y. Shan, K. Chen and Y. Wu, "Explore data placement algorithm for balanced recovery load distribution", Proceedings of the 2023 USENIX Annual Technical Conference, pp. 233-240, 2023.

[4] A. Y. Aldailamy, A. Muhammed, R. Latip, et al., "Online dynamic replication and placement algorithms for cost optimization of online social networks in two-tier multi-cloud", Journal of Network and Computer Applications, vol. 224, p. 103827, 2024.

[5] J. Aznar-Poveda, T. Pockstaller, T. Fahringer, et al., "SDKV: A smart and distributed key-value store for the edge-cloud continuum", Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing, pp. 1-8, 2023.

[6] P. Chen, M. Zheng, X. Du, et al., "A Data Replication Placement Strategy for the Distributed Storage System in Cloud-Edge-Terminal Orchestrated Computing Environments", IEEE Internet of Things Journal, vol. 12, no. 15, pp. 31824-31842, 2025.

[7] J. A. Poveda, M. F. Ebner, T. Fahringer, et al., "SmartKV: A cost-effective and low-latency geo-distributed key-value store for the computing continuum", Future Generation Computer Systems, vol. 171, p. 107857, 2025.

[8] L. Ren, S. Li and Z. Liu, "Variational Autoencoder-Based Query Plan Anomaly Detection and Cost Deviation Warning for Database Systems", 2024.

[9] S. Dang, C. Chen and K. S. Chua, "Reinforcement Learning-Driven Dynamic Cache Management with Hotspot-Aware Replacement", 2024.

[10] X. Li and Z. Song, "Reinforcement Learning-Based Elastic Scaling Policy Optimization for Microservices", 2024.

[11] L. Ren, Y. Liu and Y. Zhao, "Learning-Based Intelligent Agents for Backend Resource Scheduling and Operational Decision Making", 2025.

[12] J. MacCormick, N. Murphy, V. Ramasubramanian, et al., "Kinesis: A new approach to replica placement in distributed storage systems", ACM Transactions on Storage, vol. 4, no. 4, pp. 1-28, 2009.

[13] K. Liu, J. Peng, J. Wang, et al., "Scalable and adaptive data replica placement for geo-distributed cloud storages", IEEE Transactions on Parallel and Distributed Systems, vol. 31, no. 7, pp. 1575-1587, 2020.

[14] J. S. Costa Filho, D. M. Cavalcante, L. O. Moreira, et al., "An adaptive replica placement approach for distributed key-value stores", Concurrency and Computation: Practice and Experience, vol. 32, no. 11, p. e5675, 2020.

[15] M. Zheng, X. Du, Z. Lu, et al., "A balanced and reliable data replica placement scheme based on reinforcement learning in edge-cloud environments", Future Generation Computer Systems, vol. 155, pp. 132-145, 2024.

[16] K. Lu, N. Zhao, J. Wan, et al., "Rlrp: High-efficient data placement with reinforcement learning for modern distributed storage systems", Proceedings of the 2022 IEEE International Parallel and Distributed Processing Symposium, pp. 595-605, 2022.

[17] L. Shi, Z. Wang and X. Li, "Novel data placement algorithm for distributed storage system based on fault-tolerant domain", Journal of Shanghai Jiaotong University (Science), vol. 26, no. 4, pp. 463-470, 2021.

[18] B. G. Chun, F. Dabek, A. Haeberlen, et al., "Efficient Replica Maintenance for Distributed Storage Systems", Proceedings of the 3rd USENIX Symposium on Networked Systems Design and Implementation, vol. 6, pp. 4-4, 2006.