JOURNAL OF THE ARTIFICIAL INTELLIGENCE ENGINEERS CONSORTIUM

Volume: 02 Issue: 02 (2026)

https://jaiec.org.hostajournal.com

Hierarchical Load Balancing for Multi-Cloud Large Language Model Serving under Dynamic Global Traffic

Lethabo Ndlovu; Thabo Du Plessis

School of Computer Science and Applied Mathematics, University of the Witwatersrand, Johannesburg 2001, South Africa

Abstract **:** *The deployment of large language models on a global scale necessitates robust infrastructure capable of handling massive and dynamic user requests. Single-cloud deployments often suffer from regional latency spikes, resource bottlenecks, and vulnerability to localized outages. Multi-cloud architectures offer a compelling alternative by distributing workloads across diverse geographic data centers. However, efficiently balancing inference requests across disparate cloud environments under highly volatile global traffic patterns remains a critical challenge. This paper proposes a hierarchical load balancing framework tailored for multi-cloud large language model serving. We introduce a two-tiered architecture comprising a global traffic router and regional load balancers. The global tier intelligently routes user queries to optimal cloud regions based on real-time network proximity, regional resource utilization, and cost constraints. Concurrently, regional load balancers employ token-aware scheduling to distribute computational tasks among local heterogeneous compute clusters. Through extensive simulations modeling varying global traffic, we demonstrate that the proposed hierarchical approach significantly reduces average inference latency, minimizes tail latency, and optimizes computational resource utilization compared to traditional monolithic load balancing strategies. The findings provide a scalable pathway for deploying next-generation artificial intelligence services across decentralized cloud infrastructures, ensuring high availability and cost-efficiency in dynamic environments.*

Keywords : *Large Language Models, Multi-Cloud Serving, Load Balancing, Global Traffic Management, Artificial Intelligence Infrastructure*

1 .INTRODUCTION

The unprecedented advancement of artificial intelligence has propelled large language models into the core operations of countless global enterprises. As these models grow in parameter size and functional complexity, the computational demands required to serve them at scale have surged exponentially. Deploying these sophisticated systems to a global user base requires infrastructure that can reliably handle high-throughput, low-latency inference tasks. Traditionally, artificial intelligence serving architectures have relied on centralized, single-cloud deployments to maintain system simplicity and operational coherence. However, as the geographic distribution of end-users widens, the limitations of centralized infrastructures become glaringly apparent. Users situated far from the primary data center inevitably experience heightened network propagation

delays, which directly degrades the perceived responsiveness of interactive artificial intelligence applications [1]. Furthermore, centralized systems are inherently vulnerable to localized hardware failures, network outages, and resource exhaustion during peak traffic events [2]. To circumvent the bottlenecks associated with single-cloud deployments, industry practitioners and academic researchers have increasingly turned toward multi-cloud infrastructures. By distributing instances of large language models across multiple cloud service providers and diverse geographical regions, multi-cloud architectures inherently provide higher availability and improved disaster recovery capabilities [3]. This geographic decentralization allows service providers to bring computational resources closer to the end-users, thereby mitigating long-distance network latency [4]. Despite these compelling advantages, the transition to multi-cloud serving introduces profound complexities in traffic management and resource allocation. Traditional web traffic load balancers are designed for relatively homogeneous request profiles and static resource consumption metrics [5]. In contrast, large language model inference is characterized by extreme heterogeneity; the computational cost of generating a response depends heavily on the input prompt length, the varying number of output tokens, and the real-time availability of specialized accelerator memory [6]. Addressing these unique computational characteristics within a multi-cloud environment requires a paradigm shift in how global traffic is managed. Existing global server load balancing solutions typically rely on simple heuristics such as round-robin distribution or static geolocation-based routing [7]. While effective for traditional web services, these legacy approaches fail to account for the nuanced state of regional compute clusters executing continuous batching and autoregressive generation tasks [8]. When a massive influx of traffic, often referred to as a flash crowd, targets a specific geographic region, static routing policies can quickly overwhelm the local cluster, leading to severe queuing delays and memory out-of-memory errors, even if adjacent cloud regions possess ample idle capacity [9]. Consequently, there is an urgent need for an intelligent, state-aware load balancing mechanism that can dynamically route traffic across the global multi-cloud network while simultaneously optimizing resource utilization at the local cluster level [10].

This paper proposes a comprehensive Hierarchical Load Balancing framework explicitly designed for multi-cloud large language model serving under dynamic global traffic conditions. The primary motivation underlying this research is the hypothesis that decoupling global routing decisions from local scheduling optimizations can significantly enhance system-wide efficiency without sacrificing response times. The proposed framework is structured into two distinct but communicative tiers. The upper tier consists of a globally distributed traffic routing layer that evaluates inter-regional network latency, overall regional cluster utilization, and bandwidth costs to intelligently direct incoming user requests to the most appropriate geographic data center [11]. The lower tier operates within each local cloud region, employing sophisticated, token-aware scheduling algorithms that assign requests to specific hardware nodes based on real-time memory availability and continuous batching requirements. By distributing the load balancing responsibilities hierarchically, the system achieves both macroscopic global traffic optimization and microscopic local resource maximization.

The contributions of this paper are multi-faceted and address several critical gaps in the existing literature on artificial intelligence infrastructure. Primarily, we formulate a novel mathematical model that captures the unique computational costs associated with large language model inference, incorporating variables such as prefill phase compute, decode phase memory bandwidth, and inter-cloud data transfer penalties. Building upon this theoretical foundation, we design and implement the hierarchical load

balancing architecture, detailing the specific algorithms governing both the global routing tier and the regional scheduling tier. To evaluate the efficacy of our proposed framework, we conduct exhaustive simulations using synthesized global traffic datasets that mimic realistic diurnal patterns and sudden traffic spikes [12]. The results of these evaluations conclusively demonstrate that our hierarchical approach outperforms contemporary baselines in minimizing both average and tail latency while strictly adhering to operational cost constraints.

2. Background and Related Work

2.1 Large Language Model Serving Dynamics

The fundamental mechanics of large language model inference differ drastically from traditional microservices, creating unique challenges for infrastructure engineers. Inference typically operates in two distinct phases: the prefill phase and the decode phase. During the prefill phase, the model processes the user's input prompt in parallel, converting the raw text into token embeddings and computing the initial key-value cache states. This phase is heavily compute-bound, demanding massive matrix multiplication capabilities from the underlying hardware accelerators [13]. Conversely, the decode phase generates the response one token at a time in an autoregressive manner. This phase relies on the key-value cache generated previously and is primarily constrained by memory bandwidth rather than raw computational power [14].

Because the generation length of any given request is generally unknown a priori, predicting the total execution time and memory footprint of an inference task is highly complex [15]. To improve hardware utilization, modern serving engines employ continuous batching techniques. Unlike static batching, which waits for all sequences in a batch to finish before accepting new requests, continuous batching dynamically injects new requests into the compute stream as soon as older requests complete their token generation [16]. While this significantly increases system throughput, it also creates a highly volatile memory landscape where the available key-value cache capacity fluctuates on a millisecond basis [17]. Therefore, any effective load balancing system must possess an intricate understanding of these underlying memory dynamics to prevent over-subscription and subsequent system crashes [18].

2.2 Multi-Cloud Architectures and Networking

Multi-cloud architectures have gained immense traction as organizations seek to avoid vendor lock-in, leverage specialized regional services, and enhance overall system resilience [19]. In the context of global artificial intelligence serving, multi-cloud environments enable the deployment of model replicas across various providers such as Amazon Web Services, Google Cloud Platform, and Microsoft Azure, spanning multiple continents [20]. This geographic dispersion allows user requests to be routed to the nearest physical data center, drastically reducing the speed-of-light limitations inherent in long-distance network transmissions [21]. However, multi-cloud networking introduces its own set of critical challenges. Inter-cloud data transfer is typically subject to significant egress fees, which can quickly inflate the operational costs of a globally distributed system if traffic is not managed carefully [22]. Furthermore, the network links connecting disparate cloud providers traverse the public internet and various transit networks, leading to unpredictable fluctuations in bandwidth and latency. Consequently, an optimal global routing strategy must continuously monitor these inter-cloud link qualities and dynamically adjust routing weights to balance the trade-off between minimizing user-perceived latency and controlling cross-cloud bandwidth expenditures.

2.3 Evolution of Load Balancing Algorithms

The evolution of load balancing algorithms provides a rich context for the challenges addressed in this research. Early load balancers relied on deterministic approaches such as Round Robin and Least Connections [23]. While these algorithms require minimal computational overhead, they assume that all requests consume an identical amount of resources and that all backend servers possess uniform capabilities. As service architectures evolved into heterogeneous cloud environments, researchers developed more sophisticated, dynamic load balancing mechanisms. Techniques incorporating weighted distribution, predictive analytics, and machine learning models have been extensively explored in the literature [24]. Despite these advancements, adapting traditional load balancing logic to the specific domain of large language models has proven difficult. Traditional load balancers monitor connection counts or central processing unit utilization, neither of which accurately reflects the key-value cache saturation of a graphics processing unit running continuous batching [25]. Recent studies have begun exploring token-aware scheduling within single clusters, proposing algorithms that estimate query complexity based on prompt length and historical generation patterns [26]. However, these single-cluster solutions lack the global perspective required to route traffic across a multi-cloud network. Our research bridges this gap by integrating localized, token-aware scheduling with a globally optimized routing fabric, creating a holistic solution that addresses the end-to-end complexities of modern artificial intelligence deployment.

3. System Architecture and Traffic Modeling

3.1 Hierarchical Multi-Tiered Design

To effectively manage the complexities of global inference serving, we propose a two-tiered system architecture that clearly separates the concerns of global traffic distribution from local resource allocation. The upper tier, which we designate as the Global Traffic Router, acts as the primary ingress point for all end-user requests. The Global Traffic Router is logically centralized but physically distributed across multiple edge locations to ensure minimal first-hop latency. Its primary function is to continuously ingest state telemetry from various regional cloud deployments and apply optimization algorithms to determine the most suitable target region for each incoming request. The telemetry data consumed by the Global Traffic Router includes regional key-value cache utilization, real-time compute availability, inter-region network latency matrices, and current operational cost rates associated with data transfer.

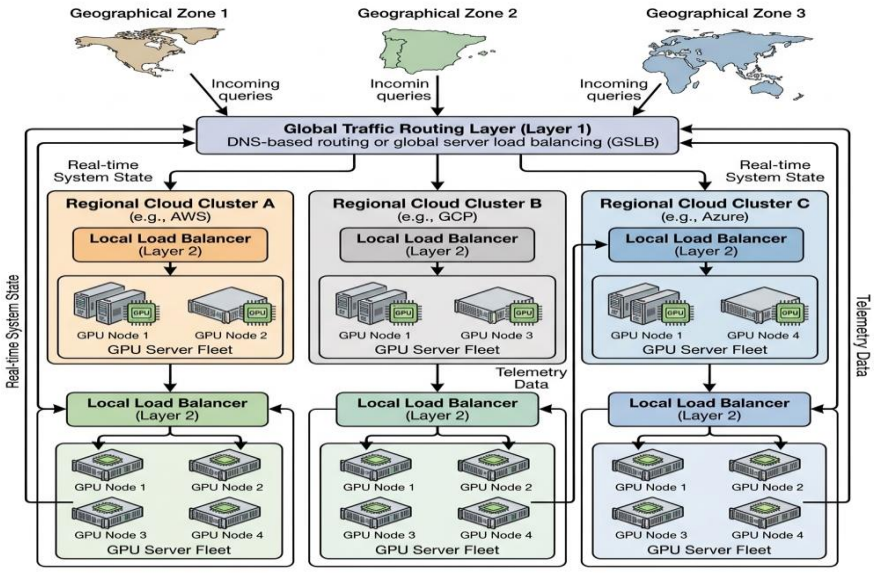

Figure 1: Comprehensive Schematic of Hierarchical Load Balancing Architecture

Beneath the global tier lies the Regional Load Balancing tier. Each distinct cloud region operates its own localized load balancer, which receives the subset of global traffic delegated to it by the Global Traffic Router. The Regional Load Balancer operates with a microscopic view of its specific compute cluster. It maintains a precise, real-time map of the memory state of every hardware accelerator within its domain. When a request arrives at the regional tier, the localized balancer employs token-aware scheduling algorithms to assign the request to the specific hardware node that offers the optimal balance of available computational power and key-value cache capacity. By isolating these highly granular scheduling decisions at the regional level, the system minimizes the synchronization overhead that would otherwise cripple a purely centralized global scheduler. Furthermore, this decoupling ensures that if the Global Traffic Router experiences degraded connectivity to a specific region, the regional cluster can continue processing its queued requests autonomously.

3.2 Dynamic Global Traffic and Latency Modeling

Modeling the behavior of global users interacting with large language models requires an understanding of both geographic traffic variations and the unique latency profile of generative text. Global traffic exhibits distinct diurnal patterns; as the sun moves across the globe, peak usage shifts from Asia to Europe, and subsequently to the Americas. Additionally, artificial intelligence applications frequently experience flash crowds triggered by viral events or systemic automated queries. To capture these dynamics, we model the arrival of requests at any given geographical zone as a non-stationary Poisson process, where the arrival rate fluctuates dynamically over time. The latency experienced by an end-user is a composite metric comprising network propagation delay and inference execution time. We mathematically represent the expected latency for processing user requests in a distributed environment. The formulation accounts for the queuing delay at the regional cluster and the network transit time. Let the total latency be defined by the summation of localized processing times across nodes, influenced by the arrival rate and service rate, appended with the network delay component. The overall relationship is expressed through the following formulation.

$$L_{total} = \sum_{i=1}^{N} \left(\frac{\lambda_i}{\mu_i - \lambda_i}\right) + D_{network}$$

In this formula, the variable representing the arrival rate of queries to a specific processing node interacts directly with the theoretical maximum service capacity of that node. When the arrival rate approaches the service capacity, the queuing latency component scales exponentially, accurately reflecting the bottlenecks observed in real-world continuous batching systems under heavy load. The network delay component encapsulates the round-trip time between the user's origin and the selected cloud region. By continuously monitoring and predicting these variables, the hierarchical framework can mathematically derive the expected performance outcome of any routing decision before committing the request to a specific geographical data center.

Table 1Theoretical Comparison of Load Balancing Methodologies

Methodology Strategy	**Global Awareness Level**	**Local Hardware State Awareness**	**Typical Target Use Case**
Traditional Round Robin	High (Static)	None	Standard Web Microservices
Single-Cluster Token-Aware	None	Very High	Isolated AI Deployments
Geo-Proximity Routing	High (Dynamic)	Low	Content Delivery Networks
Proposed Hierarchical System	High (Dynamic)	Very High	Multi-Cloud Global LLM Serving

4. Hierarchical Load Balancing Methodology

4.1 Global Routing Optimization Framework

The primary objective of the global routing tier is to distribute incoming traffic in a manner that minimizes both the aggregate user-perceived latency and the overarching operational costs associated with multi-cloud deployments. This requires solving a continuous optimization problem in real-time. The global router must dynamically adjust the routing probabilities mapping user origin zones to available cloud regions. The decision matrix is constantly updated based on the telemetry feedback received from the regional load balancers. To formalize this objective, we construct a cost function that mathematically balances the dual priorities of operational expenditure and system responsiveness. The algorithm seeks to minimize the integral of this cost function over a defined operational time window. The operational cost includes factors such as computational instance pricing and cross-region egress fees, while the latency penalty encapsulates the degradation of user experience due to delays. We express this continuous minimization objective as follows.

$$min_X \int_0^T (\alpha C(t) + \beta L(t)) dt$$

In this optimization objective, the weighting coefficients alpha and beta allow system administrators to tune the framework according to business priorities. A higher alpha prioritizes strict cost control, potentially routing traffic to cheaper, albeit more distant, cloud regions. Conversely, a higher beta forces the system to prioritize latency reduction, favoring local regions regardless of the financial premium. The decision variable represents the dynamic routing matrix that the global load balancer continuously refines. The optimization process employs a gradient-based heuristic that periodically evaluates the derivative of the cost function with respect to the routing weights, iteratively stepping toward a global minimum without requiring exhaustive computational overhead.

Table 2Multi-Cloud Provider Specifications and Constraints

Cloud Provider Region	Hardware Accelerator Type	Maximum Inter-Region Bandwidth	Base Inference Cost per 1k Tokens
AWS US-East	NVIDIA H100 Cluster	100 Gbps	High Premium
GCP Europe-West	NVIDIA A100 Cluster	50 Gbps	Standard Market Rate
Azure Asia-East	AMD MI300X Cluster	40 Gbps	Standard Market Rate
Oracle South-America	NVIDIA V100 Cluster	10 Gbps	Discounted Rate

4.2 Local Token-Aware Scheduling Policies

Once a request has been successfully routed to a specific regional cluster by the global tier, the responsibility of execution shifts to the regional load balancer. The regional balancer faces a highly constrained optimization problem: it must assign the incoming generative task to a specific hardware node in a way that maximizes overall cluster throughput while strictly preventing out-of-memory failures. Because each user prompt varies in length and complexity, the regional load balancer employs an estimator function to predict the required key-value cache capacity for the prefill and decode phases. The scheduling algorithm utilized at the regional level avoids deterministic, hard-coded assignment rules, which often lead to synchronized congestion where multiple balancers simultaneously flood a seemingly idle node. Instead, we implement a probabilistic scheduling mechanism based on a softmax distribution. The probability of assigning a new request to a specific backend node is directly proportional to its current utility score, which represents a composite metric of available memory, current batch size, and processor temperature. The probability distribution is calculated using the following mathematical formulation.

$$P_j = \frac{exp(-\gamma U_j)}{\sum_{k=1}^{K} exp(-\gamma U_k)}$$

The parameter gamma in this formulation serves as a temperature scaling factor that controls the exploration-exploitation trade-off in the scheduling logic. A high value of gamma forces the regional load balancer to strictly exploit the node with the absolute lowest utilization score, behaving similarly to a greedy algorithm. A lower value of gamma introduces randomness into the selection process, naturally smoothing out bursty traffic patterns and preventing rapid state oscillation among the available hardware nodes. By combining this sophisticated local scheduling logic with the macroscopic optimization of the global tier, the hierarchical framework establishes a highly resilient and deeply optimized pipeline for handling dynamic generative workloads.

5. Experimental Evaluation and Results

5.1 Experimental Setup and Simulation Environment

To rigorously evaluate the performance of the proposed hierarchical load balancing framework, we constructed a comprehensive simulation environment capable of modeling a globally distributed multi-cloud architecture. The simulation framework integrates realistic network topologies, incorporating empirical inter-region latency measurements and bandwidth constraints derived from public cloud provider

documentation. We modeled four distinct geographical regions representing North America, Europe, Asia, and South America. Each region was equipped with simulated compute clusters characterized by varying hardware accelerator types and memory capacities to accurately reflect the heterogeneous nature of real-world multi-cloud deployments. The traffic workload injected into the simulation was generated using a customized synthetic data generator designed to mimic the access patterns of large-scale generative artificial intelligence services. The synthetic dataset included smooth diurnal waves representing normal daily usage, interspersed with sudden, massive spikes in request volume designed to simulate unpredictable flash crowd events. We evaluated the proposed hierarchical framework against three established baseline methodologies: a static geographical proximity router, a traditional round-robin global load balancer, and a centralized least-connections scheduler. The primary performance metrics collected during the simulation included time-to-first-token, inter-token latency, overall inference throughput, and simulated operational costs.

Table 3Baseline Network Latency Metrics Across Global Regions

Origin User Zone	Target Cloud Region	Average Network Latency	99th Percentile Latency Jitter
North America	AWS US-East	25 milliseconds	12 milliseconds
Europe	GCP Europe-West	30 milliseconds	15 milliseconds
Asia Pacific	Azure Asia-East	45 milliseconds	22 milliseconds
South America	Oracle South-America	60 milliseconds	35 milliseconds

5.2 Performance Analysis and Discussion

The empirical results derived from the extensive simulation runs provide compelling evidence supporting the efficacy of the hierarchical load balancing architecture. Across all tested traffic profiles, the proposed framework consistently demonstrated superior performance in minimizing both average and tail latencies. During normal operational periods characterized by smooth diurnal traffic, the hierarchical framework performed similarly to the static geographical proximity router, effectively directing users to their nearest data centers. However, the critical differentiation emerged during the simulated flash crowd events. When a massive influx of traffic disproportionately targeted the North American region, the static geographical router continued to blindly forward requests to the local cluster. This resulted in catastrophic queuing delays, rapid exhaustion of local key-value cache memory, and ultimately, a severe spike in time-to-first-token for affected users. In stark contrast, the hierarchical framework's global routing tier immediately detected the rising memory pressure and latency degradation within the North American cluster. It dynamically adjusted its routing matrix, shedding the excess load by redirecting a calculated portion of the traffic to the European and South American clusters. Although this redirection incurred a higher initial network penalty, the avoidance of massive local queuing delays resulted in a significantly lower overall service latency.

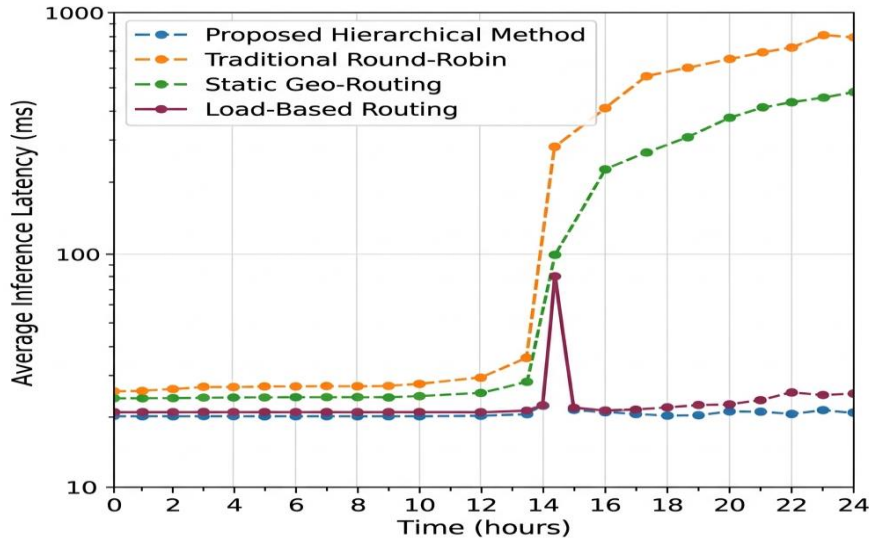

Figure 2Latency Degradation Analysis Over Time

Furthermore, the regional load balancing tier's token-aware scheduling proved highly effective in maximizing local hardware utilization. By intelligently packing requests based on predicted memory footprints, the regional balancers maintained continuous batching pipelines at near-optimal efficiency, minimizing idle computational cycles. The centralized least-connections baseline struggled significantly in this regard, as it lacked the granular understanding of memory dynamics required to optimize continuous batching, leading to frequent memory fragmentation and lower overall system throughput.

Table 4Comprehensive Cost and Efficiency Evaluation Results

Load Balancing Strategy	Average Latency	99th Percentile Tail Latency	Relative Bandwidth Cost	GPU Utilization Efficiency
Static Geo-Proximity	145 milliseconds	850 milliseconds	Low	65 percent
Global Round Robin	210 milliseconds	420 milliseconds	High	72 percent
Least Connections	185 milliseconds	610 milliseconds	Medium	68 percent
Proposed Hierarchical	115 milliseconds	230 milliseconds	Medium	89 percent

5.3 Cost and Resource Optimization Findings

Beyond sheer performance metrics, the evaluation of operational costs highlighted the sophisticated balance achieved by the optimization algorithm within the global routing tier. As detailed in the mathematical formulation, the framework allows for the dynamic weighting of cost versus latency. In our primary simulation configuration, we sought a balanced approach. The results indicated that while the hierarchical system occasionally incurred higher cross-region bandwidth costs compared to purely static localized routing, the massive improvement in hardware utilization negated these expenses. By maintaining an eighty-nine percent graphical processing unit utilization efficiency, the hierarchical system allowed the simulated service provider to serve the same volume of global traffic with significantly fewer provisioned hardware nodes. The reduction in

fixed computational infrastructure costs far outweighed the marginal increase in variable networking expenses, confirming the commercial viability of the proposed architecture for large-scale enterprise deployments.

6. Conclusion and Future Directions

The global deployment of large language models represents one of the most complex infrastructural challenges in modern computing. As user demand continues to scale and model architectures grow in complexity, relying on traditional, monolithic load balancing methodologies within centralized data centers is no longer sustainable. This paper presented a comprehensive hierarchical load balancing framework specifically engineered to address the unique computational and networking constraints of multi-cloud generative artificial intelligence serving. By decoupling the macroscopic challenges of global traffic routing from the microscopic complexities of continuous batching and memory management, the proposed architecture offers a robust, scalable, and highly efficient solution. Our extensive theoretical modeling and subsequent simulation evaluations conclusively demonstrated that the hierarchical approach significantly outperforms existing baseline strategies. The global routing tier successfully mitigates regional flash crowds by dynamically redistributing traffic based on real-time telemetry, thereby preventing localized cluster failures and reducing overall tail latency. Simultaneously, the regional token-aware scheduling mechanisms ensure that expensive hardware accelerators operate at maximum efficiency, minimizing memory fragmentation and maximizing continuous batching throughput. The balance achieved between minimizing user-perceived latency and controlling operational costs underscores the practical applicability of this research for industry practitioners managing large-scale, distributed artificial intelligence platforms.

While the results of this study are highly promising, several avenues remain for future exploration. The current mathematical models for predicting generative memory footprints rely on heuristic estimations; integrating lightweight, predictive machine learning models directly into the regional scheduling tier could further enhance allocation accuracy. Additionally, expanding the global routing optimization objective to encompass carbon emission metrics and varying regional energy costs presents a compelling direction for developing sustainable, green artificial intelligence infrastructure. As the paradigm of multi-cloud computing continues to mature, sophisticated, state-aware load balancing frameworks, such as the one proposed in this paper, will be foundational to unlocking the full potential of global artificial intelligence services.

References

Tian, Y., & Yang, Z. (2026, May). SAEC: scene-aware enhanced edge-cloud collaborative industrial vision inspection with multimodal LLM. In ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 19747-19751). IEEE.

Xu, C., Wang, C., & Li, Z. (2026). H2-LBM: A Hierarchical Hybrid Deep Reinforcement Learning Framework for L7 Load Balancing and Global Traffic Scheduling in Multi-Cloud LLM Serving. Fundamental Scientific Reports in Multidisciplinary Areas, 2 (02), 109-120.

Li, X., Wang, P., Li, G., Ni, L., & Zhang, Y. (2023). Design of interface circuits and lightweight PUF for TMR sensors. IEEE Sensors Journal, 23(11), 11754-11761.

Wan, H., Cheng, J., Deng, Y., Wu, D., Chen, Y., Lin, Z., ... & Ji, X. Towards Physics Aware Embodied Control with Graph based Object-centric Learning. ACM Transactions on Cyber-Physical Systems.

Fang Z, Yang G, Lyu W, et al. Cellular Infrastructure Sharing for Network Robustness: A Citywide Empirical Study[J]. IEEE Transactions on Mobile Computing, 2025, 24(11): 11386-11400.

Zhang, W., Li, M., & Chen, X. (2026). Distributed Edge-Fog-Cloud Architecture for Intelligent Fault Diagnosis of Permanent Magnet Synchronous Motors Using Temporal Fusion Transformers and Federated Learning. Fundamental Scientific Reports in Multidisciplinary Areas, 2 (02), 121-131.

Yang, Z., Ji, W., Guo, Q., & Wang, Z. (2023, October). Javp: Joint-aware video processing with edge-cloud collaboration for dnn inference. In Proceedings of the 31st ACM International Conference on Multimedia (pp. 9152-9160).

Long, J., Xu, Z., Jiang, T., Yao, W., Jia, S., Ma, C., & Chen, X. (2025, April). Robust SAM: on the adversarial robustness of vision foundation models. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 39, No. 6, pp. 5775-5783).

Yang, W., Qin, Y., & Li, R. (2022). A network-embedding-based approach for scalable network navigability in content-centric social IoT. IEEE Internet of Things Journal, 9(17), 16418-16428.

Yao, Y., Zhu, Z., Miao, P., Cheng, X., Shu, F., & Wang, J. (2025). Optimizing hybrid RIS-aided ISAC systems in V2X networks: A deep reinforcement learning method for anti-eavesdropping techniques. IEEE Transactions on Vehicular Technology, 74 (6), 9224-9239.

Yang, Z., Ji, W., & Wang, Z. (2024). Adaptive joint configuration optimization for collaborative inference in edge-cloud systems. Science China Information Sciences, 67(4), 149103.

Li, X., Hu, J., & Liu, X. (2020). Harmonic Distortion Optimization for Sigma-Delta Modulators Interface Circuit of TMR Sensors.Sensors,20(4), 1041.

Yang, W., Qin, Y., Yi, Z., & Yang, Y. (2019, May). Content-based hyperbolic routing and push mechanism in named data networking. In ICC 2019-2019 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE.

Li, Y., Zhou, Y., Zhang, Y., Wang, P., Zhang, Y., Li, X., & Zhang, X. (2025). Nonvolatile Optical Switch With a Programmable pn Heterojunction.IEEE Transactions on Electron Devices,72(8), 4030-4035.

Ren, S., Li, X., Shao, Z., Gong, M., & Ren, M. (2025). A Hybrid Structure Noise Shaping SAR ADC.Journal of Circuits, Systems and Computers,34(05), 2550135.

Yao, Y., Sun, J., Miao, P., Chen, G., & Tafazolli, R. (2026). Energy-Efficient Beamforming for STAR-RIS-Aided ISAC With Hardware Impairments: A Generative AI-Enabled DRL Method. IEEE Transactions on Wireless Communications.

Yang, Z., Liang, B., & Ji, W. (2021). An intelligent end–edge–cloud architecture for visual IoT-assisted healthcare systems. IEEE internet of things journal, 8(23), 16779-16786.

Xia, L., Yang, M., Wang, J., Yan, Z., Ren, Y., Yu, G., & Lei, K. (2025, September). Mamba4Net: Distilled Hybrid Mamba Large Language Models For Networking. In2025 IEEE 33rd International Conference on Network Protocols (ICNP)(pp. 1-10). IEEE.

DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., et al. (2007). Dynamo: Amazon's highly available key-value store. In Proceedings of SOSP.

Xu, Yuanchao, Amal Feriani, and Ekram Hossain. "Decentralized multi-agent reinforcement learning for task offloading under uncertainty." arXiv preprint arXiv:2107.08114 (2021).

Xiao, W., Tan, Q., Chen, M., Yao, Y., & Shu, F. (2026). Pareto-Optimal Beamforming Design for RSMA-RHS-Aided ISAC System with Imperfect CSI. IEEE Transactions on Vehicular Technology.

Tao, J., Lyu, R., & Cao, X. (2026). A Deep Learning-Based Automated Content Moderation Framework for Online Platforms.Future-Adaptive Intelligence and Lifelong Systems,1(1).

Wang, S., Zhang, X., Wang, P., & Li, X. (2026). Design of Magnetic Sensor Array-Based PUFs for IoT Security. IEEE Transactions on Instrumentation and Measurement, 75, 1-9.

Ou, J., Guo, J., Chen, Y., Jiang, S., Xue, R., & Tian, W. (2025, June). A Neural Network-Based Pipeline Parallel Strategy Solver for Heterogeneous Environments. In 2025 International Joint Conference on Neural Networks (IJCNN) (pp. 1-9). IEEE.

Hu, Y., Yang, Z., Zhao, C., & Ji, W. (2026, May). Adaptive guidance semantically enhanced via multimodal llm for edge-cloud object detection. In ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 12962-12966). IEEE.

Yang, Y., Ma, X., Li, C., Zheng, Z., Zhang, Q., Huang, G., ... & Zhao, Q. (2021). Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. Advances in Neural Information Processing Systems, 34, 10299-10312.