Recent Research on Cloud-Edge Computing for LLM

Heejae Park, Seungyeop Song, Seongryool Wee, and Laihyuk Park
Department of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul, 01811, Korea
Email: {prkhj98, sysong, holylaw, lhpark}@seoultech.ac.kr

***Abstract*—Large Language Models (LLMs) offer remarkable capabilities across various natural language processing (NLP) tasks. However, their high computational complexity and significant memory requirements during the training and inference stages hinder their practical application. Cloud-edge computing has emerged as a promising paradigm to overcome these limitations by enabling the collaborative execution of LLM inference across cloud and edge servers. This paper surveys recent research on cloud-edge collaborative frameworks for LLMs, with a focus on system architecture, optimization objectives, and learning-based offloading strategies.**

***Index Terms*—LLM, cloud-edge computing, task offloading.**

I. Introduction

Large-scale language models (LLMs) have attracted significant attention due to their outstanding performance in natural language understanding and generation [1], [2]. However, their high computational complexity and significant memory requirements during the training and inference stages hinder their practical application. These limitations make it difficult to efficiently deliver LLM services on resource-constrained devices [3].

To overcome these limitations, cloud-edge computing has emerged as an active research area, aiming to combine the powerful computational capabilities of the cloud with the low-latency and local processing capabilities of edge servers [4]. This collaborative architecture enables dynamic workload distribution and real-time service delivery.

To provide valuable perspectives for future research, this paper explores edge-cloud collaboration strategies for LLM services, focusing on offloading decision-making and resource allocation considering latency, energy efficiency, and quality of service.

II. Background

A. Large Language Model

LLMs have revolutionized the field of artificial intelligence, particularly in natural language processing (NLP). Most LLMs are based on the Transformer architecture, which employs a self-attention mechanism [5]. The self-attention operation is mathematically expressed as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \qquad (1)$$

where Q, K, and V are the query, key, and value matrices, respectively, and d_k is the dimensionality of the key.

The adaptability of LLMs makes them suitable for various industries and use cases, but this versatility comes at a cost. These models demand significant computational and memory resources, especially during inference, which involves processing user inputs to generate intelligent responses.

B. Cloud-Edge Computing

Cloud-edge computing is a collaborative architectural paradigm that combines the high-capacity computational infrastructure of cloud servers with the low computation latency and proximity advantages of edge servers. Cloud servers possess extensive processing power and memory, making them suitable for handling large-scale and compute-intensive tasks. In contrast, edge servers are deployed closer to end-users, allowing for faster response times and reduced transmission delays. By offloading the LLM tasks to the cloud or edge servers, it can reduce the computational load on local devices. However, this approach introduces additional communication delay.

III. Research Trends

The authors in [6] propose a cloud-edge collaborative framework to support the efficient deployment of LLMs. The system consists of multiple LLM users, cloud computing center, and edge server. The objective of this study is to determine an optimal offloading policy for large-model inference tasks, considering limited computational resources at the endpoint and leveraging edge or cloud infrastructures. To solve this problem, authors adopt an active inference framework, rooted in Bayesian inference and variational free energy minimization. Unlike traditional reinforcement learning, the system uses a Partially Observable Markov Decision Process (POMDP) model and minimizes variational free energy to decide optimal actions for LLM deployment and resource management. The state space consists of remaining computing resource, the remaining bandwidth resource, and the remaining graphics memory resource. The action consists of offloading the LLM inference task to a selected server, along with allocating computational resources, channel bandwidth, and graphics memory required for task execution.

The authors in [7] propose EdgeShard, a collaborative inference framework designed to efficiently run LLMs across edge devices. The system model assumes a layered LLM architecture with N layers, where each layer has an activation

size and memory requirement. The network consists of interconnected devices, including edge devices and more powerful cloud servers. EdgeShard operates in three stages: profiling, scheduling optimization, and collaborative inference. In the profiling stage, the system gathers layer-wise execution time, memory usage, and bandwidth information using dynamic model loading to accommodate memory-limited devices. The scheduling stage generates a device-aware model partition plan, assigning layers to edge devices based on resource availability and system constraints. Finally, in the collaborative inference stage, devices execute the model with preallocated KV-cache memory to ensure efficient and accurate output. Both sequential and pipeline parallel inference are supported. The authors formulate the problem of minimizing LLM inference latency and optimizing throughput, and solve it using dynamic programming algorithm.

The authors in [8] introduce PerLLM, a personalized inference scheduling framework designed to optimize LLM services through edge-cloud collaboration. They formulate multi-objective optimization task that aims to minimize the total energy cost of LLM inference across edge-cloud infrastructure, including transmission, inference, and idle energy. The problem considers latency constraint, bandwidth constraint, computing constraint, and assignment constraint. The latency constraint ensures timely service completion from the user's perspective, whereas bandwidth and computing power constraints reflect the limitations of available resources. Assignment constraint guarantees only one server can be chosen for each service. To solve the scheduling problem under multiple constraints, the authors model it as a Combinatorial Multi-Armed Bandit (CMAB) problem. They propose a novel algorithm called Constraint Satisfaction Upper Confidence Bound (CS-UCB), which selects service-to-server assignments by maximizing expected reward while satisfying processing time, bandwidth, and computation constraints. The state space consists of the current computing and bandwidth resources of each server, while the action space involves assigning each service to a specific server.

The authors in [9] propose a cloud–edge collaborative framework for enabling multimodal LLM (MLLM)-based Advanced Driver Assistance Systems (ADAS) in IoT-enabled vehicular networks. State-of-the-art MLLM model (CogVLM2) is deployed at the edge, while ChatGPT-4o is deployed at the cloud. Authors used the BDD-X dataset to fine-tune the CogVLM2 model and leveraged few-shot learning to enhance the performance of ChatGPT-4o. In addition, they formulate three models: 1) service latency model, 2) energy consumption model, and 3) QoS (Quality of Service) model. The service latency model considers upload latency, inference latency, and download latency. The energy consumption model considers inference energy consumption and communication energy consumption. The QoS model computes the ADAS task success rate of CogVLM2 and ChatGPT-4o. Based on these models, authors formulate the utility function to minimize service latency and service energy consumption while improving QoS. To solve this function, Deep Deterministic Policy Gradient (DDPG) based solution is proposed. The state space of DDPG is computational load, the data size of task and result, remaining computational resources of the local device and edge servers, latency, required energy, and distance between nodes. The action space consists of where to offload the task.

IV. Conclusion

This paper examines the background of LLM and cloud-edge computing. In addition, we analyze the system model, optimization problem, and solution proposed in recent research. Future work could focus on designing graph neural network (GNN)-based task offloading in the Cloud-Edge environment, which can effectively capture the relationships between communication/computing nodes.

Acknowledgment

This work was partly supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2025-16070295) and IITP (Institute for Information & Communications Technology Planning & Evaluation) grant funded by the Korea government(MSIT) (No.RS-2024-00437252, Development of anti-sniffing technology in mobile communication and AirGap environments).

References

[1] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang *et al.*, "A survey on evaluation of large language models," *ACM transactions on intelligent systems and technology*, vol. 15, no. 3, pp. 1–45, 2024.

[2] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, "A comprehensive overview of large language models," *ACM Transactions on Intelligent Systems and Technology*, 2023.

[3] Y. Shen, J. Shao, X. Zhang, Z. Lin, H. Pan, D. Li, J. Zhang, and K. B. Letaief, "Large language models empowered autonomous edge ai for connected intelligence," *IEEE Communications Magazine*, vol. 62, no. 10, pp. 140–146, 2024.

[4] Y. Zheng, Y. Chen, B. Qian, X. Shi, Y. Shu, and J. Chen, "A review on edge large language models: Design, execution, and applications," *ACM Computing Surveys*, vol. 57, no. 8, pp. 1–35, 2025.

[5] H. Zhou, C. Hu, Y. Yuan, Y. Cui, Y. Jin, C. Chen, H. Wu, D. Yuan, L. Jiang, D. Wu *et al.*, "Large language model (llm) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities," *IEEE Communications Surveys & Tutorials*, 2024.

[6] Y. He, J. Fang, F. R. Yu, and V. C. Leung, "Large language models (llms) inference offloading and resource allocation in cloud-edge computing: An active inference approach," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 11 253–11 264, 2024.

[7] M. Zhang, X. Shen, J. Cao, Z. Cui, and S. Jiang, "Edgeshard: Efficient llm inference via collaborative edge computing," *IEEE Internet of Things Journal*, 2024.

[8] Z. Yang, Y. Yang, C. Zhao, Q. Guo, W. He, and W. Ji, "Perllm: Personalized inference scheduling with edge-cloud collaboration for diverse llm services," *arXiv preprint arXiv:2405.14636*, 2024.

[9] Y. Hu, D. Ye, J. Kang, M. Wu, and R. Yu, "A cloud–edge collaborative architecture for multimodal llm-based advanced driver assistance systems in iot networks," *IEEE Internet of Things Journal*, vol. 12, no. 10, pp. 13 208–13 221, 2025.