2025 IEEE International Conference on Edge Computing and Communications (EDGE) | 979-8-3315-5559-7/25/$31.00 ©2025 IEEE | DOI: 10.1109/EDGE67623.2025.00019

Enabling Fairness Across Multi-modal and Multi-agent Applications

Rui Zhang
Department of Computer Science and Engineering
University of California, Santa Cruz
Santa Cruz, USA
rzhan229@ucsc.edu

Liting Hu
Department of Computer Science and Engineering
University of California, Santa Cruz
Santa Cruz, USA
lhu82@ucsc.edu

***Abstract*—Modern multi-agent systems leverage a diverse set of AI models, including Large Language Models (LLMs), Vision-Language Models (VLMs), etc., to perform complex multi-modal tasks. However, fair model serving in such heterogeneous environments remains a significant challenge. Existing scheduling methods primarily focus on single-modality fairness, failing to account for varying computational costs across different models and the hierarchical structure of multi-agent applications. In this work, we introduce Hierarchical Multi-Modality Fair Scheduling (HMFS), a novel approach that ensures fairness across applications, agents, and tasks while maintaining high resource utilization.**

To enable cross-modality fairness, we propose a Unified Token Representation, which normalizes token costs across different transformer-based models by leveraging latent space embedding dimensions and computational intensity factors. Using this unified metric, we design a Hierarchical Multi-Modality Fair Scheduling algorithm that dynamically prioritizes requests at both application and agent levels, ensuring equitable access to compute resources.

***Index Terms*—Fairness, Scheduling, Transformer, Edge-Cloud Communication, Multi-agent, Multi-modality**

I. Introduction

Modern applications increasingly rely on multi-agent systems to tackle complex, distributed tasks through agent collaboration. These systems are widely deployed in edge computing environments, such as autonomous driving [1, 2], robotics [3, 4], and content generation [5]. For instance, in autonomous driving, multiple agents—including perception, planning, and control modules—collaborate in real-time to process sensor data, make driving decisions, and execute controls, all within the constraints of low-latency edge inference.

Recently, the integration of foundational AI models such as LLMs and VLMs has significantly transformed multi-agent edge computing systems. Instead of relying on rule-based policies, agents can now offload reasoning and decision-making to pre-trained AI models [6, 7, 8, 9]. However, this shift introduces new challenges for model inference, as edge-based systems must now handle heterogeneous requests with varying model demands, dynamic resource constraints, and strict SLO requirements. While existing model-serving frameworks optimize throughput and latency, fairness in multi-modal, multi-agent inference remains an open challenge.

Fair scheduling for transformer-based model serving faces key challenges. First, output lengths are unpredictable, making it difficult to estimate resource utilization before execution. Second, server processing rates vary over time, meaning resource availability fluctuates dynamically. Token-based fairness mechanisms, such as Virtual Token Counter (VTC) scheduling [10], address these issues by allocating service based on token consumption, eliminating the need for predefined output lengths or fixed resource capacity. However, these methods are designed for single-modality LLM inference and do not extend to multi-modal, multi-agent edge workloads, where requests span text, images, and video generation.

Applying the fairness scheduling in LLM serving to multi-modal systems can be challenging. First, the input and output have diverse types instead of text tokens which means that we need a new abstraction towards the input and output. Second, for different modality, the input and output cost are not the same. For example, in order to generate an image, it is way more expensive than generating a single word. Third, Even if we achieve fairness between applications, since each application is consist of multiple agents, they can also introduce unfairness within an application.

To tackle the challenges, we propose Hierarchical Multi-modal Fairness Scheduling (HMFS), a novel fairness-aware scheduling mechanism designed for multi-modal, multi-agent model serving which can hierarchically allocate fairness across different models, agents, and applications. For the challenges introduced by the variety of models, we extend the concept of token and generalize the fairness across different types of transformer-based models by abstracting their token-processing characteristics into a universal fairness metric. For the multi-agent unfairness problem, we introduce a hierarchical approach that enforces fairness at both application and client levels.

The key contributions of this paper are:

- A Unified Token-Based Fairness Model: We propose a unified fairness model can generalize to the different modality.
- Hierarchical Weighted Fairness Mechanism: Based on the extended concept of token, we design a new algorithm that can hierarchically achieve fairness across different levels.

2767-9918/25/$31.00 ©2025 IEEE
DOI 10.1109/EDGE67623.2025.00019

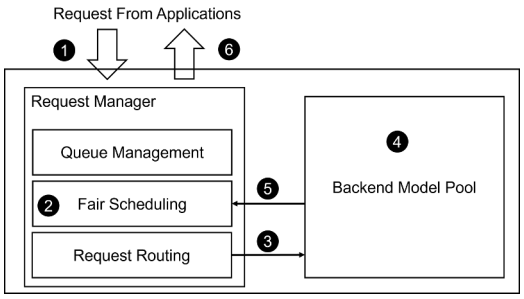

Fig. 1. System Overview. 1. Agents in the edge submit requests. 2. The request manager tracks each request's token usage and selects the next request using hierarchal fairness scheduling. 3. The request is routed to the associated model instance in the model pool. 4. The model processes the request and generates output iteratively. 5. The model reports intermediate token usage to the request manager. 6. The final output is returned to the requesting agent.

II. Background

A. Transformer-based Model Serving

As the transformer-based LLM model succeed in handling text related task and become the main stream of Language model family tree, more and more models in other discipline like image generation, text generation and audio generation are developed using the transformer architecture. These models utilize a self-attention mechanism to efficiently process sequences, making them particularly well-suited for autoregressive generation, where output tokens are generated sequentially. Since decoding is sequential and each output token depends on all prior tokens, inference time varies significantly across requests, making fair scheduling a critical challenge.

B. Fairness in Model Serving

Achieving efficient multi-modal model inference with Service Level Objective (SLO) guarantees requires fairness among different applications in a shared serving infrastructure. In a multi-agent system, different agents request heterogeneous models, competing for limited GPU resources. Without fairness mechanisms, some agents or clients may monopolize resources, leading to starvation of others. The Virtual Token Counter (VTC) algorithm[10] was the first fair scheduling approach designed for continuous batching in LLM serving. VTC maintains a virtual counter of tokens processed per client and prioritizes clients with the lowest counters in each batching iteration. However, VTC was designed for single-modality LLM inference and does not extend to multi-agent, multi-modal environments. It does not differentiate between LLMs, VLMs and other models, each of which has different token costs. Also, it does not account for multi-agent interactions, where fairness must be ensured across multiple levels.

III. Design

In this section, we first introduce the overall design of our system. Then we will illustrate our key techniques and our proposed scheduling algorithm.

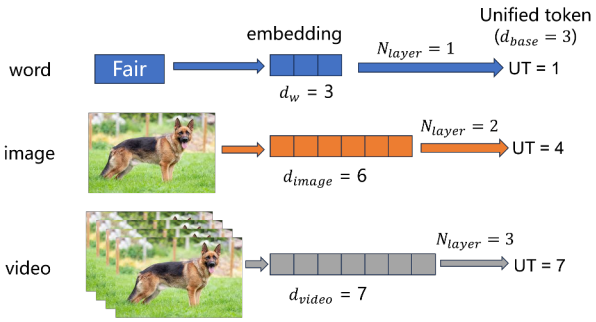

Fig. 2. Unified token count for different modalities

A. System Overview

Our system is designed to provide fair scheduling across multiple applications with different modalities, addressing the challenges of heterogeneous workload and enable fairness across various transformer-based models. Fig. 1 provides a high-level illustration of our system design.

B. Unified Token

Efficient scheduling in multi-modal, multi-agent model serving requires a unified metric for resource consumption across diverse models. Traditional token-based fairness mechanisms like VTC work for LLMs but fail in multi-modal settings, where LLMs, VLMs, etc. have varying computational costs per token. Raw token counts unfairly prioritize low-cost models, causing resource starvation. We propose the Unified Token (UT) representation, which normalizes token costs based on latent space embeddings and computational intensity. This ensures fair scheduling across models, aligning with transformer-based computation and memory usage.. We define the Unified Token Count $T(n, \tau_r)$ for a request r with n raw tokens as:

$$T(n, \tau_r) = n * \frac{d_{\tau_r}}{d_{base}} * N_{\tau_r}$$

where τ_r represents the model type of the request r, d_{τ_r} is the embedding dimension of model τ_r, d_{base} is a reference embedding dimension as scale factor, N_{τ_r} is the layer numbers of the models.

This formulation ensures that tokens from different models are weighted appropriately, that fairness is enforced across heterogeneous workloads. By defining fairness at the latent space level, we ensure that multi-modal workloads are scheduled in a unified, resource-aware manner, preventing unfair prioritization of low-cost tokens while maintaining efficient GPU utilization. Fig.2 illustrates the core idea of the unified token count.

C. Hierarchical Fair scheduling

Enabling fairness in a multi-agent, multi-modal model serving environment requires balancing resource at multiple levels. Traditional fairness methods, such as token-based scheduling,

Algorithm 1 Hierarchical Multi-modality Fair Scheduling

Input: Request trace, input token weight w_p, output token weight w_q, unified token f_{UT}.

▷ Initialize fairness counters:

1: $C_a \leftarrow 0$ for each application a
2: $CT_{a,t} \leftarrow 0$ for each agent t in application a
3: Let Q denote the waiting queue, dynamically changing.

▷ Monitoring Stream for Incoming Requests:

4: **while** True **do**
5: **if** new request r from application u arrives **then**
6: **if** $\nexists r' \in Q, app(r') = u$ **then**
7: **if** $Q = \emptyset$ **then**
8: Let $l \leftarrow$ last application left Q
9: $C_a[u] \leftarrow \max\{C_a[u], C_a[l]\}$
10: **else**
11: $P \leftarrow \{i \mid \exists r' \in Q, app(r') = i\}$
12: $C_a[u] \leftarrow \max\{C_a[u], \min\{C_a[i] \mid i \in P\}\}$
13: **end if**
14: $CT_{a,t}[u][t] \leftarrow C_a[u]$ for each task in u
15: **end if**
16: $Q \leftarrow Q + r$
17: **end if**
18: **end while**

▷ Processing New Requests:

19: **if** add_new_request() **then**
20: $a^* \leftarrow \arg\min C_a$
21: $t^* \leftarrow \arg\min CT_{a^*}$
22: Let r be the earliest request in Q from a^*, t^*
23: $C_a[a^*] \leftarrow C_a[a^*] + w_p * T(n_{input}, \tau_r)$
24: $CT_a[a^*][t^*] \leftarrow CT_a[a^*][t^*] + w_p * T(n_{input}, \tau_r)$
25: $Q \leftarrow Q - r$
26: **end if**

▷ Asynchronous Fairness Token Update:

27: $C_a[i] \leftarrow C_a[i] + w_q * \sum_{r|app(r)=i} T(n_{output}, \tau_r)$
28: $CT_{a,t}[i][j] \leftarrow CT_{a,t}[i][j] + w_q * \sum_r T(n_{output}, \tau_r)$ for $app(r) = i, task(r) = j$

focus on individual request fairness but fail to address hierarchical fairness across applications and agents. To address these challenges, we propose a Hierarchical Multi-Modality Fair Scheduling algorithm, leveraging the Unified Token (UT) representation to ensure fairness across heterogeneous model workloads. Algorithm 1 describes the procedure.

The scheduling algorithm operates in two streams. The first one is monitoring stream(line 4-18) which monitor the incoming requests and dynamically add to the waiting queue Q. Each request is associated with an application a and an agent t within that application. Then the scheduler maintains two fairness counters over application and agent levels with initialization using counter lift describing in the paper(citation). The second stream(line 19-26) is processing stream which selects the earliest request with lowest multi-level counter for processing and performs the token update. The fairness token counter will update asynchronously using the intermediate token count while generating the token.

IV. Conclusion and Futrue Work

In this work, we introduced Hierarchical Multi-Modality Fair Scheduling (HMFS), a novel scheduling approach for multi-agent, multi-modal model serving by leveraging the Unified Token Representation. For future work, we plan to conduct a comprehensive evaluation of our scheduling algorithm under different workload conditions to analyze its impact on fairness, throughput, and latency. Additionally, we aim to extend our framework by integrating resource-aware scheduling, where fairness is balanced against GPU memory, computational and SLO constraints, and batch efficiency. By combining fair scheduling with adaptive resource allocation, we seek to further optimize performance and ensure efficient model serving in large-scale AI systems.

References

[1] P. Palanisamy, "Multi-agent connected autonomous driving using deep reinforcement learning," in *2020 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2020, pp. 1–7.

[2] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295*, 2016.

[3] J. Ota, "Multi-agent robot systems as distributed autonomous systems," *Advanced engineering informatics*, vol. 20, no. 1, pp. 59–70, 2006.

[4] S. Vorotnikov, K. Ermishin, A. Nazarova, and A. Yuschenko, "Multi-agent robotic systems in collaborative robotics," in *Interactive Collaborative Robotics: Third International Conference, ICR 2018, Leipzig, Germany, September 18–22, 2018, Proceedings 3*, Springer, 2018, pp. 270–279.

[5] D. Huang, J. M. Zhang, M. Luck, Q. Bu, Y. Qing, and H. Cui, "Agentcoder: Multi-agent-based code generation with iterative testing and optimisation," *arXiv preprint arXiv:2312.13010*, 2023.

[6] C.-M. Chan *et al.*, "Chateval: Towards better llm-based evaluators through multi-agent debate," *arXiv preprint arXiv:2308.07201*, 2023.

[7] T. Liang *et al.*, "Encouraging divergent thinking in large language models through multi-agent debate," *arXiv preprint arXiv:2305.19118*, 2023.

[8] Q. Wu *et al.*, "Autogen: Enabling next-gen llm applications via multi-agent conversation," *arXiv preprint arXiv:2308.08155*, 2023.

[9] D. P. Panagoulias, M. Virvou, and G. A. Tsihrintzis, "Evaluating llm–generated multimodal diagnosis from medical images and symptom analysis," *arXiv preprint arXiv:2402.01730*, 2024.

[10] Y. Sheng *et al.*, "Fairness in serving large language models," in *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, 2024, pp. 965–988.