Efficiently Serving Large Multimodal Models Using EPD Disaggregation

Gursimran Singh, Xinglu Wang, Ivan Hu, Timothy Yu, Linzi Xing, Wei Jiang, Zhefeng Wang,
Xiaolong Bai, Yi Li, Ying Xiong, Yong Zhang, Zhenan Fan

MLSys Santa Barbara, CA
ICML International Conference On Machine Learning — Accepted (poster)

Link to the paper

HUAWEI SFU

I. Motivation

Existing LMM serving systems **couple Encoding (E) and Prefill (P) stages** on the same device → E-P interference

- Fig. 1: Compute interference reduces throughput
- Fig. 2: Memory contention limits batch sizes.

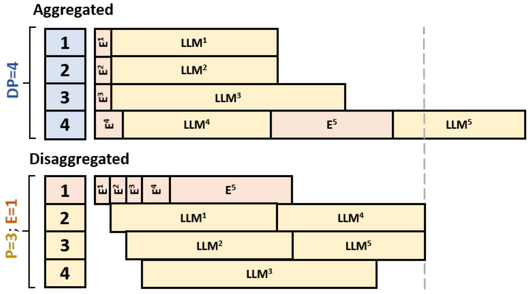


Figure 1: Disaggregating E/P stages avoids interference and enhances both throughput and latency performance.

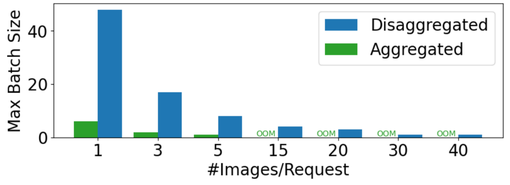


Figure 2: Given requests with same #Images/Request, a disaggregated system allows for a larger max batch size.

Contributions:

- A novel LMM inference system that *disaggregates E and P stages*, enabling efficient parallelization strategies.
- Intra-request parallelism (IPR) to split the E phase of a single request into multiple independent sub-tasks that can be *executed in parallel across devices*.
- Features like a black-box optimizer for configuration tuning, and dynamic role switching across pipeline stages (E,P,D).

II. System Design

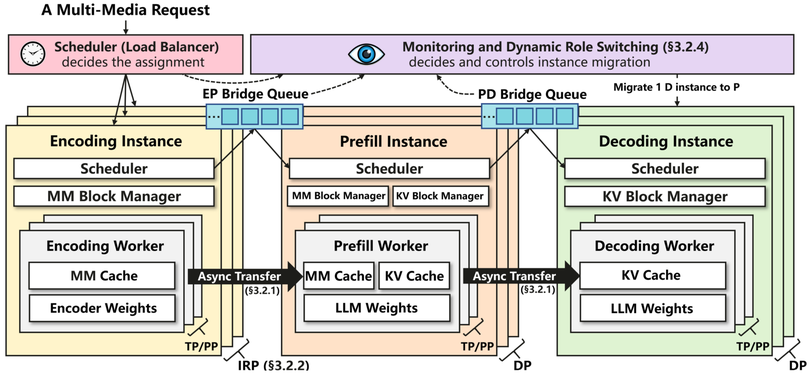


Figure 3: System architecture of the proposed EPD Disaggregated Inference.

Three-stage decoupling:

- *Encoding (E) → Prefill (P) → Decode (D)*
- *Each stage runs* on dedicated devices

Asynchronous transfers:

- *EP-migration (E→P MM cache)*
- *PD-migration* (P→D KV cache)

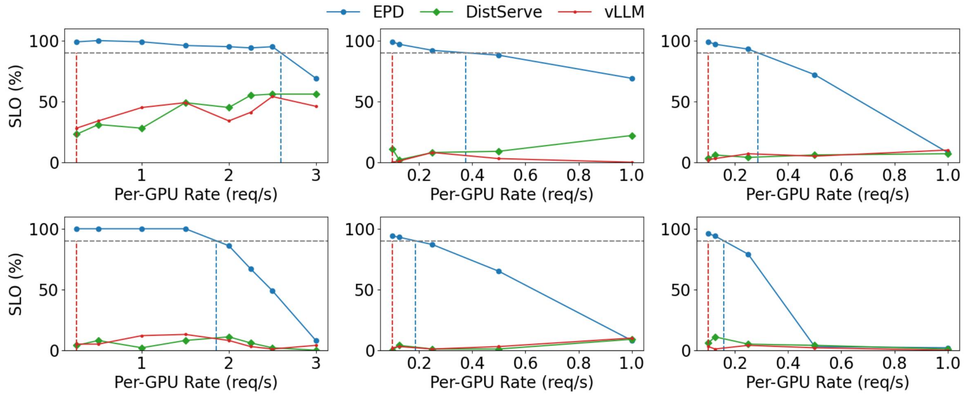
Key components:

- *Intra-Request Parallelism (IRP)* accelerates high resolution image encoding by sharding single requests across multiple GPUs.
- *Asynchronous Token Transfer* with chip-level interconnects (e.g., NVLink) and high-speed networks for MM and KV tokens.
- *MMBlockManager* (interface-compatible with KVBlockManager [1]) for dynamic MM cache management, enabling async. cache transfer and execution.
- *Black-box Resource Optimizer* inherited from [2] automatically tunes batch sizes and parallelization strategies specific to workload composition.
- *Dynamic Role Switching* monitors queues and reallocates GPUs between stages.

Key Benefits

- *Specialization*: Allows components to focus on a single task, improving efficiency and scalability. Each stage can be scaled independently based on workload.
- *Improved Parallelism*: Enables better utilization of hardware by decoupling tightly-bound processing phases, thereby, allowing parallelizing individual requests.
- *Bottleneck Resolution*: Allows addressing performance issues at specific stages.

IV. Experimental Evaluation



(a) MiniCPMv-2.6 (b) InternVL2-8B (c) InternVL2-26B

Figure 5: SLO attainment v.s. arriving request rate. #Image/Request=2 for top row, and 4 for bottom row. Dash vertical line indicates goodput (uses Synthetic dataset).

Gains:

- **71% faster** first token (TTFT)
- **15× less memory** in encoding stage
- **22× larger batches**
- **10× more images/request**
- Dynamic switching **cuts latency 2.4×**

Real-World Ready:

- Handles **8K images & video QA** (NextQA)
- **90% SLO attainment** under heavy load
- Support both **NPU** and **GPU** devices
- Supports **audio**, **images**, and **video**.

Baselines:

- DistServe [2] (+ multimodal)
- vLLM [1] (monolithic).

Metrics:

- Latency: average TTFT, TPOT
- Goodput [2]: SLO attainment (>90%)
- Memory: Batch size, Images/request

Datasets:

- Synthetic (controlled),
- NextQA (real-world video QA).
- Video MME

References:

[1] Kwon et al., "Efficient Memory Management for Large Language Model Serving with PagedAttention", ACM SOSP 2023.
[2] Zhong et al., "DistServe: Disaggregating Prefill and Decoding for Goodput-Optimized Large Language Model Serving", OSDI (2024)