ModServe: Resource Disaggregation for Scalable Multimodal Model Serving

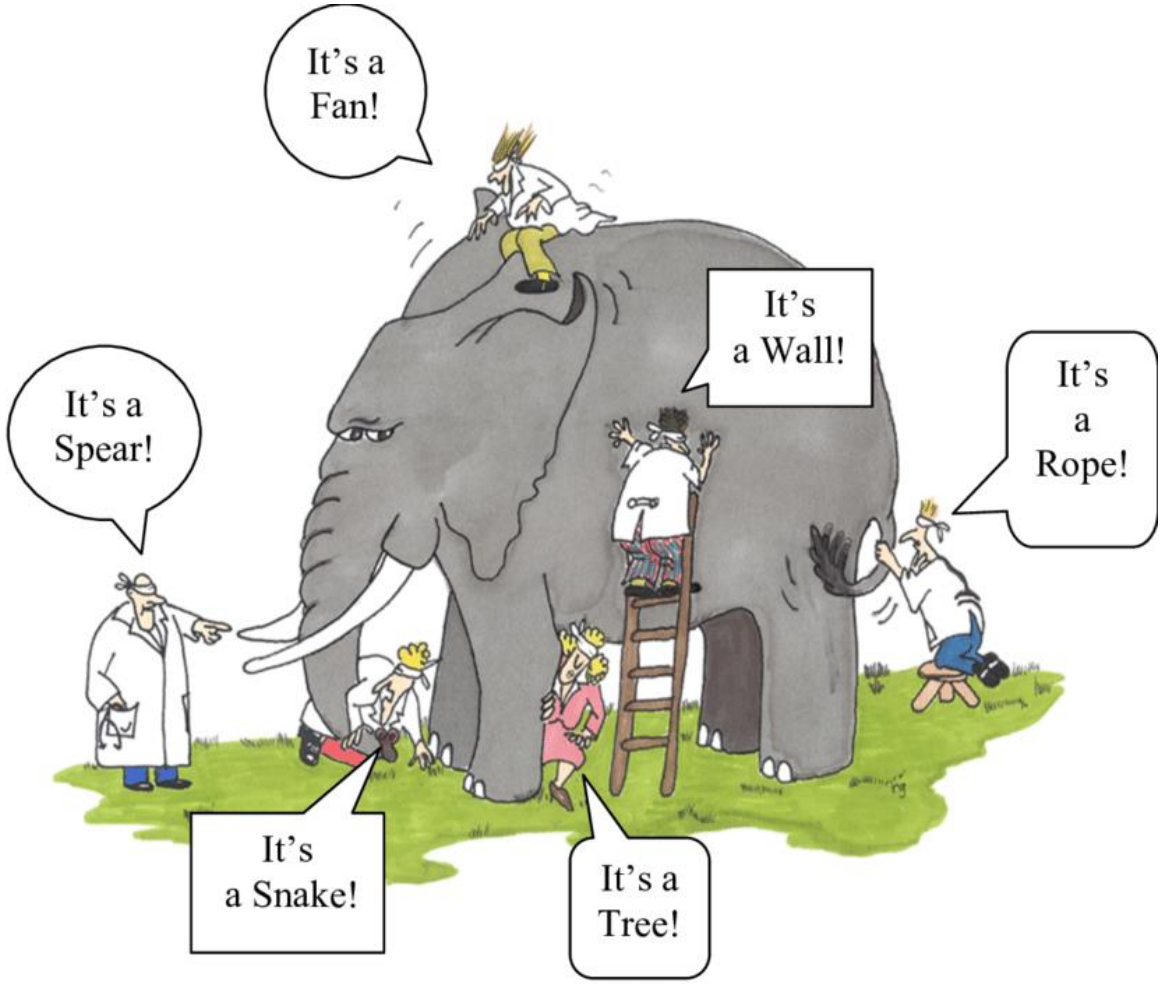
Haoran Qiu[1], Anish Biswas[2]; Zihan Zhao[3]; Jayashree Mohan[2], Alind Khare[4]
Esha Choukse[1], Íñigo Goiri[1]; Zeyu Zhang[3], Haiying Shen[3]
Chetan Bansal[4], Ram Ramjee[2], Rodrigo Fonseca[1]

[1]Microsoft Azure Research, [2]MSR–India, [3]University of Virginia, [4]M365 Research

Microsoft

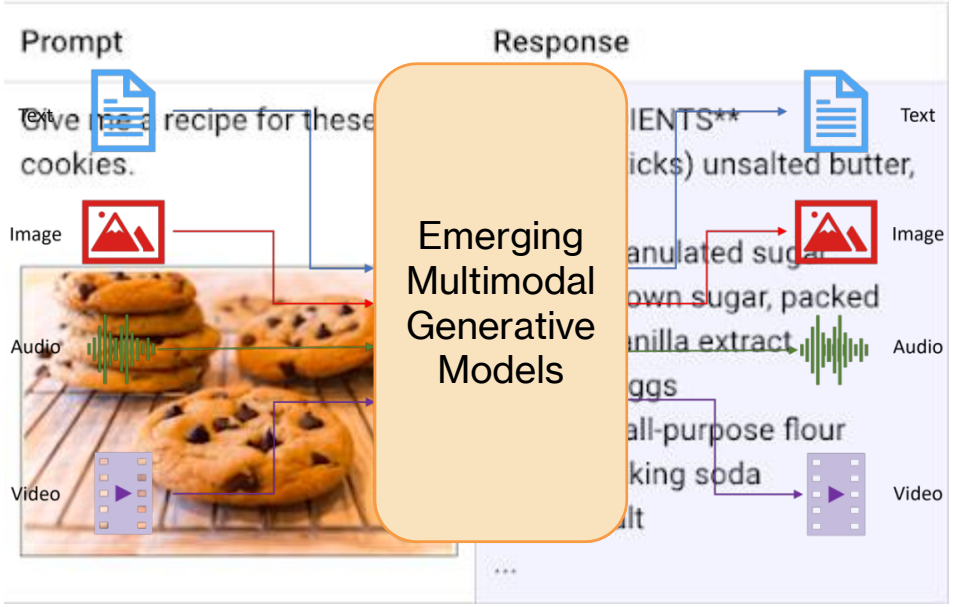
University of Virginia

A Multimodal World



Important to capture the overall sense of data through different modalities

"A multimodal model is capable of *understanding* and *processing* input data of different types and *generating* almost any output."



Credit: https://huggingface.co/learn/computer-vision-course/en/unit4/multimodal-models/a_multimodal_world

This Work

- What are **unique characteristics** of multimodal model inference?

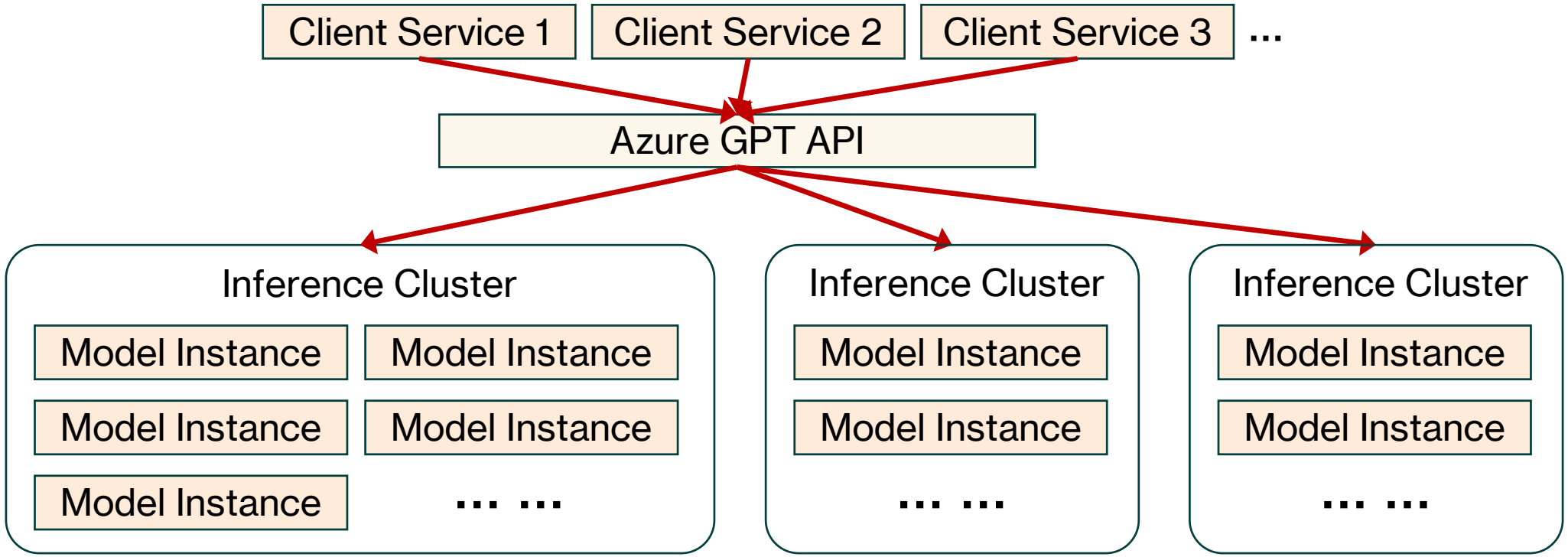
- **Characterization**
  - Production multimodal serving traces
  - Open-source models

- **ModServe**: our proposed solution for efficiently serving multimodal models at scale

Production Traces

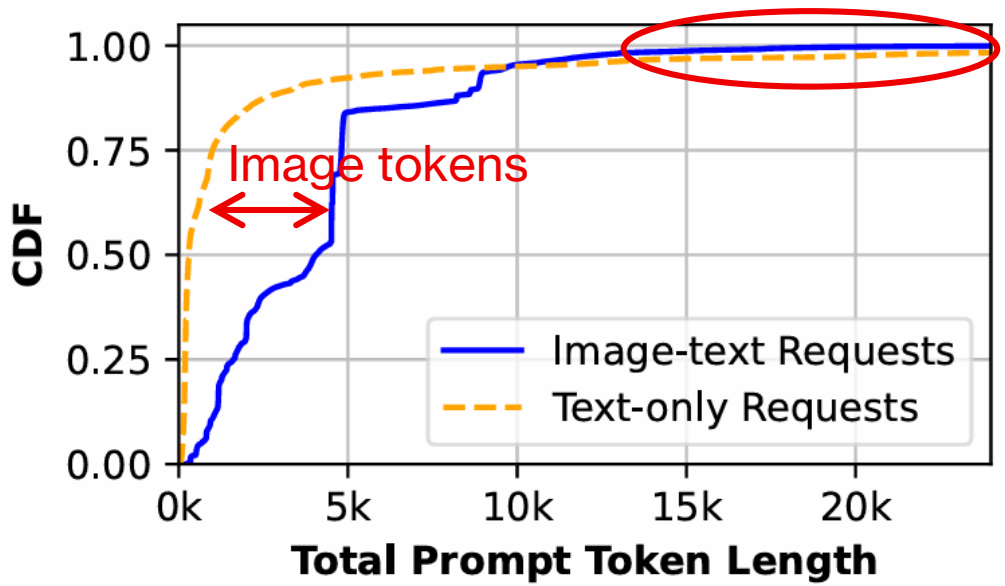
Microsoft Azure

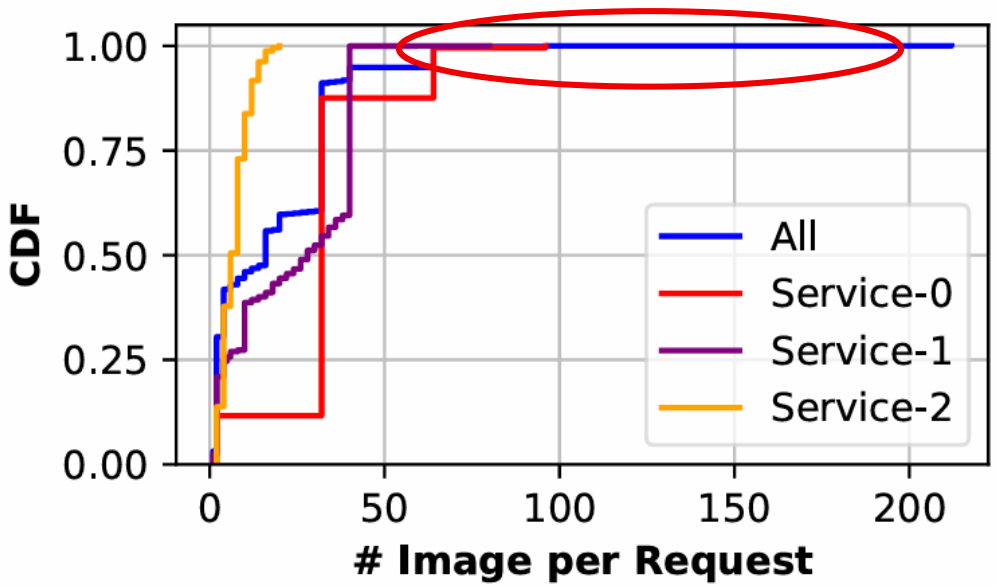
- Azure OpenAI offer multimodal model APIs such as GPT4, GPT4o, GPT5



Token / Image Distribution

- Image requests have higher median / P25 / P75 prompt tokens
- Text requests have higher tail prompt tokens
- There are emerging video analysis workloads (>100 images per request)

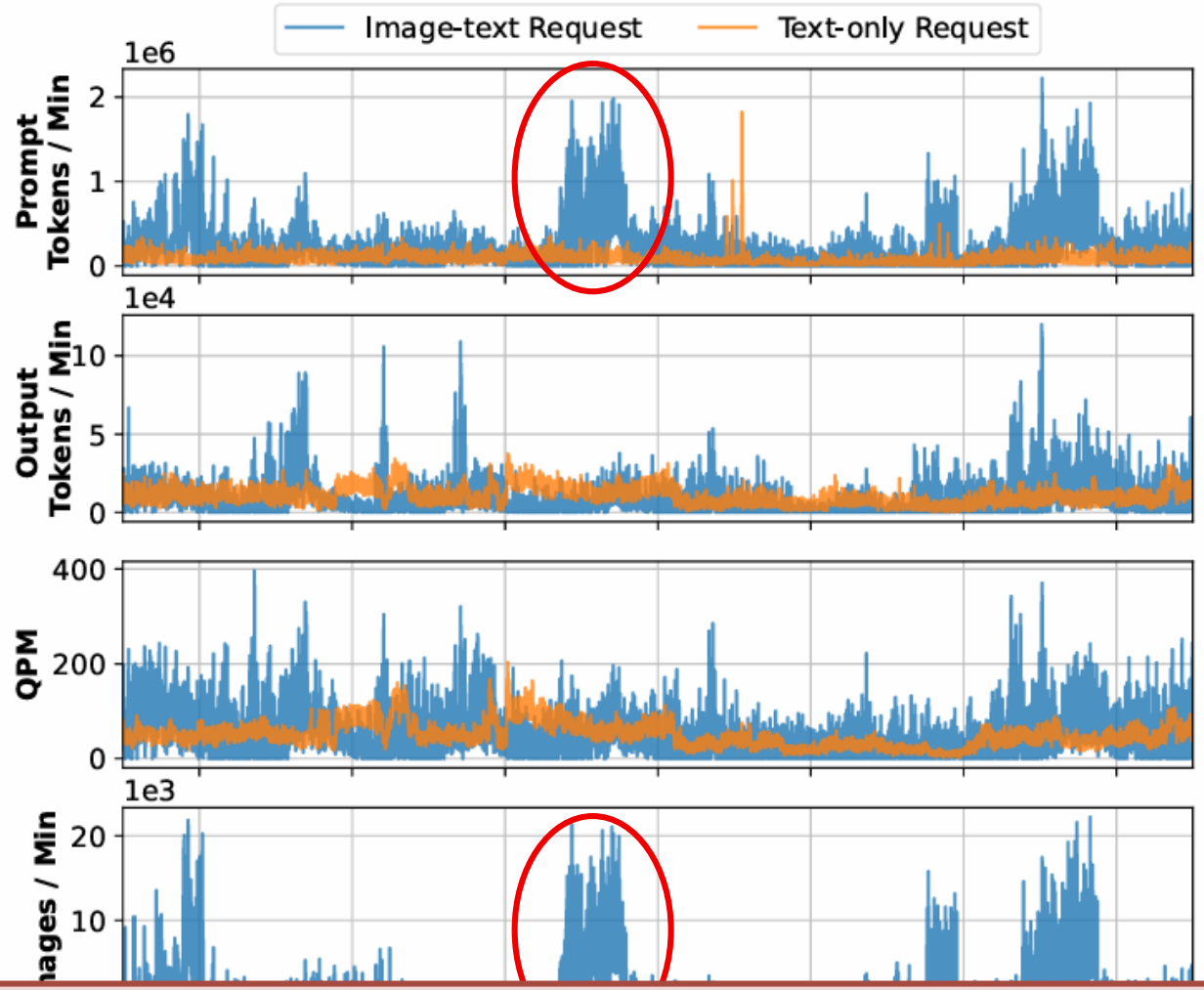




Request heterogeneity is increased with the new multimodal capabilities.

Arrival Patterns (over a week)

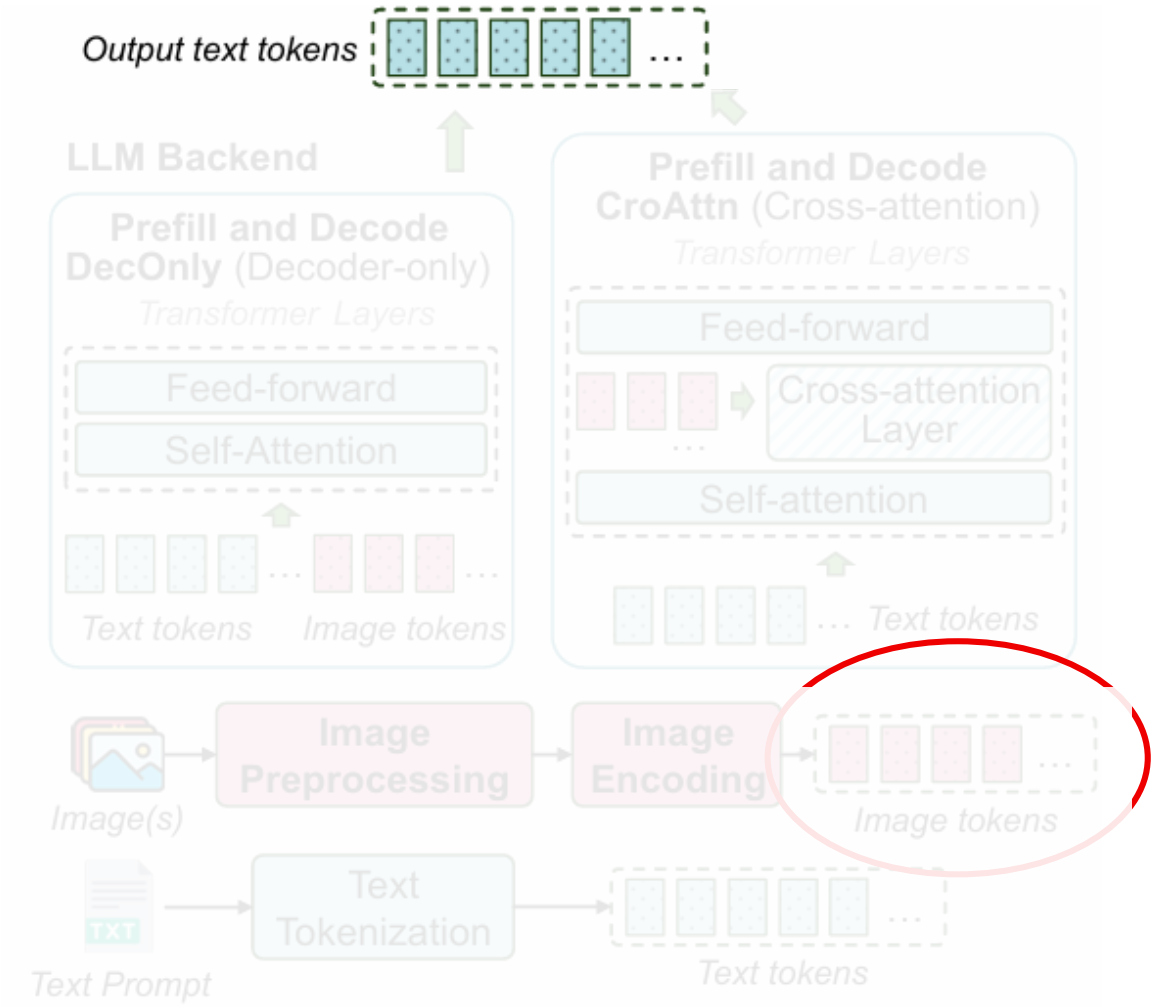
- Image workloads show drastically different traffic patterns from textual workloads
- Image prompt tokens bursts are driven by the **number of images**, rather than the arrival rate
- Multimodal traffic has higher variability than textual traffic



Multimodal traffic overall is more **unpredictable** than text-only requests.

No Longer a Single Monolith

- ***Text-Image-to-Text:***
  - Different stages: (1) image preprocessing → (2) encoding → (3) LLM prefill → LLM decode
- ***Two dominant architectures:***
  - Decoder-only
  - Cross-attention-based



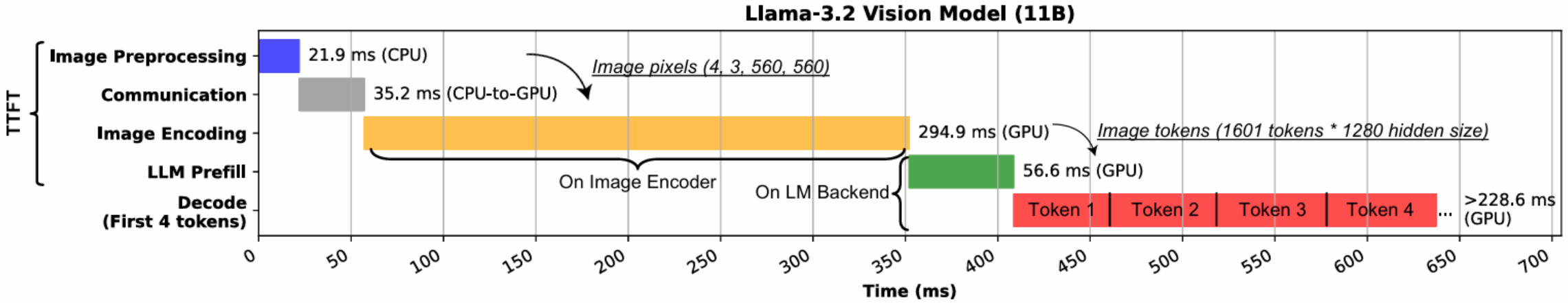
LMM Characteristics – Setup

- Open-source models

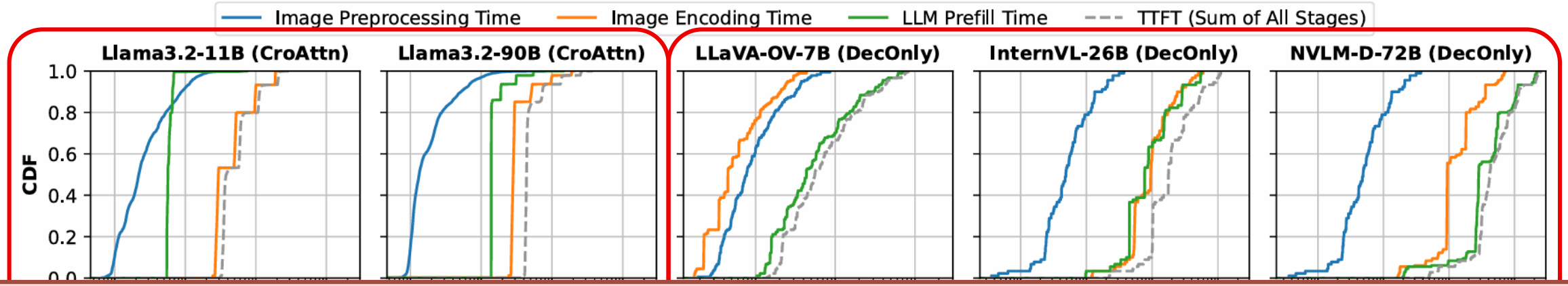
Model	Architecture	Image Encoder
Llama3.2-11B	Cross-attention	ViT-H/14 (630M)
Llama3.2-90B		
LLaVA-OneVision-7B	Decoder-only	SigLIP (400M)
LLaVA-OneVision-72B		
InternVL-2.5 26B		InternViT (6B)
NVLM-D 72B		

- Setup:
  - Hardware: Azure VM with 8xA100s NVIDIA GPUs
  - Dataset: ShareGPT-4o Image datasets

Time-to-First-Token Breakdown Analysis



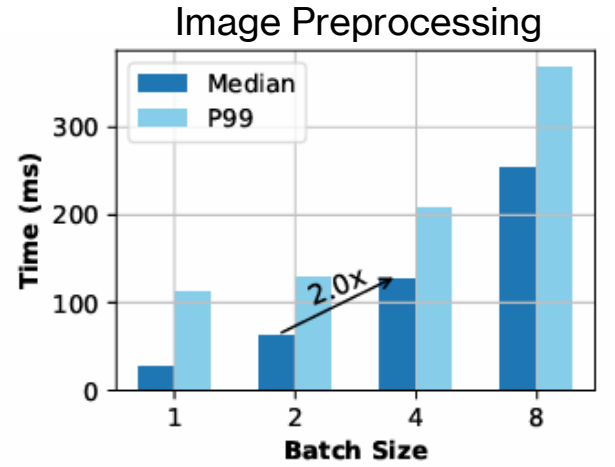
- TTFT breakdown of each model:

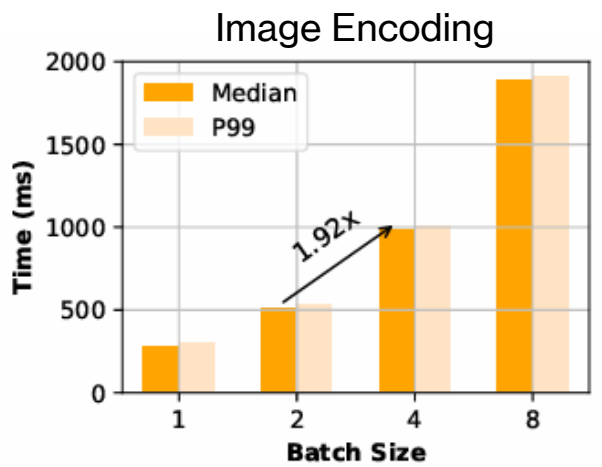


A major portion of the TTFT is spent on **image encoding**, particularly for CroAttn models, making image encoding optimization critical to meet TTFT SLOs.

Compute-Intensive Image Encoding

- Image preprocessing time and encoding time **linearly increase** with the batch size





- SM core utilization during image encoding stages are **saturated**

Image preprocessing and encoding are both **compute-intensive** similar to LLM prefill stage → Demands **scale-out** or **parallelization** for image heavy requests

Diverse Sensitivity to Batching

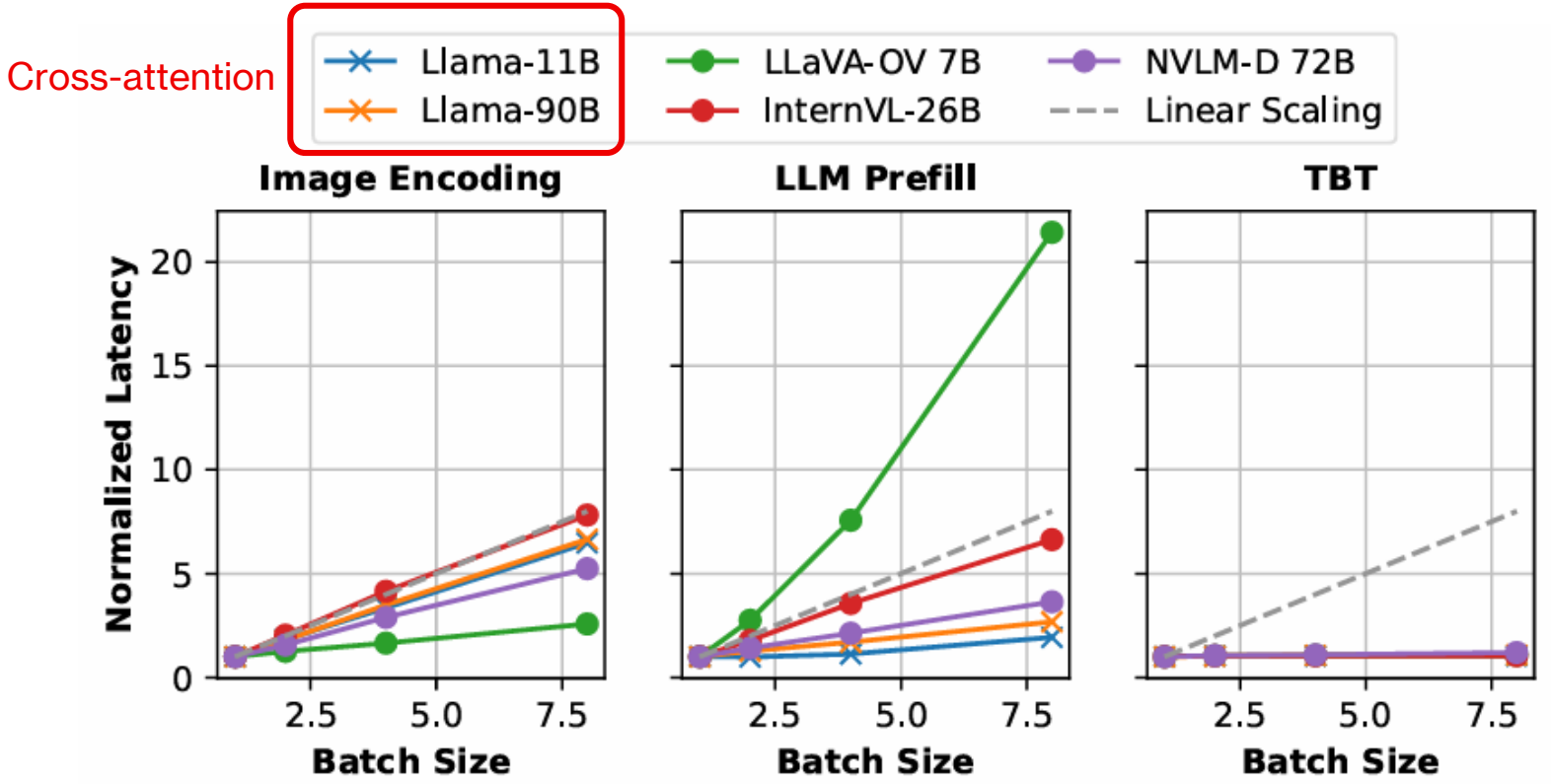
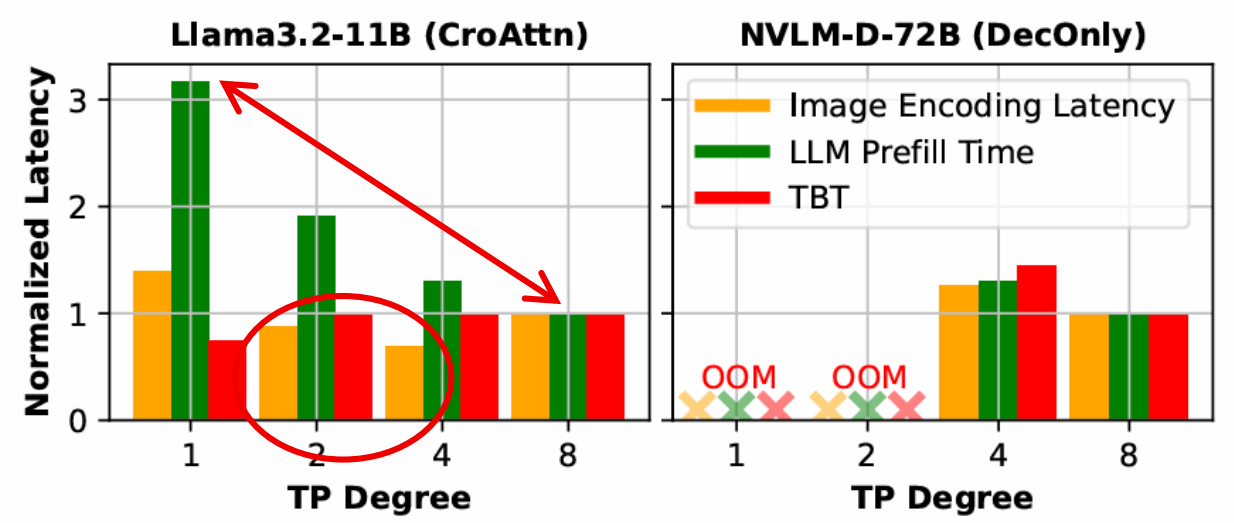


Image encoding, LLM prefill, and decode, each has **different sensitivity to batching**, and is model-dependent

Different Model Parallelism Strategies



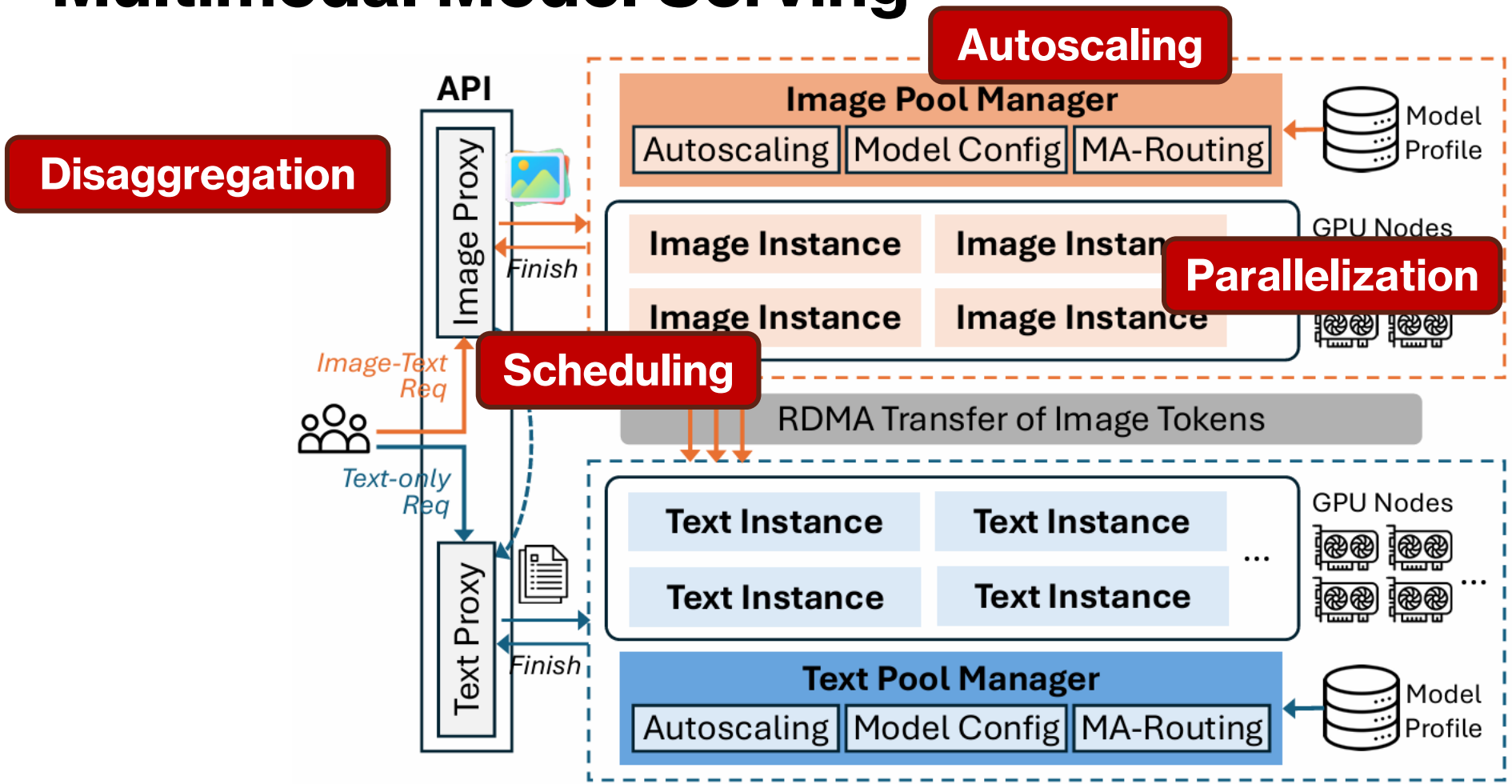
- **Tensor Parallelism (TP)** reduces compute latency while increasing the communication overhead
  - Commonly used when partitioning a model across devices within a server

Treating the image encoder and LLM backend as a **monolith** limits parallelism flexibility and degrades performance.

Systems implications & challenges?

- **Decoupling** – Stage disaggregation is a well-known principle
  - Image encoder, LLM prefill, (and optionally LLM decode)
- **Parallelization**
  - Image preprocessing/encoding parallelization for image-heavy requests
- **Modality-aware scheduling**
  - How to tackle mixed text-only vs. multimodal requests and provide proper performance isolation?
- **Stage-aware autoscaling**
  - Minimize wasted resources while meeting traffic bursts

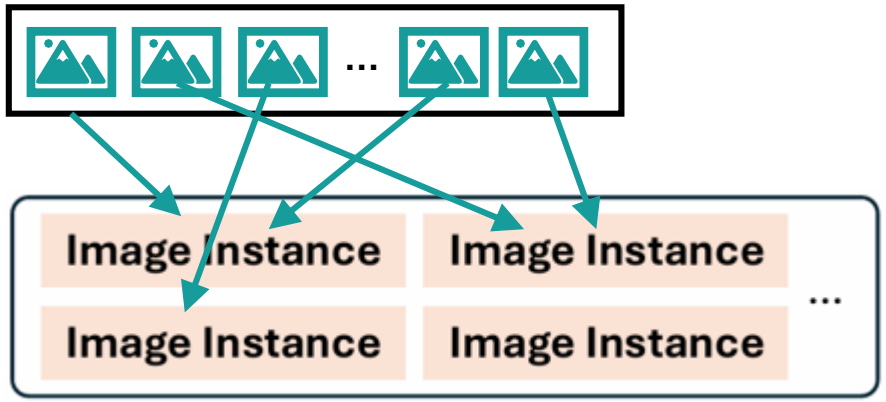
ModServe: Scalable and Resource-Efficient Large Multimodal Model Serving



Production Trace Analysis	Open-source LMM Characterization	ModServe Design	Evaluation

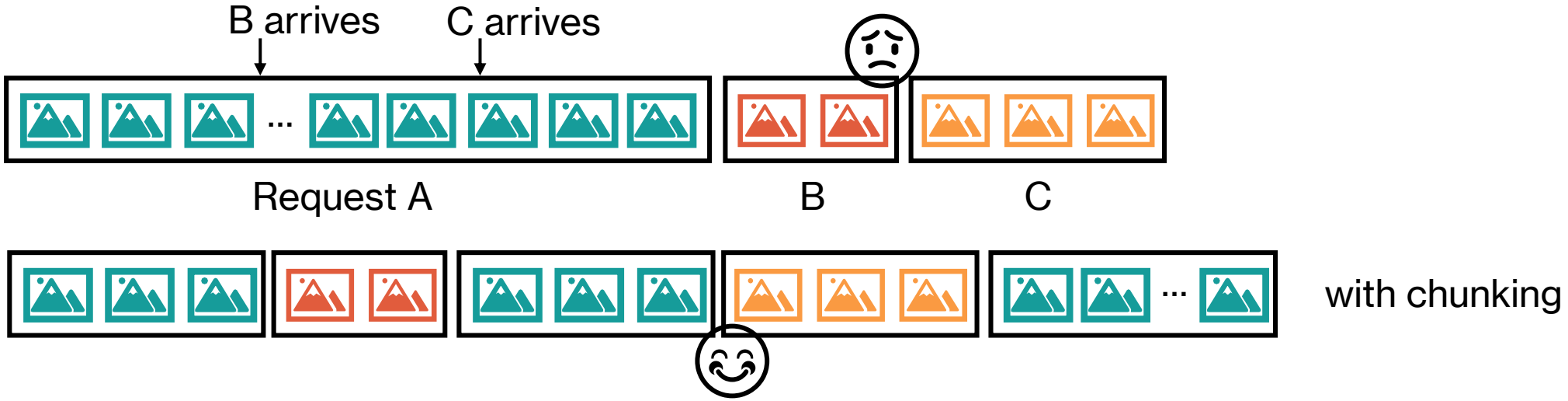
Request Routing

- Text-only requests are routed to a Text Instance directly
  - With the **lowest load (pending tokens)**
- Image requests are first routed to Image Instances for image encoding
  - Each image is routed to an instance with the **lowest load (pending image tokens), in parallel**
  - Then the request is routed to a Text Instance for text generation



Batch Scheduling

- For mixing text-only requests and image requests:
  - Short text-only requests are prioritized (lower TTFT SLOs)
  - Least slack-aware scheduling
- For image-heavy requests with many images:
  - Request chunking to avoid head-of-line blocking

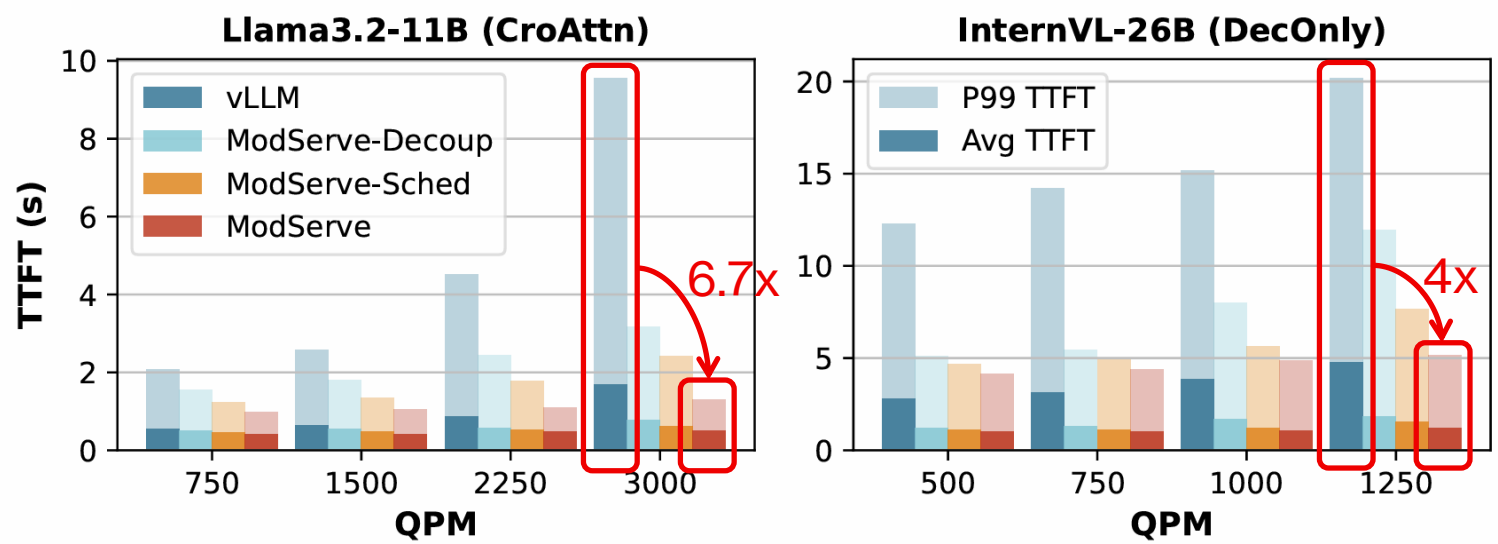


Evaluation Setup

- Implementation on top of **vLLM** v0.7.2
- **128 A100 GPUs** connected with NVLink within each server, and InfiniBand across servers
- **Llama3.2-11B** and **InternVL-26B** for representative cross-attention vs. decoder-only LMMs
- Production LMM inference traces
- ShareGPT-4o image datasets

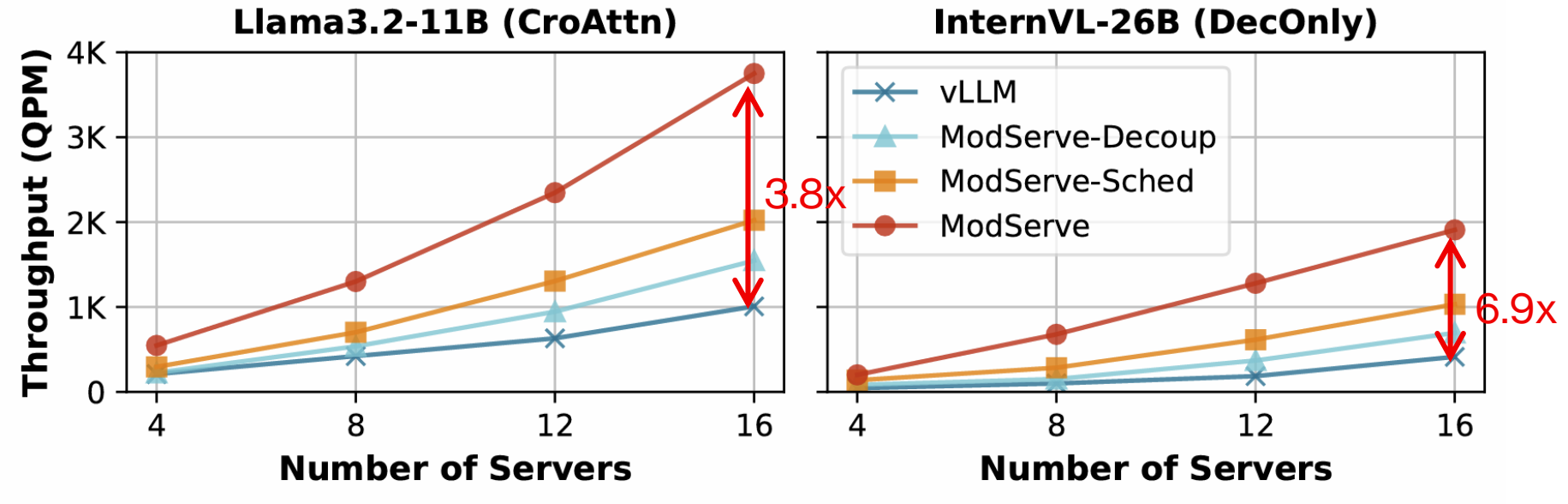
Latency Reduction

- Compared to vLLM (monolith), **ModServe reduces TTFT** by up to **6.7x** and **4x**, for Llama and InternVL
- No impact on TBT latency



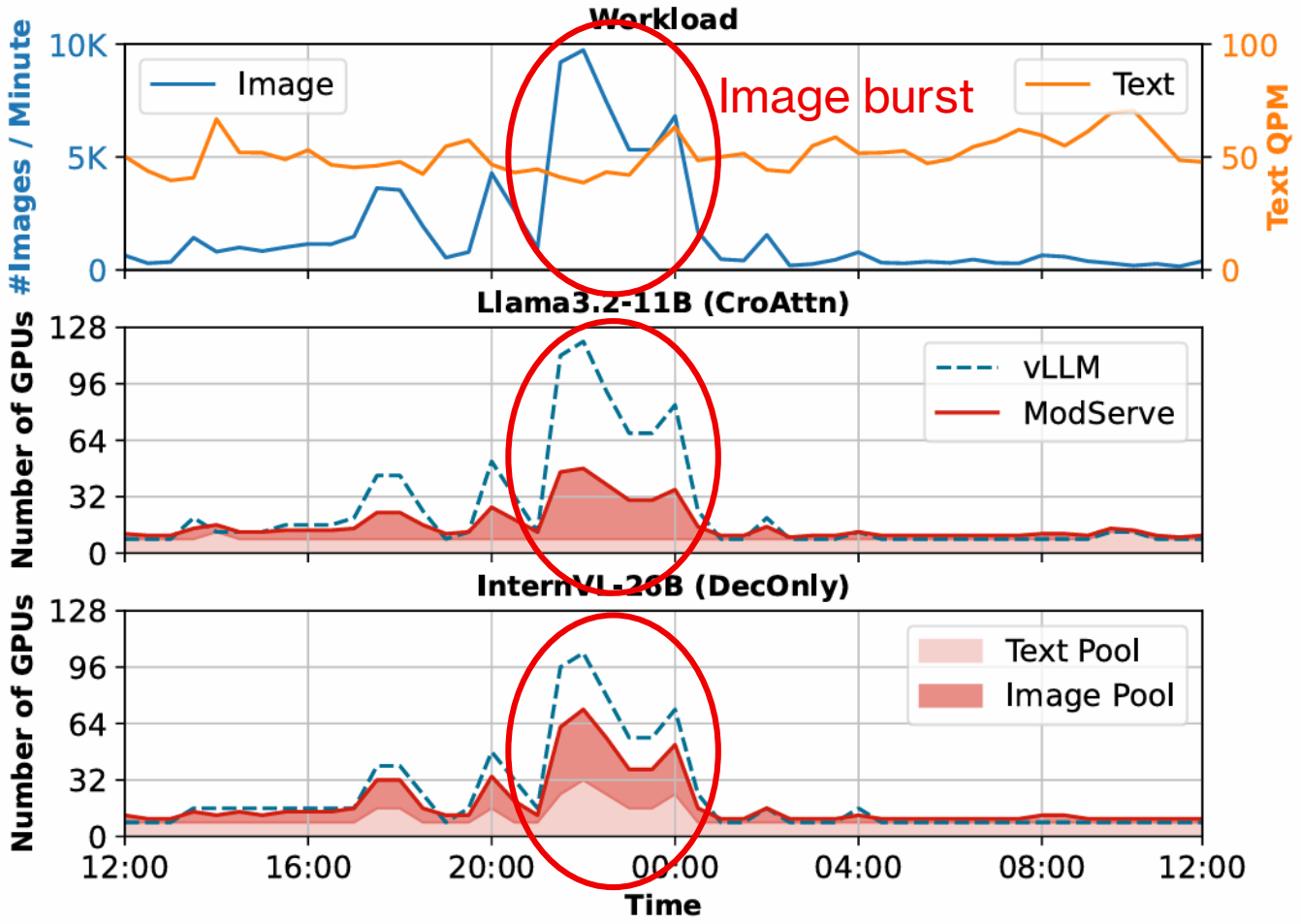
Increasing Throughput

- Max throughput under the TTFT and TBT SLOs
- ModServe achieves up to **3.8x** and **5.5x** throughput improvement over vLLM
- InternVL enjoys **larger benefits** from decoupling because decoder-only model introduces higher contention between LLM backend and encoders



Dynamic Resource Provisioning

- Larger GPU savings at bursts
- **25-42%** GPU usage savings on average
- **Less** GPU savings for decoder-only models
  - Image bursts translated to larger prompt tokens
  - Needs more #LLMs



Summary

- Unique LMM serving challenges from (1) **heterogeneous** multimodal model stages/components and (2) **imbalanced** multimodal traffic
- **Characterization** on open-source LMMs and production traces
- ***ModServe***: Scalable and resource-efficient LMM serving at scale
  - Disaggregation, modality-aware policies

- **Traces are open-sourced!**
  https://github.com/Azure/AzurePublicDataset
- **More in the paper!**

Overview People Publications Projects News & features

https://www.microsoft.com/en-us/research/group/azure-research-systems/

Overview

Azure Research – Systems is a research group in Azure Core that brings forward-looking, world-class systems research directly into Azure. The group was seeded from the Cloud Efficiency team, which migrated from the Systems Research Group at Microsoft Research, for a closer integration with Azure.

Our group's main mission is to improve the cost efficiency of Microsoft's online services and datacenters. We pursue this mission by working closely with the company's product groups to (1) propose and lead joint projects that improve efficiency, and (2) do research on potential future efficiency improvements.

Impact of our research

Some of our main tech transfers and corresponding papers

- We coordinated an effort between Microsoft, NVIDIA, and OpenAI to document and mitigate the effects of the large power swings caused by modern AI training jobs at scale, described in our joint report from August 2025. This resulted in a new power regulation feature in the GB200, and will go into production in our AI datacenters.
- Our hybrid (software + firmware) implementation of per-VM power capping went into production in Cobalt servers in August 2025.
- Our approach for power oversubscription of GPU clusters (described in our ASPLOS 2024 paper) went into production in November 2024.

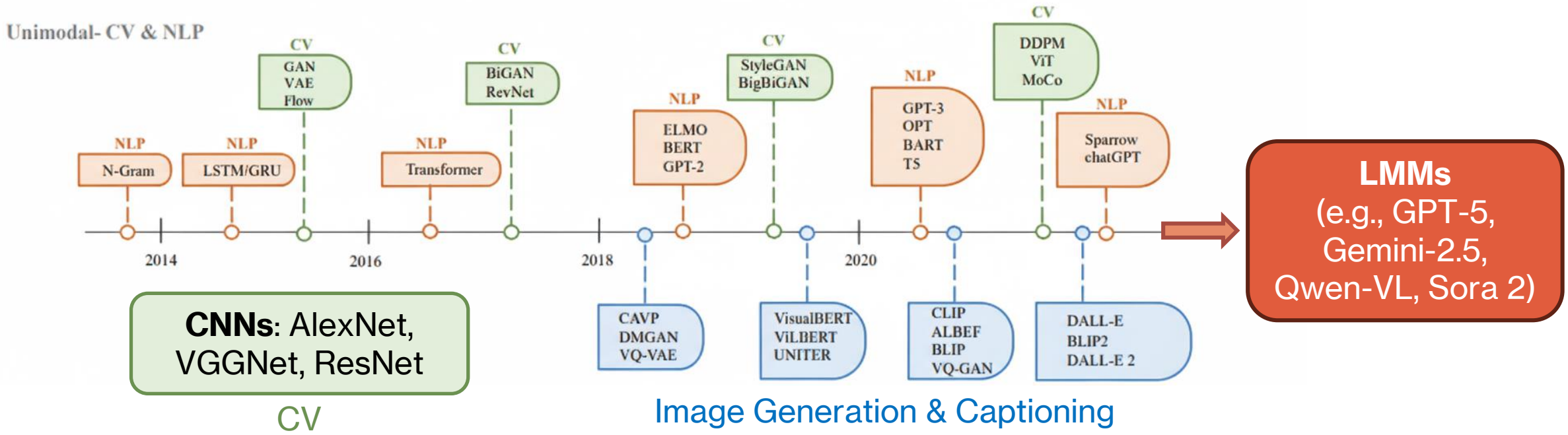
Thank you!

Haoran Qiu (haoran.qiu@microsoft.com)

Backup Slides

Evolution of Vision-Language Models

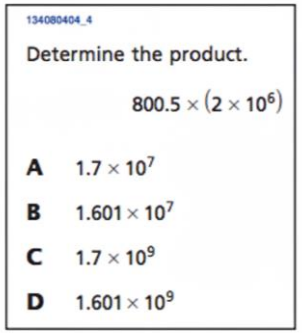
- Large multimodal models (LMMs)
- **All-to-all** interactions (e.g., image-to-text, text-to-video, audio-to-audio)



Source: https://arxiv.org/pdf/2303.04226.pdf

How are LMMs different from Text LLMs?

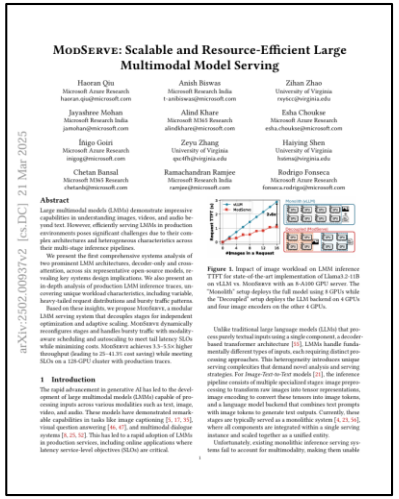
- Models that can **process** inputs beyond text



Please answer the math question in the picture.

Answer: D
Explanation: "This response represents the correct product of the two given numbers. Here's one way to solve it: 800.5 x (2 x 10^6) = 1,601 x 10^6 = 1.601 x 10^9

- Models that can **generate** outputs beyond text

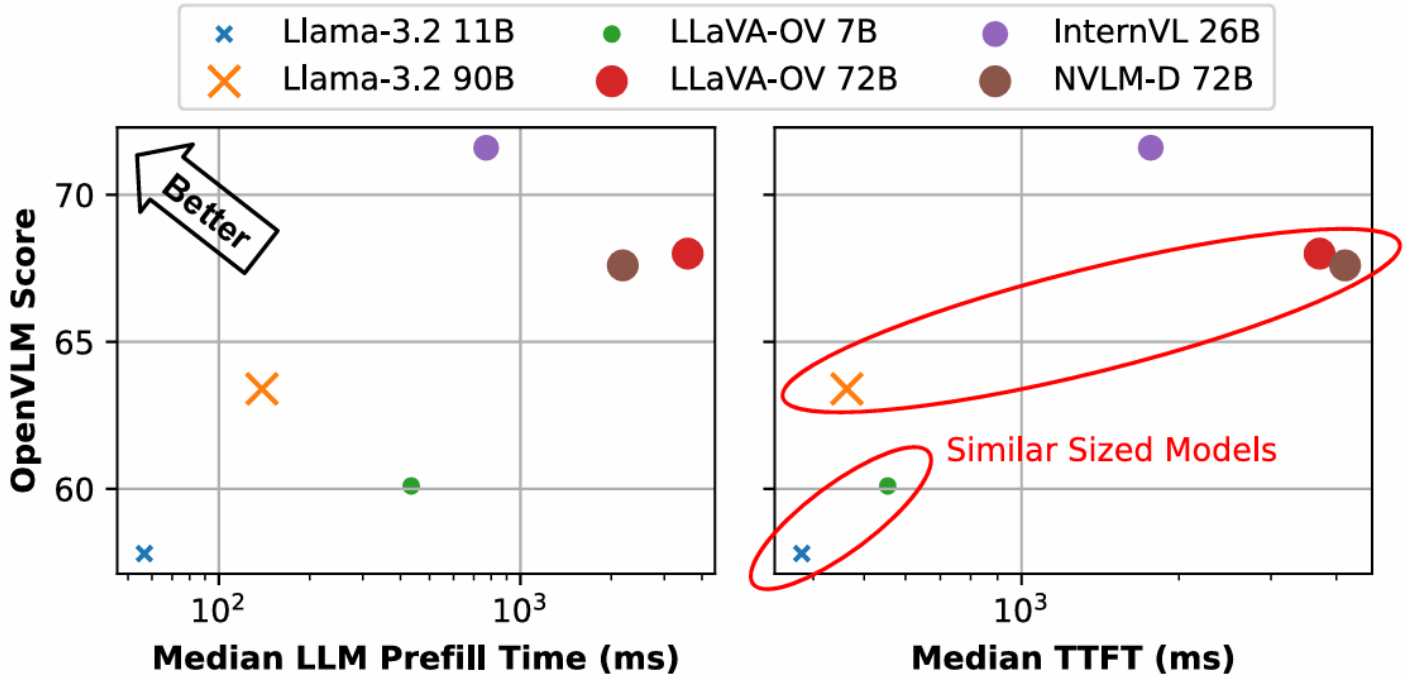

MoonCake: Scalable and Resource-Efficient Large Multimodal Model Serving

Please generate a podcast on the topic of LMM inference based on the given PDF.


....

Important Metrics: TTFT / Accuracy

- TTFT: time to first token (responsiveness)
- Accuracy: quality of the output



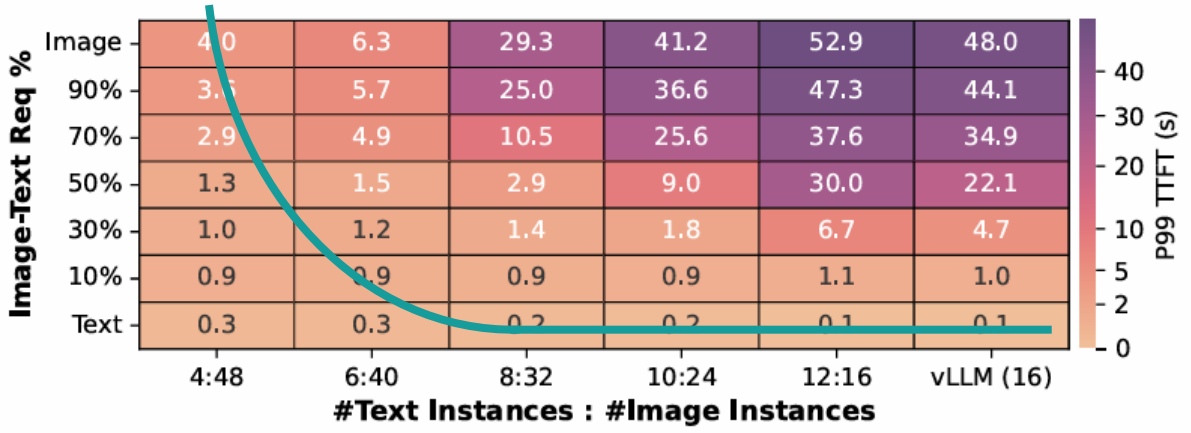
Cross-attention LMMs are more efficient compared to similar-sized decoder-only models; while decoder-only LMMs achieves slightly better accuracy.

Image Token Transfer

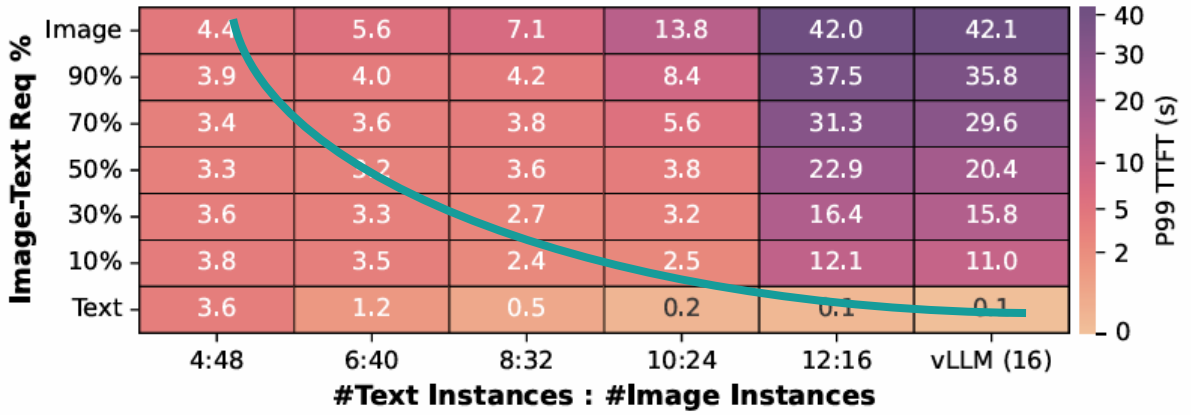
- Pull-based RDMA request from Text Instance to Image Instance(s)
- Asynchronous transfer across GPUs; Parallel across Image Instances
- Can overlap with LLM prefill, but only partially
- P99 transfer time ~5ms on InfiniBand

Impact of Image Request Percentages

- Balancing multimodal workloads
- Increasing in image% → prefer higher Image Instance ratios
- Optimal instance ratio → **3-10x** TTFT reduction from vLLM
- The optimality line shift to **higher** Text Instance ratios for decoder-only models



(a) Llama3.2-11B CroAttn LMM.



(b) InternVL-26B DecOnly LMM.