Characterizing CPU-Induced Slowdowns in Multi-GPU LLM Inference

Euijun Chung, Yuxiao Jia, Aaron Jezghani, Hyesoon Kim
Georgia Institute of Technology

2026 IEEE International Symposium on Workload Characterization
Best Paper Candidate

HPArch
@ Georgia Tech

Georgia Tech

Growing Attention to CPUs in AI Infrastructure

White Paper

Data Center & Artificial Intelligence
Agentic AI Infrastructure Sizing

intel

Agentic AI Requires M

Avoiding GPU Idle Time in Modern In

Authors
Steve Fowler
Josh Segovia
Lawrence Leung
Laurie Fordham
Stephen Holt

Executive Su

Inference has
shifting from G
These agentic
introduce CPU
and planning,
utilization.[2] GP
any delays cau
idle, resulting i
examines how
thereby reduc

Red Hat Blog | By product | By topic | Podcasts | More blogs

The CPU is back: Rethinking the CPU GPU split for LLM inference

August 6, 2026 | 9-minute read | AI inference

< Back to all posts

For the past 3 years, graphics processing units (GPUs) have dominated the large
(LLM) conversation. In traditional chatbot applications, central processing units (
fraction of the total compute per request, while GPUs do the heavy lifting. Howev
single model answering a single question. A growing reliance on tool calls, multiste
orchestration across small, specialized models changes the math on where compu
has called out this shift noting that the CPU-to-GPU ratio is moving from 1:8 in tra
1:1, and in some cases 4:1 in agentic deployments.

Here we'll examine why the assumptions making GPUs the obvious choice for LLM
being renegotiated, what's driving renewed demand for CPU-based serving, and
about where the industry is heading.

IEEE Spectrum

Type to search

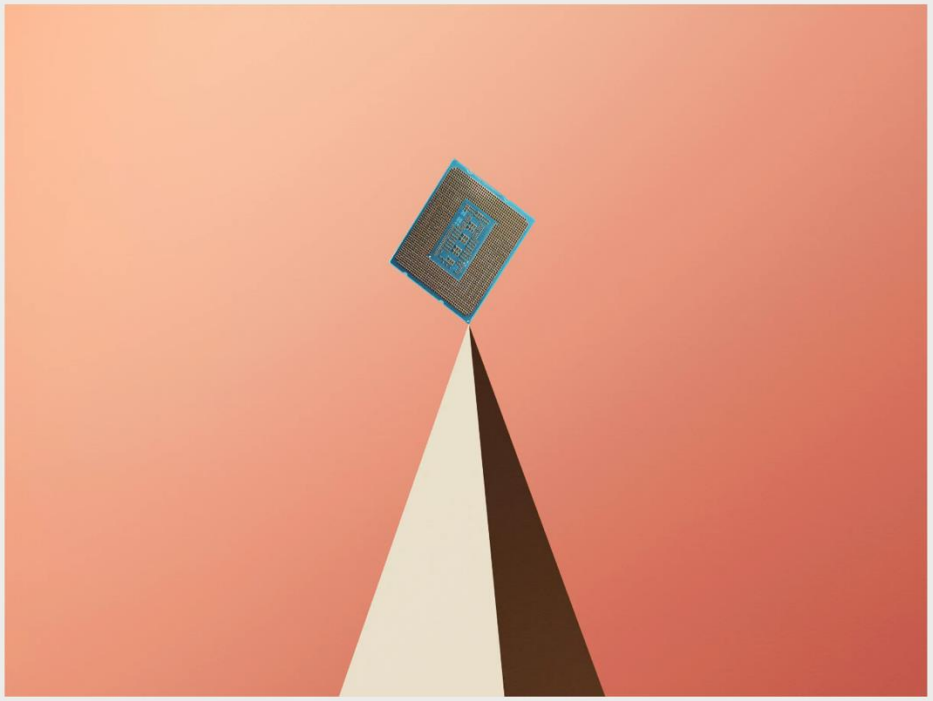
NEWS | AI

The CPU Comeback Is Upon Us >

Agentic AI has made CPUs the new performance bottleneck

BY MATTHEW S. SMITH | 16 AUG 2026 | 5 MIN READ

Matthew S. Smith is a contributing editor for IEEE Spectrum and the former lead reviews editor at Digital Trends.



How important is CPU in agentic AI workload?

CPUs are getting important in AI workloads.

How *exactly* does **CPU** affect **end-to-end performance** in AI workloads?

In this work, we first characterize the **CPU's role in LLM inference**, and investigate how CPU can affect **the end-to-end performance**.

Georgia Tech

CPU's Job in Multi-GPU LLM Inference

1. Input processing (Tokenization)

"Who won the 2025 World Series?"

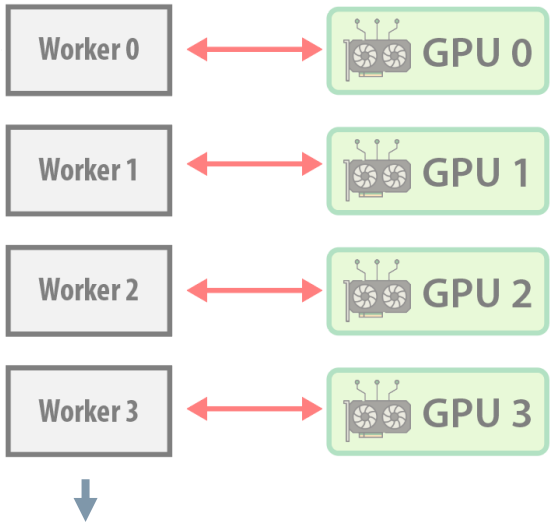
↓

["Who", "won", "the", "202", "5", "World", "Series", "?"]

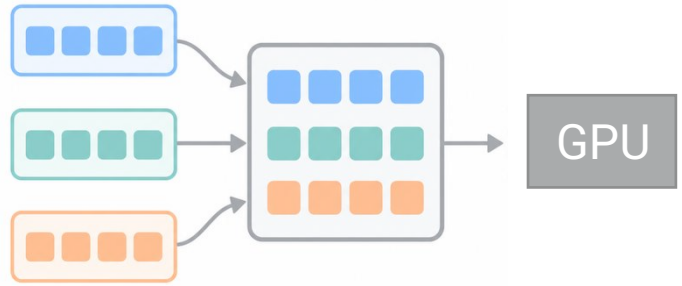
↓

[20600, 3313, 290, 1323, 20, 5922, 14932, 30]

2. Execution of GPU workers



3. Request scheduling & work dispatch



→ All processes share the **same limited CPU resources**.

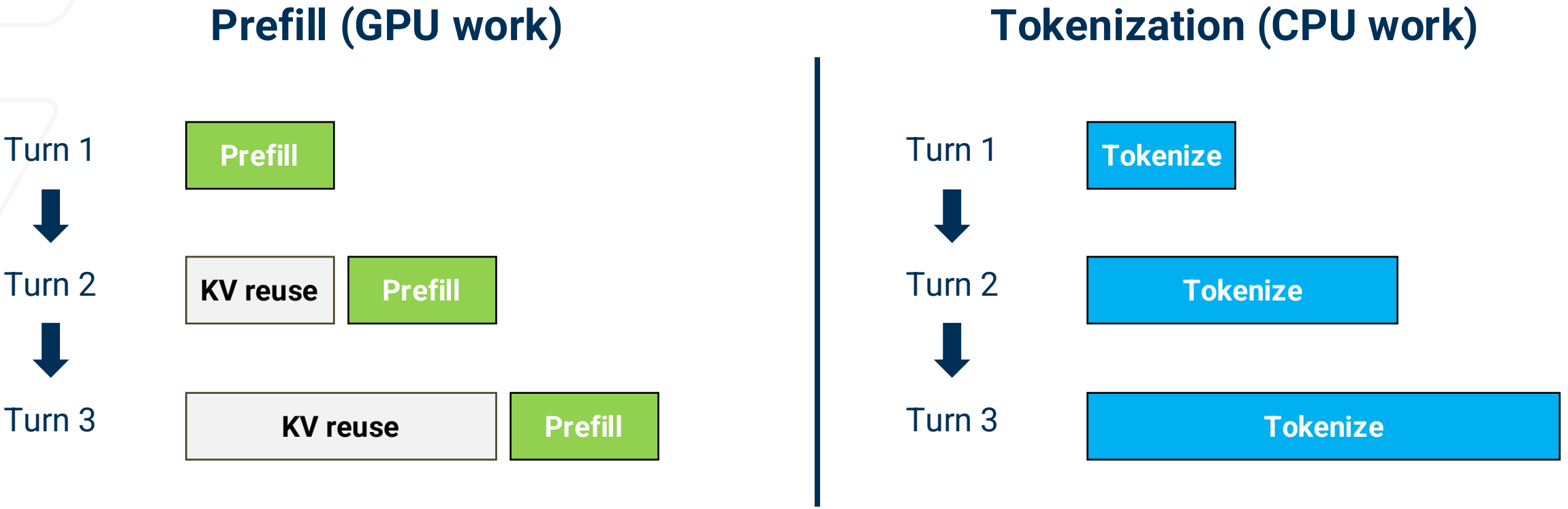
Georgia Tech

Growing Context in Multi-Turn Agentic Workloads

Each request includes the **accumulated conversation history**.

GPU Work Remains Similar, While CPU Work Grows

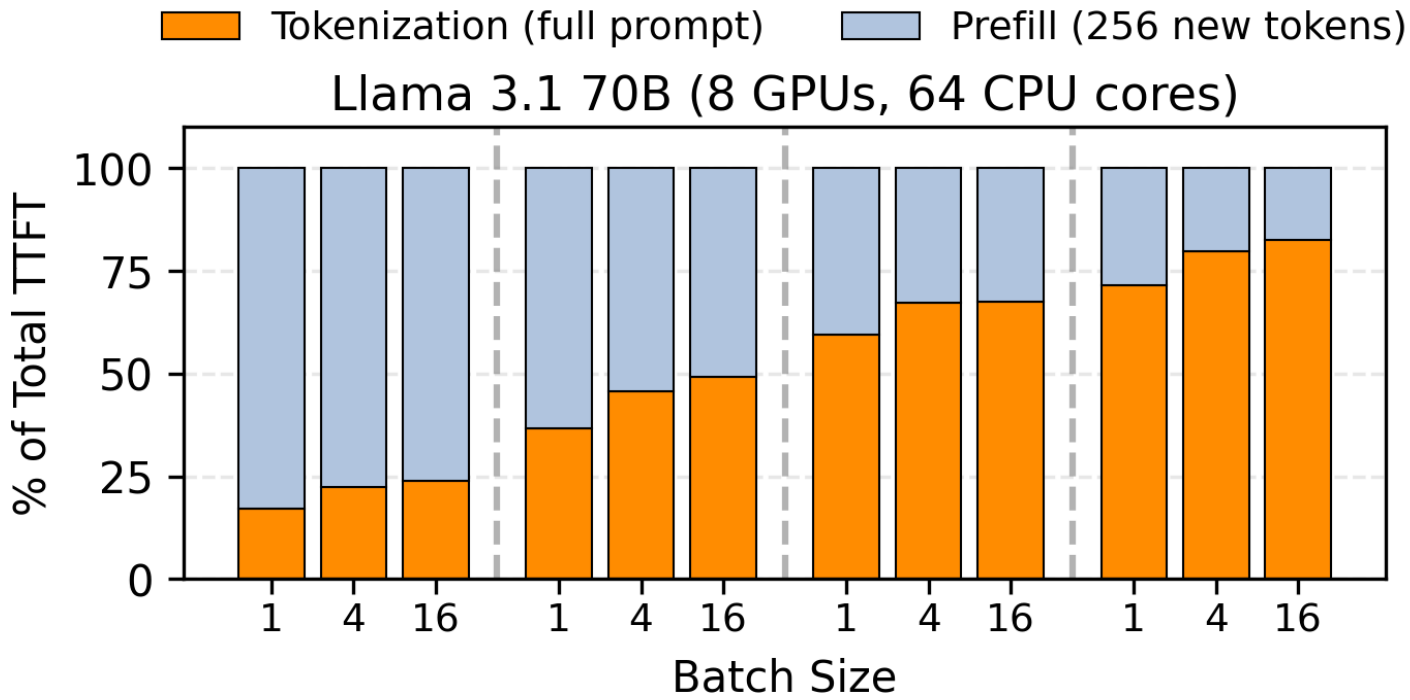
Each turn requires tokenizing the **entire accumulated context**.



In **agentic AI** workloads, **GPU** only processes **new tokens**, while **CPU** processes **the full context**.

Tokenization Can Dominate First-Token Latency

We measure **tokenization** and **model execution time** separately.



Tokenization accounts for up to **80% of time-to-first-token (TTFT)** in our experiments.

CPU's Job in Multi-GPU LLM Inference

1. Input processing (Tokenization)

"Who won the 2025 World Series?"

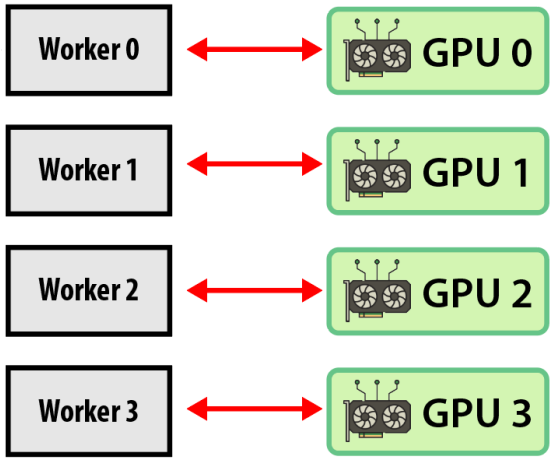
↓

["Who", "won", "the", "202", "5", "World", "Series", "?"]

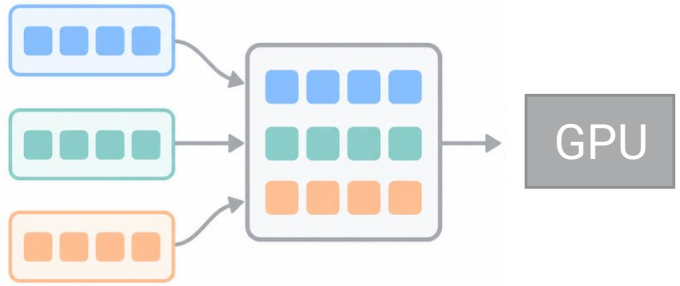
↓

[20600, 3313, 290, 1323, 20, 5922, 14932, 30]

2. Execution of GPU workers



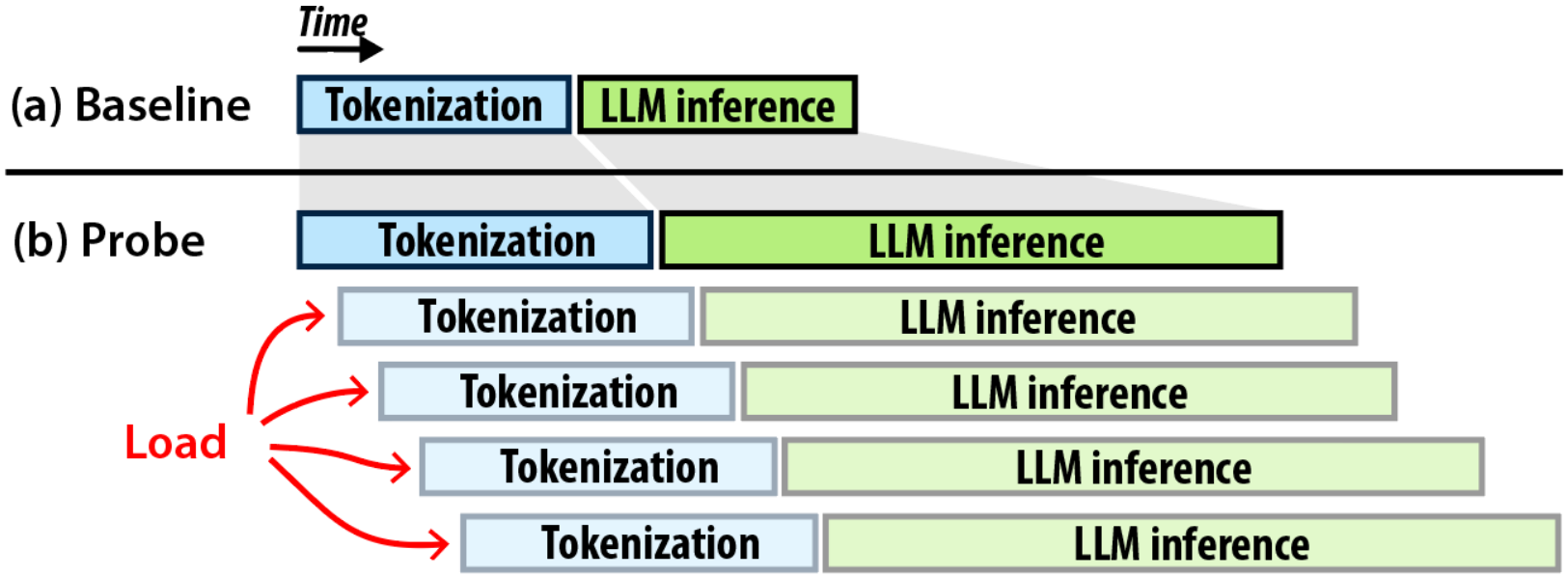
3. Request scheduling & work dispatch



Georgia Tech

How Does Tokenization Slowdown End-to-End Serving?

Probe-under-load experiment: vary CPU cores while keeping GPU fixed.

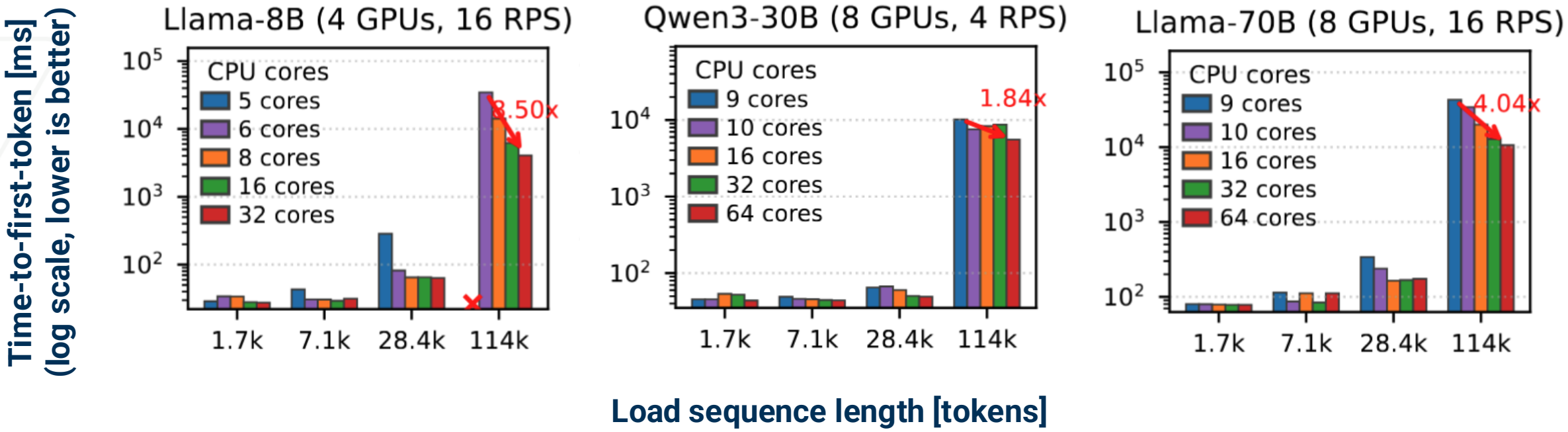


1. How do **concurrent tokenization threads** compete for CPU resources?
2. How does CPU oversubscription affect **end-to-end performance**?

Limited CPU Resources Increase Request Latency

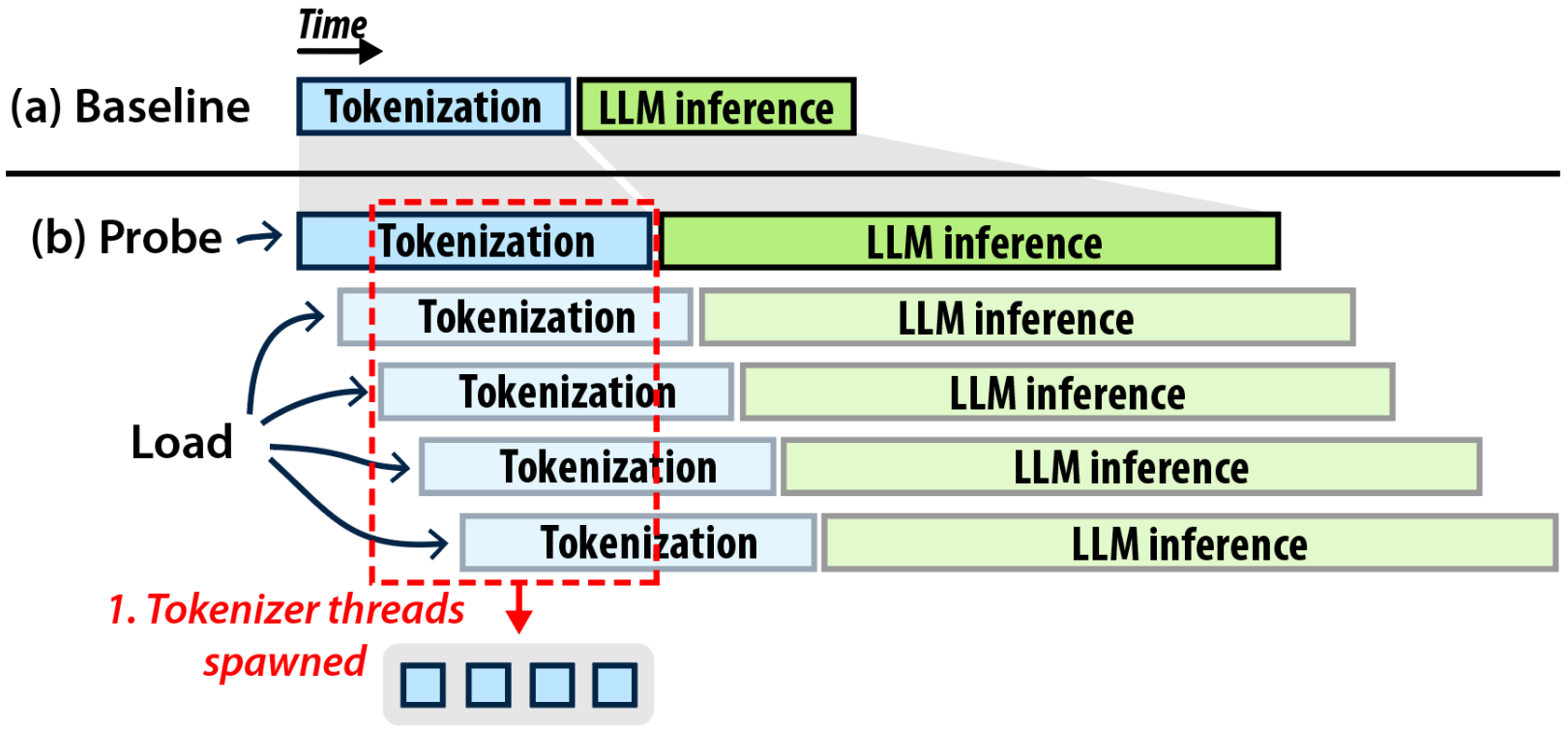
We measure TTFT for a fixed **2.8K-token probe** under concurrent load.

- Red × indicates a **probe timeout (200 s)**

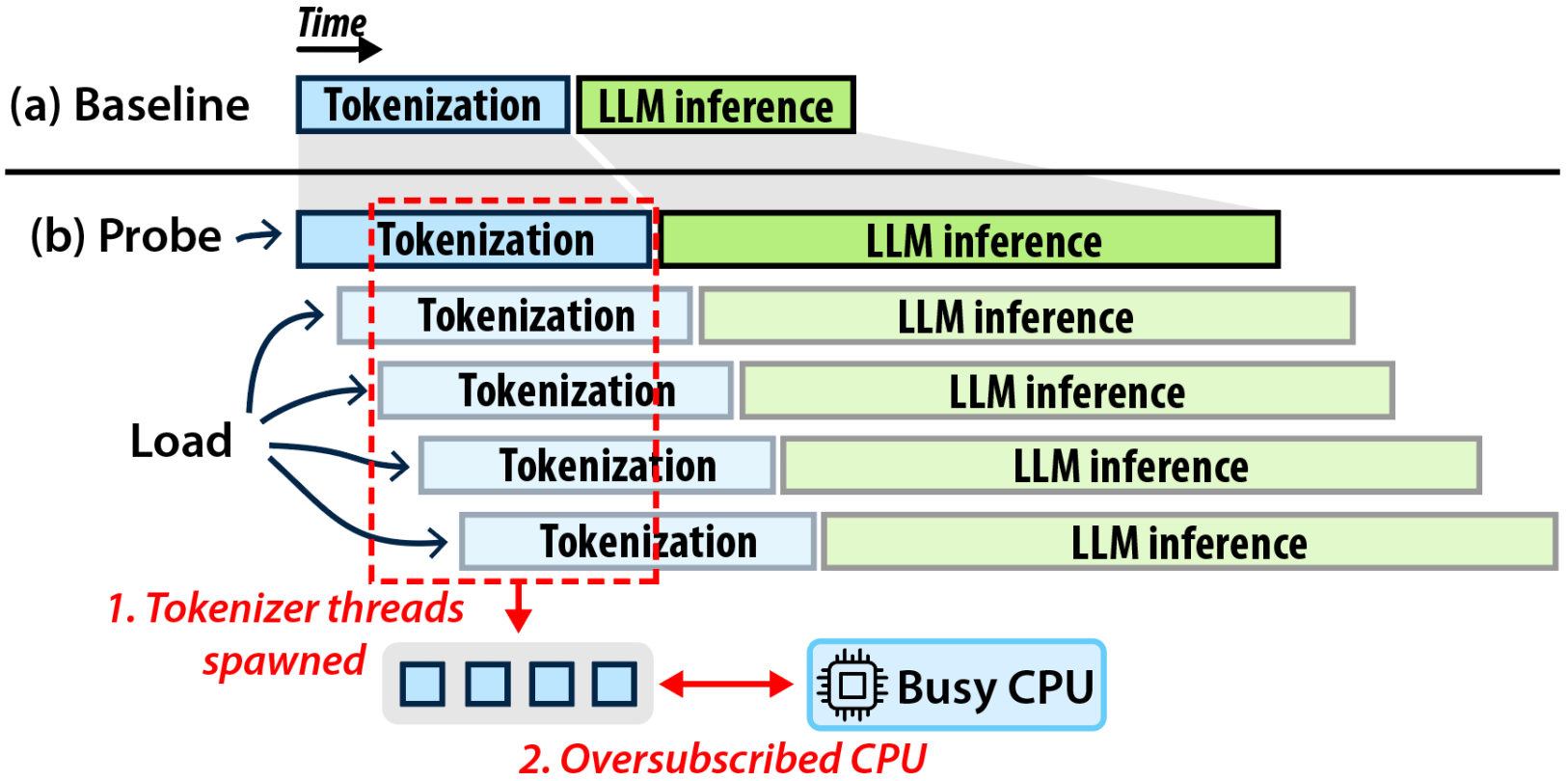


Under the same load, **fewer CPU cores** lead to **up to 7.11× higher TTFT**.

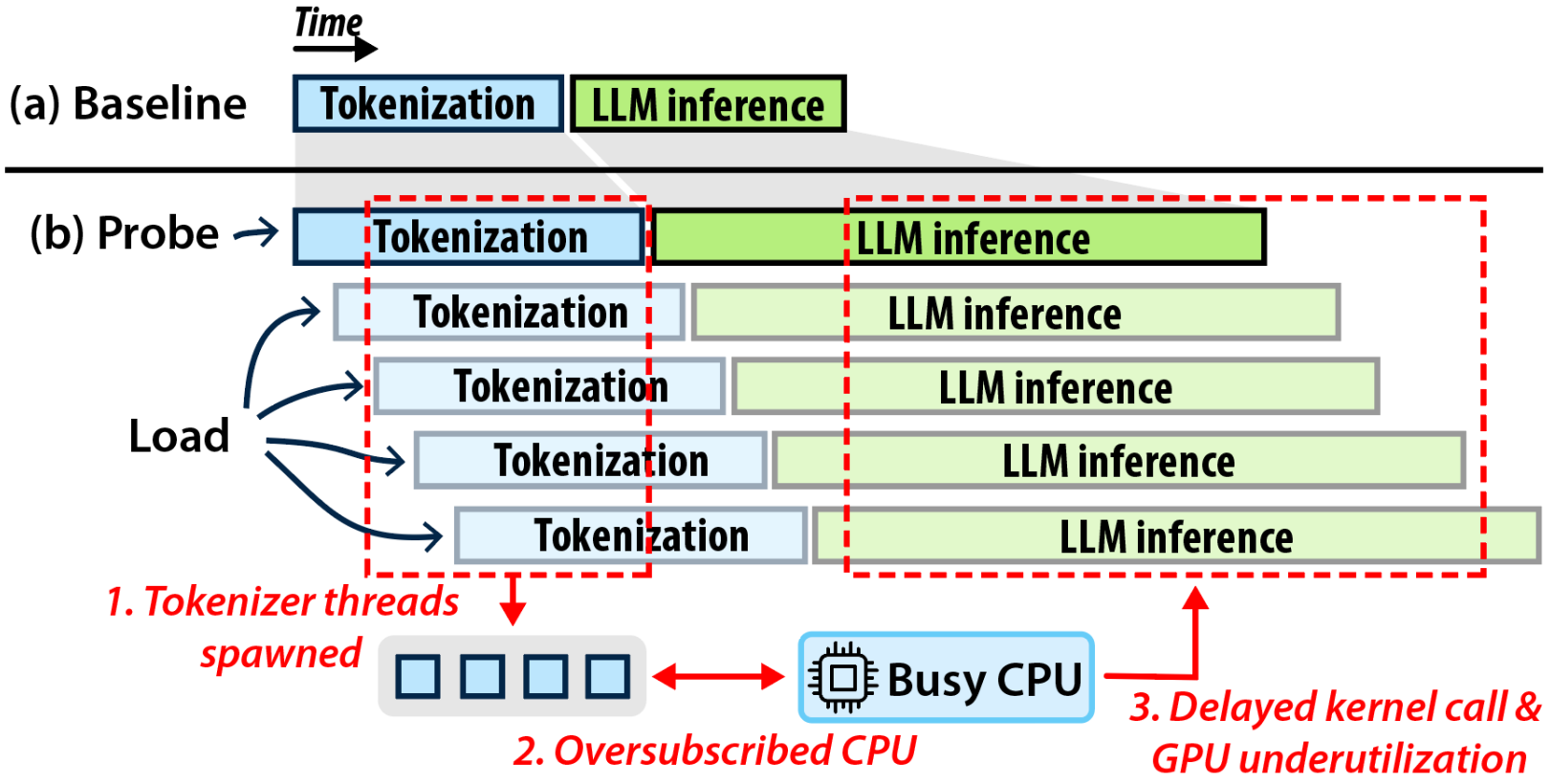
Tokenization Threads Compete for CPU Resources



Tokenization Threads Compete for CPU Resources



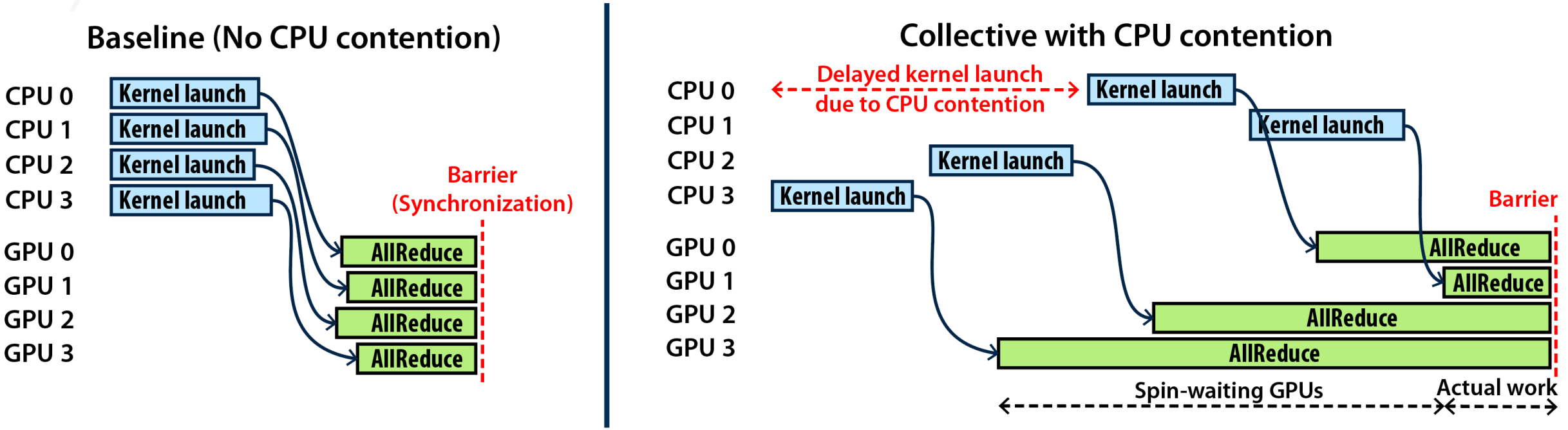
Tokenization Threads Compete for CPU Resources



Georgia Tech

Delayed Kernel Dispatch Leaves GPUs Waiting

Collective kernels (e.g., Allreduce) require every GPU rank to participate.

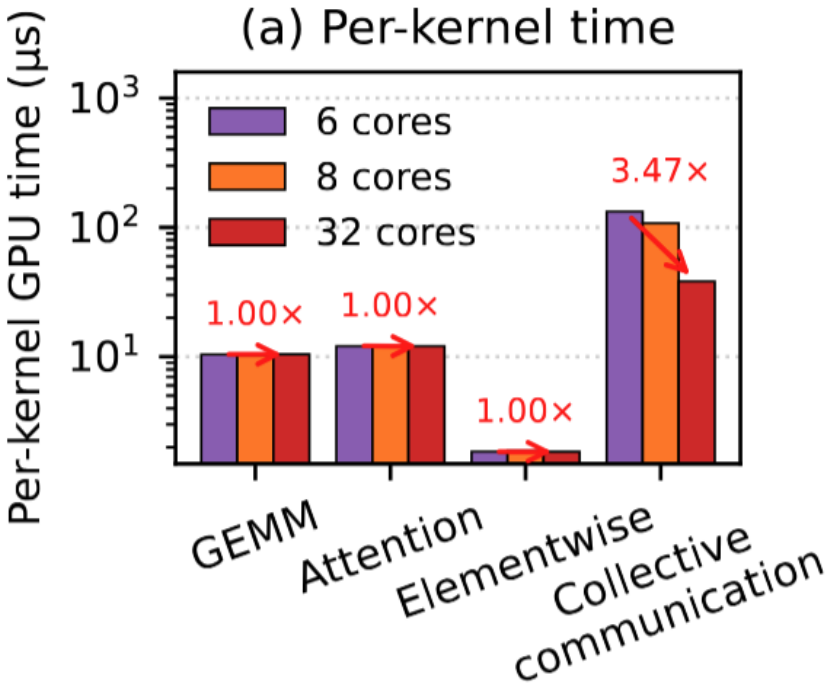


(A) Aligned collective kernel launches

(B) Delayed collective kernel launches, peers wait

CPU oversubscription **delays kernel launches**,
causing **stalls across GPUs** during collective operations.

Collective Latency Is Sensitive to CPU Allocation



Increasing CPU cores from **6 → 32** reduces collective latency by **3.47×.**
Compute kernel latency only changes by **less than 1%**.

Georgia Tech

CPU's Job in Multi-GPU LLM Inference

1. Input processing (Tokenization)

"Who won the 2025 World Series?"

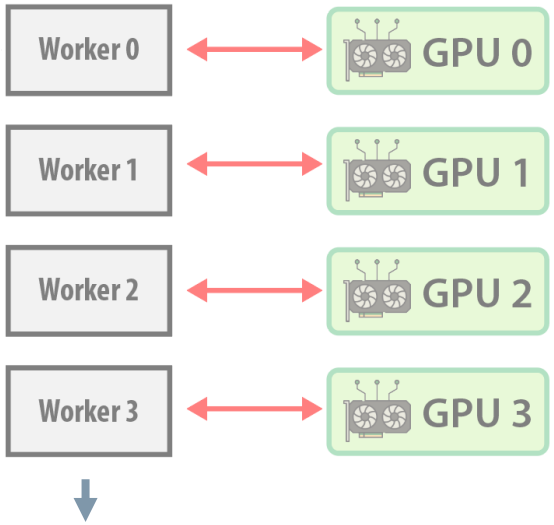
↓

["Who", "won", "the", "202", "5", "World", "Series", "?"]

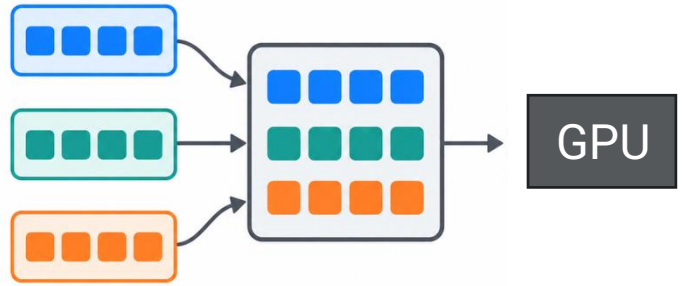
↓

[20600, 3313, 290, 1323, 20, 5922, 14932, 30]

2. Execution of GPU workers



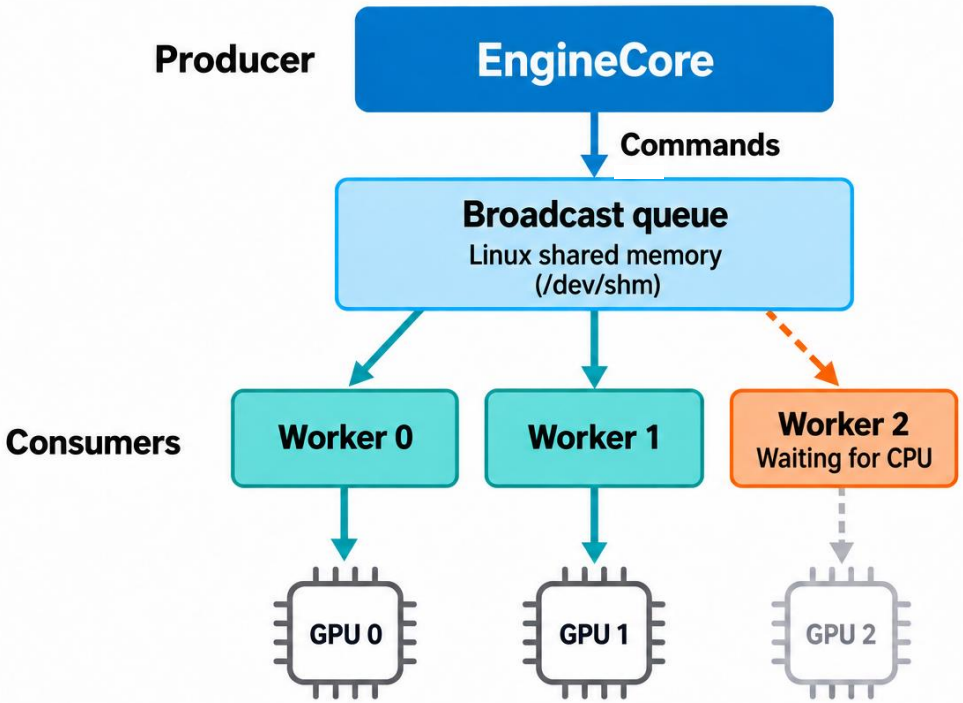
3. Request scheduling & work dispatch



Georgia Tech

CPU Contention Delays Inter-process Communication (IPC)

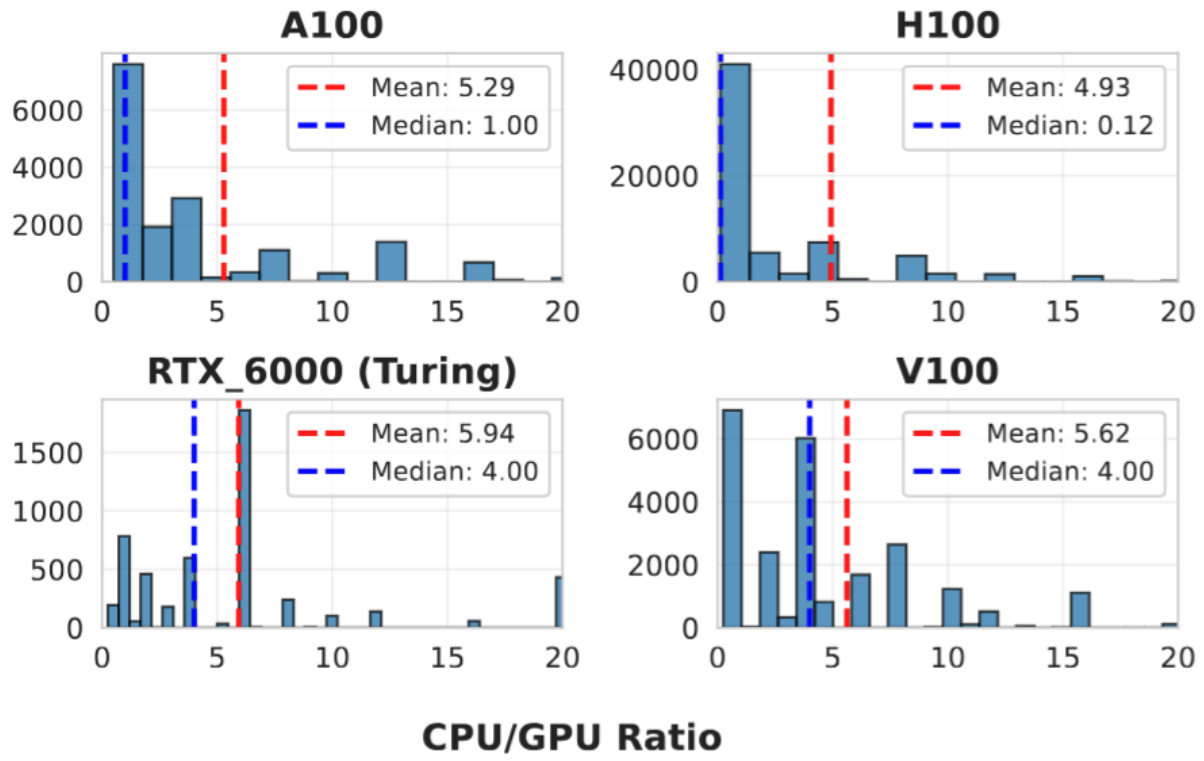
Busy CPU can delay **broadcasts** from the **main engine** to **GPU workers**.



Please refer to the paper for a more detailed analysis and evaluation of the IPC delays.

Case Study: Low CPU Allocations in Real Clusters

Instructional cluster:
users manually specify CPU:GPU ratio



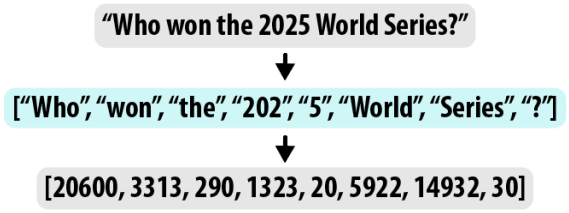
Across **4.65M GPU allocation records** from university clusters, users often allocate only **1–4 CPU cores per GPU**.

More results in the paper

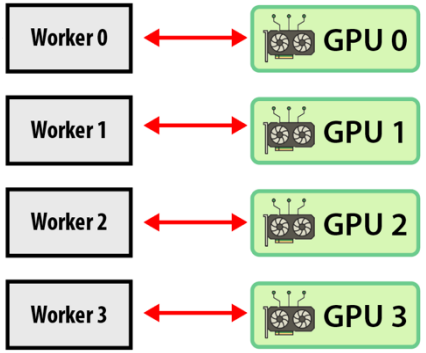
- Evaluation results on **more models** and **configurations**
- **Detailed breakdown** of per-request TTFT latency
- **CPU** and **GPU utilization** reports
- Discussion of the **costs and trade-offs** of **CPU overprovisioning**

Conclusion & Future work

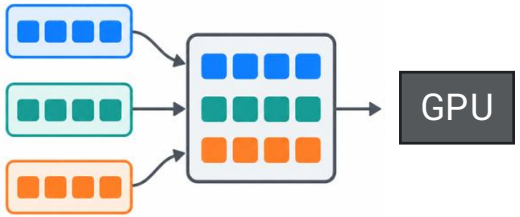
Input processing (Tokenization)



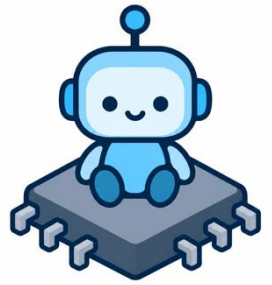
Execution of GPU workers



Request scheduling & work dispatch



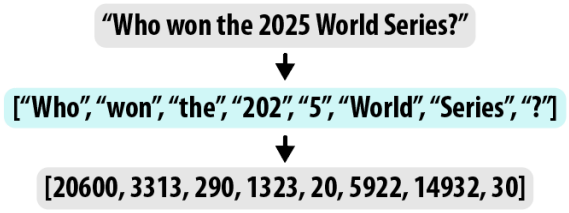
Agentic tools in LLM serving



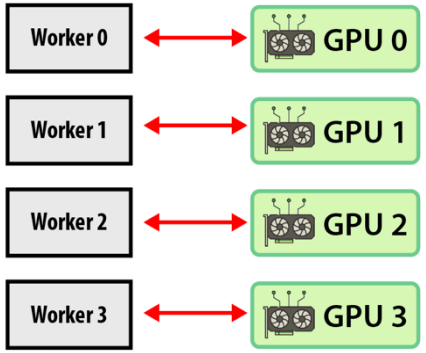
- Growing context in agentic serving increases **CPU tokenization work**
  - Limited CPU resources lead to **up to 7.11× higher TTFT**
- CPU contention delays **tokenization, kernel dispatch, and scheduling broadcasts**, causing end-to-end slowdowns
- **CPU-intensive agentic tools** may add further host-side contention

Questions?

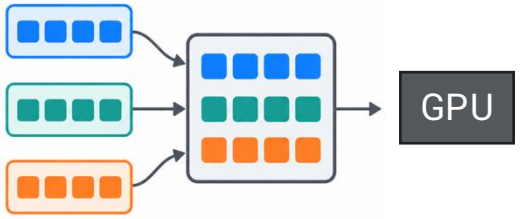
Input processing (Tokenization)



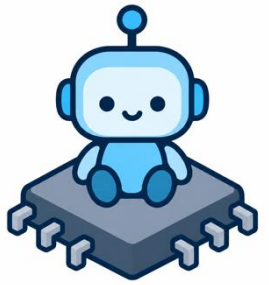
Execution of GPU workers



Request scheduling & work dispatch



Agentic tools in LLM serving



Presenter: Euijun Chung (euijun@gatech.edu)

Paper

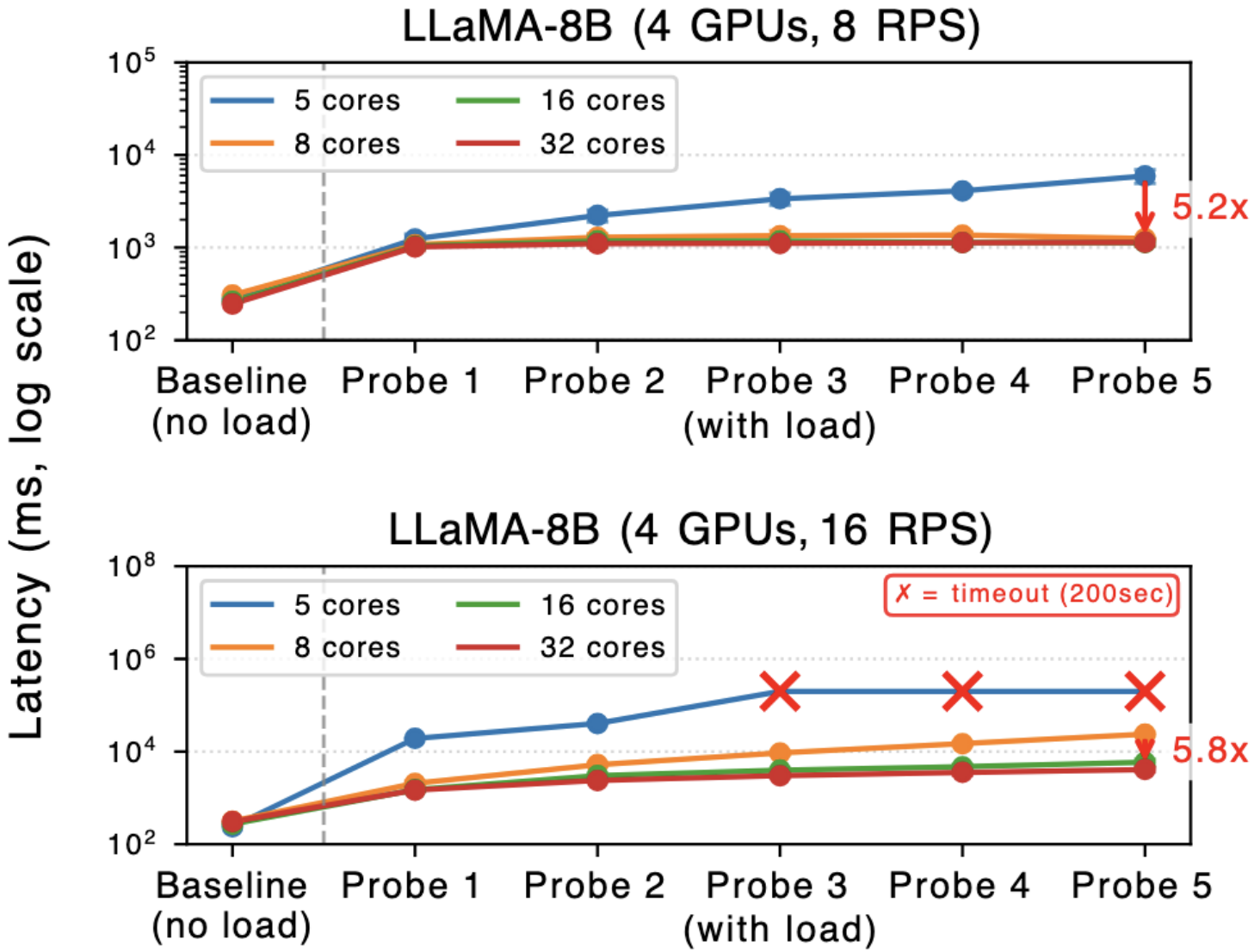
Backup slides

Evaluation Setups

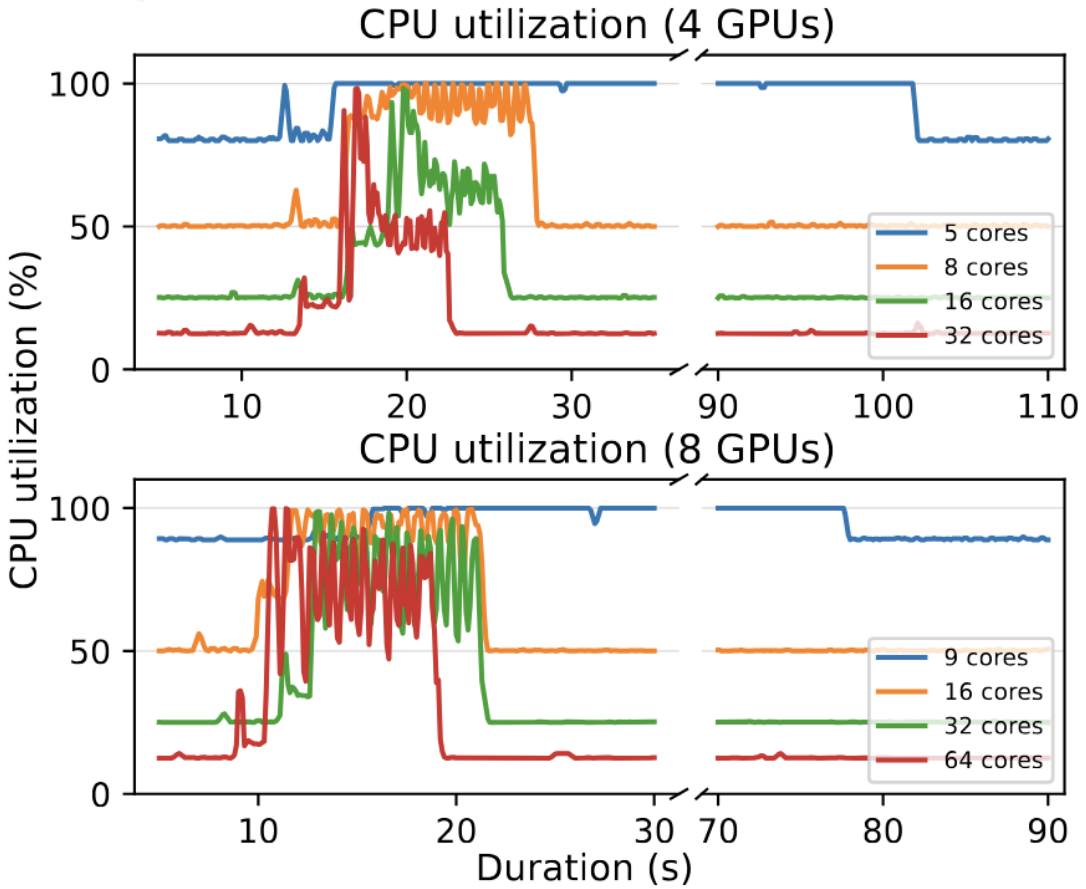
TABLE I
LIST OF CPU-GPU HETEROGENEOUS SYSTEM SETUPS USED IN THIS PAPER'S EVALUATION.

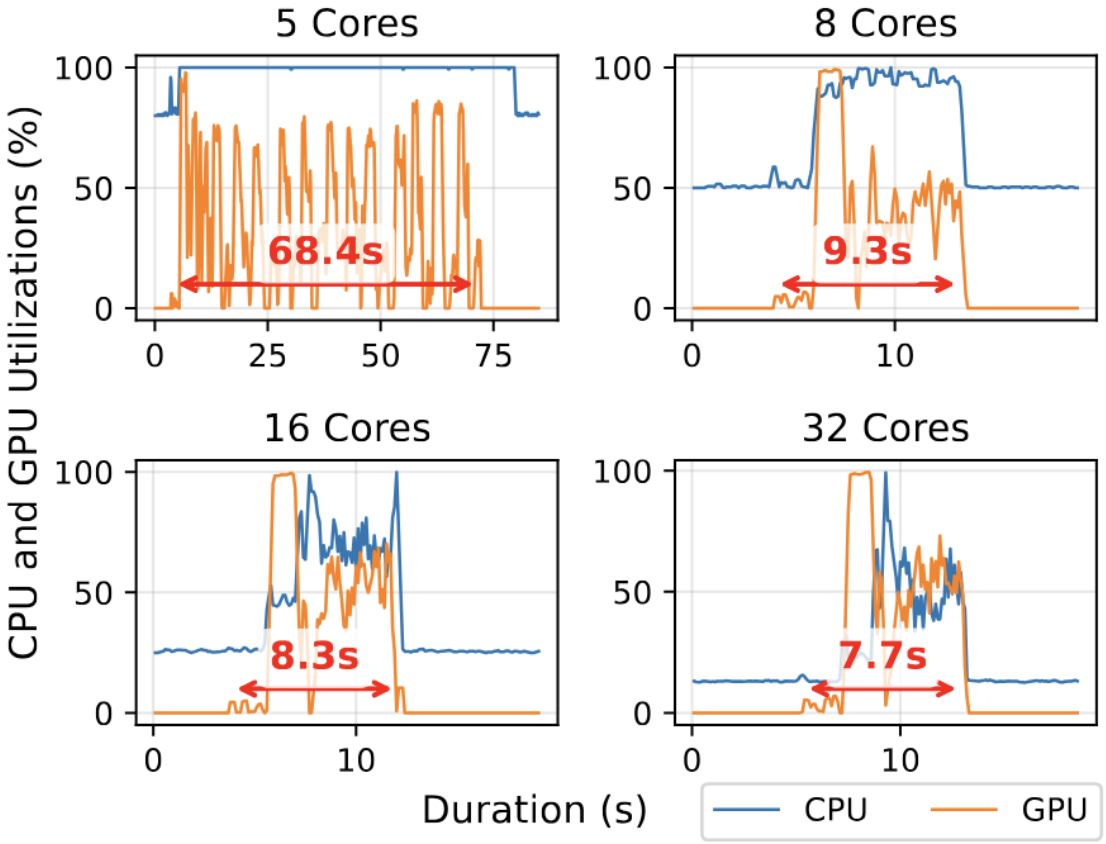
System (GPU)	Architecture (Compute Capability)	CPU Model	#CPU Cores	#GPUs per Node	Inter-GPU Interconnect
H100	Hopper (9.0)	Intel Xeon Platinum 8462Y+	64	8	NVLink 4.0 (900 GB/s bidirectional)
H200	Hopper (9.0)	Intel Xeon Platinum 8562Y+	64	8	NVLink 4.0 (900 GB/s bidirectional)
RTX Pro 6000	Blackwell (12.0)	Dual Intel Xeon 6737P	64	8	No NVLink (PCIe 5.0, 64 GB/s)

Probe latency breakdown



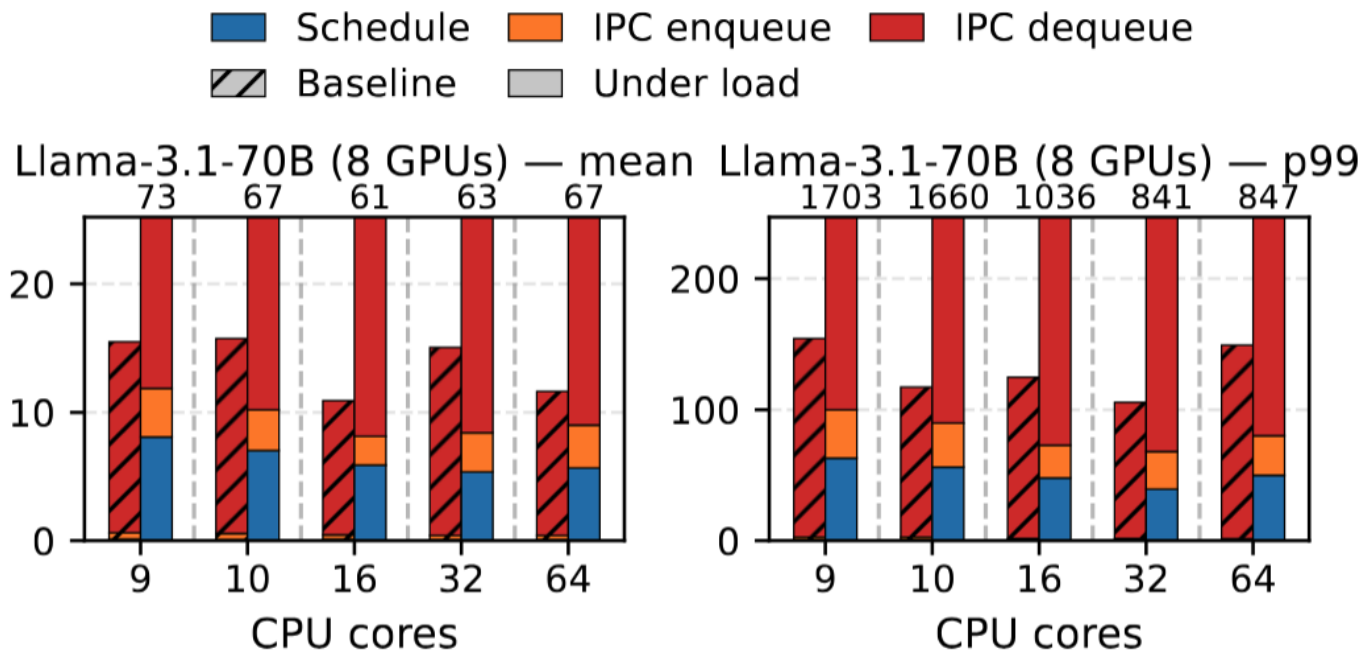
CPU/GPU utilization





Georgia Tech

CPU Contention Increases Broadcast-Path Delays



Under load, CPU-side per-step time rises from **2–16 ms** to **46–73 ms**.

→ Additional CPU cores reduce tail (p99) delays, but the mean stays.