Characterizing Power Management Opportunities for LLMs in the Cloud

Pratyush Patel

Esha Choukse Chaojie Zhang Íñigo Goiri Brijesh Warrier
Nithish Mahalingam Ricardo Bianchini


ChatGPT
Gemini
GitHub
Copilot

ChatGPT Gemini

GitHub Copilot

Microsoft places huge cap-ex bets on datacenters for cloud and AI

CFO

Google Cloud braces for AI compute costs, ramps up data center inves

Zuckerberg's Meta Is Spending Billions to Buy 350,000 Nvidia H100 GPUs

In tc
deve

Amazon Aims for AI Supremacy With $8B Data Surge in Ohio

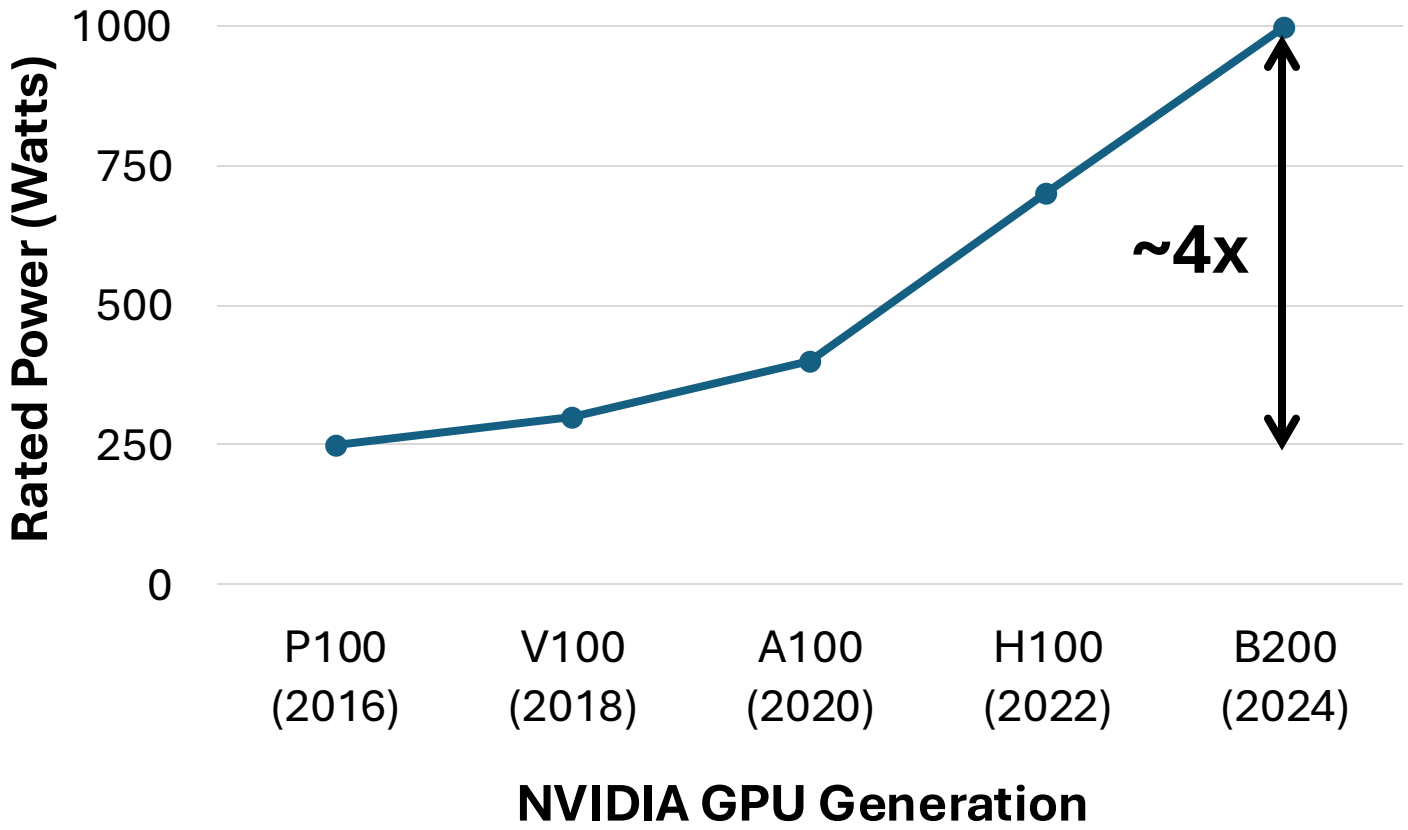
Amazon Web Services also is building $35B in new data center capacity in Virginia.



ChatGPT Gemini

GitHub Copilot


GPU clusters for LLMs are incredibly power hungry



ChatGPT Gemini

GitHub Copilot

Big Tech's Latest Obsession Is Finding Enough Energy

The AI boom is fueling an insatiable appetite for electricity, which is

Data Centers in Demand Despite Global Power Limitations

U.S. Power Grid Struggles to Keep Up with Data Center Growth

Power output will need to double to keep pace with voracious demand for electricity.

Addressing the power wall for LLMs at scale

Profile power usage patterns of training and inference workloads in production clusters

Analyze design implications for power management in cloud deployments

Build a power oversubscription framework that safely adds ~30% more servers in inference clouds

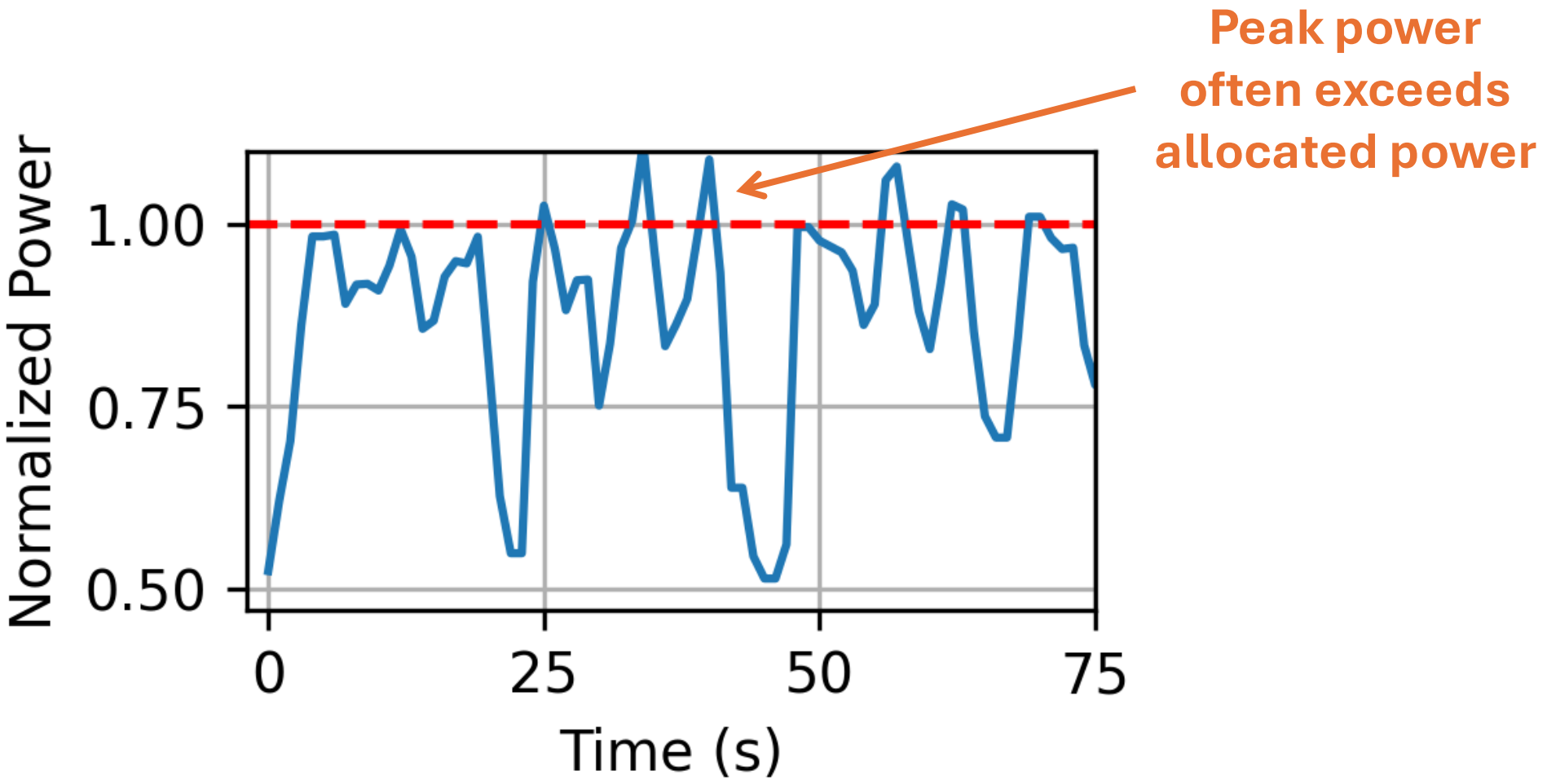
Characterizing Power Management Opportunities for LLMs in the Cloud

Power usage patterns of LLMs in production

Design implications for cloud deployments

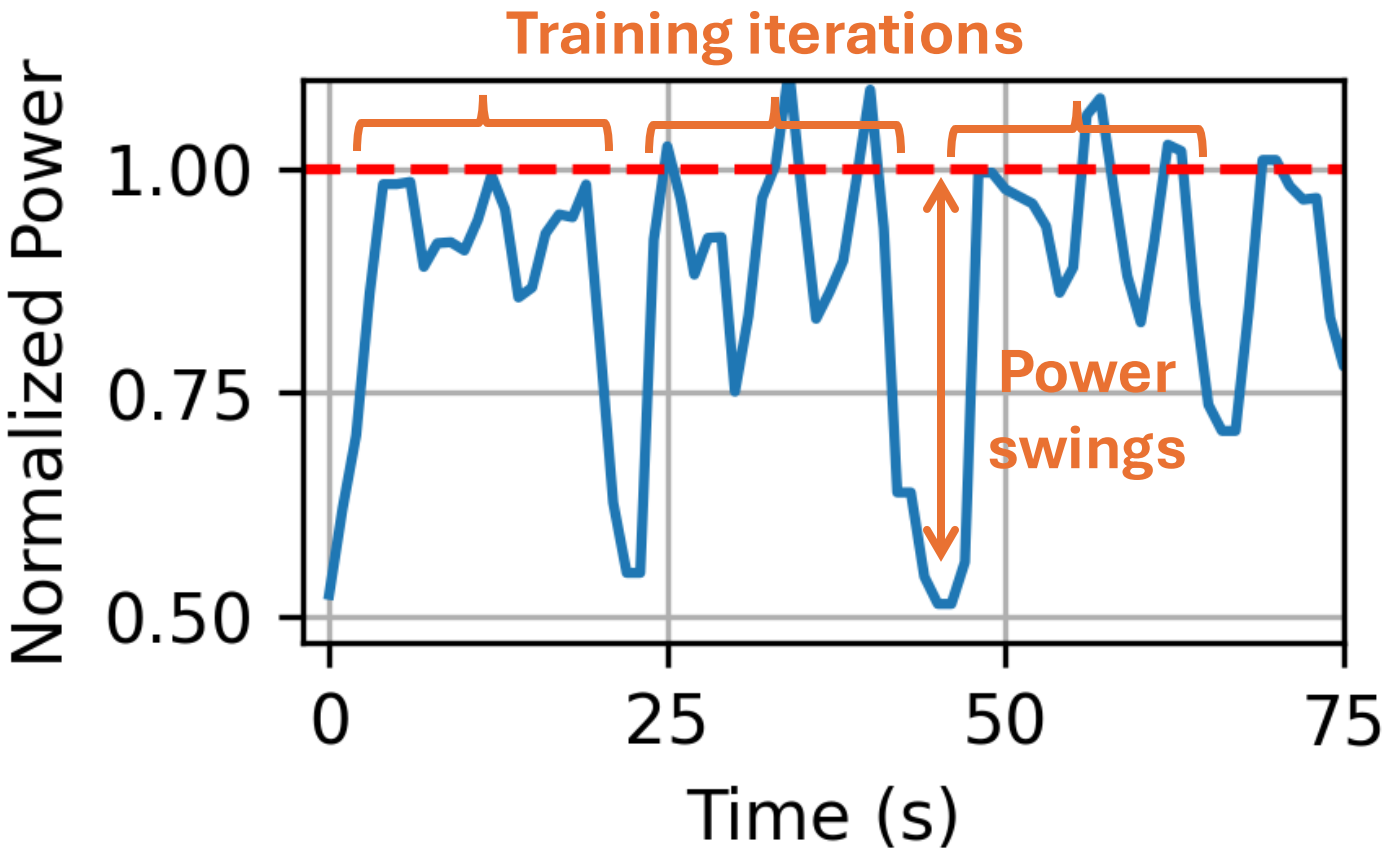
Power oversubscription for LLM inference clouds

Training power usage patterns



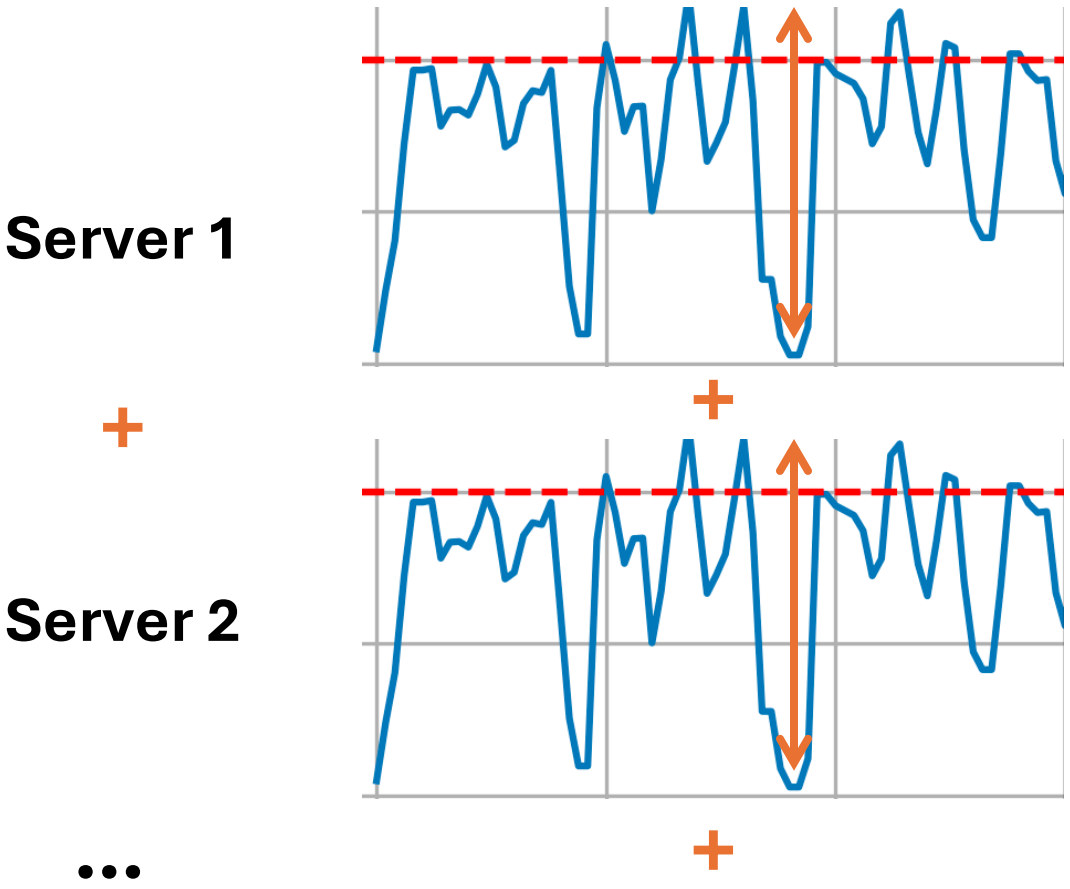
LLM fine-tuning on 8 A100 GPUs

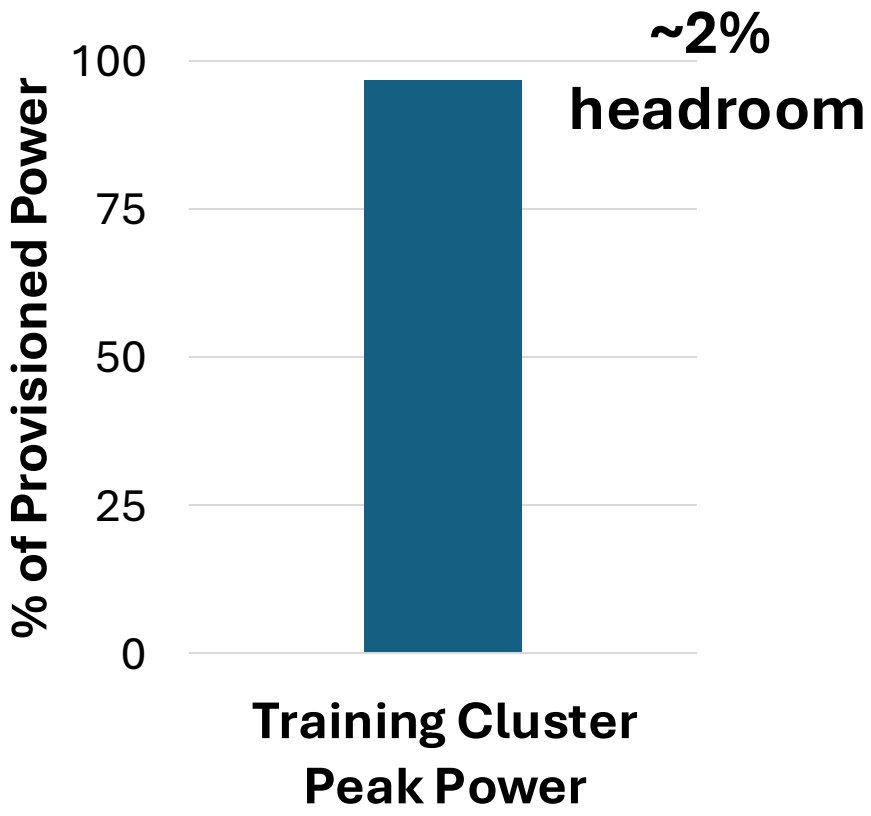
Training power usage is periodic



LLM fine-tuning on 8 A100 GPUs

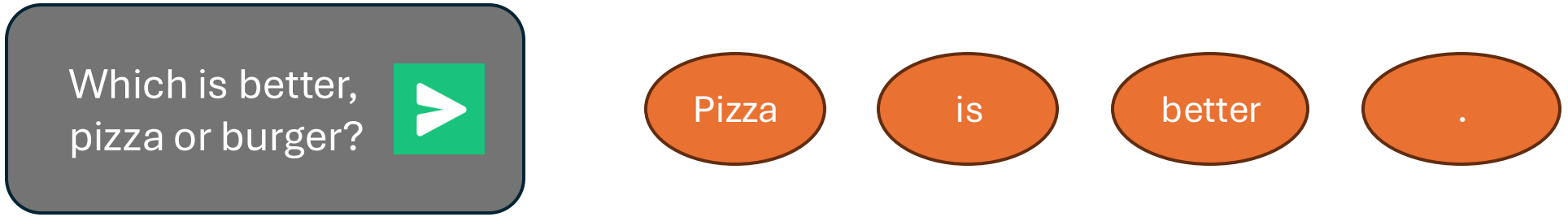
Training clusters have little power headroom





Due to synchronized computation and communication across thousands of GPUs

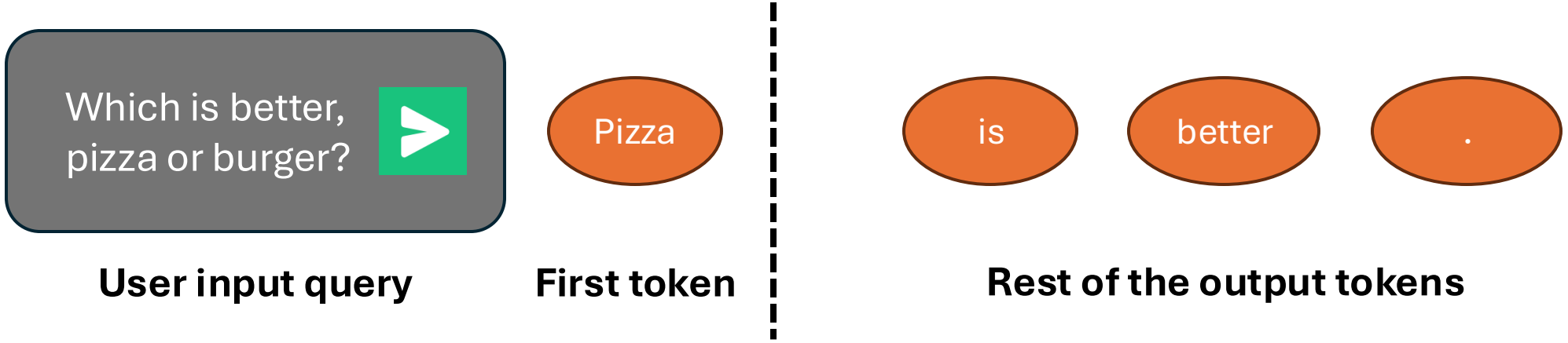
Inference requests have two compute phases



User input query

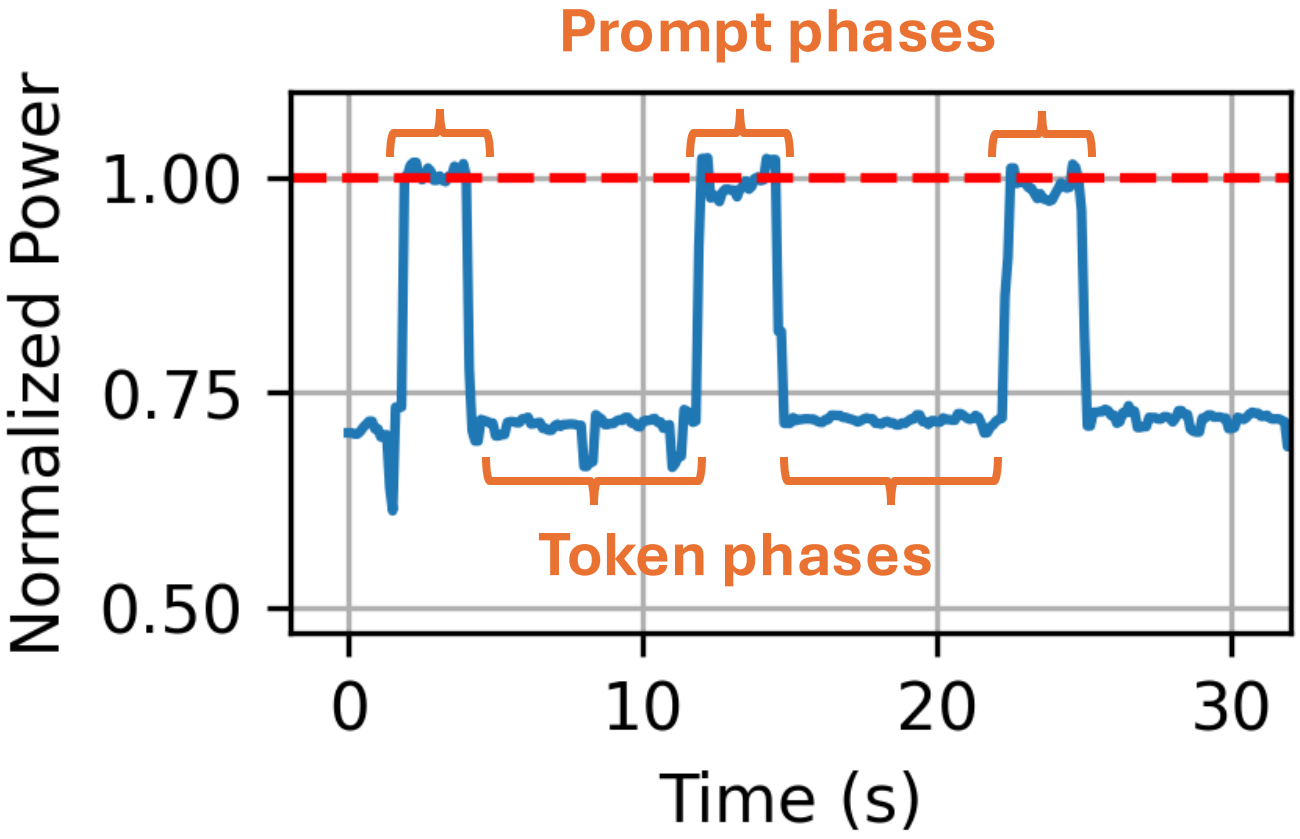
LLM response (output tokens)

Inference requests have two compute phases



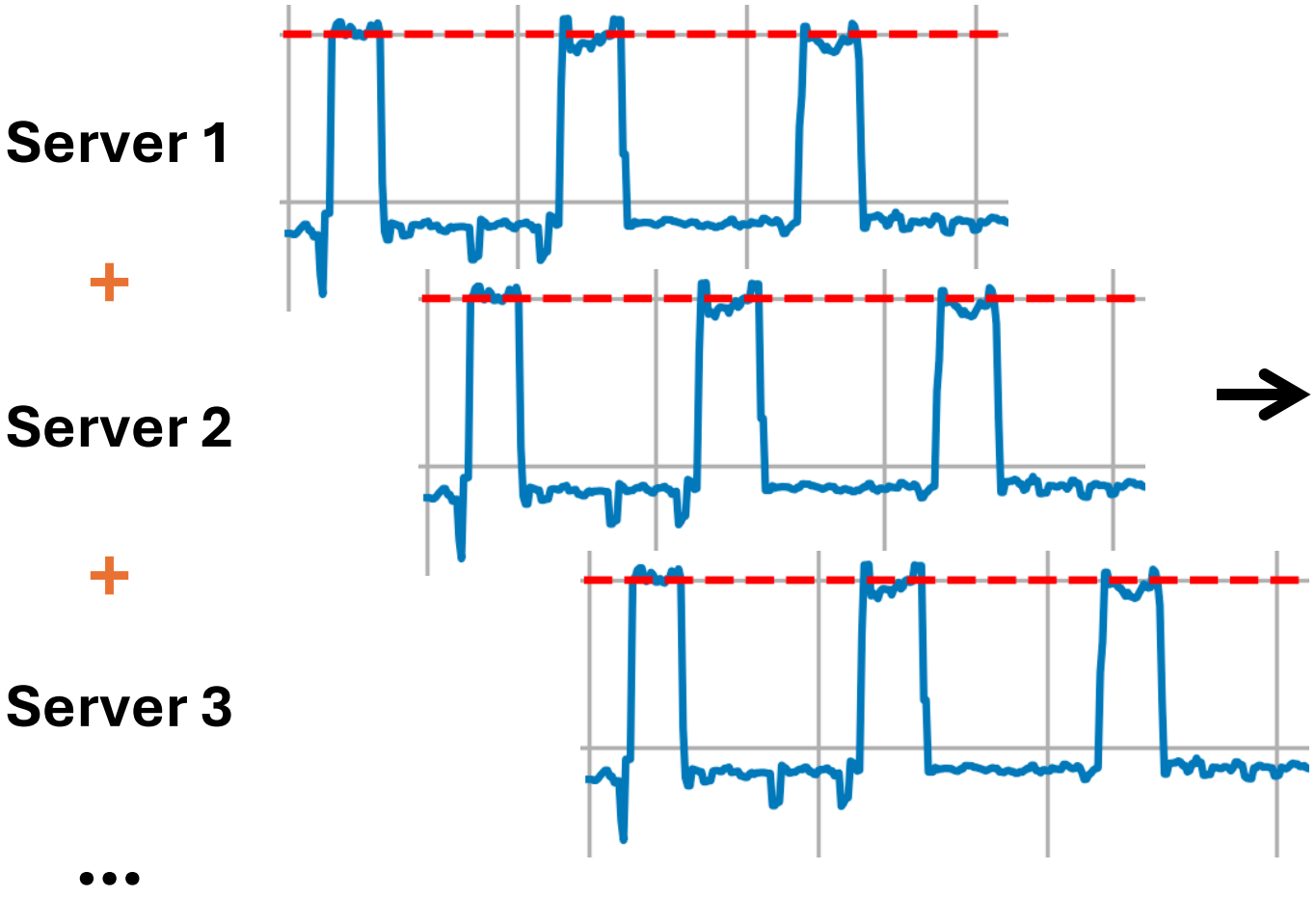
Prompt phase	Token phase
User input processed in parallel	Serialized token generation
Compute intensive	Memory intensive

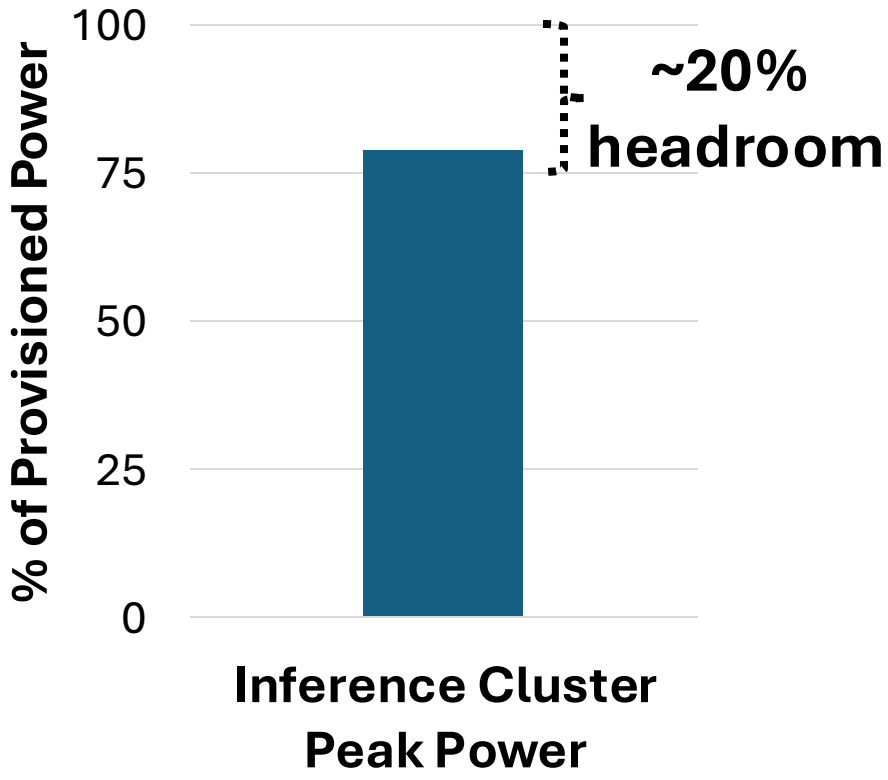
Each phase has distinct power draw patterns



3x LLM inference requests on 8 A100 GPUs

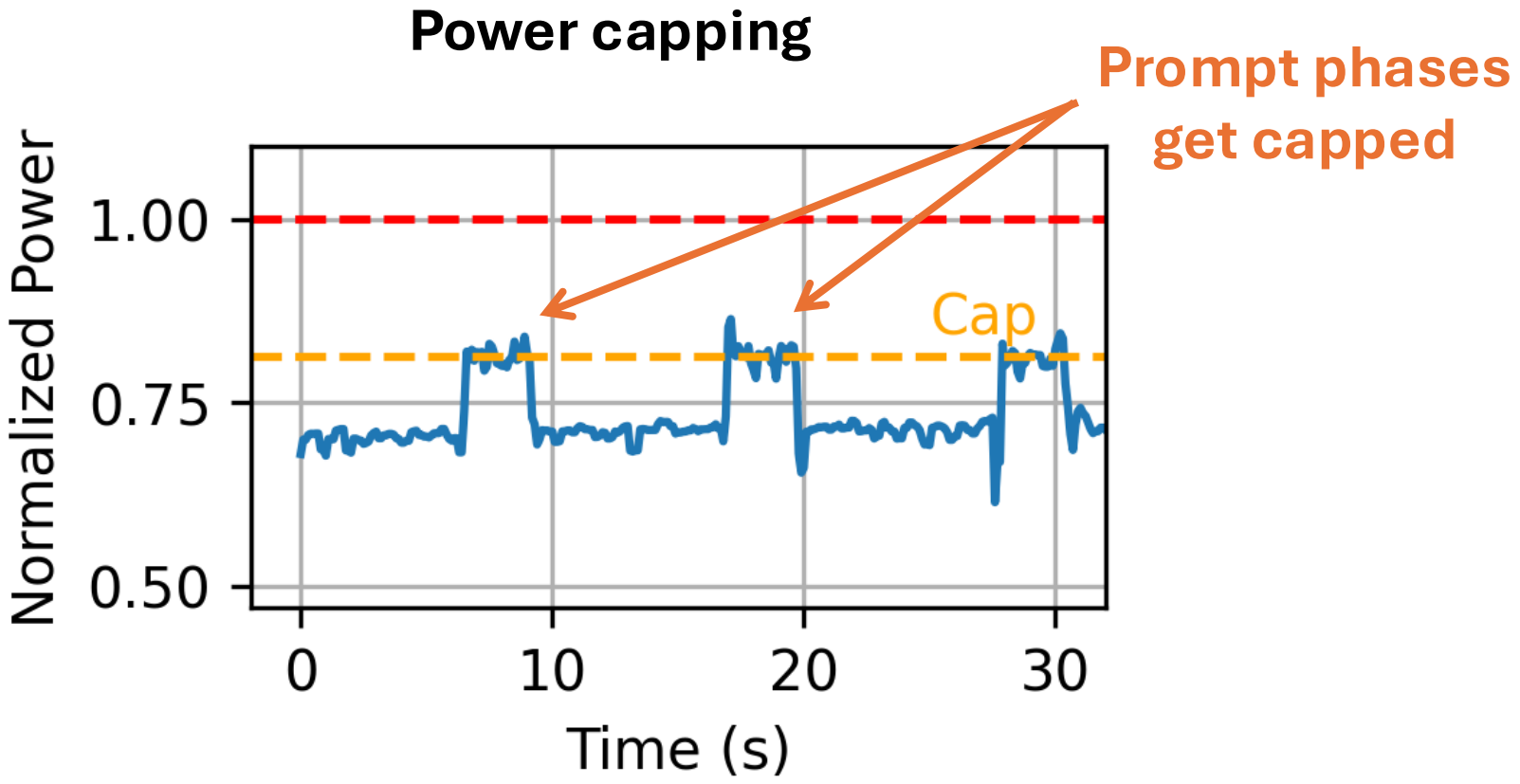
Inference clusters underutilize power



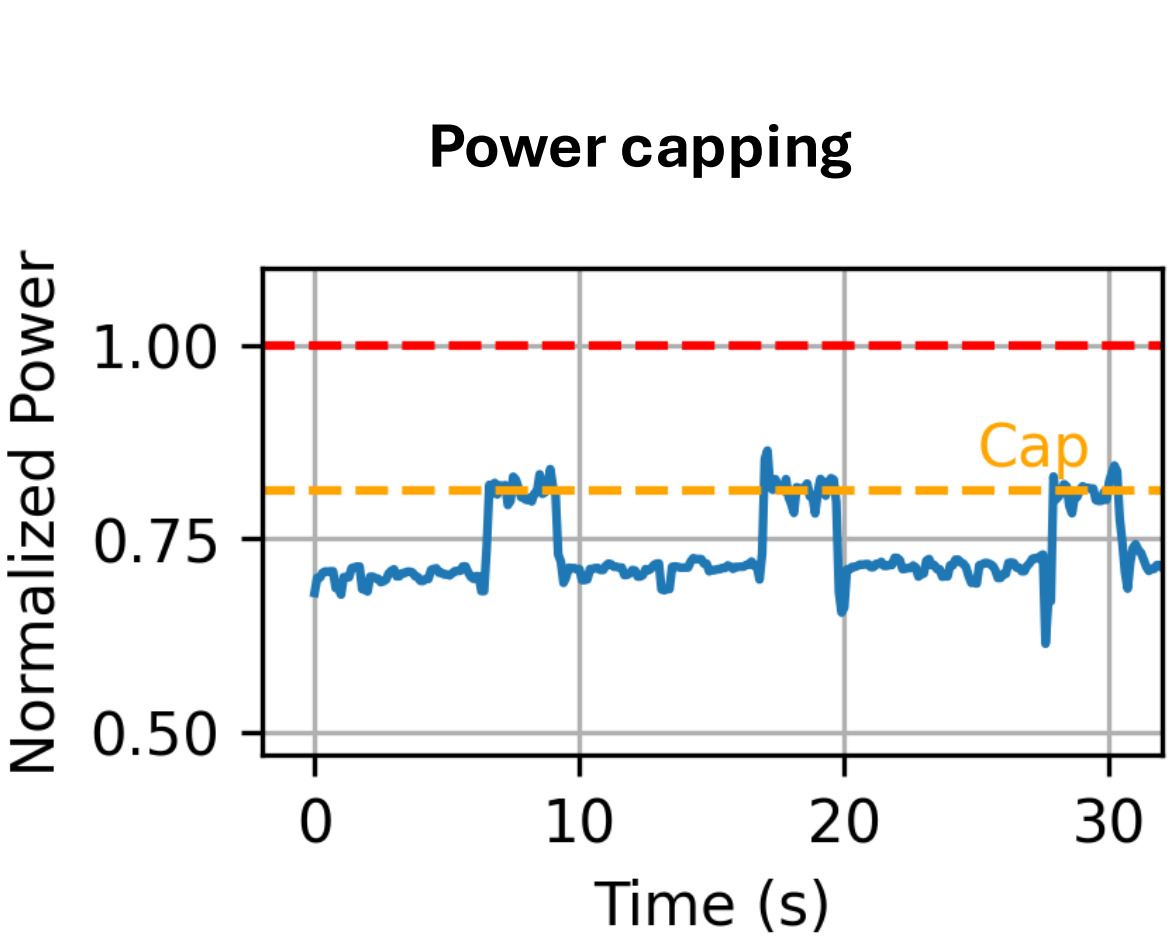


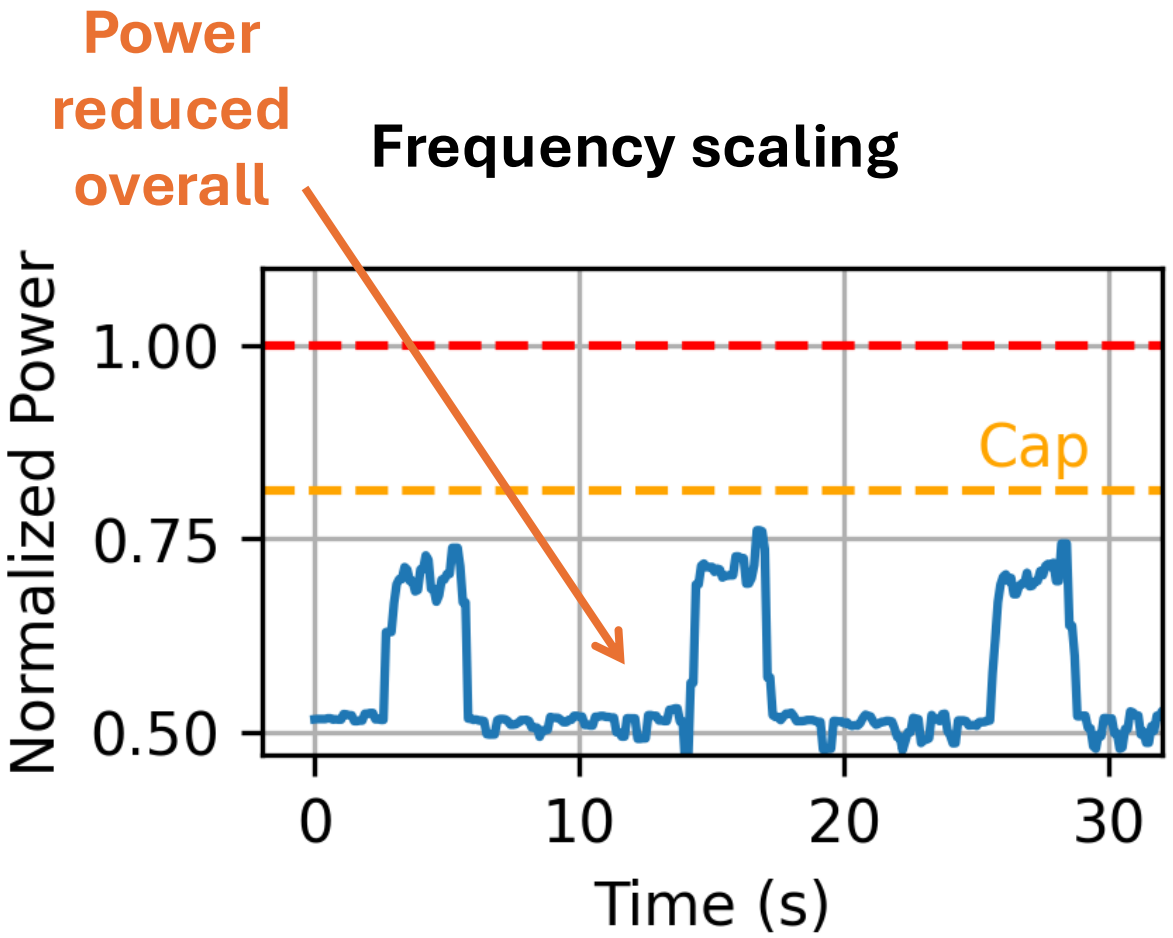
Due to the statistical multiplexing of many prompt and token phases

GPU power management knobs in the cloud

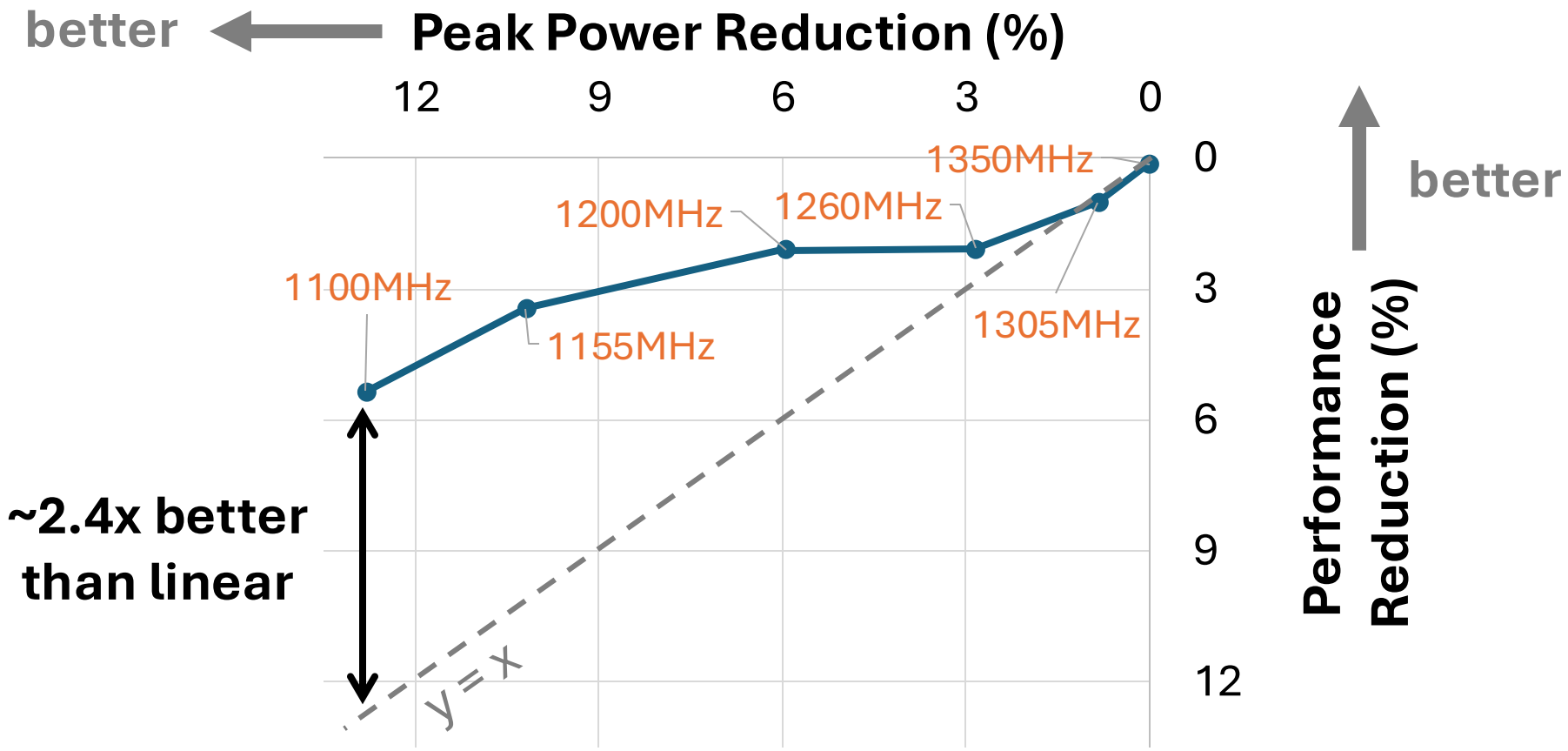


GPU power management knobs in the cloud



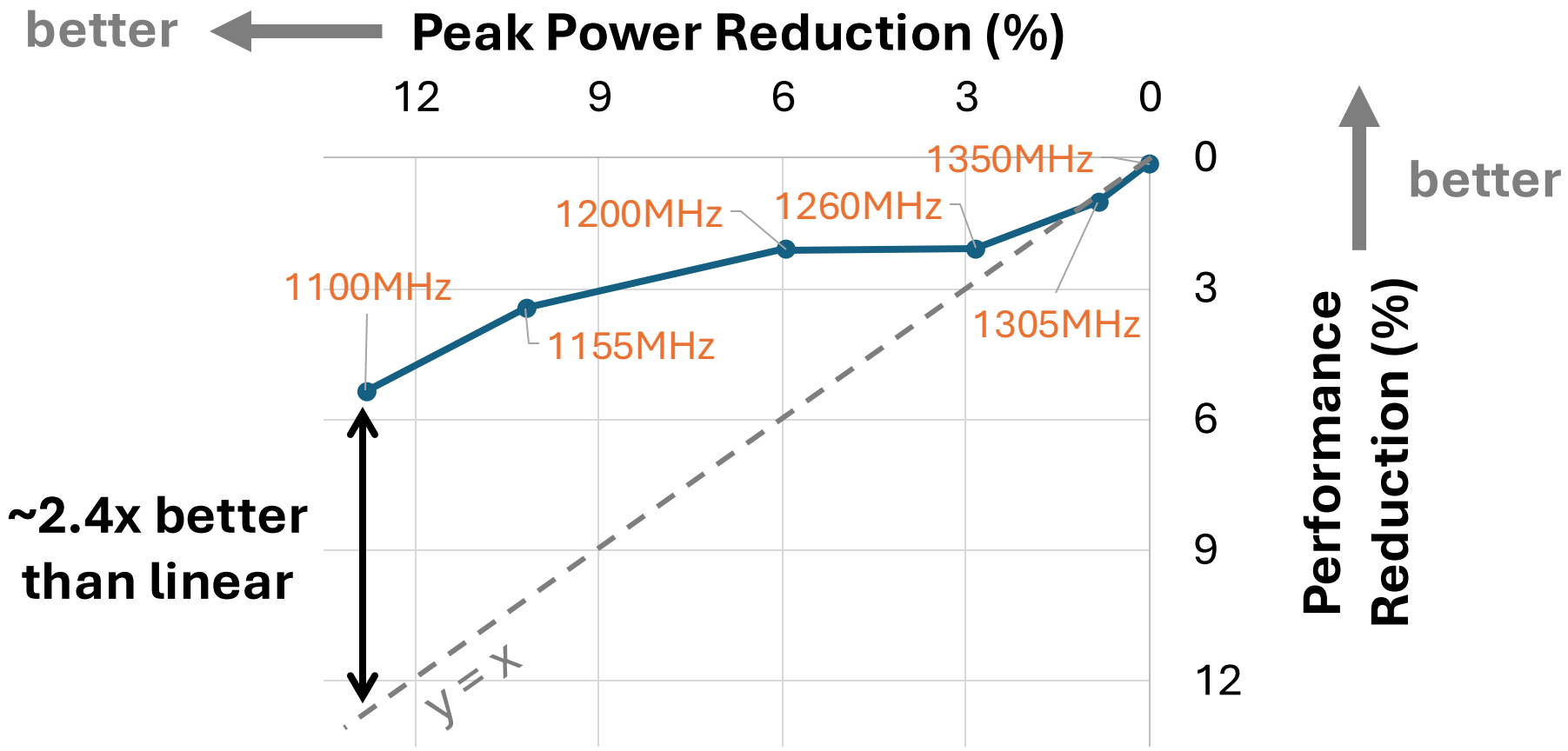


Performance impact of frequency scaling



LLM inference on 8 A100 GPUs

Frequency scaling is effective for inference



Can reclaim substantial power with low performance loss

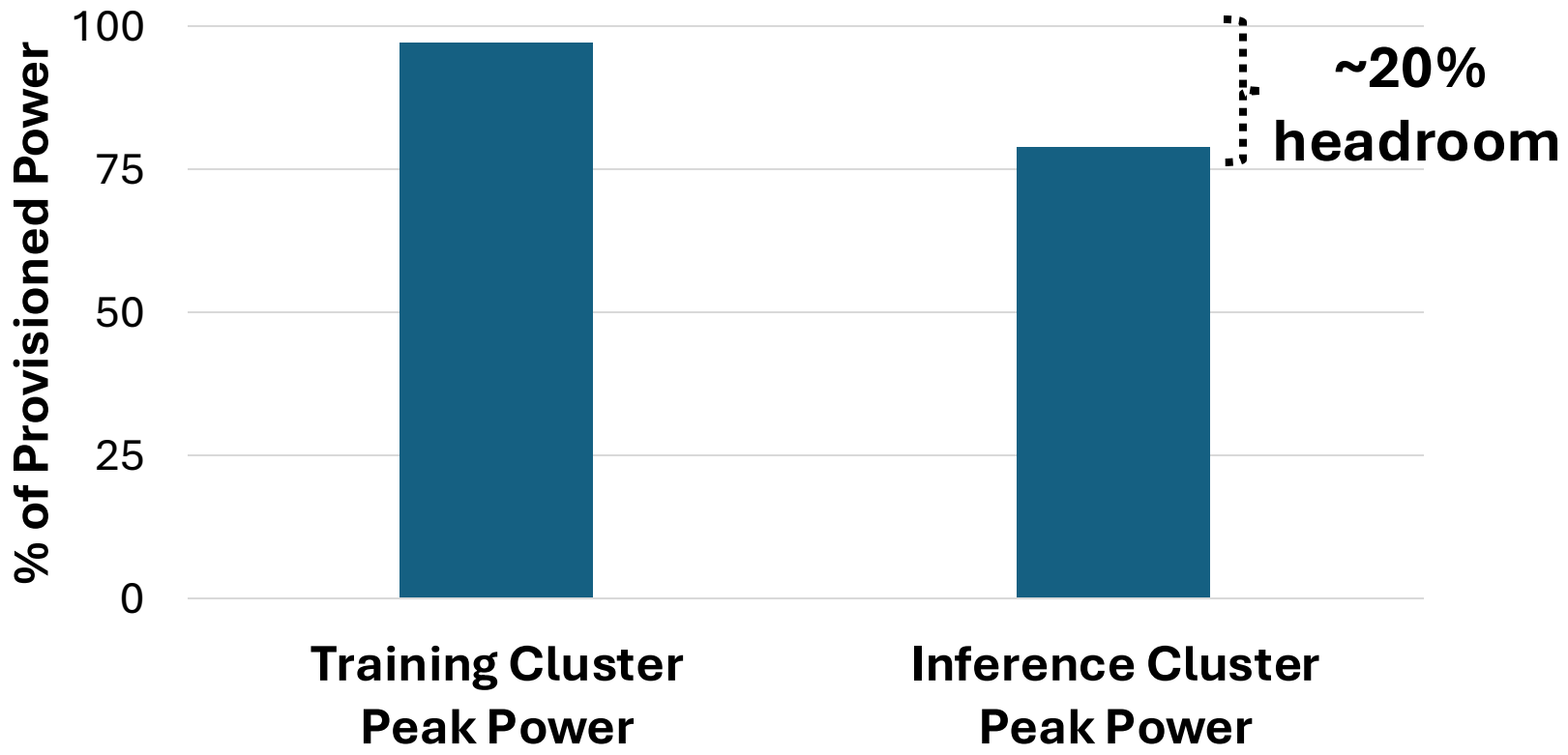
Characterizing Power Management Opportunities for LLMs in the Cloud

Power usage patterns of LLMs in production

Design implications for cloud deployments

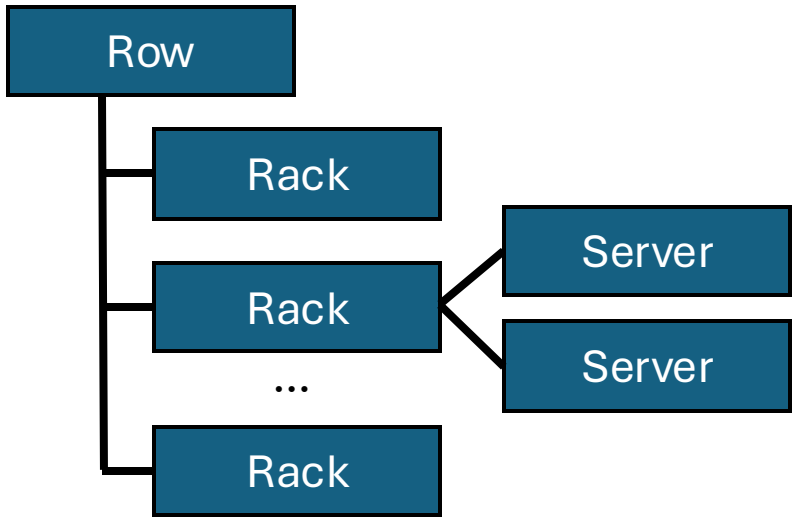
Power oversubscription for LLM inference clouds

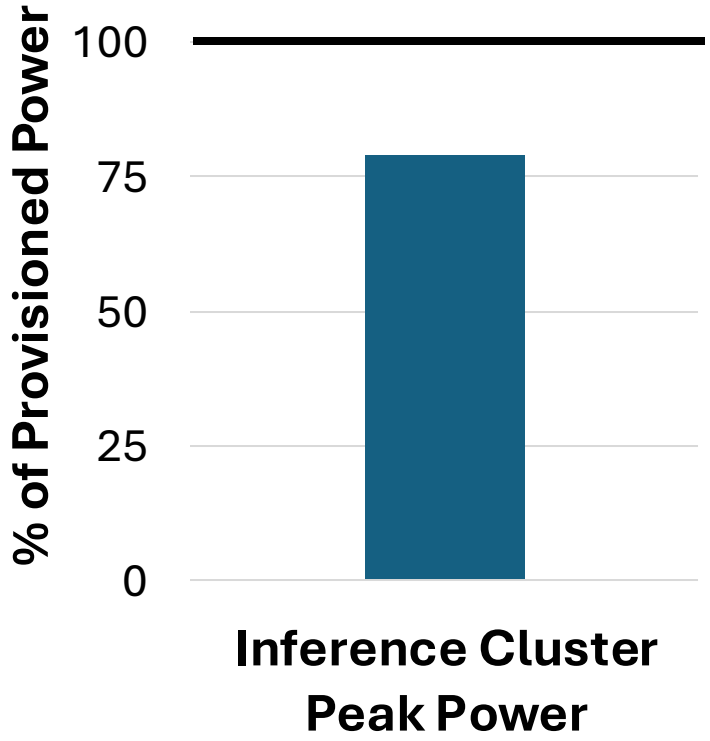
Deploy more servers under a power budget?



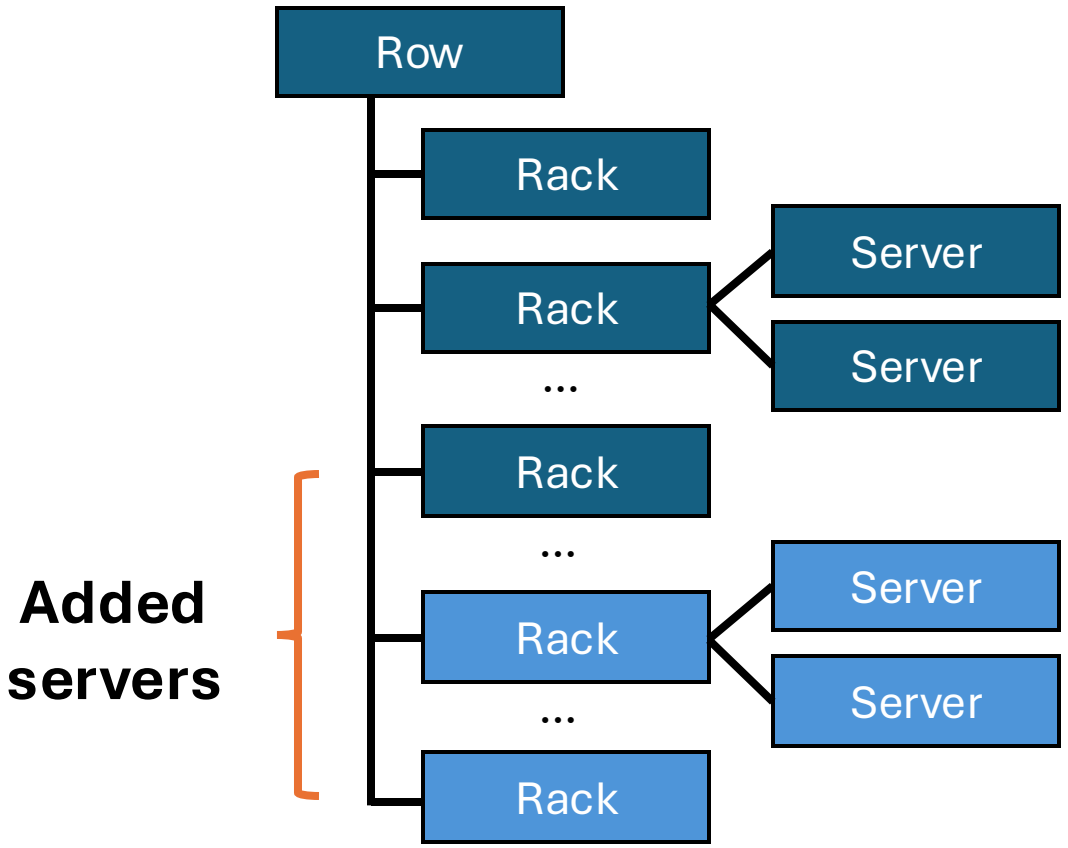
Inference makes up most of the LLM compute demand

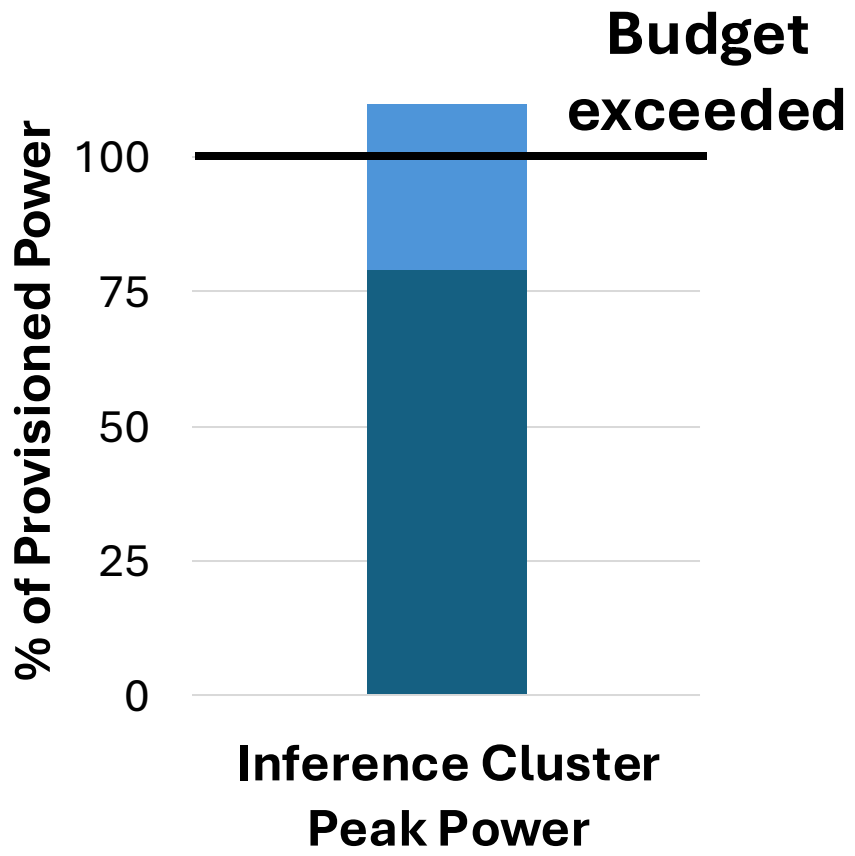
Deploy more servers in inference clusters?





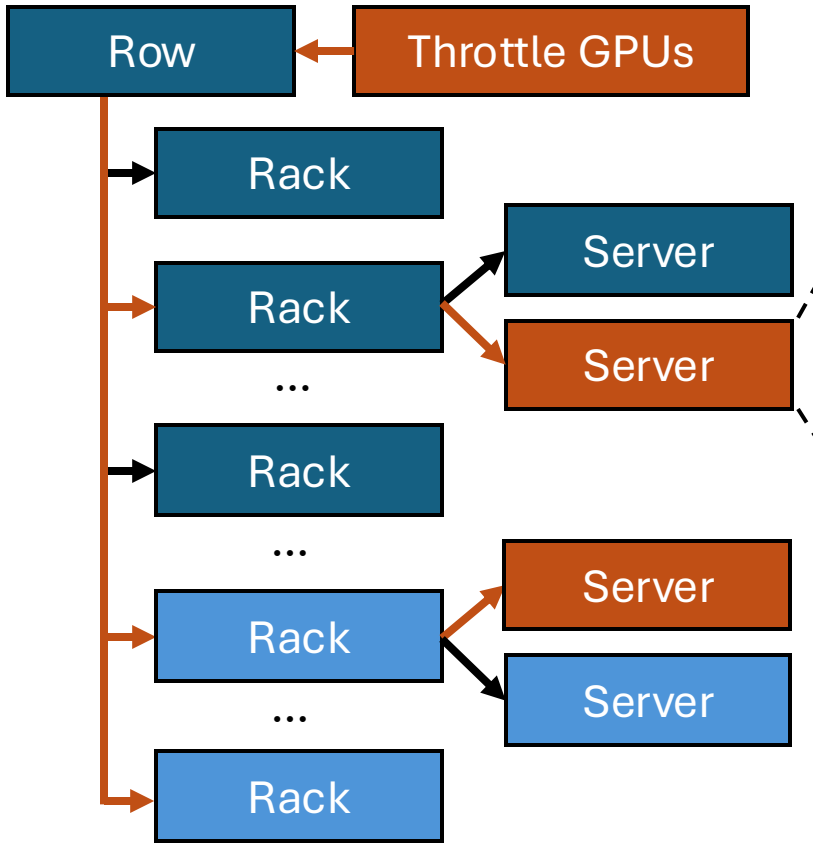
More servers could exceed the power budget

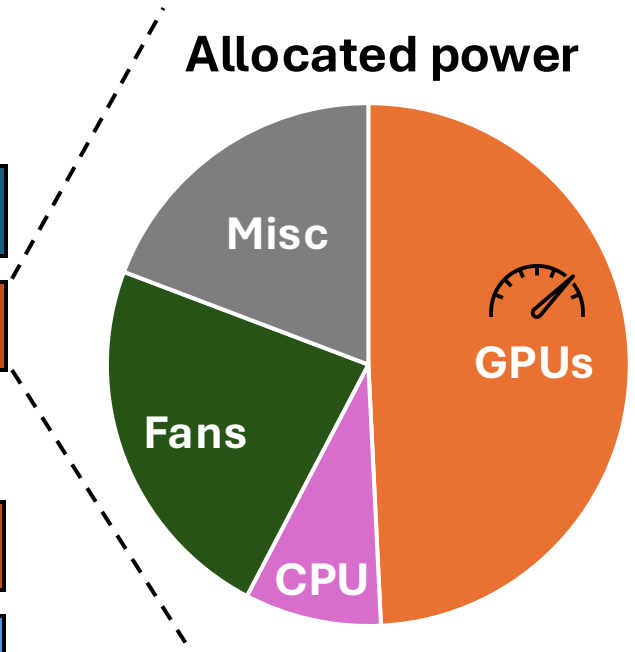


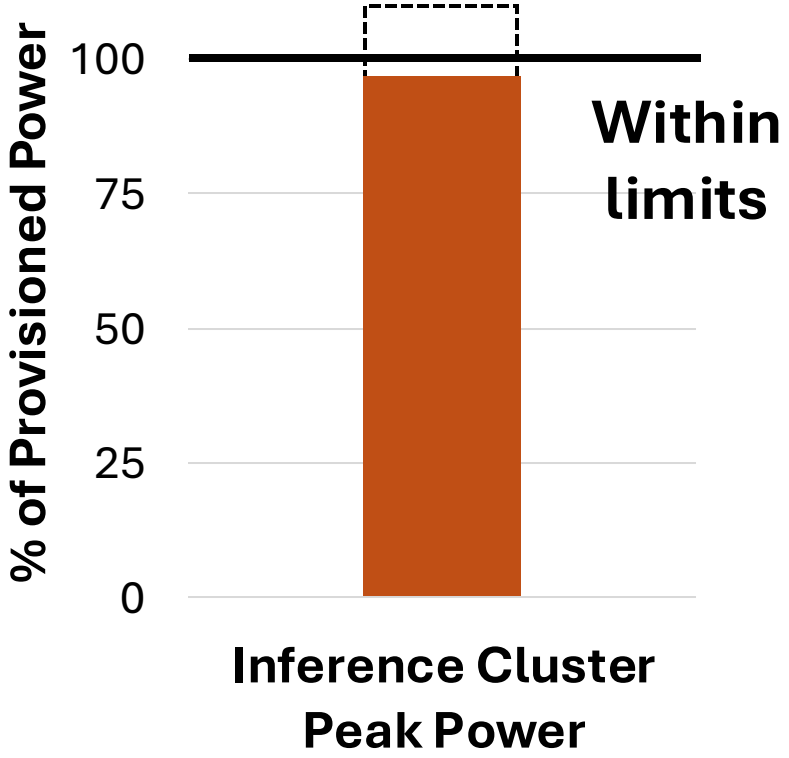


Need to quickly reduce cluster power usage to prevent power failures

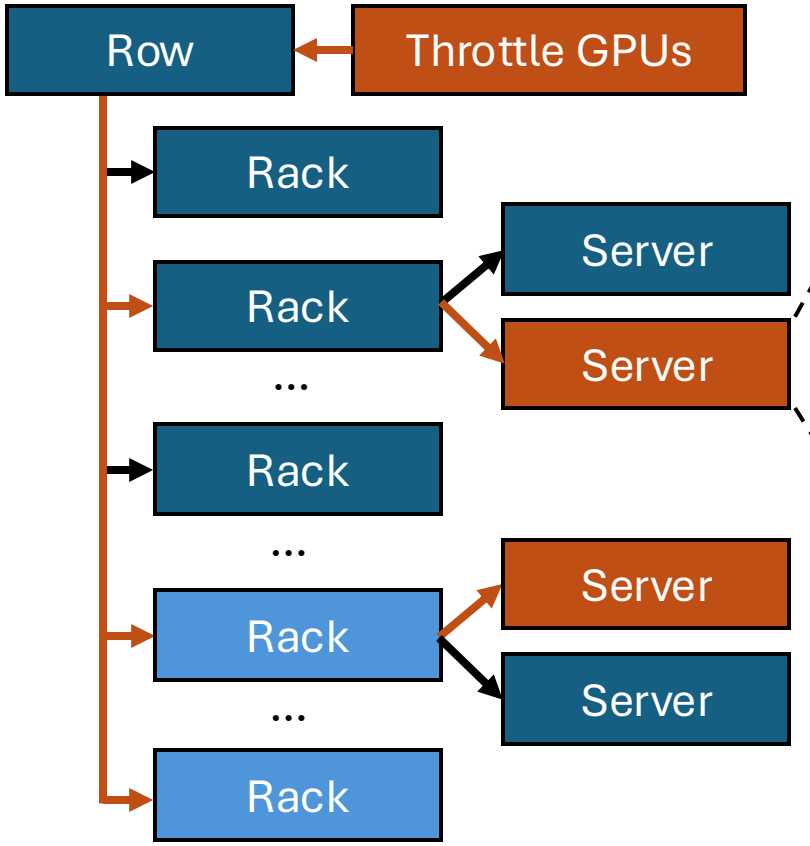
GPU power throttling could help!

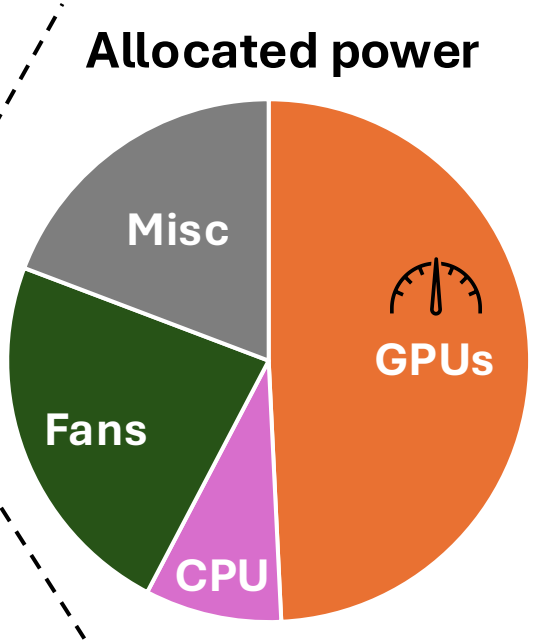






Cloud GPU throttling knobs are too slow

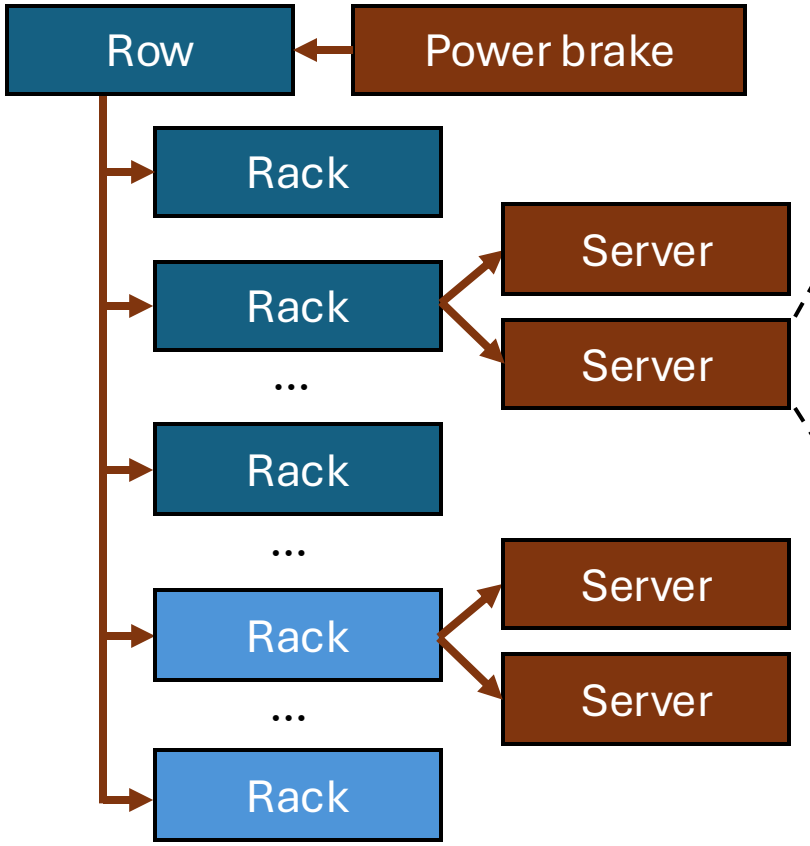


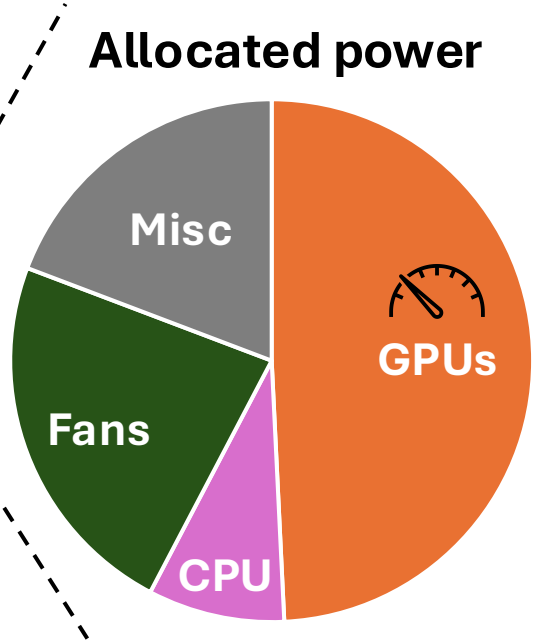


~40 seconds to throttle a server

Much slower than the ~10 seconds deadline to reduce power usage

Power brake works but is too extreme





~5 seconds to throttle all servers

Quickly throttles *all* GPUs to a very low power and performance

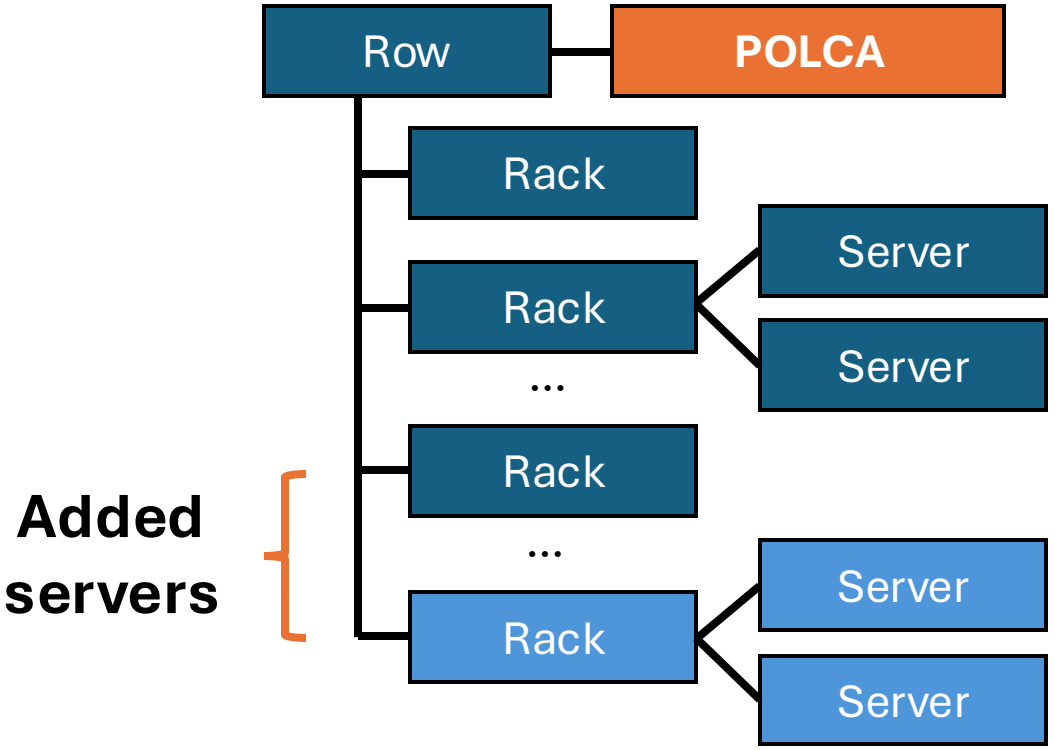
Characterizing Power Management Opportunities for LLMs in the Cloud

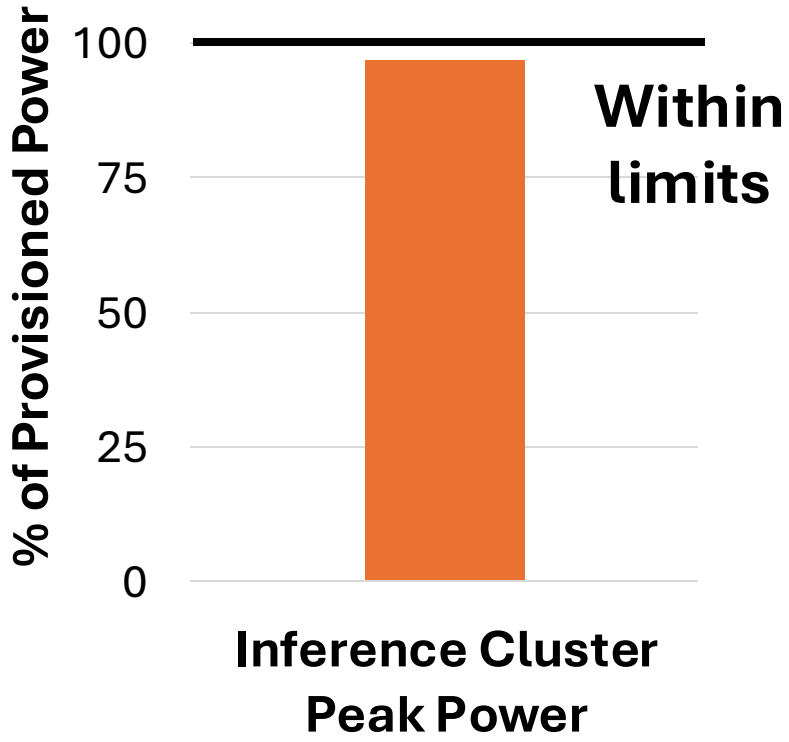
Power usage patterns of LLMs in production

Design implications for cloud deployments

Power oversubscription for LLM inference clouds

POLCA helps safely deploy more servers





With minimal performance impact on latency-critical LLM inference workloads

Inputs: workload priorities

POLCA

Diverse workloads & pricing tiers

Gemini

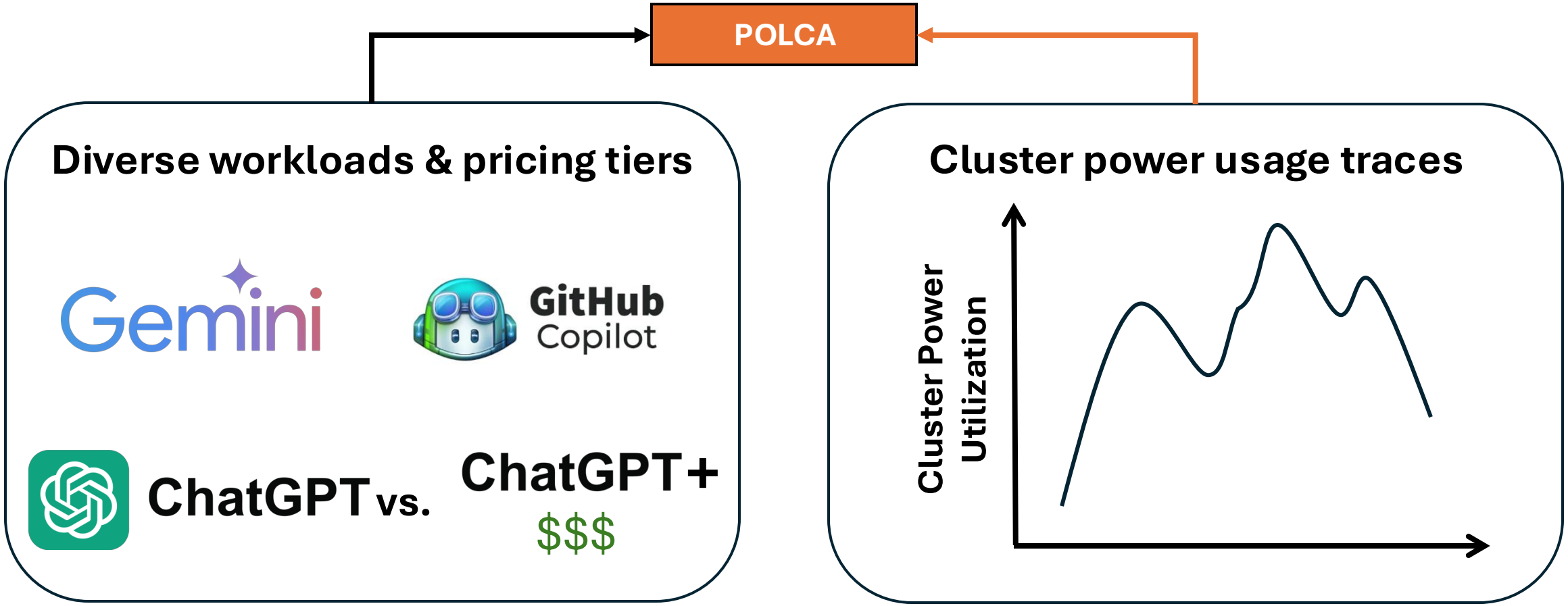
GitHub Copilot

ChatGPT vs. ChatGPT+

$$$

To capture the latency sensitivity of different workloads

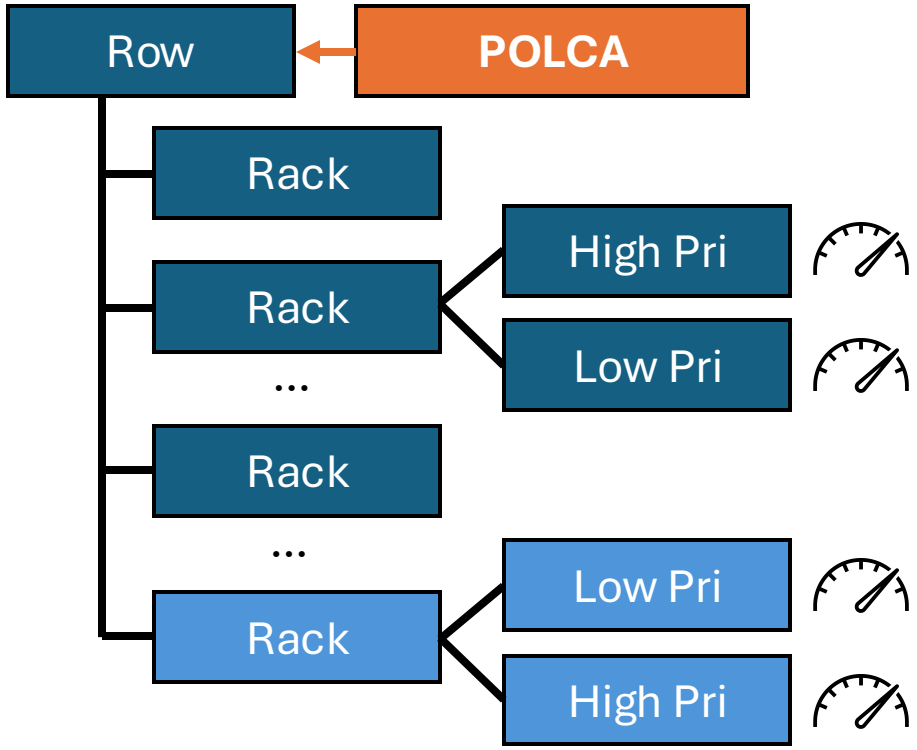
Inputs: workload priorities and power traces

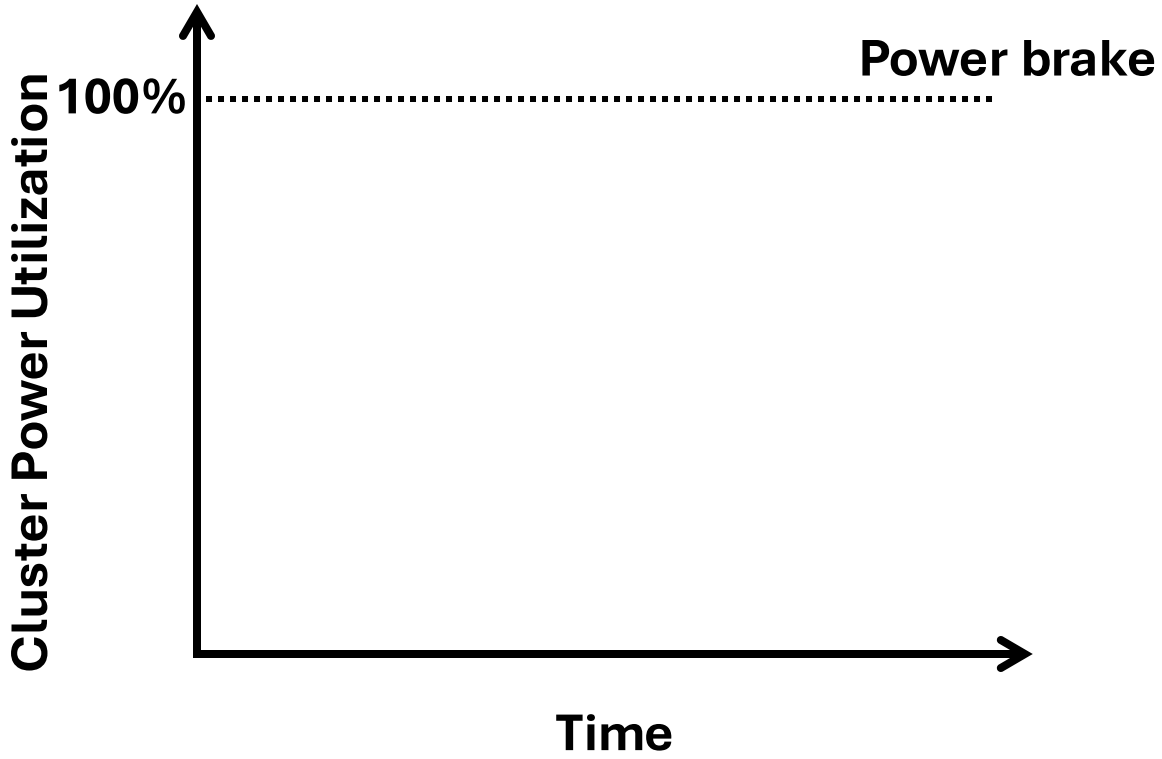


To capture the latency sensitivity of different workloads

To infer workload variability and power usage patterns

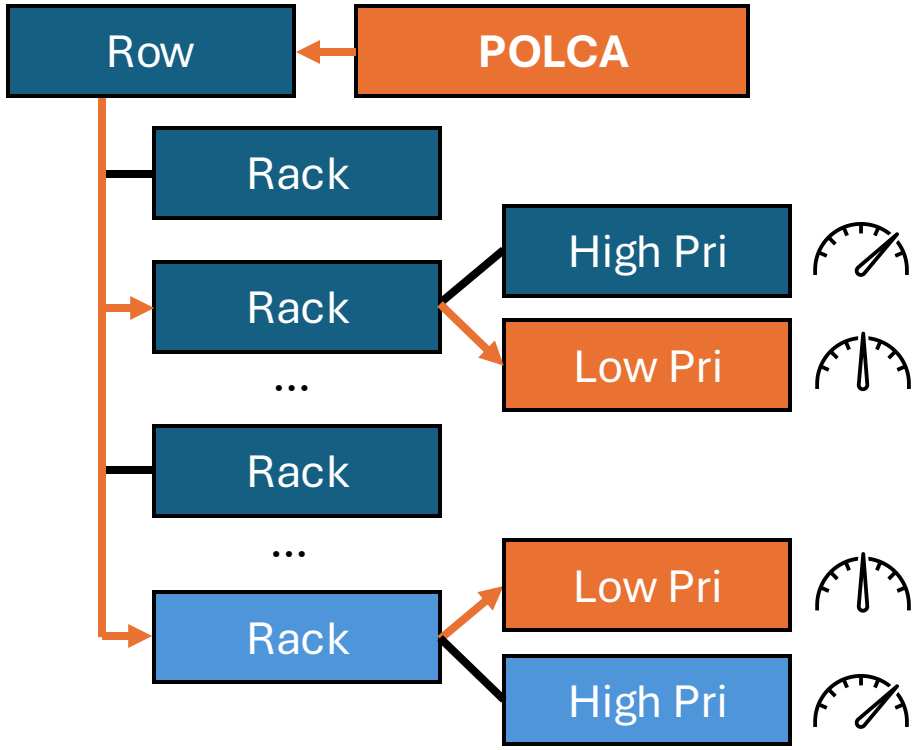
Leverage a proactive power throttling policy

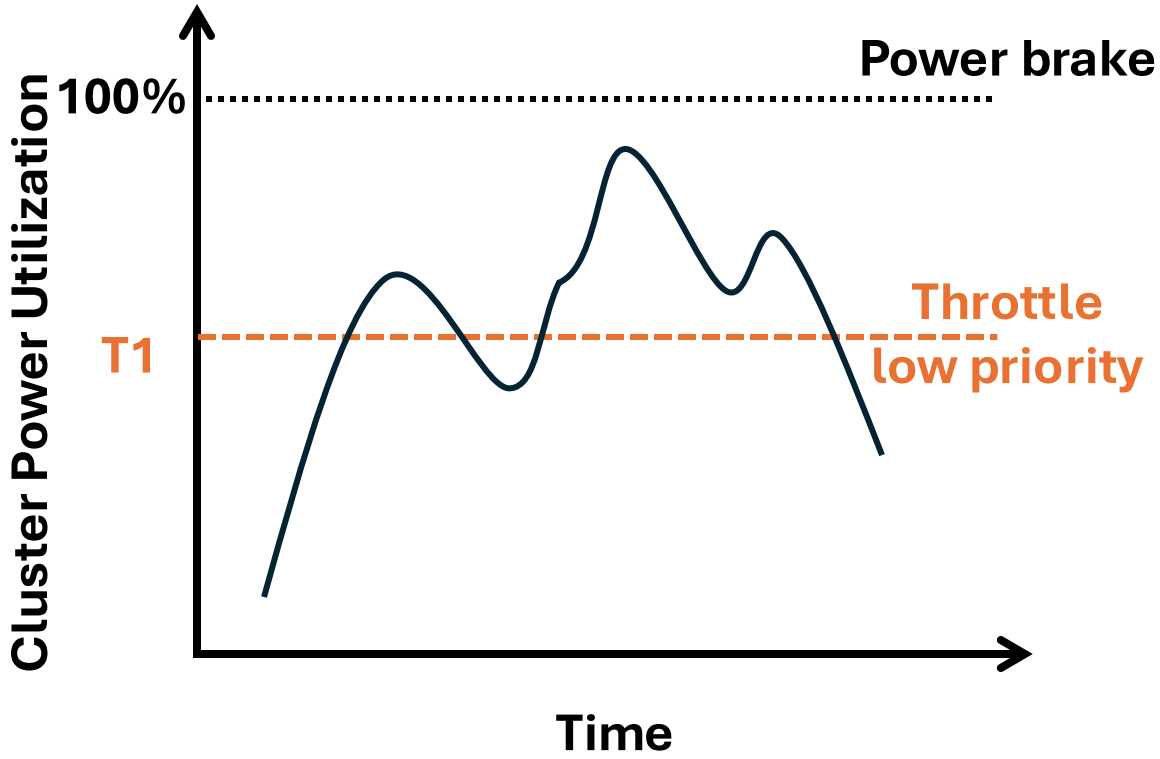




To ensure safety with slow GPU power throttling interfaces in the cloud

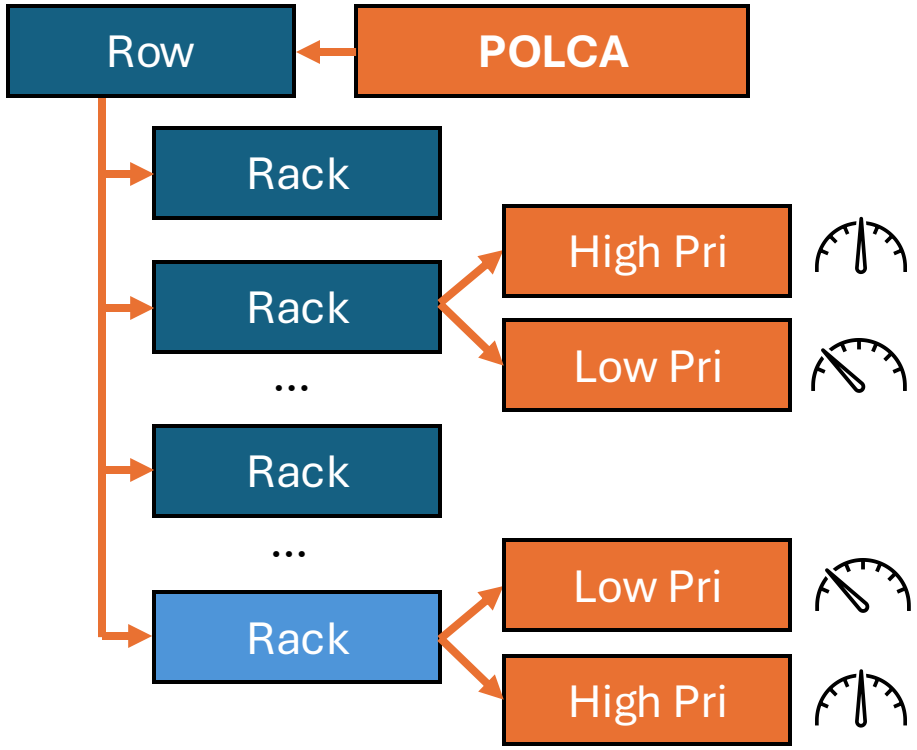
Configure priority-aware thresholds & actions

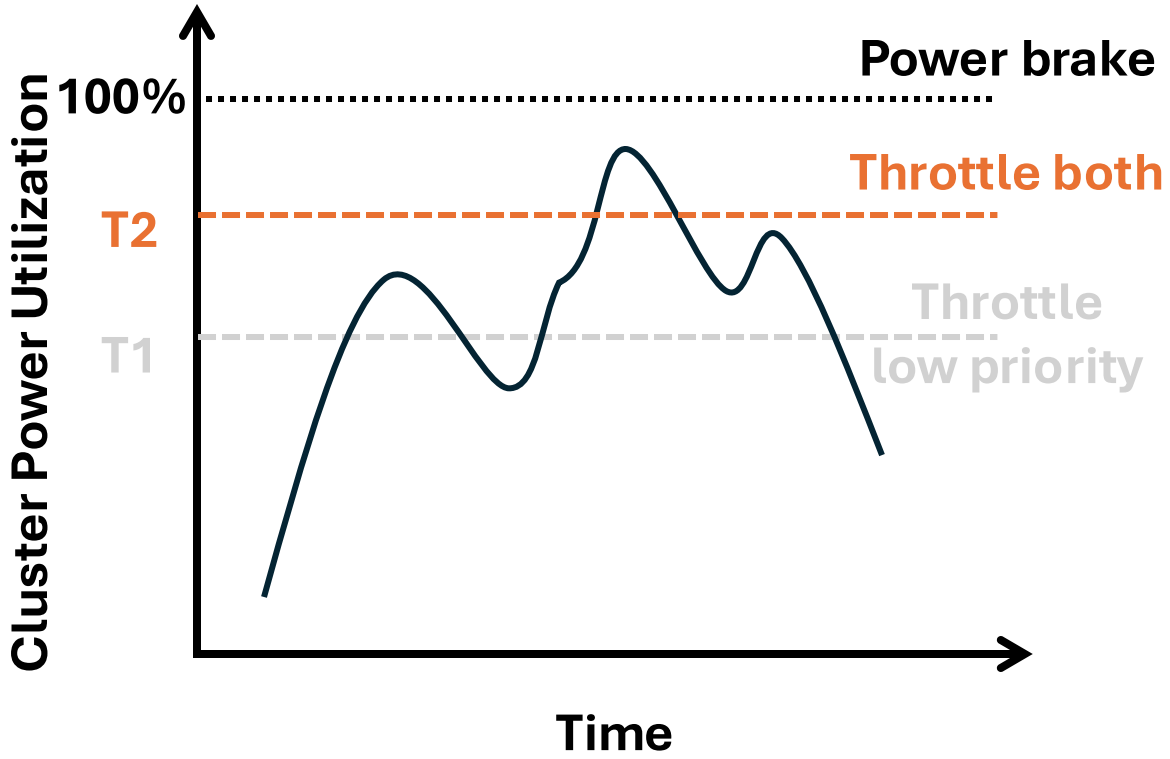




Preserve higher priority performance by aggressively throttling lower priority

Configure priority-aware thresholds & actions





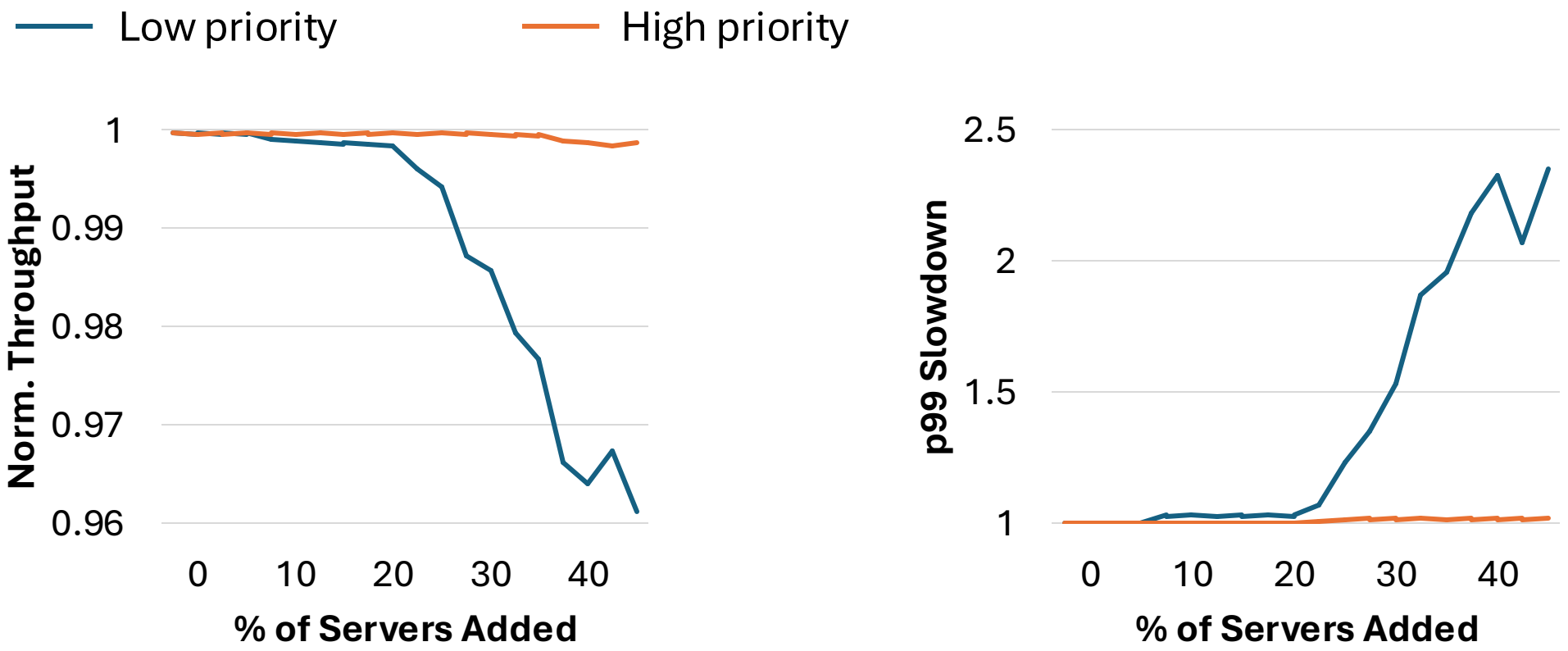
Preserve higher priority performance by aggressively throttling lower priority

Evaluation on six-week long production traces

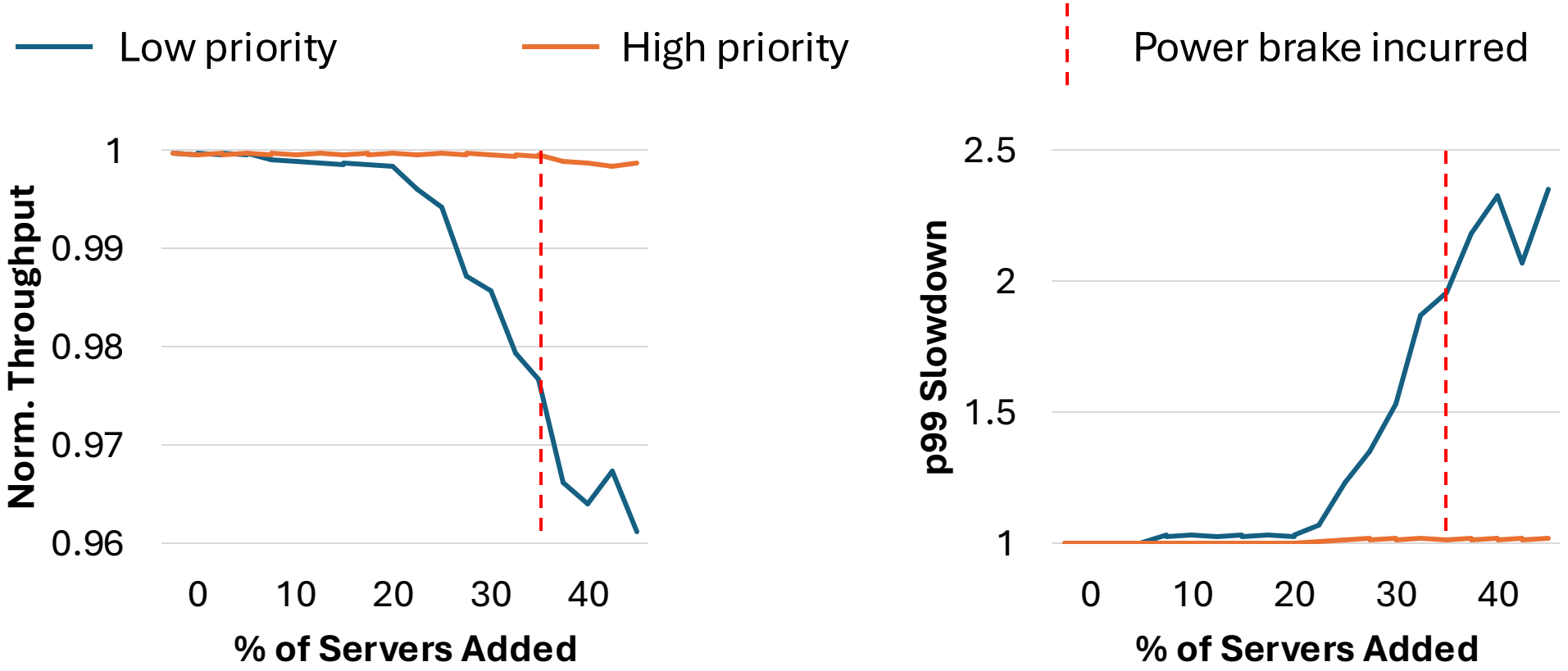
Workload	Prompt size	Output size	Fraction
Summarize	2k-8k	256-512	25%
Search	512-2k	1k-2k	25%
Chat	2k-4k	128-2k	50%

Replicated production power usage patterns using open-source models

Add servers and check performance impact



POLCA can safely deploy ~30% more servers



With less than 1.5% tail latency impact for high-priority workloads

Characterizing Power Management Opportunities for LLMs in the Cloud

Power usage characterization of training and inference workloads in production clusters

Design implications for power management in cloud scale deployments

Power oversubscription framework that safely adds ~30% more servers in LLM inference clouds

aka.ms/LLMPower

Thanks!

pratyush@cs.uw.edu