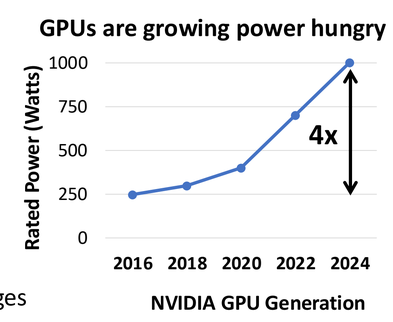
GPU Power Management Enables Rapid Deployment of Large Language Models

Characterizing Power Management Opportunities for LLMs in the Cloud

Power is a key bottleneck for LLM deployments at scale

Big Tech's Latest Obsession Is Finding Enough Energy

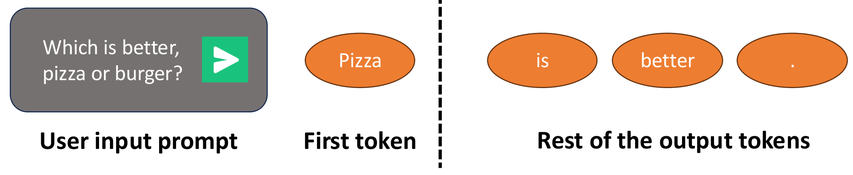
The AI boom is fueling an insatiable appetite for electricity, which is creating risks to the grid and the transition to cleaner energy sources



- The world adds a new datacenter every 3 days
- Datacenter electricity usage set to double by 2026
- 2-to-6 year construction delays due to power supply shortages

Power supply cannot keep up with the explosion in demand for LLMs

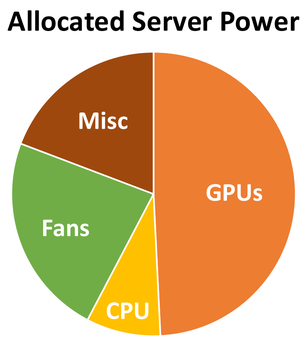
Background: LLM inference has two distinct phases



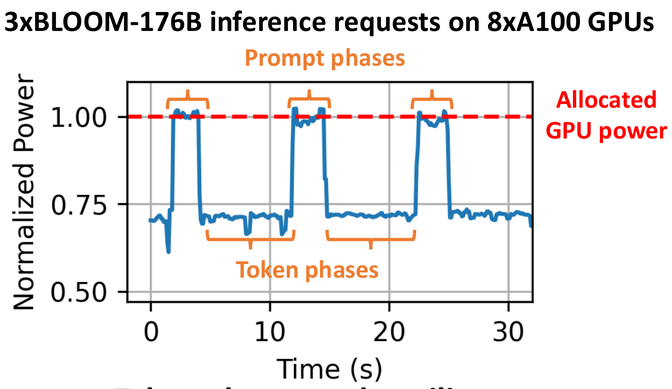
Prompt phase	Token generation phase
All input tokens processed in parallel	Serialized token generation
Compute intensive	Memory intensive (uses KV cache)

Phases are fundamental to transformer-based generative LLM inference

Prompt phases are power intensive, token phases not

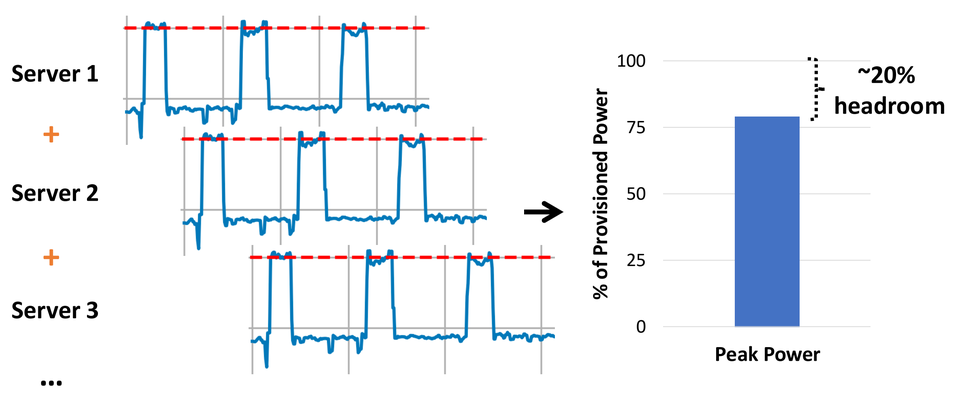


GPUs account for nearly half of the allocated power



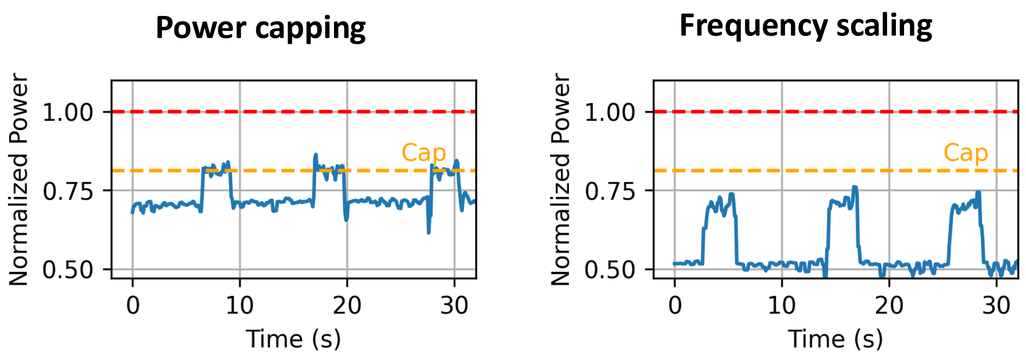
Token phases underutilize allocated GPU power

Production LLM inference clusters underutilize power



Prompt and token phases are statistically multiplexed across the cluster

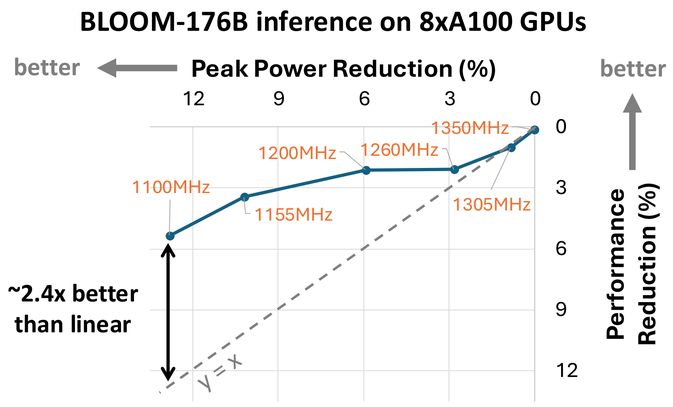
GPU power throttling knobs have different trade-offs



Power capping limits power draw and mainly throttles prompts

Frequency scaling locks GPU clock and throttles the entire request

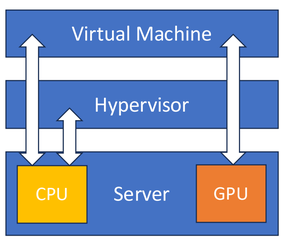
GPU power throttling is effective for LLM inference



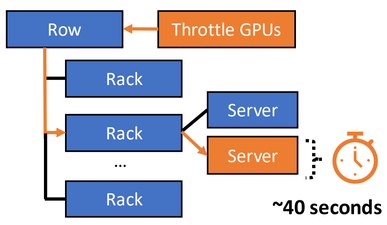
Frequency scaling reclaims power with low performance impact

POLCA manages GPU power to safely deploy ~30% more servers in existing and upcoming LLM inference clouds

Challenge 1: Slow out-of-band GPU management



Virtualized GPUs cannot be controlled with in-band knobs



Power throttling delay far exceeds the 10s deadline to shed power

Challenge 2: Diverse and evolving LLMs

ChatGPT · Microsoft 365 Copilot · Gemini · GitHub Copilot

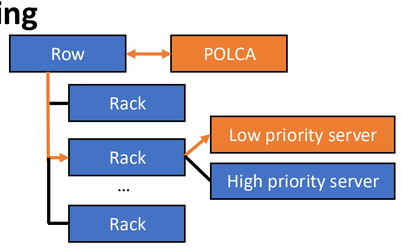
GPT-3 · GPT-3.5 · GPT-4 · LLaMA · LLaMA-2 · LLaMA-3

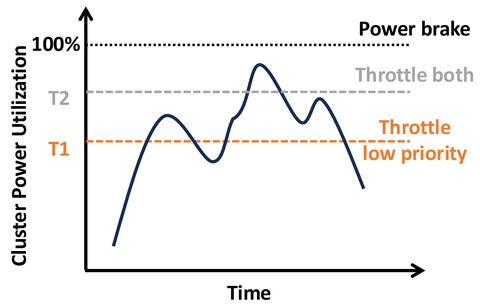
3 years later?

Approach: Configurable and proactive priority-aware throttling

POLCA Inputs

1. Historical power traces
2. Workload priorities
3. Row-level power draw

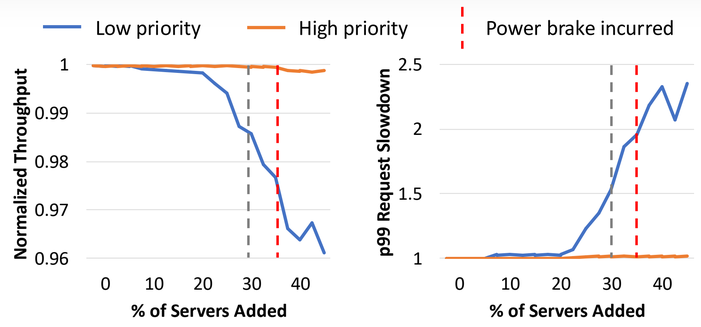




Result: Deploy 30% more servers with < 1.5% p99 impact on high-priority workloads

Production power usage patterns with open-source LLMs

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



In the paper: in-depth training and inference characterization, design implications for LLM clusters, etc.

aka.ms/LLMPower

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