THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

Where Does the Energy Go?

Profiling LLM Agent Inference on Blackwell GPUs

Qi Luo, Kunlin Li, Ziwen Wang, Yun Chen*

Microelectronics Thrust, Function Hub, The Hong Kong University of Science and Technology (Guangzhou)

*Corresponding author

NEURAL INFORMATION PROCESSING SYSTEMS

NeurIPS 2026 Workshop on Resource-Aware Agentic AI

Why profile agent energy?

Agent inference is not single-pass inference

Many turns

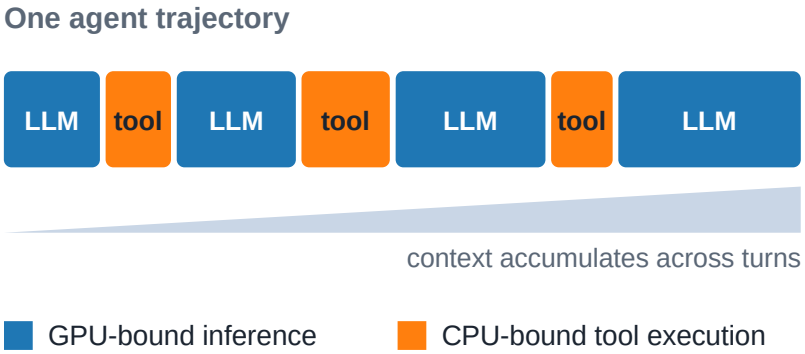
Each request depends on the trajectory so far, and context accumulates.

Two phases

GPU-bound inference bursts alternate with CPU-bound tools: shell, file I/O, tests.

A blind spot

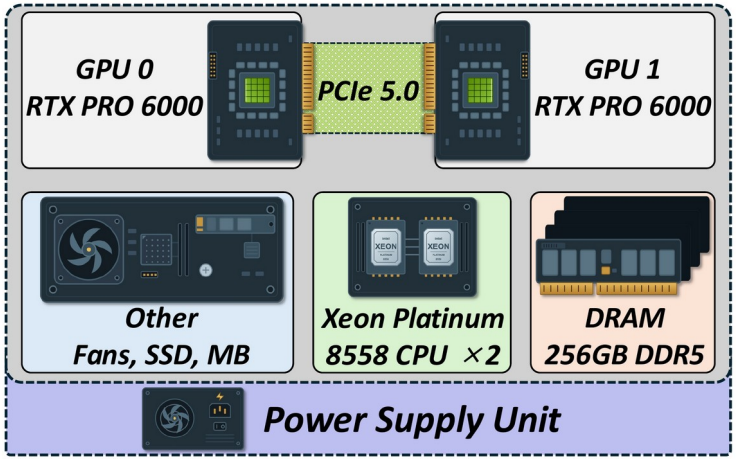
GPU-only telemetry misses non-GPU hardware; replayed traces miss energy between calls.



How much energy does agent inference consume, and where does it go?

Full-stack energy measurement

Three interfaces in one timing loop on a dual-GPU Blackwell server



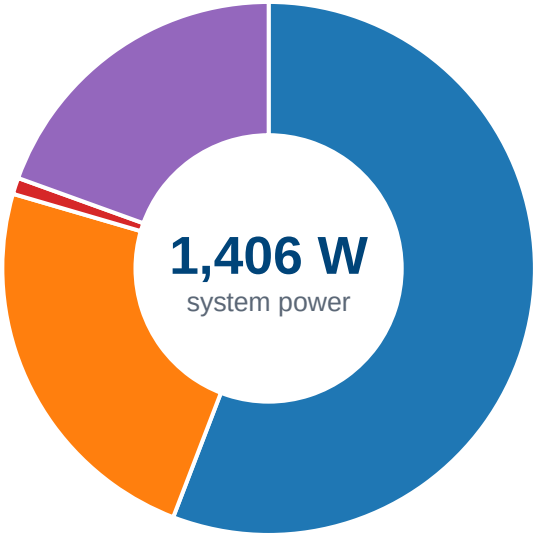
NVML 10 Hz	Cumulative per-GPU energy counters
RAPL 10 Hz	CPU package and DRAM energy counters
IPMI 1 Hz	CPU, memory, fan and total-system power from the BMC

Others = system − GPU − CPU − DRAM: fans, VRM and PSU losses, baseboard, SSDs, BMC.

- **Agentic coding** — SWE-bench Verified, 50 tasks
- **Math reasoning** — AIME and MATH-500, thinking on/off
- **Continuous serving** — ShareGPT at 1–16 req/s

GPU telemetry misses 41–45%

System power during agentic coding on SWE-bench Verified: 1,406 W on average



GPUs	785 W ·	55.9%
CPUs	333 W ·	23.7%
DRAM	14 W ·	1.0%
Others	274 W ·	19.5%

Others: fans, VRM and PSU conversion losses, PCIe links, baseboard.

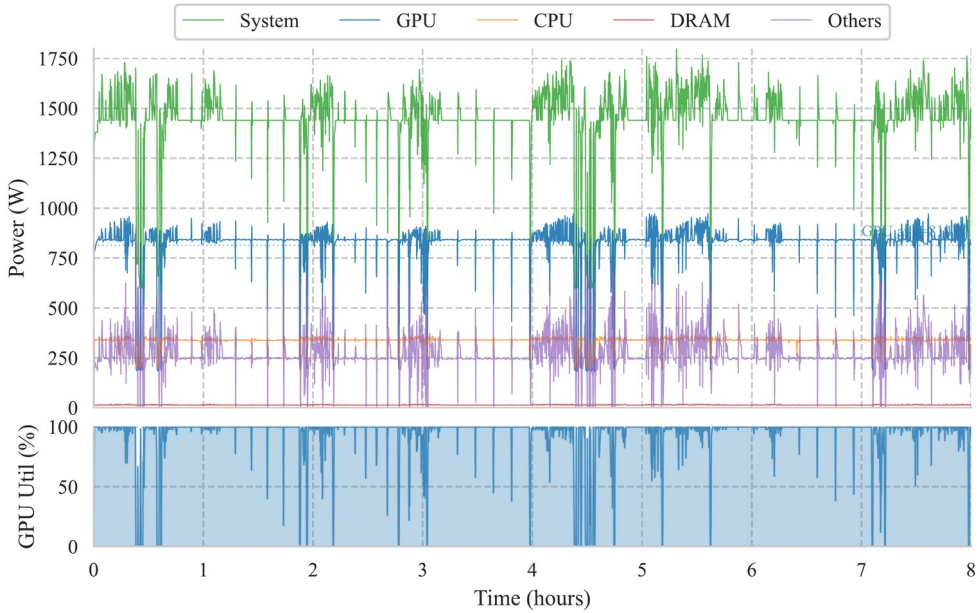
41–45%

of system energy is outside GPU telemetry

GPU share: 55–59% across agentic coding, reasoning and serving.

GPU power bursts and idles

System power stays elevated between bursts: the CPUs keep running tools



~800 W

GPU power during inference

~150 W

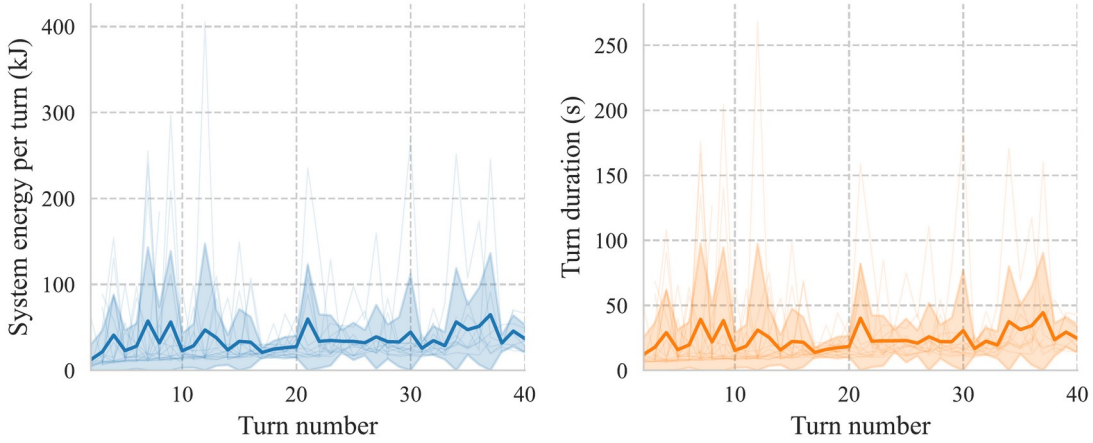
GPU power during tool execution

46.7%

of tool calls are shell commands, with the GPU largely idle

Energy tracks time, not power

Per-turn system energy and duration across SWE-bench tasks



Light lines: individual tasks; bold: mean ±1σ.

Energy follows duration

Both tend to rise over the course of a trajectory.

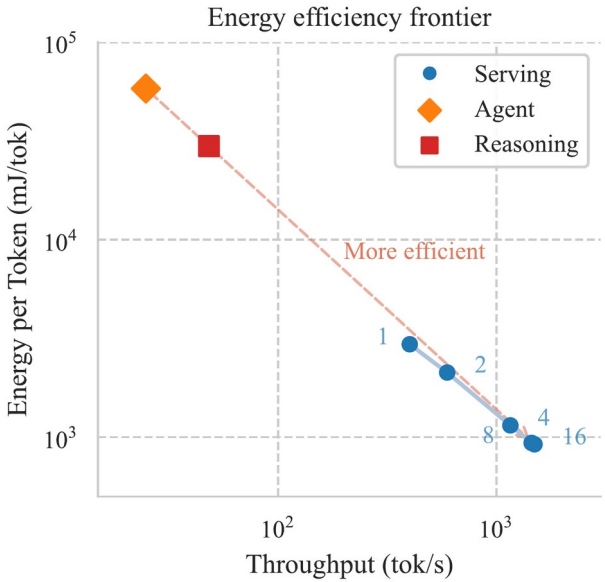
Power stays flat

Average system power changes little across turns.

Longer turns, not higher power: consistent with longer prefill as context accumulates, a target for context management.

63× more energy per token

A sequential agent vs. saturated serving, on the same server



System energy per output token	mJ/tok
Agent, SWE-bench	58,383
Reasoning, AIME (thinking on)	29,802
Serving, 1 req/s	2,955
Serving, 16 req/s	920

Why the gap: no batching, prefill that grows with context, tool time between calls.

Batching: 3.2× better efficiency from 1 to 16 req/s; GPU power plateaus by 4 req/s.

Thinking: 21–75% more tokens, under 1% change in per-token energy.

Takeaways

What the measurements suggest for resource-aware agents

41–45%

of system energy is outside GPU telemetry

Account for the full system

63×

more energy per token: sequential agent vs. saturated serving

Manage context across turns

<1%

per-token change from thinking, which adds 21–75% more tokens

Budget output tokens

400 W

cap costs reasoning no throughput; 300 W cuts it 3.1×

Pick power caps per workload

Paper: arXiv:2609.29707

Scope: one dual-GPU server and one 27B model; no external PDU validation.



THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

Thank you

Project page