NEURIPS 2026 WORKSHOP ON RESOURCE-AWARE AGENTIC AI

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

Agents break the single-pass energy picture

LLM agents reason, plan and call tools over many turns. Context accumulates, and GPU-bound inference bursts alternate with CPU-bound tool execution: shell commands, file I/O, test runs. GPU-only measurements miss the non-GPU hardware; replayed request traces miss the energy between model calls.

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

41–45%
of system energy is missed by GPU-only telemetry

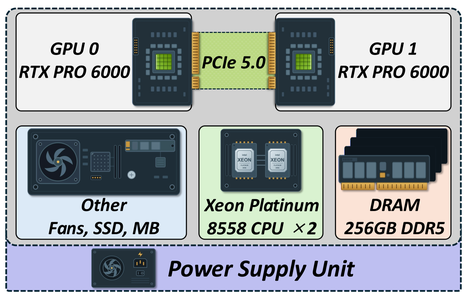
63×
more energy per output token: sequential agent vs. saturated serving

<1%
per-token energy change from thinking, which adds 21–75% more tokens

3.2×
better energy efficiency from continuous batching, 1 to 16 req/s

Full-stack measurement

Three interfaces sampled in one timing loop on a dual-GPU 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 BMC sensors

$P_{others} = P_{system} - P_{GPU} - P_{CPU} - P_{DRAM}$ covers fans, VRM and PSU conversion losses, the baseboard, SSDs and the BMC. RAPL CPU package power and the IPMI CPU rail differ by only 0.1–0.6%.

Platform and workloads

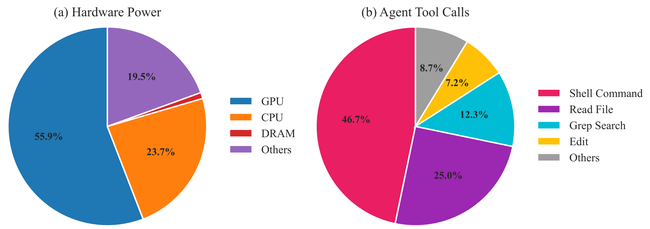
Server. 2× NVIDIA RTX PRO 6000 Blackwell Server Edition (96 GB, up to 600 W each), 2× Intel Xeon Platinum 8558, 256 GB DDR5. **Model.** Qwen3.8-27B on vLLM 0.19.1, tensor-parallel across both GPUs, BF16, 131k-token context.

- **Agentic coding.** 50 SWE-bench Verified tasks in the official Docker harness; multi-turn tool calling with extended thinking; one agent, sequential.
- **Mathematical reasoning.** AIME 2022–2024 (90 problems) and MATH-500, thinking enabled vs. disabled.
- **Continuous serving.** ShareGPT prompts at 1–16 requests per second under continuous batching.

Energy figures are total system energy unless marked GPU-only: per output token (mJ/tok) and per task (kJ).

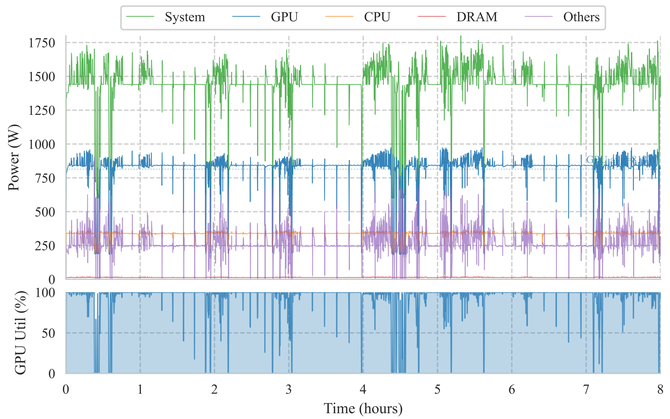
Agentic coding on SWE-bench Verified

1 GPU telemetry alone misses about 44% of system energy



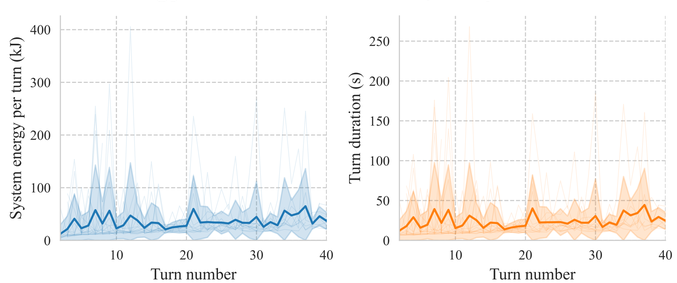
System power averages **1,406 W**: GPUs 785 W, CPUs 333 W, DRAM 14 W and Others 274 W (fans, VRM and PSU losses, baseboard). Shell commands, 46.7% of tool calls, often keep the CPU active while the GPU is largely idle.

2 GPU power bursts and idles; system power stays elevated



Over a selected 8 h window, GPU power rises to about 800 W during inference and falls to about 150 W during tool execution. System power generally remains elevated between bursts because the CPUs keep working.

3 Per-turn energy tracks turn duration, not power

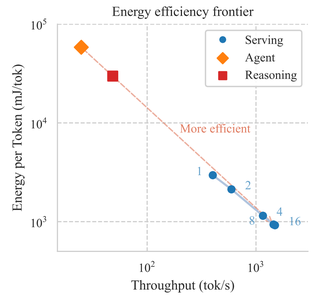


Per-turn system energy and duration. Light lines: individual tasks; bold: mean ±1σ.

Energy and duration per turn rise together along a trajectory while average system power changes little. Longer execution is the primary source of the increase, consistent with longer prefill as context accumulates.

Reasoning, serving and power caps

4 Agent vs. saturated serving: 63× energy per token



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, 4 req/s	1,147
● Serving, 16 req/s	920

Per task: 2,989 kJ for one SWE-bench task; 313.2 kJ per AIME problem and 53.6 kJ per MATH-500 problem with thinking enabled.

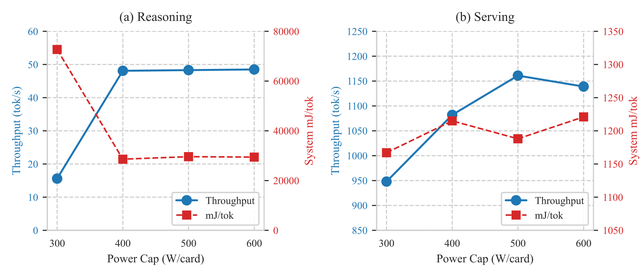
The gap reflects no batching across requests, prefill that grows with context, and tool time between model calls. With continuous batching, GPU power largely plateaus by 4 req/s while throughput climbs from 401 to 1,494 tok/s.

5 Thinking costs tokens, not per-token efficiency

DATASET	THINKING	TOKENS / PROBLEM	SYSTEM MJ/TOK	ENERGY / PROBLEM
AIME	enabled	10,506	29,802	313.2 kJ
AIME	disabled	5,991	29,751	178.2 kJ
MATH-500	enabled	1,800	29,759	53.6 kJ
MATH-500	disabled	1,488	29,682	44.2 kJ

Thinking adds 75% more tokens per problem on AIME and 21% on MATH-500; per-token energy differs by less than 0.4%.

6 Capping reasoning: no loss at 400 W, a cliff at 300 W



(a) Reasoning, single request. (b) Serving, 4 req/s. Throughput on the left axis, system energy per token on the right.

Reasoning draws about 420 W per card: 500 W and 600 W caps never bind, and 400 W trims power with no throughput loss. At 300 W, throughput falls 3.1× and energy per token rises 2.5×. Serving at 4 req/s is most efficient at 300 W.

Directions these results point to

- Manage context: summarization, selective retrieval
- Budget tokens: early stopping, budget-aware decoding
- Account for the full system, not GPU telemetry alone
- Select power caps per workload

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

arXiv:2609.29707