Energy-Aware LLM Inference on Consumer Hardware

RTX 3060 vs Intel i5-13th Gen

CSCE 585 – Milestone 1 (Week 4)
Suprawee Pongpeeradech

Motivation & Problem

- Why does this matter? Small labs and edge devices care about energy per request and latency, not just raw speed.
- Goal: Measure CPU vs GPU trade-offs in Joules per token, p95 latency, and cost per 1k tokens.
- Setup: llama.cpp on CPU and CUDA (RTX 3060) with TinyLlama-1.1B or similar model.
- Workload: Short to long prompts (64–1024 tokens) with steady and burst arrivals.

Tools

llama.cpp – Efficient local inference for LLMs with CPU and GPU backends.

NVIDIA NVML / pyNVML – GPU power and energy telemetry APIs.

Intel RAPL / Intel Power Gadget – CPU package power measurement tools.

Initial Hypotheses

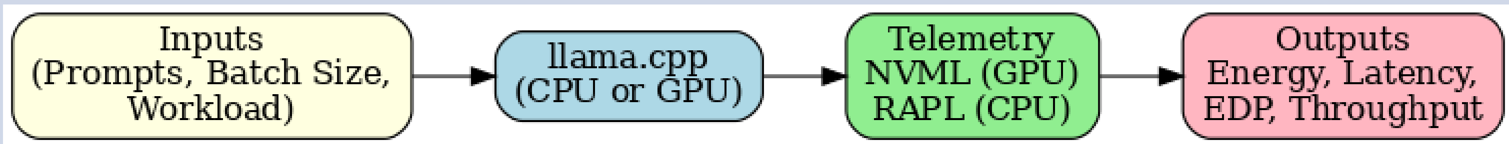
H1: GPU is more efficient for medium loads ($\leq$512 tokens, batch $\geq$2).

H2: CPU may be better for tiny loads (batch=1, very short prompts).

H3: Energy grows faster than linear with longer context; latency shows a knee.

H4: Energy-Delay Product reveals the sweet spot balancing energy and latency.

Design, Metrics & Plan



Telemetry: NVML for GPU, RAPL/Power Gadget for CPU; run 3–5 trials.

Metrics: Energy per request/token, p95 latency, throughput per cost; focus on Energy first.

This week: Lock configs, set experiment plan, check power readings.

Next: Run batching/context sweeps; make latency vs energy figure and table.

THANK YOU