More Answers per Chip

Memory-Pressure-Aware Request Shaping for energy-efficient LLM inference

Joel Maldonado

Neural Command, LLC (NRLC.ai) · Croutons

joel@nrlc.ai

11 November 2026 · **PDF**

Abstract

Energy per answer, not energy per token, is the figure AI data centres will increasingly be judged on. Measurements on a production-class inference stack show that accelerator energy follows busy time: a three-term model of idle, busy and per-token draw explains 99.8% of energy, while tokens alone explain 58%. Working memory, not compute, is the binding limit, and 4.7–15.7% of accelerator energy goes to work that did not need doing [**1**]. We propose *Memory-Pressure-Aware Request Shaping* (MPRS), a control layer in front of any serving engine that attributes energy to each request, forecasts KV-cache pressure at arrival, rewrites prompts so shared context can be reused, verifies answer reuse, and schedules work to minimise energy per answer under quality and latency limits. We set out five objectives with pass criteria fixed before any data is collected, ending in a 10% reduction in accelerator energy per answer on replayed production traffic at equal quality. Results will be published whether they pass or fail.

Keywords: LLM inference, energy per answer, KV cache, request scheduling, prefix caching, data-centre efficiency `cs.DC · cs.LG`

1 The gap

Large language model (LLM) inference is becoming the main load in new AI data centres, yet the figure operators, utilities and regulators now care about, energy per useful answer, is controlled by nothing in the serving stack. Serving engines schedule work inside one engine to raise throughput [**2**, **3**, **8**]. API gateways route on price per token [**9**]. Neither sees the accelerator's memory state, and neither is accountable for energy per answer.

2 What we measured

99.8%	of accelerator energy explained by idle + busy + per-token draw. Tokens alone: 58%. [**1**]
3.3×	less energy per request at peak load than at low load, on the same hardware. [**1**]
0% → 55%	reuse of earlier computation when a shared document moves from mid-prompt to the front. [**1**]

Our measurements on a production-class inference stack [**1**] show why this gap matters. A three-term model (idle draw, a fixed draw while the accelerator is busy, and a small per-token cost) explains 99.8% of measured accelerator energy; token counts alone explain 58%. Energy follows busy time, not tokens. The same hardware used 3.3 times less energy per request at peak load than at low load. The working memory of active requests (the KV cache) was above 90% full a third of the time and reached 100%, so requests were paused and recomputed while compute still had room [**13**]. Long-form summaries were 7% of requests and 37% of energy. Between 4.7% and 15.7% of accelerator energy went to work that did not need doing: identical retried calls, repeated questions, and shared documents placed mid-prompt, which gave 0% reuse of earlier computation against 55% when the same document came first.

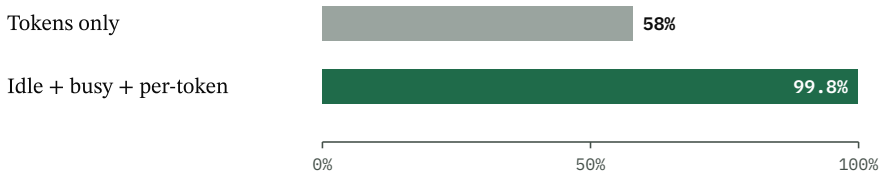


Figure 1. Share of measured accelerator energy explained by two models. Energy tracks busy time, not tokens [**1**].

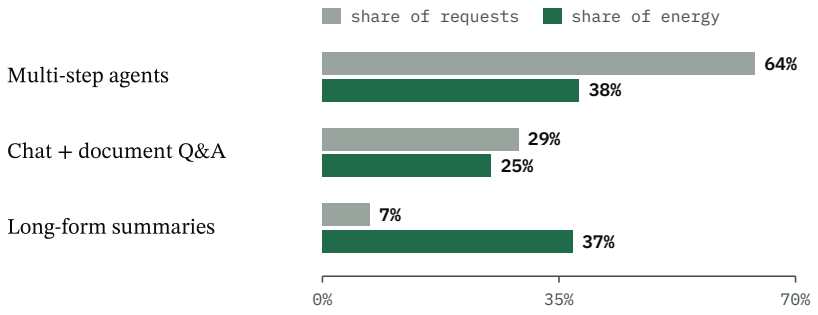


Figure 2. Share of requests vs share of energy by workload class. Summaries are a small share of traffic and a large share of energy [**1**].

3 Proposed system: Memory-Pressure-Aware Request Shaping

MPRS is a closed-loop control layer that sits in front of any serving engine and reshapes the request stream before it reaches the accelerator, with energy per answer as its objective and answer quality as a hard constraint. It joins four mechanisms that have not been combined:

(1) per-request energy attribution, fitted online from hardware counters and engine logs, that prices each request in busy-time joules rather than tokens;

(2) a forecaster that predicts KV-cache pressure from features known at request arrival (workload class, prompt structure, expected output length) early enough to act;

(3) quality-preserving prompt canonicalization, which moves shared context into a stable prefix the engine can reuse, plus verified reuse of repeated answers;

(4) admission, batching and routing that move long-generation and agent fan-out work into the windows and model sizes where energy per answer is lowest, within latency targets.

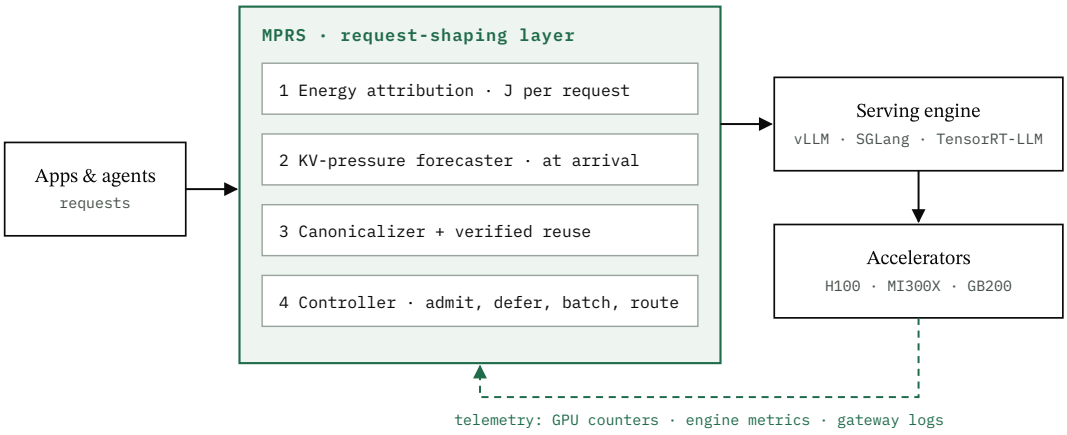


Figure 3. MPRS sits between clients and any serving engine. It acts on requests before they reach the accelerator and learns from telemetry operators already collect.

Today's prefix caches reuse only what clients happen to send in the same order; MPRS makes reuse a property of the request stream. Today's gateways optimise cost per token; our data show tokens explain barely half of energy. No current system ties a request-level action to a verified change in energy per answer.

Because MPRS runs outside the engine and needs only telemetry operators already collect (GPU counters, engine metrics, gateway logs), it can be adopted without retraining models or replacing infrastructure.

4 Pre-registered objectives

Each objective has a pass criterion fixed before data collection, under the protocol of our published benchmarks: pre-registered hypotheses, paired bootstrap intervals, and negative results reported [**4**, **5**, **6**].

O1. Portable energy attribution. Fit the idle + busy + per-token model on at least three accelerator families and two serving engines. Pass: at least 95% of measured energy explained on each. Challenge: busy draw shifts with batch mix and power capping. Approach: add batch-occupancy and clock terms only where held-out fit improves [**11**].

O2. Early forecasting of memory pressure. Predict at arrival whether admitting a request pushes KV occupancy past 90% during its lifetime. Pass: AUROC of at least 0.85 on held-out traffic. Challenge: output length, the main driver of KV growth, is unknown at arrival. Approach: class-conditional length distributions, conformal bounds, and online correction as tokens stream.

O3. Quality-preserving canonicalization. Move shared documents into a stable prefix. Pass: at least 50% prefix reuse on traffic with shared context, with no significant accuracy loss. Challenge: position changes what models use [**7**], and in our chunking study removing a passage's subject cost 1.4 points [**5**]. Approach: reorder only segments whose answer-relevant content does not depend on order, verified per workload class by paired evaluation.

O4. Verified answer reuse. Serve repeated and near-repeated questions from cache only when an independent check confirms equivalence. Pass: at most 1% answer disagreement on audited hits. Challenge: semantic matching yields confident false hits. Approach: a two-stage gate (canonical form, then a small verifier model), with the verifier's own energy charged against the saving.

O5. Closed-loop control. Combine O1–O4 into a controller that admits, defers, batches and routes requests to minimise energy per answer under p95 latency and quality limits. Pass: at least 10% lower accelerator energy per answer on replayed production traffic. Challenge: actions interact; deferral raises peak occupancy and can cause the preemption it was meant to avoid. Approach: model-predictive control on the O2 forecast, with a safe fallback to pass-through.

Table 2. Objectives and pass criteria, fixed before data collection.

	Objective	Pass criterion
O1	Portable energy attribution	≥ 95% of measured energy explained on each of ≥ 3 accelerator families, 2 engines
O2	Early KV-pressure forecast	AUROC ≥ 0.85 for crossing 90% occupancy, held-out traffic
O3	Quality-preserving canonicalization	≥ 50% prefix reuse on shared-context traffic, no significant accuracy loss
O4	Verified answer reuse	≤ 1% answer disagreement on audited cache hits
O5	Closed-loop control	≥ 10% lower accelerator energy per answer at equal quality and p95 latency

Key risk: the avoidable share we measured depends on how often traffic repeats [**1**]. If it repeats less, O4 adds little, and we test whether O2 and O3 alone recover energy. A negative result on any objective will be published.

5 Why it matters

US data centres used 176 TWh in 2023, 4.4% of national electricity, and are projected to use 325–580 TWh by 2028 [**10**]. Power, not chips, now limits where AI capacity is built, and regulators are moving toward output metrics. Malaysia, one of the fastest-growing AI data-centre markets, ties incentives to declared PUE, WUE and CUE, and requires large energy users to report and carry out energy saving measures under its 2024 energy efficiency act [**15**].

0.170 Wh facility energy per answer in the reference 20 MW hall. Published median for a Gemini text prompt: 0.24 Wh. [**1**, **12**]

For a reference 20 MW inference hall, our benchmark implies 20,300–67,900 GJ of avoidable facility energy a year and 0.7–2.2 MW of accelerator capacity freed [**1**].

Table 1. Energy saving measures in EECA 2024 format for a reference 20 MW inference hall (70% load, PUE 1.4). Each counts only after replay shows answer quality holds [**1**].

	Measure	Facility GJ/yr	t CO_2/yr	Water m^3/yr
ESM 1	Remove identical retried calls	9,086	1,913	3,966
ESM 2	Place shared documents first	11,250	2,369	4,911
	Conservative total (ESM 1 + 2)	**20,336**	**4,282**	**8,877**
ESM 3	Cache repeated questions (upper bound)	≤ 47,594	≤ 10,021	≤ 20,775
	Benchmark-mix total	**67,930**	**14,303**	**29,652**

6 Related work

Serving engines (vLLM, SGLang, TensorRT-LLM) optimise inside one engine [**2**, **3**, **8**]; gateways (LiteLLM, Portkey, Helicone) optimise price per token [**9**]; facility tools measure buildings, not answers. Recent measurement work diagnoses where inference energy goes [**11**, **12**] and how long contexts strain GPU memory [**13**]. MPRS uses engine caches and gateway logs as inputs and adds the missing layer: one that is accountable for energy per answer. Croutons, our inference-management product, already reads these logs and telemetry on the operator's own machine [**14**].

7 Open method

−89% input tokens with four retrieved passages, at 98.5% accuracy vs 95.1% for the full article. [**4**]

Everything in this agenda is open. Method, data and code will be published under CC BY 4.0 as each objective is run, in the same way as our token-efficiency benchmark, which reported a result against our own product: our atomic-fact rewrite scored 5.4 points below full text, while passage retrieval cut input tokens 89% and raised accuracy to 98.5% [**4**].

References

[1] Croutons (Neural Command, LLC). *Croutons Malaysia: energy per AI answer, in MITI and EECA terms.* Benchmark and report, November 2026. https://croutons.ai/malaysia

[2] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, I. Stoica. *Efficient Memory Management for Large Language Model Serving with PagedAttention.* SOSP 2023. arXiv:2309.06180. https://arxiv.org/abs/2309.06180

[3] L. Zheng et al. *SGLang: Efficient Execution of Structured Language Model Programs.* NeurIPS 2024. arXiv:2312.07104. https://arxiv.org/abs/2312.07104

[4] J. Maldonado. *How Many Tokens Does an Answer Need? A Token-Efficiency Benchmark.* NRLC.ai, November 2026. https://nrlc.ai/research/token-efficiency-benchmark/

[5] J. Maldonado. *Do Headings Survive Chunking? A Re-test on 800 Questions.* NRLC.ai, 11 November 2026. https://nrlc.ai/research/headings-chunking-retest/

[6] NRLC.ai. *Does AI search optimization work? A randomized trial.* Public pre-registered plan, 2026. https://nrlc.ai/en-us/insights/ai-search-optimization-randomized-trial/

[7] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, P. Liang. *Lost in the Middle: How Language Models Use Long Contexts.* TACL 2024. arXiv:2307.03172. https://arxiv.org/abs/2307.03172

[8] A. Agrawal et al. *Taming Throughput-Latency Tradeoff in LLM Inference with Sarathi-Serve.* OSDI 2024. arXiv:2403.02310. https://arxiv.org/abs/2403.02310

[9] I. Ong et al. *RouteLLM: Learning to Route LLMs with Preference Data.* arXiv:2406.18665, 2024. https://arxiv.org/abs/2406.18665

[10] A. Shehabi et al. *2024 United States Data Center Energy Usage Report.* Lawrence Berkeley National Laboratory, December 2024. https://www.energy.gov/articles/doe-releases-new-report-evaluating-increase-electricity-demand-data-centers

[11] J.-W. Chung, R. Wu, J. J. Ma, M. Chowdhury. *Where Do the Joules Go? Diagnosing Inference Energy Consumption.* arXiv:2601.22076, 2026. https://arxiv.org/abs/2601.22076

[12] C. Elsworth et al. *Measuring the environmental impact of delivering AI at Google Scale.* arXiv:2508.15734, 2025. https://arxiv.org/abs/2508.15734

[13] S. Luo, H. Shen. *No Buffer, No Bottleneck: Efficient Zero-Copy KV Cache Offloading for Long-Context LLMs.* OSDI 2026. https://www.usenix.org/system/files/osdi26-luo.pdf

[14] Croutons. *Documentation: Crouton Units and the croutons CLI.* 2026. https://croutons.ai/docs

[15] Croutons. *Building data centres in Malaysia: power, water and approval rules* (MITI guideline, DESAC, EECA 2024). 2026. https://croutons.ai/malaysia/building-data-centers

---

© 2026 Neural Command, LLC · Text and figures **CC BY 4.0** · Benchmark figures: Croutons Malaysia [**1**]