Energy Efficient or Exhaustive? Benchmarking Power Consumption of LLM Inference Engines

Chenxu Niu[1], Wei Zhang[2], Yongjian Zhao[1], Yong Chen[1]

Texas Tech University[1]

Lawrence Berkeley National Laboratory[2]

Overview

- **Motivation**: Why does LLM inference energy matter?
- **Research Questions**: What did we want to find out?
- **Methodology**: How did we measure and analyze it?
- **Evaluation & Results**: What did we find?
- **Key Insights**: What are the trade-offs and conclusions?
- **Future Work**

Motivation

- Large Language Models (LLMs) like the GPT-series and LLaMA-series have revolutionized the field of natural language processing.
- As models grow in parameter size, inference becomes a key bottleneck in real-world deployments.
- LLM inference is both computationally intensive and latency-sensitive, which is critical for real-time applications.

Motivation: The Hidden Cost

- Prior research has primarily focused on the energy costs of training and fine-tuning LLMs.
- However, recent evidence shows that the inference process now dominates the energy footprint, consuming nearly 90% of energy in large-scale deployments.
- Inference is a continuous process that directly impacts operational costs and environmental footprint.

Motivation: Inference Engines

To optimize performance, several inference engines have been developed. We chose four representative ones:

- **Transformers:** A highly flexible framework used as our baseline for comparison.

- **DeepSpeed:** A Microsoft engine focused on improving the scalability of the largest models.

- **TensorRT-LLM:** NVIDIA's specialized engine, optimized for low-latency and high-performance on NVIDIA GPUs.

- **vLLM:** An engine designed for high-throughput serving, featuring the "PagedAttention" technique.

Research Questions

- During the **setup stage**, what is the power consumption and latency to initialize engines and load models?
- During the **token generation stage**, how does energy efficiency vary across engines and hardware components (GPU, CPU, DRAM)?
- What is the relationship between **energy efficiency and throughput**?
- Is there a **single inference engine** that is the most energy-efficient in all scenarios?

Methodology

We break down the entire inference process into two distinct stages:

- Setup Stage: Includes engine initialization and model loading.
- Token Generation Stage: Where the actual inference takes place.

$$E_{\text{total}} = E_{\text{Setup}} + E_{\text{TG}}$$
$$= E_{\text{IE}} + E_{\text{LM}} + T \cdot E_{\text{PT}}$$

Experimental Setup & Tools

Hardware Platform:

- NSF REPACSS Data Center (built 4 month ago)
- Powered by variable energy sources: including wind and solar.
- Will be a part of NSF ACCESS: https://allocations.access-ci.org/resources

Experimental Setup & Tools

Hardware Platform:

- GPUs: 4 x NVIDIA H100 (94GB memory each)
- CPUs: 2 x Intel Xeon Gold 6426Y
- RAM: 503GB

Software & Models:

- Models: Llama 3.1-8B, Llama 3.2-1B, and Llama 3.2-3B
- Dataset: Alpaca (containing 52,002 prompts)

Experimental Setup & Tools

Measurement Tools:

- IPMI (Total System Power)
- NVIDIA Management Library (GPU Power)
- Intel RAPL (CPU & DRAM Power)

Evaluation & Results

Sta

Thi

en

Table 1. Energy Consumption and Latency of Loading Inference Engines and Model Loading for Different Model Sizes

Phase	Engine/Model	Metrics				
		Latency (s)	Total Energy (J)	GPU Energy (J)	CPU Energy (J)	DRAM Energy (J)
E_{IE}	**vLLM**	48.39	27209.42	7298.76	11985.02	982.83
	Transformers	2.89	1632.91	413.02	518.35	70.81
	DeepSpeed	2.92	1691.98	419.71	538.71	71.98
	TensorRT-LLM	30.21	18722.34	4566.90	8627.28	627.02
E_{LM}	**vLLM – 1B**	3.81	2302.98	691.22	1028.89	80.73
	vLLM – 3B	9.11	5502.72	1792.73	2328.41	194.54
	vLLM – 8B	11.64	7184.06	2480.81	2659.39	251.56
	Transformers – 1B	1.29	748.43	229.19	323.24	27.85
	Transformers – 3B	1.75	1020.65	311.28	469.02	38.37
	Transformers – 8B	3.31	1963.59	586.35	726.50	84.18
	DeepSpeed – 1B	1.23	718.59	218.47	316.14	28.38
	DeepSpeed – 3B	1.77	1024.17	313.37	474.21	39.11
	DeepSpeed – 8B	3.23	1951.83	574.56	712.26	82.07
	TensorRT-LLM – 1B	2.62	1734.79	522.29	762.74	56.21
	TensorRT-LLM – 3B	4.27	3022.91	917.63	1492.41	102.67
	TensorRT-LLM – 8B	7.92	4892.89	1492.7	2088.12	162.83

Why the Huge Difference in Setup Time?

- vLLM & TensorRT-LLM perform extensive pre-optimization during setup.
- vLLM: Sets up "PagedAttention" for memory management and configures distributed inference.
- TensorRT-LLM: Requires model compilation, layer fusion, and hardware-specific CUDA kernel generation.
- Transformers & DeepSpeed use dynamic computation graphs with fewer optimizations, enabling faster deployment.

Evaluation & Results

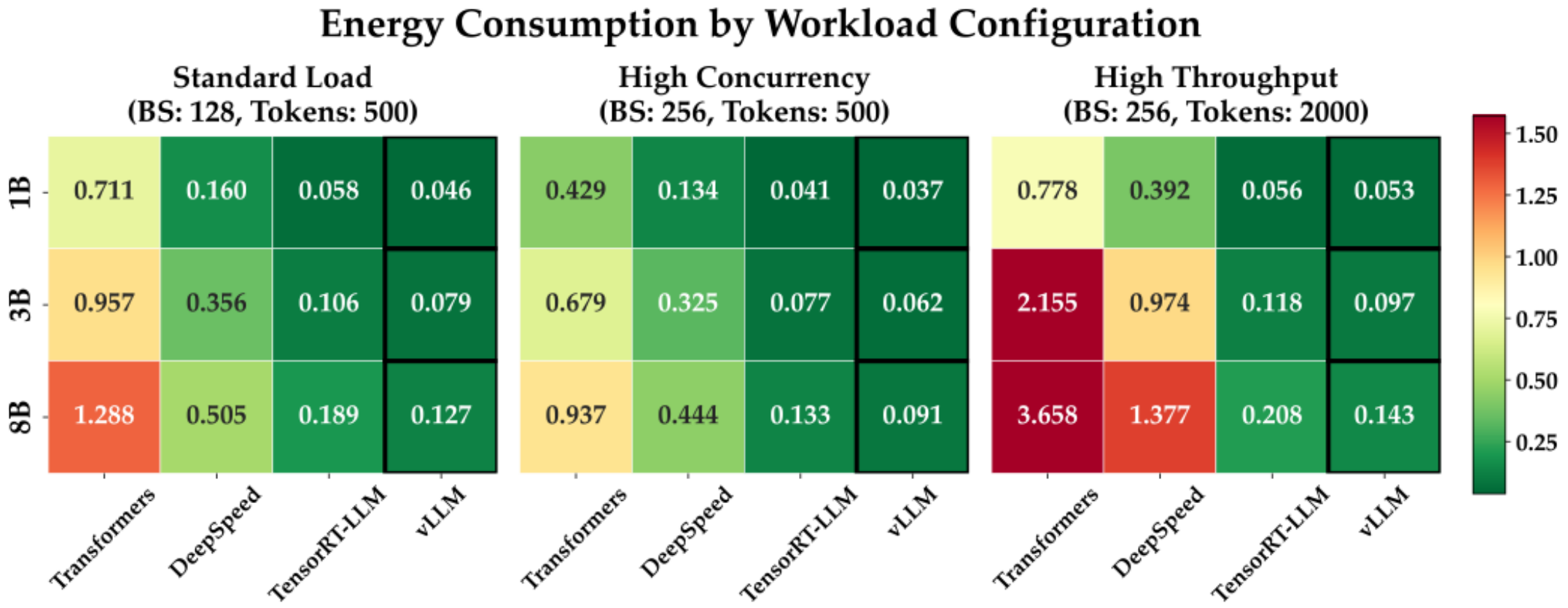
Stage 2: Token Generation Energy Efficiency

We simulated three real-world workload configurations:

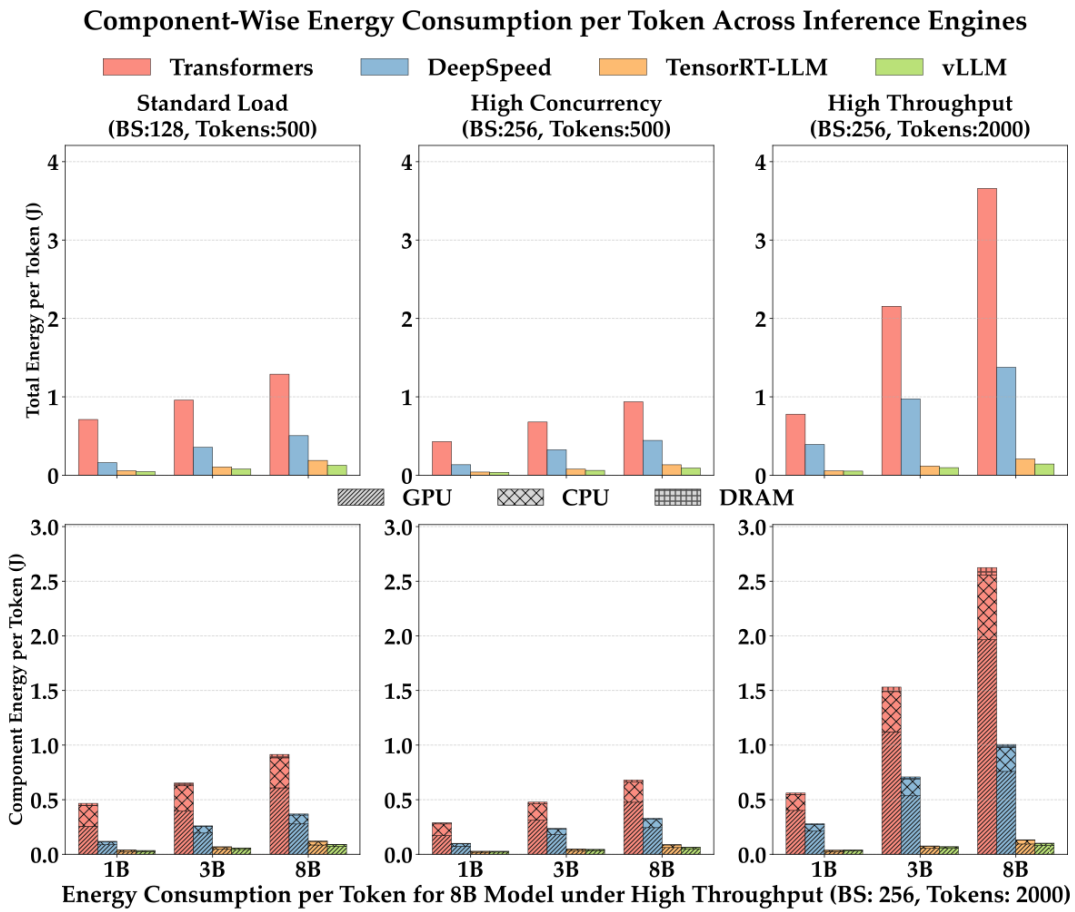
- Standard Load: Batch Size (BS): 128, Output Tokens: 500
- High Concurrency: BS: 256, Output Tokens: 500
- High Throughput: BS: 256, Output Tokens: 2000

Evaluation & Results: Heatmap of Energy per Token

Stage 2: Token Generation Energy Efficiency



Deeper Dive: Component-wise Breakdown



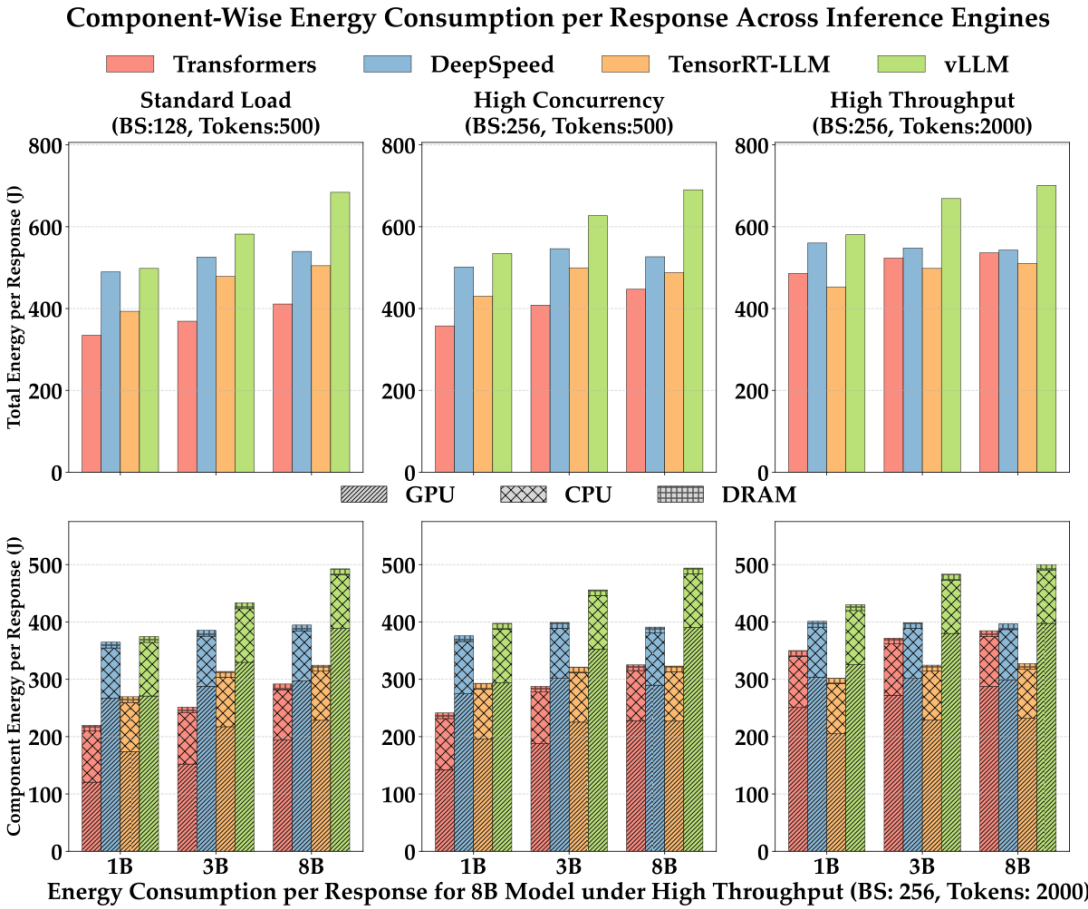
Energy Consumption per Token for 8B Model under High Throughput (BS: 256, Tokens: 2000)

Inference Engine	Total (J)	GPU (J)	CPU (J)	DRAM (J)
Transformers	3.658	1.964	0.593	0.068
DeepSpeed	1.377	0.759	0.221	0.025
TensorRT-LLM	0.208	0.094	0.035	0.004
vLLM	0.143	0.081	0.019	0.002

Key Insights:

- The GPU is the dominant energy consumer, accounting for over 50% of the total energy.
- Under High Throughput, vLLM's GPU energy consumption is only 0.081 J/token, which is just 4% of what Transformers consumes.
- A similar trend is observed for CPU and DRAM, where vLLM also maintains the lowest energy usage.

Evaluation & Results: Energy per Response



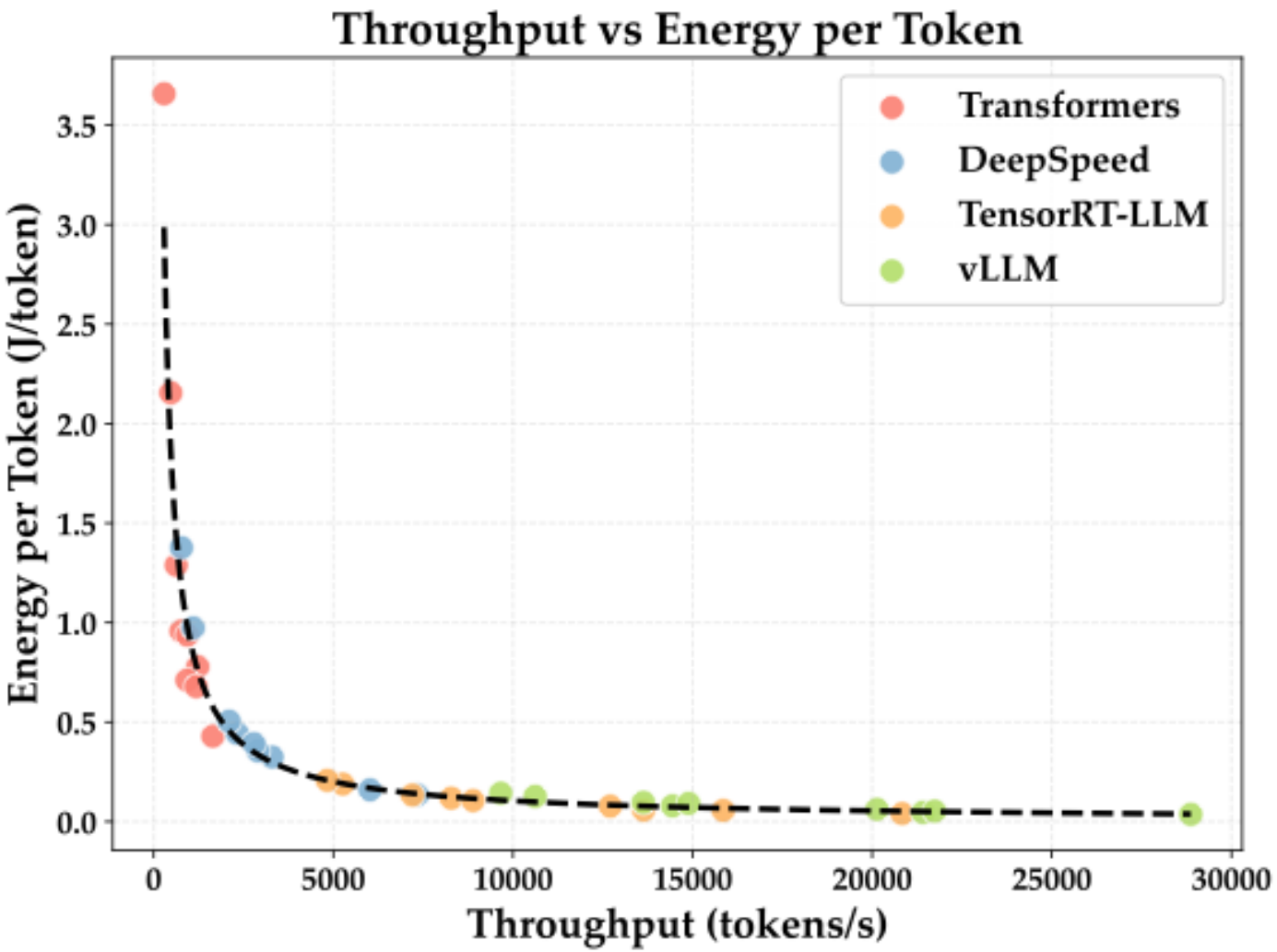
Energy Consumption per Response for 8B Model under High Throughput (BS: 256, Tokens: 2000)

Inference Engine	Total (J)	GPU (J)	CPU (J)	DRAM (J)
Transformers	536.126	287.858	86.883	9.947
DeepSpeed	542.900	299.366	87.305	10.046
TensorRT-LLM	510.368	232.081	85.277	10.167
vLLM	700.849	397.513	92.280	10.282

Key Insights:

- The Paradox: While vLLM is most efficient per token, it consumed the highest total energy per response.
- The Reason: Different engines have different ending policies.
- vLLM generates the largest number of tokens per response, thus increasing its total energy consumption.

The Relationship Between Efficiency and Throughput



Key Insights:

- Hypothesis Validated: Higher throughput improves energy efficiency by reducing the per-token energy cost.
- The Reason: The fixed idle power of the system is amortized over more tokens generated per unit of time.

Conclusion

- We conducted the first comprehensive benchmark of power consumption across several widely used LLM inference engines.

- We provided a fine-grained breakdown analysis across two lifecycle stages and key hardware components (GPU, CPU, DRAM).

Take-away Insights

Question: Is There a Single Best Solution?

Answer: No.

Our evaluation shows that no single inference engine universally optimizes energy efficiency across the entire lifecycle of inference.

The optimal choice is dependent on the specific use case.

Take-away Insights

It’s a Trade-Off.

- For Latency-Sensitive or On-Demand Environments:

- Recommendation: Transformers, DeepSpeed.
- Reason: They offer the most efficient setup in both latency and energy consumption.

- For High-Throughput, Intensive Inference Environments:

- Recommendation: vLLM, TensorRT-LLM
- Reason: They dominate in energy efficiency per token, especially under heavy workloads.

Future Work

- Extend this study to larger-scale LLM models and multi-node GPU clusters.
- Analyze the impact of distributed inference and inter-GPU communication on energy consumption.
- Propose and develop a novel, energy-efficient inference engine or framework that integrates the strengths of existing systems.

https://github.com/chenxuniu/LLM-Inference-Engine-Benchmark

Thanks! Any Questions?

https://repacss.org/