2024 IEEE 15th International Green and Sustainable Computing Conference (IGSC) | 979-8-3315-0786-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/IGSC64514.2024.00016

Sustainable LLM Serving: Environmental Implications, Challenges, and Opportunities

(Invited Paper)

Yi Ding
Purdue University
West Lafayette, IN, USA
yiding@purdue.edu

Tianyao Shi
Purdue University
West Lafayette, IN, USA
shi676@purdue.edu

***Abstract*—Large language models (LLMs) have been widely used for their ability to handle complex natural language tasks with high accuracy. The lifecycle of LLMs development and deployment encompasses both training and serving phases. Although training takes months and consumes significant amounts of energy, recent studies show that the energy consumption of LLM serving has now surpassed that of training, leading to significant environmental impacts, especially in terms of carbon footprints. While much prior work has focused on improving LLM performance, the specific challenge of reducing the carbon footprint of LLM serving has been largely overlooked. This paper identifies key challenges and outlines research directions for making LLM serving more sustainable, aiming to inspire further environmentally responsible advancements in the field.**

I. Introduction

Large language models (LLMs) have been widely adopted due to their ability to perform complex natural language processing tasks with high accuracy [1]. Models such as OpenAI's GPT-4, Google's Gemini, and Meta's LLaMA are now integrated into applications ranging from chatbots to data analytics. The LLM lifecycle involves both training and serving. Although training takes months and consumes significant amounts of energy, recent studies show that the energy consumption of LLM serving has now surpassed that of training, leading to significant environmental impacts, especially in terms of carbon footprints measured in CO_2eq [2].

The carbon footprints of LLM serving are categorized into *embodied* and *operational*. On one hand, the deployment of LLM serving systems requires advanced hardware infrastructure such as GPUs and machine learning accelerators (e.g., TPUs). Previous studies have shown that the manufacturing process of these high-performing hardware devices generates significant embodied carbon footprints [3]. The key idea of reducing embodied carbon is to extend hardware lifetimes to amortize its embodied carbon over a longer period. On the other hand, operational carbon footprints are generated from the energy consumption required to run LLMs during their serving phase. For example, serving a single prompt in ChatGPT produces more than 4 grams of CO_2eq [4], which is over 20 times the operational carbon footprint of a web search query [5]. Therefore, to effectively reduce the total carbon footprints of LLM serving, we must consider both embodied and operational carbon footprints holistically.

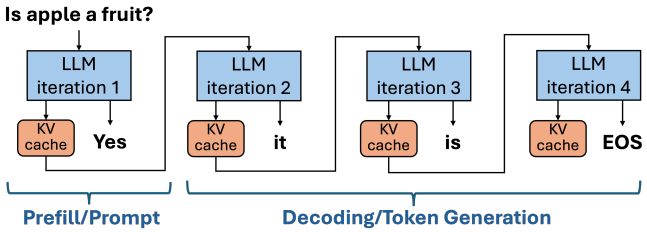

Fig. 1. An LLM serving example.

II. Challenges

LLM serving differs significantly from traditional cloud applications like microservices and serverless workloads. Unlike these lightweight applications, LLM serving is highly compute- and memory-intensive, requiring substantial compute resources such as GPUs or TPUs due to the large scale of models, which have billions of parameters. We summarize the unique characteristics of LLM serving as follows.

1) *High compute and memory intensity.* Existing performance optimization techniques for LLM serving highly rely on the latest and most advanced hardware [6], as older hardware often cannot meet the compute demands and latency SLOs of LLM models. While reusing older hardware like legacy servers and discarded smartphones can reduce carbon footprints for lightweight microservices [7], [8], this approach does not generalize to LLM serving, where high compute and memory requirements make meeting latency SLOs challenging.

2) *Two-phase execution.* LLM serving involves two phases: *prefill* and *decoding* [9]. In prefill, all input tokens are processed in parallel to generate the first token, and the resulting context is stored in a key-value (KV) cache. In decoding, subsequent tokens are generated using the last token and the KV cache. Figure 1 shows an example of LLM serving. The prefill phase is compute-bound, while the decoding phase is memory-bound, each with distinct latency SLOs, complicating efforts to minimize carbon footprints.

Based on these characteristics, we list three main challenges in minimizing the carbon footprints of LLM serving.

Limited understanding of tradeoffs between performance and carbon. Unlike traditional cloud applications, where performance is measured by a single metric like end-to-end latency, LLM serving involves two distinct metrics:

979-8-3315-0786-2/24/$31.00 ©2024 IEEE
DOI 10.1109/IGSC64514.2024.00016

time to first token (prefill phase) and time to generate each token (decoding phase). Each phase has its own latency SLOs, compute and memory needs, and carbon footprints, making the performance-carbon relationship unique. Furthermore, different optimization strategies affect the compute and memory intensities of LLM components variably, posing the challenge of understanding their performance and carbon impact across different hardware and application-level configurations.

Conflict between high compute/memory requirements and embodied carbon reduction. Reducing embodied carbon involves extending hardware lifetimes by reusing older hardware. However, the trend of training larger models (e.g., Meta's LLaMA 3 with 450 billion parameters) on advanced GPUs and TPUs increases embodied carbon. Additionally, the varying compute and memory demands of different LLM components further complicate extending hardware lifetimes.

Unreliable carbon intensity forecasts. Serving LLMs in low carbon intensity regions is complicated by the tradeoffs between operational and embodied carbon. Hardware with low embodied carbon may be located in a high carbon intensity region. Performance and energy efficiency may decline when shifting workloads between distant regions, not to mention reliability issues during network communication and data transfers. Furthermore, carbon intensity forecasts are uncertain [10], complicating the decision-making of resource allocation.

III. Research Opportunities

To tackle the abovementioned challenges, we present several research opportunities below.

Unified benchmarking. While the open source community has begun developing performance optimization tools to enable automatic performance and energy measurement of LLM serving, more can be done to integrate sustainability by incorporating carbon accounting methodologies and telemetry into these tools. Additionally, since these tools are developed by both academia and industry, their assumptions about LLM models and library dependencies vary. To ensure fair comparisons, each LLM optimization tool should be evaluated using consistent LLM models and libraries. Thus, it is crucial to build a generalized benchmarking framework that bridges different LLM optimization techniques with various models and libraries, facilitating seamless integration and evaluation.

Reusing old GPUs. Heavy reliance on the latest GPUs to improve LLM serving performance significantly raises embodied carbon footprints. This trend encourages the industry to discard under-performing and even mediocre-performing GPUs before their actual lifetimes end, making it harder to reduce embodied carbon. To address this, we must shift away from the belief that only the newest GPUs can boost performance. Instead, hardware, systems, and application co-optimization is needed to simultaneously improve performance and reduce embodied carbon. Additionally, reusing old GPUs can potentially increase latency and energy consumption, leading to higher operational carbon. Therefore, the tradeoffs between embodied and operational carbon must be managed.

Embodied carbon from storage. Storage plays a critical role in LLM serving, encompassing areas like model storage, temporary storage for input data and preprocessing, KV-cache, logging, monitoring, and postprocessing. While LLMCarbon has explored the energy consumption of data storage and transfer [11], the embodied carbon of storage has been overlooked. The embodied carbon of storage becomes significant when offloading techniques are used due to the insufficient memory capacity of modern GPUs to handle LLMs with hundreds of gigabytes in size. In these cases, embodied carbon comes not just from memory and GPUs but also SSDs and HHDs. Recent studies show that storage contributes 33% of operational and 61% of embodied carbon in Azure's cloud [12], yet storage emissions specific to LLMs remain under-explored.

Minimizing total carbon under uncertainties. Minimizing the total carbon footprint of LLM serving involves several uncertainties, including inaccurate performance and energy profiling, unreliable performance and energy modeling, uncertain embodied carbon accounting, unpredictable carbon intensity, and inconsistent communications between servers. Therefore, robust scheduling is needed for reliable lower carbon.

Acknowledgment

This work is supported by NSF CCF-2413870.

References

[1] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang, "HuggingGPT: Solving AI tasks with ChatGPT and its friends in hugging face," *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[2] A. A. Chien, L. Lin, H. Nguyen, V. Rao, T. Sharma, and R. Wijayawardana, "Reducing the carbon impact of generative AI inference (today and in 2035)," in *Proceedings of the 2nd Workshop on Sustainable Computer Systems (HotCarbon)*, 2023.

[3] U. Gupta, M. Elgamal, G. Hills, G.-Y. Wei, H.-H. S. Lee, D. Brooks, and C.-J. Wu, "ACT: Designing sustainable computer systems with an architectural carbon modeling tool," in *ISCA*, 2022.

[4] V. Wong, "Gen AI's environmental ledger: A closer look at the carbon footprint of ChatGPT," https://piktochart.com/blog/carbon-footprint-of-chatgpt/, 2023.

[5] S. Griffiths, "Why your internet habits are not as clean as you think," https://www.bbc.com/future/article/20200305-why-your-internet-habits-are-not-as-clean-as-you-think, 2020.

[6] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, and I. Stoica, "Efficient memory management for large language model serving with pagedattention," in *SOSP*, 2023.

[7] J. Wang, D. S. Berger, F. Kazhamiaka, C. Irvene, C. Zhang, E. Choukse, K. Frost, R. Fonseca, B. Warrier, C. Bansal, J. Stern, R. Bianchini, and A. Sriraman, "Designing cloud servers for lower carbon," in *ISCA*, 2024.

[8] J. Switzer, G. Marcano, R. Kastner, and P. Pannuto, "Junkyard computing: Repurposing discarded smartphones to minimize carbon," in *ASPLOS*, 2023.

[9] P. Patel, E. Choukse, C. Zhang, A. Shah, Í. Goiri, S. Maleki, and R. Bianchini, "Splitwise: Efficient generative LLM inference using phase splitting," in *ISCA*, 2024.

[10] A. Li, S. Liu, and Y. Ding, "Uncertainty-aware decarbonization for datacenters," in *Proceedings of the 3rd Workshop on Sustainable Computer Systems (HotCarbon)*, 2024.

[11] A. Faiz, S. Kaneda, R. Wang, R. C. Osi, P. Sharma, F. Chen, and L. Jiang, "LLMCarbon: Modeling the end-to-end carbon footprint of large language models," in *The Twelfth International Conference on Learning Representations*, 2024.

[12] S. McAllister, F. Kazhamiaka, D. S. Berger, R. Fonseca, K. Frost, A. Ogus, M. Sah, R. Bianchini, G. Amvrosiadis, N. Beckmann *et al.*, "A call for research on storage emissions," in *Proceedings of the 3rd Workshop on Sustainable Computer Systems (HotCarbon)*, 2024.