Chapter 23
On-Site Deployment of LLMs



Zachary Schillaci

Abstract As consumer electronics and tensor computation for *machine learning* (ML) continue to advance, model execution and training become more accessible. NVIDIA introduced the RTX 4090 graphics cards, marketed initially as gamer-oriented products, in late 2022. Though relatively expensive for consumer use, their *manufacturer's suggested retail price* (MSRP) of 1600 USD makes them affordable as a professional tool. These cards' extensive video random access memory (vRAM), computational power comparable to last-generation flagship professional cards, and ability to use single-byte floats enable a pair of them to train, fine-tune, and run on-premises *Large Language Models* (LLMs) with up to 7 billion parameters per card. Until this release, such a feat would have required data center-level equipment. Although the RTX 4090 and H100 GPU represent a qualitative step forward, iterative improvements combined with the speculated lowering of computational precision to half-byte floats could make larger models even more accessible for on-premises use. This development might, in one aspect, lower the entry barrier for cyberattackers, simplifying the process for *advanced persistent threats* (APTs) to camouflage their activities amidst unsophisticated attackers or those employing generative LLMs for non-malicious purposes. Conversely, as an alternative to cloud-hosted models, on-site LLMs may limit the possibility of private information leakage or model poisoning while offering specialized capabilities for legitimate users.

23.1 Introduction

The decision between on-premise and cloud-based *Large Language Model* (LLM) hosting solutions will likely mirror the adoption trends of conventional software products. On-premise solutions offer total control over configuration and security

Z. Schillaci (✉)
Effixis SA, Lausanne, Switzerland
e-mail: zachary.schillaci@effixis.ch

© The Author(s) 2024

A. Kucharavy et al. (eds.), *Large Language Models in Cybersecurity*,
https://doi.org/10.1007/978-3-031-54827-7_23

but have drawbacks such as significant upfront capital expenditure, increased maintenance costs, and limited scalability. In the context of self-hosting LLMs, the initial setup costs may be more substantial than traditional software implementations, primarily due to the need for specialized, high-performance GPU cards. Conversely, cloud-based solutions present a scalable and cost-effective alternative appropriate for various applications. These solutions encompass the managed hosting of open-source models (e.g., HuggingFace Inference API, Microsoft Azure ML, Together AI) and the utilization of API endpoints of proprietary models (e.g., OpenAI ChatCompletions).

For applications that prioritize security and have ample capital at their disposal, such as those in the banking, cybersecurity, and defense sectors, locally hosted solutions will likely be the preferred and perhaps legally required choice.

23.2 Open-Source Development

The on-site deployment of LLMs necessitates using either self-trained or open-source models. For most users, however, training a foundation model from scratch will be challenging, given both the exorbitantly high cost of pre-training and the growing abundance of open-source, commercial-use models. Furthermore, given the availability of efficient fine-tuning techniques such as *parameter-efficient fine-tuning* (PEFT) [1], such as *low-rank adaptation* (LoRA) [2], it has become much more feasible to adapt an open-source base model to one's specific needs.

In recent months, Meta has emerged as a pioneer in open-source LLMs. Beginning with the initial leak of Llama [3] in March 2023, Meta has since intentionally released two new models to the family: Llama-2 [4] and Code Llama [5], both of which have generous licenses covering most applications of commercial use. In the case of Llama-2, Meta released both instruction- and chat-tuned models, mimicking the exact offerings of OpenAI's Completions and ChatCompletions API endpoints, respectively [4]. The most significant variant of Llama-2 is a massive 70 billion parameter model (Llama-2 70b) comparable to OpenAI's GPT-3.5 on non-coding tasks [4]. With the release of Code Llama, which comes in variants up to 34 billion parameters and includes specifically-tuned Python versions, the coding performance is as good or better than GPT-3.5 and approaching the levels of GPT-4 [5]. The release of Llama-2 and Code Llama will undoubtedly drive open-source LLM adoption, spurring further innovation in hosting, quantizing, and fine-tuning large models.

Many small organizations and individuals are beginning to experiment with the latest open-source models. They regularly release fine-tuned, quantized, and novel variants on HuggingFace's model repository platform. In response to the surge in open-source LLM innovation, HuggingFace has created the *Open LLM Leaderboard* to evaluate and track the performance of open-source LLMs across several well-known benchmarks [6]—including the famous *Massive Multitask Language Understanding* (MMLU) dataset [7]. Similarly, *Large Model Systems Organiza-*

tion—shortened as LMSys Org—has developed a community-driven *Chatbot Arena* to rank LLMs in a tournament-style setting using the Elo rating system [8]. For LLM developers, monitoring these online leaderboards has become essential to tracking the best-performing models and identifying competing models' relative strengths and weaknesses. It should be noted that the accuracy and reliability of these benchmarks are disputed, and the robust evaluation of LLM capabilities remains an open challenge. Moreover, when open-source models are optimized specifically for these benchmarks, there is a risk of over-fitting, degrading performance on other untested yet essential tasks. For a comprehensive review of LLM evaluation methods, please refer to Ref. [9].

23.3 Technical Solution

While training LLMs from scratch remains prohibitively expensive and complex for all, but the largest AI-oriented organizations, running inference of pre-trained models remains within reach of many enterprises and motivated individuals. This is particularly the case for language models in the sub 100 billion parameter range, where running on a single consumer-grade GPU instance is possible.[1] However, the difference between, for example, experimental testing in a local development setting and deploying a production-ready solution capable of handling many concurrent users is substantial. Despite this hurdle, many organizations are investing in local model development. In response to escalating market demand, the relatively novel field of *LLM operations* (LLMOps) is experiencing significant growth and commercialization to meet these needs.

23.3.1 Serving

Serving LLMs to multiple concurrent users in a production setting is a hard technical challenge, typically requiring specialized hardware, complex software, and broad expertise covering systems design and machine learning engineering. As an example, a typical 13 billion parameter model would require nearly 26 gigabytes (GB) of memory to be loaded at its "half-precision" of 16 bits (2 bytes). Running inference of LLMs requires an additional memory overhead that scales with the total sequence length. For a 13 billion parameter model, this is estimated to be approximately 1 megabyte (MB) of state for each token in a sequence [10]. Given a typical batch size of 32 and sequence generation length of 512, this amounts to an additional per-batch memory requirement of 32 sequences $\times$ 512 tokens/sequence $\times$

[1] Note for larger models, this may require additional techniques such as quantization or CPU offloading which may come with losses in performance and speed.

1 MB/token $\approx$ 16 GB. Therefore, running such a model entirely on GPU while supporting batched inference up to a reasonable maximum sequence length would require a powerful GPU such as the NVIDIA Tesla A100 40 GB—the retail price of which is approximately 10,000 Swiss francs at the time of writing. While this is an expensive purchase for the individual user, it is well within the hardware budget of many organizations. Aside from GPUs, specialized AI hardware accelerators—such as *Google's Tensor Processing Unit* (TPU) or custom-built *field-programmable gate arrays* (FPGA)—provide alternative, albeit typically more costly solutions.

Serving LLMs in multi-user settings presents additional challenges to handle batched inference for minimal request latency properly. Performance enhancements such as flash attention, paged attention, and tensor parallelism can further speed up model inference. Other features like token streaming and sampling control are also essential for building a ChatGPT-like replacement. Open-source LLM serving frameworks address some or all these problems, including HuggingFace's text-generation-inference, vLLM, and LMSys Org's FastChat. With viable open-source options available, organizations will likely adopt these frameworks for internal use rather than building their complex solution from scratch.

23.3.2 Quantization

Quantization is a frequently used method in deep learning that reduces the memory footprint of a model by transforming some or all of the parameters to a lower precision at the expense of some degradation in performance. In the context of LLMs, models are typically trained at a "full precision" of 32 bits (4 bytes). However, once trained, LLMs can generally be run in inference at a "half-precision" of 16 bits (2 bytes) without any performance loss. While there is no clear consensus, LLMs are generally considered to be above the 1 billion parameter cutoff. At the same time, state-of-the-art open-source models, such as Meta's Llama-2 [4], can reach 70 billion parameters. Therefore, for many consumer applications, more than 16-bit quantization is required. More advanced quantization techniques can further reduce the memory footprint in such cases.

The simplest form of quantization would be to truncate all model parameters to a lower precision, either through absolute maximum or zero-point quantization. However, such a naive approach typically degrades model performance too significantly. Instead, more specialized approaches selectively truncate parameters based on the model architecture. Reference [11] proposes a novel quantization procedure, referred to as LLM.int8(), which casts the majority of parameters to 8-bit while performing 16-bit matrix multiplication for the more critical outlier parameters. This approach allows massive parameters such as BLOOM 175B to run on a single consumer-grade GPU without any performance degradation [11]. More recently, further research has suggested that 4-bit quantization is universally optimal regarding memory footprint and performance [12]. GPTQ proposes an alternative quantization method, based on Optimal Brain Quantization [13], that can quantize

massive models down to 4 and even 3 bits while significantly speeding up inference. At 4-bit precision, the footprint of LLMs may be reduced by nearly 4× the size of the 16-bit baseline, slashing the size of a 13 billion parameter model from 26 GB to only 6.5 GB. This is well within the range to fit entirely on the VRAM of consumer-grade GPUs.

23.3.3 Energy Costs

In addition to upfront hardware costs, the energy costs of training, fine-tuning, and hosting LLMs can grow prohibitively expensive. Pre-training, in particular, may require several hundred thousand or more GPU hours. In the development of Llama-2, Meta measured the GPU time required for the pre-training of different model sizes, ranging from 184,320 hours for the 7B parameter model to 1,720,320 hours for the 70B parameter model [4]. Compared to pre-training, fine-tuning is less computationally expensive as it requires on the order of 100,000 examples versus the several hundred billion or trillions of tokens needed for pre-training. However, this can only require significant GPU resources for sufficiently large models if parameter-efficient techniques are used. Hosting LLMs effectively depends on the model size, hardware setup, and the expected user load. For most imaginable internal uses, efficiently serving a moderately sized LLM to several concurrent users requires one or more high-performance GPUs. As a reference, a single NVIDIA Tesla A100 40 GB GPU has a maximum power consumption of 250 W, at least twice that of many high-end CPUs. Therefore, the operating costs of hosting LLMs are considerably higher than those of more conventional software workloads.

23.4 Risk Assessment

Despite the financial cost and technical challenge of self-hosting LLMs, specific organizations will opt to do so if they are security-constrained or require additional functionality, which is either expressly forbidden by or not technically feasible with services from cloud providers. Those who self-host LLMs to avoid safety mechanisms, such as OpenAI's built-in content moderation filter, are mainly concerned in the context of harmful use. We are now in an era where highly motivated individuals can download powerful open-source models and fine-tune them for particular harmful applications on consumer-grade hardware. This opens the door for many alarming possibilities that underscore the urgent need for responsible usage guidelines, robust monitoring, and regulation of AI technologies. Unfortunately,

given the widespread availability of this technology and the decreasing barriers to entry, there is little in the way to prevent such activity. Instead, society must emphasize education and the development of solid ethical frameworks through a combined collaboration of governments, industry stakeholders, and researchers. As reactive measures are insufficient, a proactive approach is needed to ensure that locally-hosted LLMs are used responsibly. The challenges ahead are complex, but the potential benefits of these technologies require us to face them head-on.

Mitigating potential harms and risks in the era of advanced AI models requires a comprehensive strategy encompassing, at the minimum, the following elements:

- **Education and awareness**: Continuous education and public awareness campaigns are vital to equipping individuals with the understanding and vigilance needed to counter risks. The same principles used to combat cyber threats such as phishing must be applied to the emerging challenges LLMs pose.
- **Resource monitoring**: Given the benefit of scale, massive foundation models pose the primary concern for serious misuse. At the time of writing, the most advanced foundation models—namely GPT-4, Claude 2, and Bard—remain under the strict control of private companies. This is slowly beginning to change with the public release of Llama-2 [4] and the rapid development in the open-source LLM community. Fortunately, running frontier models of the GPT-4 scale requires massive computational resources, which were not easily accessible at the time of writing. Going forward, monitoring the most powerful computational resources will ensure responsible use and prevent potential misuse.
- **Rapid harm detection**: Developing and employing automated detection systems, including those leveraging AI and LLMs, will enable swift identification and response to potential harms. These mechanisms must be robust and agile to adapt to the rapidly evolving landscape of AI technology.

References

1. Haokun Liu et al. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning, 2022.
2. Edward J. Hu et al. Lora: Low-rank adaptation of large language models, 2021.
3. Hugo Touvron et al. Llama: Open and efficient foundation language models, 2023.
4. Hugo Touvron et al. Llama 2: Open foundation and fine-tuned chat models, 2023.
5. Baptiste Rozière et al. Code llama: Open foundation models for code, 2023.
6. Edward Beeching et al. Open llm leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard, 2023.
7. Dan Hendrycks et al. Measuring massive multitask language understanding, 2021.
8. Lianmin Zheng et al. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.
9. Yupeng Chang et al. A survey on evaluation of large language models, 2023.

10. Ray Project. Numbers every LLM developer should know, 2023. GitHub repository.
11. Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale, 2022.
12. Tim Dettmers and Luke Zettlemoyer. The case for 4-bit precision: k-bit inference scaling laws, 2023.
13. Elias Frantar, Sidak Pal Singh, and Dan Alistarh. Optimal brain compression: A framework for accurate post-training quantization and pruning, 2023.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

