LLM Deployment

Yannis Bendi-Ouis
Dan Dutarte
Xavier Hinaut
-
Centre Inria de l'Université de Bordeaux

yannis.bendi-ouis@inria.fr
dan.dutarte@inria.fr
xavier.hinaut@inria.fr

Abstract. Since the release of ChatGPT [5] in November 2023, large language models (LLMs) [1] have seen considerable success, including in the open-source community, with many open-weight models available. However, the requirements to deploy such a service are often unknown and difficult to evaluate in advance. To facilitate this process, we conducted numerous tests at the Centre Inria de l'Université de Bordeaux. In this article, we propose a comparison of the performance of several models of different sizes (mainly Mistral [2][3] and LLaMa [4]) depending on the available GPUs, using vLLM [8], a Python library designed to optimize the inference of these models. Our results provide valuable information for private and public groups wishing to deploy LLMs, allowing them to evaluate the performance of different models based on their available hardware. This study thus contributes to facilitating the adoption and use of these large language models in various application domains.

1. Introduction

Following the release of ChatGPT by OpenAI in November 2023 [5], large language models (LLMs) [1] have sparked great interest in the private sector, with many companies seeking to offer services based on these models. However, training and inference of such models remain inaccessible to the general public, requiring considerable computational power and high-quality data. For example, Meta AI acquired 350,000 NVIDIA H100 GPUs in January 2024, at an estimated cost of $9 billion, specifically for training LLMs. Their training of LLaMa-3 was performed on $14 * 10^{12}$ tokens.

Some pioneering companies quickly realized they could benefit from a monopoly on this technological advancement, giving them unprecedented decision-making power. To ensure they retain this monopoly, these companies are now lobbying governments for the regulation of these models, citing the risks and potential dangers of their malicious use. They propose measures ranging from banning the training of models beyond a certain computational power to government control of GPUs, with the possibility of remote deactivation [12].

However, it is essential that these tools are not solely in the hands of a few powerful actors, capable of influencing the biases of their models that they distribute on a large scale, thus allowing them mass influence. The transparency of these models, with the opening of training data and associated weights, is the most appropriate solution to allow

any external entity to verify the reliability and security of the proposed models. Although this approach is not appreciated by the majority of these companies, some of them, such as Meta and Mistral, are investing heavily in open-weight models, freely distributing variants of their LlaMa [4] and Mistral [2][3] models.

Thanks to these efforts, many groups, both public and private, are now able to deploy powerful models, thus ensuring the sovereignty of their data and avoiding the concentration of this wealth and potential power in a single point. However, even though these models are available to everyone, it is not easy to deploy them or estimate the resources needed to do so. While it is simple to serve a model to one user, it is much more complex to deploy it for tens, hundreds, or even thousands of simultaneous users. In this context, we conducted several tests at the Centre Inria de l'Université de Bordeaux concerning the deployment of such models.

2. Objectives

The main objective of our study is to address the security and confidentiality concerns raised by the increasing use of proprietary LLM solutions - such as ChatGPT - by students and researchers at the Centre Inria de l'Université de Bordeaux. Indeed, a huge part of our students use these tools to help them in their daily work, whether for writing, programming, proofreading articles, or brainstorming.

However, the use of these proprietary solutions raises serious security and confidentiality issues. They do not guarantee the confidentiality of data, and the private interests behind them can potentially use them for commercial purposes, training, or even industrial espionage. This last point is particularly concerning for a research center like Inria, which must ensure the confidentiality of its employees' research work and is in direct competition with the companies offering these proprietary solutions.

It is therefore crucial for Inria to propose alternative solutions and preserve its digital sovereignty. Moreover, given the growing importance of this technology, more and more researchers and students wish to conduct experiments based on LLMs. For example, setting up RAG (Retrieval Augmented Generation) [6][7] systems is a common application in the business world, which involves using an LLM to "discuss" with its data. It would therefore be interesting to offer them an adapted service.

3. Prerequisites

3.1. Skills

To deploy an LLM on a GPU, certain knowledge and skills are required in Linux and Python development, as well as a strong curiosity about existing models and quantification. Although understanding the internal workings of Transformers is not necessary, it can be an asset.

The required skills include the ability to update CUDA drivers (version 12 minimum), install a version of Python (minimum 3.9), install Python dependencies, make HTTP requests, and choose the right model for your use case, quantified or not depending on your resources.

3.2. Hardware

We conducted our tests on the Plafrim computing server, equipped with two types of GPUs:

- NVIDIA V100 16 GB
- NVIDIA A100 40 GB

3.3. Software

We used vLLM [8], a Python library designed to optimize the inference of these models. This library requires at least the prior installation of Python 3.9 and CUDA 12.

The advantage of vLLM over other solutions is its ability to handle multiple requests simultaneously, without a queue and without a linear increase in computation time depending on the number of requests, but rather logarithmic.

However, depending on the available hardware, other solutions can be considered. Notably, tensorRT-LLM offers excellent performance with NVIDIA GPUs, and lllama.cpp provides remarkable performance on Macs equipped with M1, M2, or M3 chips.

3.4. Quantification

Some models can be very large, making it particularly difficult to load them into the available hardware - limited by its VRAM.

To address this problem, one of the best solutions is to quantify our models. Instead of writing the values of our weights on 16 or 32 bits, we can accept a slight loss of precision and write them on 4 or 8 bits.

This loss has been evaluated several times, and although it varies depending on the models and quantification methods used, we can affirm that it is negligible up to 6-bits and acceptable up to 4-bits. However, for a number of parameters greater than 70 billion, the models are robust enough to allow quantification below 4-bits while maintaining good coherence.

Among the different quantification methods [10], we can mention AWQ [9], GPTQ [11], and GGUF (llama.cpp).

4. Experimentation

In this study, we seek to determine the maximum load of simultaneous requests that a server equipped with V100 16 GB or A100 40 GB GPUs can support, depending on the LLM used. For this, we conducted tests by progressively increasing the number of simultaneous requests and the size of the prompts, until reaching the maximum load. For each request, we measured the time required to generate 100 tokens. And, for each model and GPU, we measured the memory load, execution speed, and number of tokens per second depending on the number of simultaneous requests and the maximum context size.

We chose to focus mainly on the models proposed by Mistral AI [2][3], due to their diversity, popularity, and skills. We also appreciate their performance on European languages, particularly French. Additionally, their Mixture of Experts [3] architecture allows for computational savings during inference, by selecting only a part of the model's weights at each step, which also reduces energy consumption.

Moreover, we included the LLaMa-3-70B [4] model from Meta, which achieves performance comparable to GPT-4 with its 70 billion parameters. This model size seems to be a good compromise between size and performance, justifying its inclusion in our study.

Thus, we tested the following models:

- Mistral-7B
- Codestral-22b
- Mixtral-8x7b
- Mixtral-8x22b
- LLaMa-3-70B

5. Results

5.1. Mistral 7B on 2 V100 16 GB

Requests	Number of tokens						
	31	63	119	296	480	822	2193
1	1.8	1.8	1.9	1.9	1.9	2.1	2.3
2	2.1	2.1	2.0	2.2	2.3	2.6	2.8
4	2.2	2.3	2.1	2.6	2.5	2.8	3.7
8	2.4	2.4	2.5	2.7	3.0	3.5	5.9
16	2.9	2.9	3.0	3.8	4.2	5.2	9.2
32	4.2	4.2	4.5	5.4	6.9	8.8	19.0
64	6.7	7.1	7.7	9.8	11.9	17.1	36.0
128	10.6	10.4	11.5	16.2	24.4	33.3	72.1

Table 1. Time (s) to resolve the request with Mistral 7B on 2 V100 16 GB

5.2. Codestral 22B AWQ 4-bits on 1 A100 40 GB

Requests	Number of tokens						
	31	63	119	296	480	822	2193
1	2.3	2.3	2.4	2.5	2.6	2.6	3.0
2	2.3	2.4	2.5	2.7	2.7	2.8	3.5
4	2.4	2.5	2.6	2.8	3.0	3.4	4.8
8	2.6	2.7	2.8	3.2	3.7	4.5	7.4
16	3.0	3.2	3.4	4.2	5.0	6.4	12.3
32	4.5	4.8	5.4	6.7	8.4	11.4	23.1
64	7.9	8.5	9.3	12.3	15.8	21.6	47.7
128	14.3	15.4	17.6	24.2	29.9	46.4	96.2

Table 2. Time (s) to resolve the request with Codestral 22B AWQ 4-bits on 1 A100 40 GB

5.3. Codestral 22B GPTQ 8-bits on 2 V100 16 GB

Requests	Number of tokens						
	31	63	119	296	480	822	2193
1	3.1	3.2	3.2	3.3	3.4	3.7	4.3
2	3.3	3.4	3.4	3.6	4.0	4.4	5.8
4	3.7	3.8	3.9	4.4	4.8	5.6	8.8
8	4.8	5.1	5.3	6.0	6.8	8.8	15
16	7.1	7.5	7.9	9.8	11.7	14.3	27.5
32	10.4	10.9	11.9	15.3	19.0	24.6	53.8
64	15.5	17.0	18.7	25.9	32.1	43.9	108.2
128	21.7	-	-	-	-	-	-

Table 3. Time (s) to resolve the request with Codestral 22B GPTQ 8-bits on 2 V100 16 GB

5.4. Codestral 22B on 2 A100 40 GB

Requests	Number of tokens						
	31	63	119	296	480	822	2193
1	2.3	2.3	2.4	2.4	2.5	2.6	2.8
2	2.3	2.3	2.4	2.5	2.6	2.7	3.3
4	2.4	2.4	2.5	2.7	2.8	3.1	4.2
8	2.5	2.6	2.8	3.1	3.4	4.1	6.3
16	2.8	2.9	3.2	3.8	4.4	5.6	10.2
32	3.3	3.7	4.0	5.2	6.4	8.8	18.1
64	4.3	4.6	5.7	8.0	10.5	15.5	36.8
128	6.8	7.8	9.4	14.5	19.6	32.9	71.1

Table 4. Time (s) to resolve the request with Codestral 22B on 2 A100 40 GB

5.5. Mixtral 8x7B AWQ 4-bits on 2 A100 40 GB

Requests	Number of tokens						
	31	63	119	296	480	822	2193
1	3.1	3.2	3.6	3.4	3.5	3.5	4.1
2	3.3	3.3	3.5	3.5	3.8	3.8	4.7
4	3.5	3.6	3.5	4.3	4.1	4.6	6.2
8	3.8	3.8	4.0	4.3	4.9	5.7	9.3
16	4.3	4.6	4.9	6.0	6.7	8.5	15.5
32	6.0	6.4	7.6	8.7	10.9	14.2	-
64	10.0	10.5	11.6	15.5	-	-	-
128	18.5	19.6	21.5	-	-	-	-

Table 5. Time (s) to resolve the request with Mixtral 8x7B AWQ 4-bits on 2 A100 40 GB

5.6. Mixtral 8x22B AWQ 4-bits on 4 A100 40 GB

Requests	Number of tokens						
	31	63	119	296	480	822	2193
1	6.0	6.0	6.3	6.8	7.0	7.6	10.9
2	7.2	7.1	7.4	8.4	8.7	10.3	16.7
4	8.0	8.1	8.7	10.3	11.9	14.2	21.3
8	9.0	9.4	10.0	13.0	16.7	19.4	36.2
16	11.0	12.2	13.2	21.1	26.5	31.9	66.4
32	16.0	17.6	22.6	33.6	37.7	56.7	-
64	28.0	31.8	35.9	56.0	71.7	-	-
128	52.0	55.3	67.0	111.7	-	-	-

Table 6. Time (s) to resolve the request with Mixtral 8x22B AWQ 4-bits on 4 A100 40 GB

5.7. LLaMa-3 70B AWQ 4-bits on 2 A100 40 GB

Requests	Number of tokens						
	21	51	97	240	398	703	1848
1	3.6	3.7	3.7	3.9	4.0	4.2	4.8
2	3.7	3.7	3.9	4.1	4.2	4.5	5.8
4	3.8	4.0	4.1	4.4	4.7	5.4	7.9
8	4.3	4.5	4.8	5.1	5.9	7.4	12.5
16	4.9	5.2	5.7	6.9	8.3	10.8	21.4
32	7.6	8.2	8.7	11.1	13.7	19.6	40.9
64	12.9	13.9	15.2	20.3	25.8	37.5	-
128	23.2	-	-	-	-	-	-

Table 7. Time (s) to resolve the request with LLaMa-3 70B AWQ 4-bits on 2 A100 40 GB

6. Discussions & Perspectives

The first thing we can notice is that the larger the context size, the slower the model is at generating 100 tokens. This is expected and is related to its complexity, which grows quadratically with the context size.

Moreover, as we can see with the Mixtral [3] and Llama [4] models, just because a model can be loaded does not mean it can be used for any context size. The context size has a quadratic cost in RAM (or VRAM), which adds to the model size, potentially significantly increasing memory requirements.

On the other hand, what we can also observe is that we do not have a linear loss of efficiency with the number of simultaneous requests: the time required to respond to a request does not double when the number of simultaneous requests doubles. However, this seems less true when the request size exceeds a certain threshold.

Finally, we can notice that although the cost of these GPUs is far from trivial (V100 16GB $\approx$ \$5000, A100 40GB $\approx$ \$8500), it is not necessary to possess an exorbitant quantity to successfully run a local alternative to ChatGPT or other proprietary solutions. In

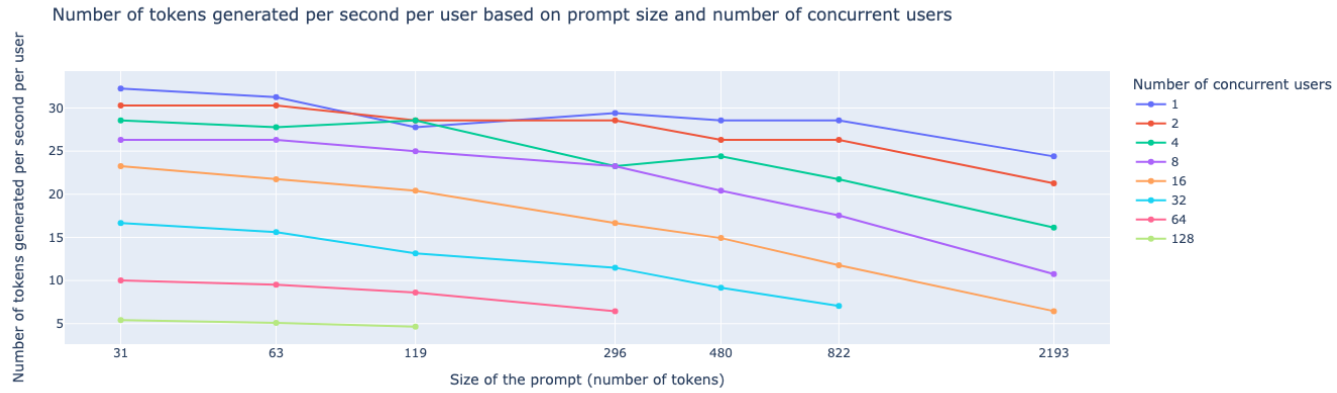

Figure 1. Number of tokens generated per second and per user for Mixtral 8x7B AWQ 4-bits on 2 A100 40GB

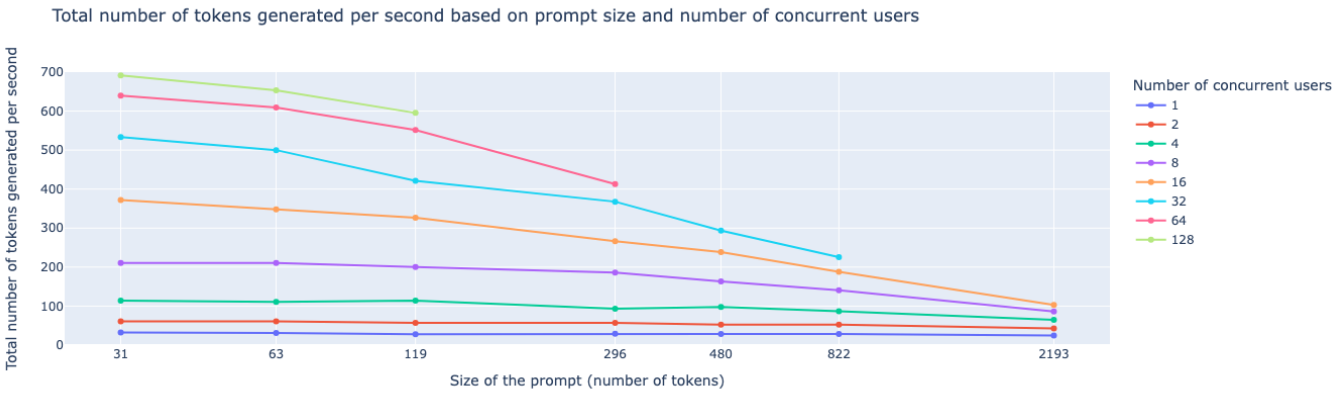

Figure 2. Total number of token generated per second for Mixtral 8x7B AWQ 4-bits on 2 A100 40GB

fact, with two A100 40GB GPUs (or a single 80GB GPU), it is already possible to run LLaMa-3-70B or Mixtral 8x7B under very good conditions. According to numerous benchmarks and user reviews, these models are serious contenders to GPT-4.

To a lesser extent, it is possible to host smaller models (in the range of 7 to 30B) and achieve extremely impressive generation speeds, especially if requests are parallelized.

For example, we can observe in Figure 1 the number of tokens generated per second and per user as a function of the number of simultaneous requests and the size of the prompt, and in Figure 2 the total number of tokens generated per second (for all users combined) as a function of the number of simultaneous requests and the size of the prompts. Thus, we can observe that for a model like Mixtral 8x7B — which has 49B parameters and only 12-13B active parameters at any given time (MoE architecture) [3] — we can achieve up to 700 tokens/second for 128 simultaneous requests with a nearly zero context (30 tokens), and we can obtain an inference speed of just under 20 tokens per second for 20 simultaneous users.

7. Conclusion

In this article, we have presented a comparative study of the performance of several large language models (LLMs) based on available hardware resources. Our results demonstrate that models such as Mistral [2][3] and LLaMa [5] can be efficiently deployed on V100 and A100 GPUs, offering competitive performance compared to proprietary solutions like ChatGPT [5].

These findings have significant implications for both the academic and industrial communities, providing valuable insights into the resources required to deploy LLMs. They also highlight the importance of transparency and digital sovereignty, enabling public and private groups to deploy open-source models without relying on proprietary solutions.

We strongly encourage various public and private groups to deploy their own LLM solutions, particularly by utilizing open-source (or open-weight) models. This approach not only reduces our digital dependence but also brings us closer to achieving sovereignty over our data.

We extend our gratitude to the Centre Inria de l'Université de Bordeaux for their support and resources, which made this study possible.

8. References

[1] *A Survey of Large Language Models*, arXiv:2303.18223.

[2] *Mistral 7B*, arXiv:2310.06825.

[3] *Mixtral of experts*, arXiv:2401.04088.

[4] *LLaMA: Open and Efficient Foundation Language Models*, arXiv:2302.13971.

[5] *GPT-4 Technical Report*, arXiv:2303.08774.

[6] *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*, arXiv:2005.11401.

[7] *From Local to Global: A Graph RAG Approach to Query-Focused Summarization*, arXiv:2404.16130.

[8] *Efficient Memory Management for Large Language Model Serving with PagedAttention*, arXiv:2309.06180.

[9] *AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration*, arXiv:2306.00978.

[10] *Benchmarking Emerging Deep Learning Quantization Methods for Energy Efficiency*, tusharma.in/preprints/Greens2024.pdf.

[11] *GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers*, arXiv:2210.17323.

[12] *Reimagining secure infrastructure for advanced AI*, https://openai.com/index/reimagining-secure-infrastructure-for-advanced-ai/