Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Towards Pareto Optimal Throughput in Small Language Model Serving

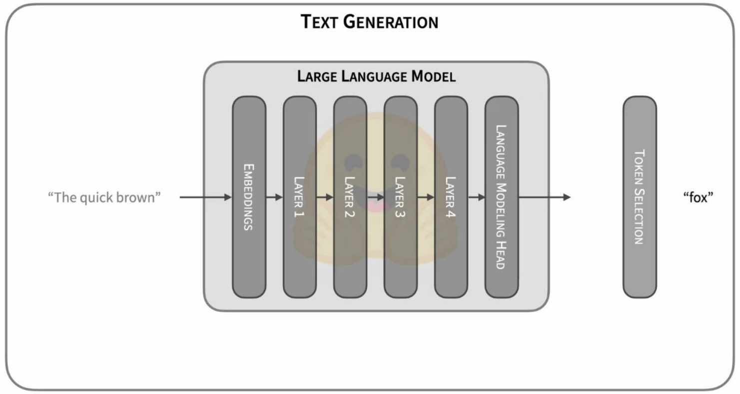
Pol G. Recasens, Yue Zhu, Chen Wang, Olivier Tardieu, Eun K. Lee, Alaa Youssef, Josep Ll. Berral, Jordi Torres

BSC & IBM Research
pol.garcia@bsc.es

Agenda

1. LLM inference
2. Motivation
3. Batching
4. Performance limiters
5. Small Language Models
6. Experimentation
7. Discussion

LLM inference



The autoregressive generation of decoder-only Transformer models can be decomposed in two phases.

- **Prefill phase**: the model generates the intermediate keys and values (KV) of the prompt tokens.
- **Autoregressive phase**: the model generates one token per iteration.

The space in the GPU HBM where we store the intermediate results is named KV cache.

https://huggingface.co/docs/transformers/llm_tutorial

Motivation

Serving Large Language Models (LLMs) is **memory intensive**.

- OPT-175B requires 350GB just to host the model weights.

The incremental decoding of autoregressive models limits the serving performance.

Matrix-vector operations in single-batch inference.

+

Large cost of loading model weights from GPU HBM to on-chip SRAM.

=

Low arithmetic intensity (ratio OPS:BYTE).

Batching

Batching increments the arithmetic intensity.

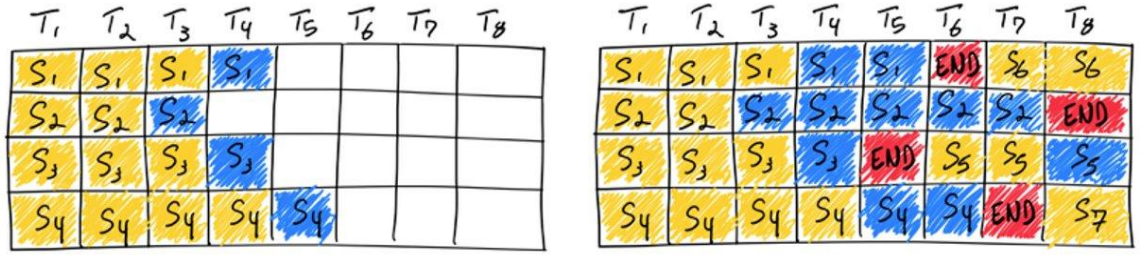
- Computing more sequences for the same transfer of weights.

-In continuous batching [1] the scheduler decides at each iteration which requests join or leave batch.

Batching techniques are employed to increase the system’s **throughput**.

- Number of requests processed per second by the engine.

- Good serving performance should maximize the throughput while providing low latency to users.



https://www.anyscale.com/blog/continuous-batching-llm-inference

[1] Yu, Gyeong-In, et al. "Orca: A distributed serving system for {Transformer-Based} generative models." *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*. 2022.

Performance limiters

Performance of an inference step on a given processor can be:

- **Memory-IO bound**: limited by the time spent accessing memory.
- **Compute bound**: limited by the time spent computing operations.

Increasing the number of concurrent requests (batch size) increases the computational cost.

- If the compute time is larger than the memory-IO time we reach a performance upper-bound.
    - Throughput plateau.

LLM serving is memory-IO bound.

- The high memory demands of the model weights and the KV cache limits the batch size.

Small Language Models

Small Language Models (SLMs, ≈ 2.7B) are increasingly important.

- Can be deployed by resource-constrained users at their local machines.
- Offer a good performance in specific tasks.

Emerging techniques for reducing memory requirements in language model serving include:

- Quantization.
- Sparsity.
- Offloading

The reduced memory footprint of SLMs allows for large batch sizes.

- Are we still in the memory-IO bound regime?

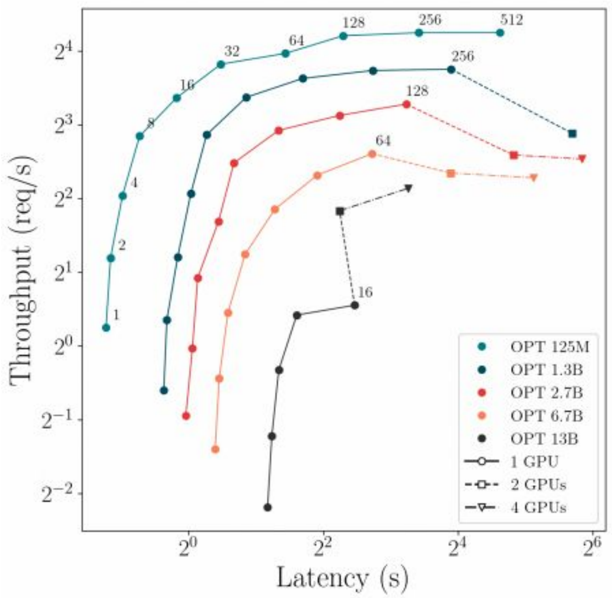
Experimentation

Serving OPT Small Language Models from 125M to 6.7B parameter range.

Requests generated from ShareGPT dataset (768 tokens/request).

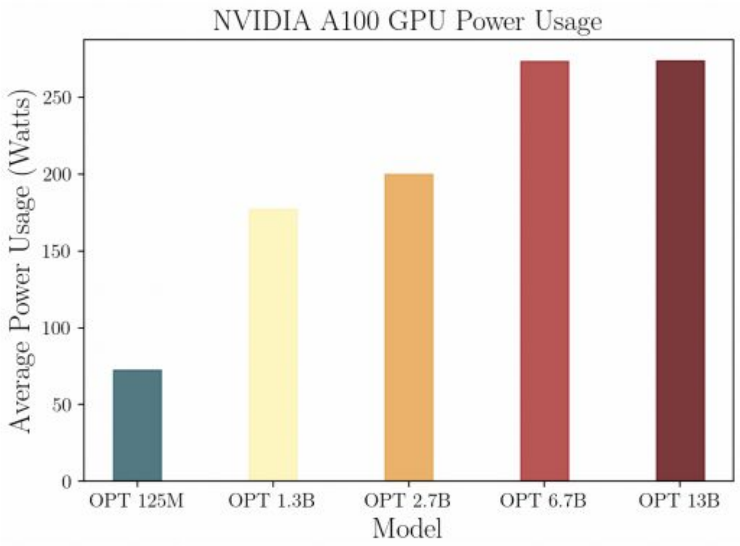
vLLM serving engine [2].

40GB A100 GPUs.

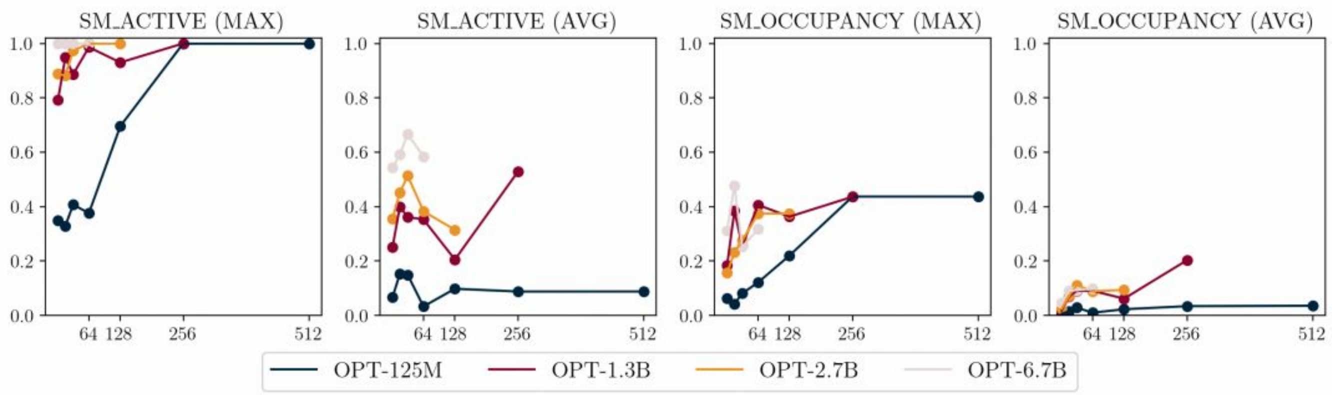


[2] Kwon, Woosuk, et al. "Efficient memory management for large language model serving with pagedattention." *Proceedings of the 29th Symposium on Operating Systems Principles*. 2023.

Experimentation



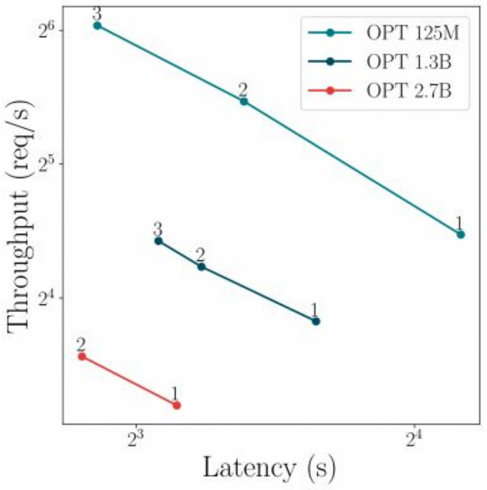
Model	Parameters	Maximum batch size
OPT-125M	250 MB	512
OPT-1.3B	2.6 GB	256
OPT-2.7B	5.4 GB	128
OPT-6.7B	13.4 GB	64
OPT-13 B	26 GB	16

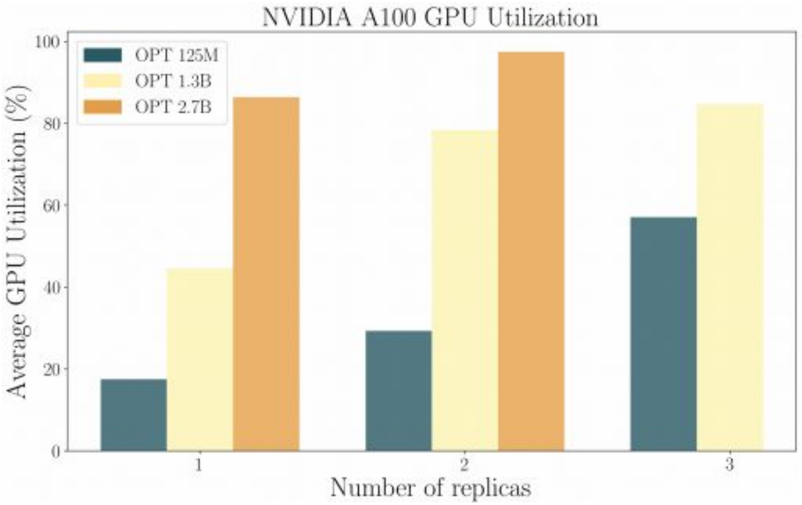


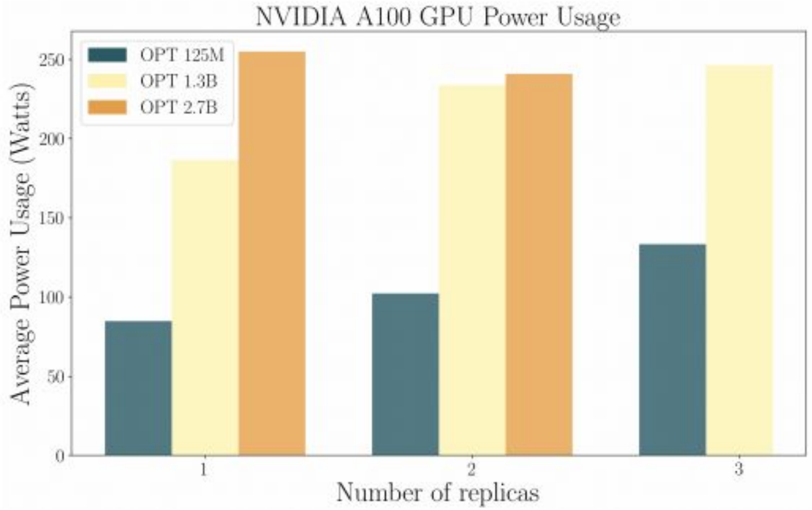
Experimentation – model replication

We observe a throughput plateau in SLMs within a single GPU.

- Overprovisioning memory to the model does not correlate to a performance improvement.
- We can limit the memory allocated to each model and run multiple instances simultaneously.







Discussion

High memory transfer, low compute

- Large amount of memory transfer with minimal computational workload in single-batch inference.
- Resulting in a memory-IO bound regime.

Discussion

High memory transfer, low compute

- Large amount of memory transfer with minimal computational workload in single-batch inference.
- Resulting in a memory-IO bound regime.

Increasing interest in reducing memory demands on serving.

- Approaches: SLMs, quantization, offloading, sparsity.
- Implicit increase of the potential batch size.

Discussion

High memory transfer, low compute

- Large amount of memory transfer with minimal computational workload in single-batch inference.
- Resulting in a memory-IO bound regime.

Increasing interest in reducing memory demands on serving.

- Approaches: SLMs, quantization, offloading, sparsity.
- Implicit increase of the potential batch size.

Reaching throughput plateau with SLMs within a single accelerator.

- Limit the memory assigned to each small model depending on its size and replicate?
- This approach can be complemented with other optimizations to further reduce the memory demand.

Towards Pareto Optimal Throughput in Small Language Model Serving

Pol G. Recasens, Yue Zhu, Chen Wang, Olivier Tardieu, Eun K. Lee, Alaa Youssef, Josep Ll. Berral, Jordi Torres

BSC & IBM Research
pol.garcia@bsc.es