UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Challenges and Opportunities on serving LLMs

Pol Garcia Recasens

(w/ Chen Wang and Yue Zhu)

CROMAI group - BSC & UPC

Visiting IBM TJ Watson (Sep'23 – Dec'23)

https://pgarec.github.io/
pol.garcia@bsc.es

IBM TJ Watson

Agenda

1. Understanding Text Generation

- Problem formulation
- Attention

2. Characterizing current serving systems

- Workflow
- Metrics
- Batching / PagedAttention

4. Evaluation

5. On-going work

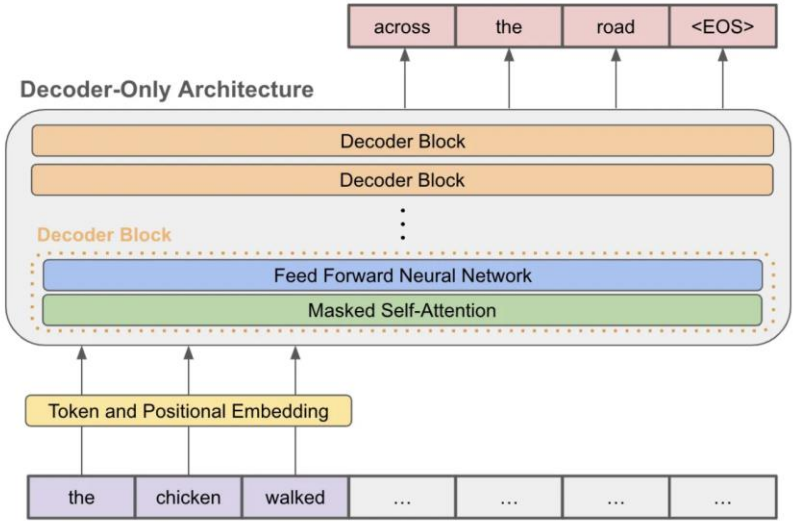
Understanding Text Generation

Core task for causal language models

Emerging properties of LLMs due to next-token prediction pre-training -- few shot learners

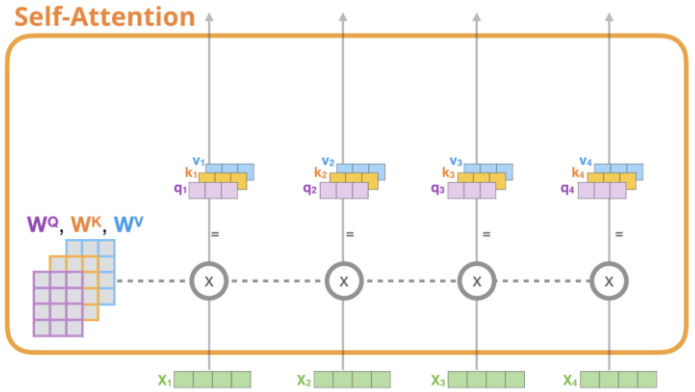
For each request

- You start with a sequence of tokens (called the "prefix" or "prompt")
- The LLM generates one token per step (forward pass), and stops when it generates a special token or reaches a maximum sequence length

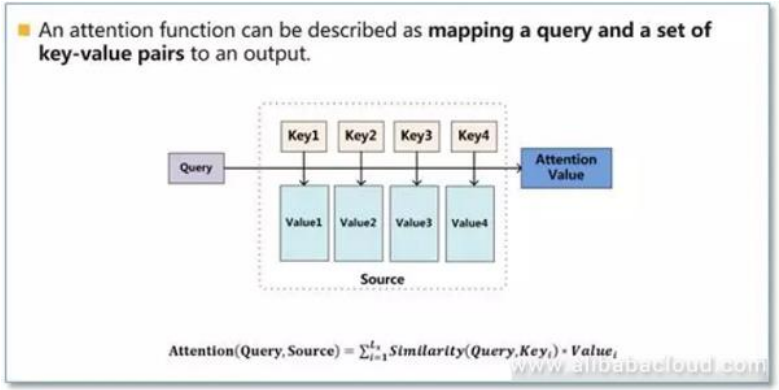


Understanding Text Generation

Attention models the context between tokens, key for long-range dependencies



$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



Attention needs keys and values of all preceding tokens -> internal states should be maintained across iterations to avoid re-computation. This scales with the number of layers and hidden dimensions.

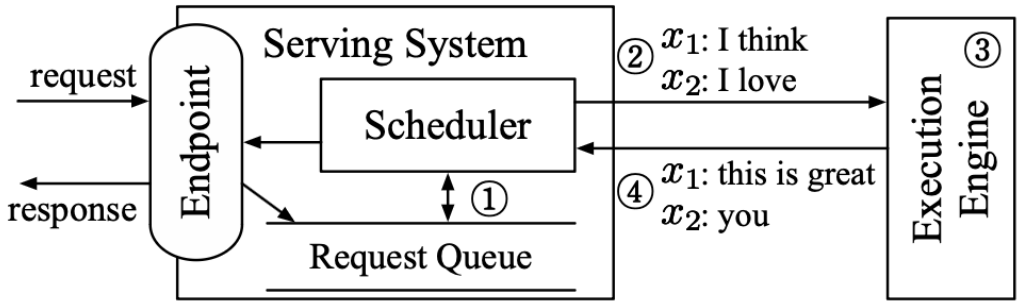
Challenge: The amount of memory consumed per prompt scales with the size of the model and the length of the input and output

Characterizing current serving systems

Users submit requests to an inference service

We are interested in maximizing the system's throughput and minimizing the user's latency

Features and optimizations provided by serving systems -> batching is key!



There are more optimizations techniques (quantization, compression, parallelization)

SOTA serving systems: *DeepSpeed-FastGen, vLLM, TGI, ORCA, AlpaServe, FlexGen*

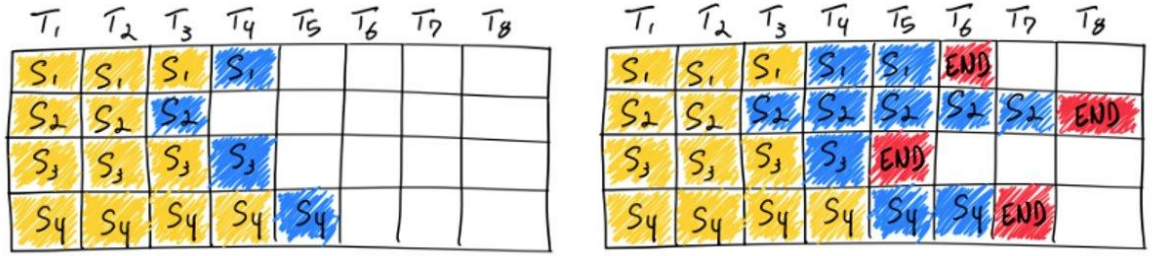
Characterizing current serving systems

Two levels of granularity

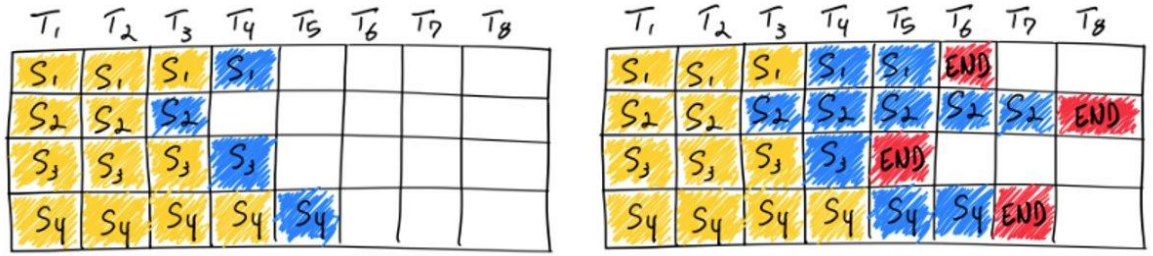
- Request-level granularity
- Iteration-level granularity

Three types of batching

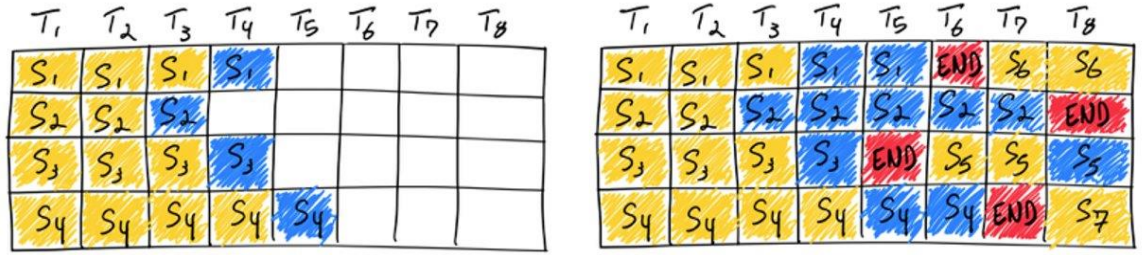
- Static batching
- Dynamic batching
- Continuous batching



Static batching



Dynamic batching



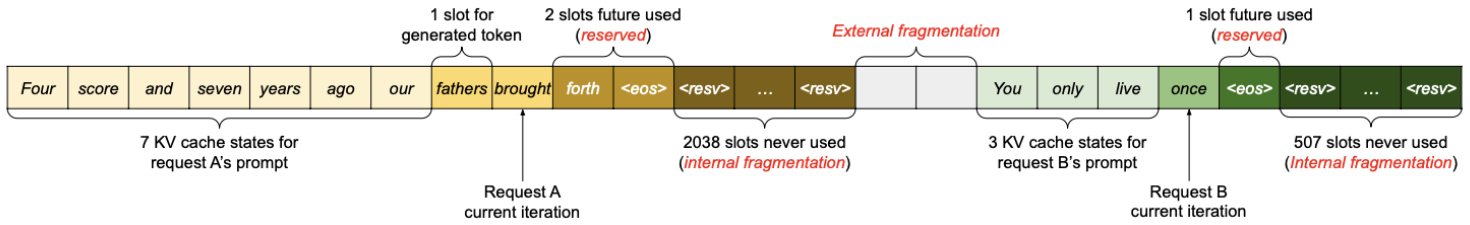
Continuous batching with Paged Attention

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

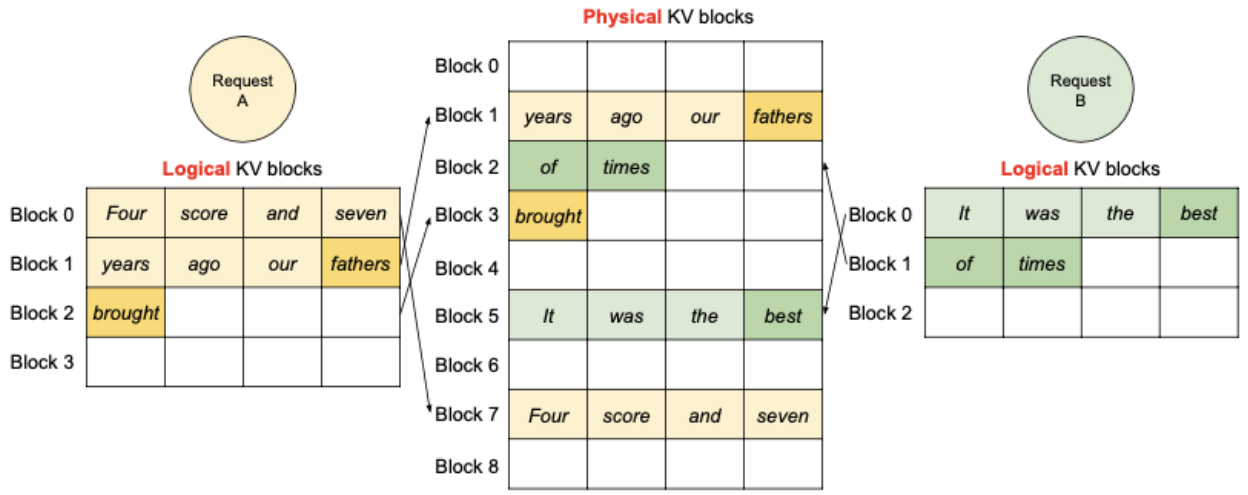
Paged Attention

Identified memory fragmentations in the KV cache management

PagedAttention: attention mechanism that allows to store memory blocks in non-contiguous space



Continuous batching



Continuous batching with Paged Attention

Evaluation: set-up

Model: facebook/OPT-125m

Dataset: 500 sentences of ShareGPT

Input length: 512 tokens

Output length: 32 / 64 /128 / 256 tokens

Arrival rate: Poisson process, infinite / r100 / r10

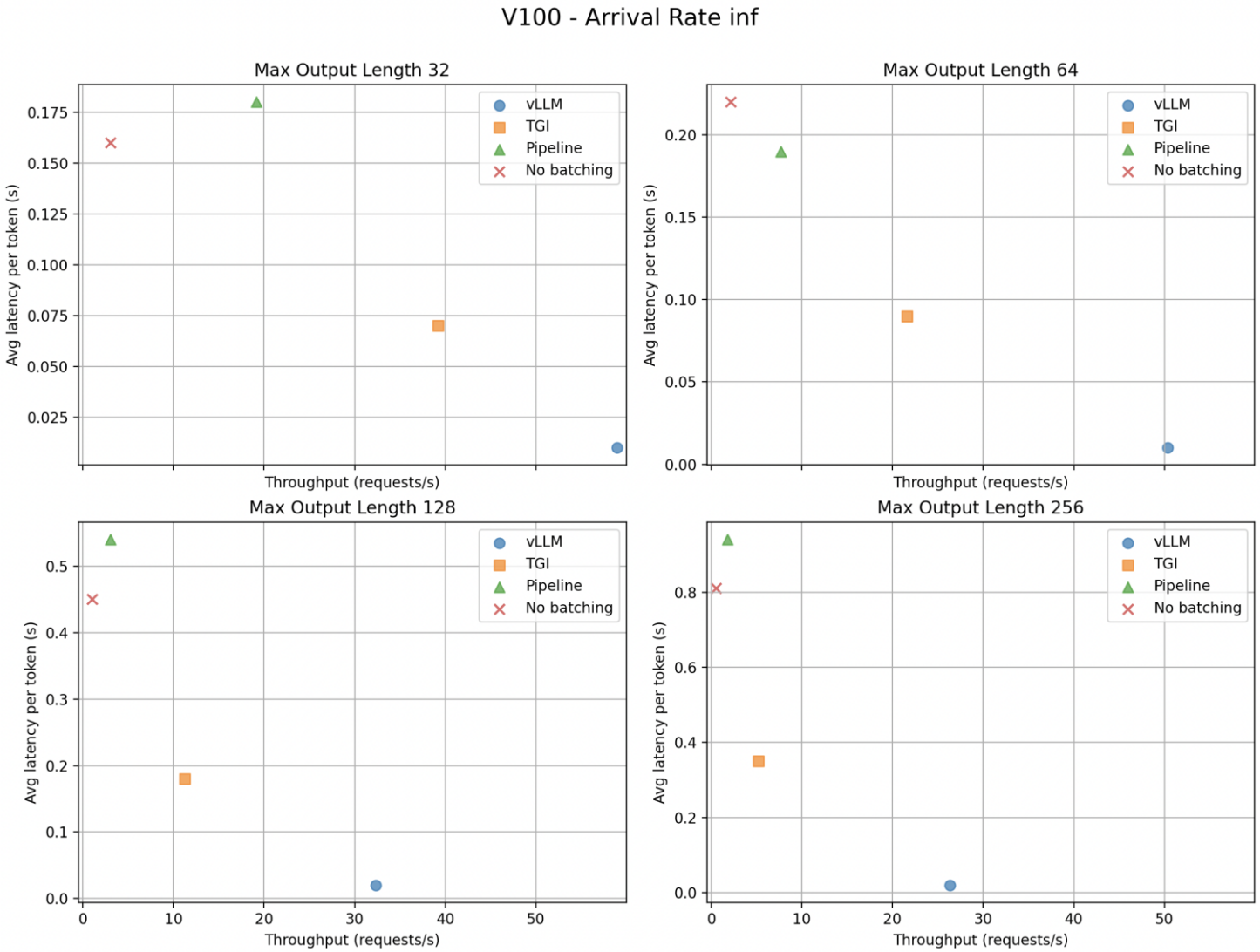
GPU: 1 NVIDIA V100

Throughput / latency per token

Frameworks

- VLLM: continuous batching with Paged Attention
- TGI: continuous batching (with Paged Attention?)
- Pipeline: dynamic batching
- No batching

Evaluation


V100 - Arrival Rate inf
Max Output Length 32
Max Output Length 64
Max Output Length 128
Max Output Length 256
vLLM
TGI
Pipeline
No batching
Avg latency per token (s)
Throughput (requests/s)

Evaluation: set-up

Model: facebook/OPT-125m, facebook/OPT-6.7b, facebook/OPT-13b

Dataset: 500 sentences of ShareGPT

Input length: 512 tokens

Output length: 32 / 64 /128 / 256 tokens

Arrival rate: Poisson process, infinite / r100 / r10

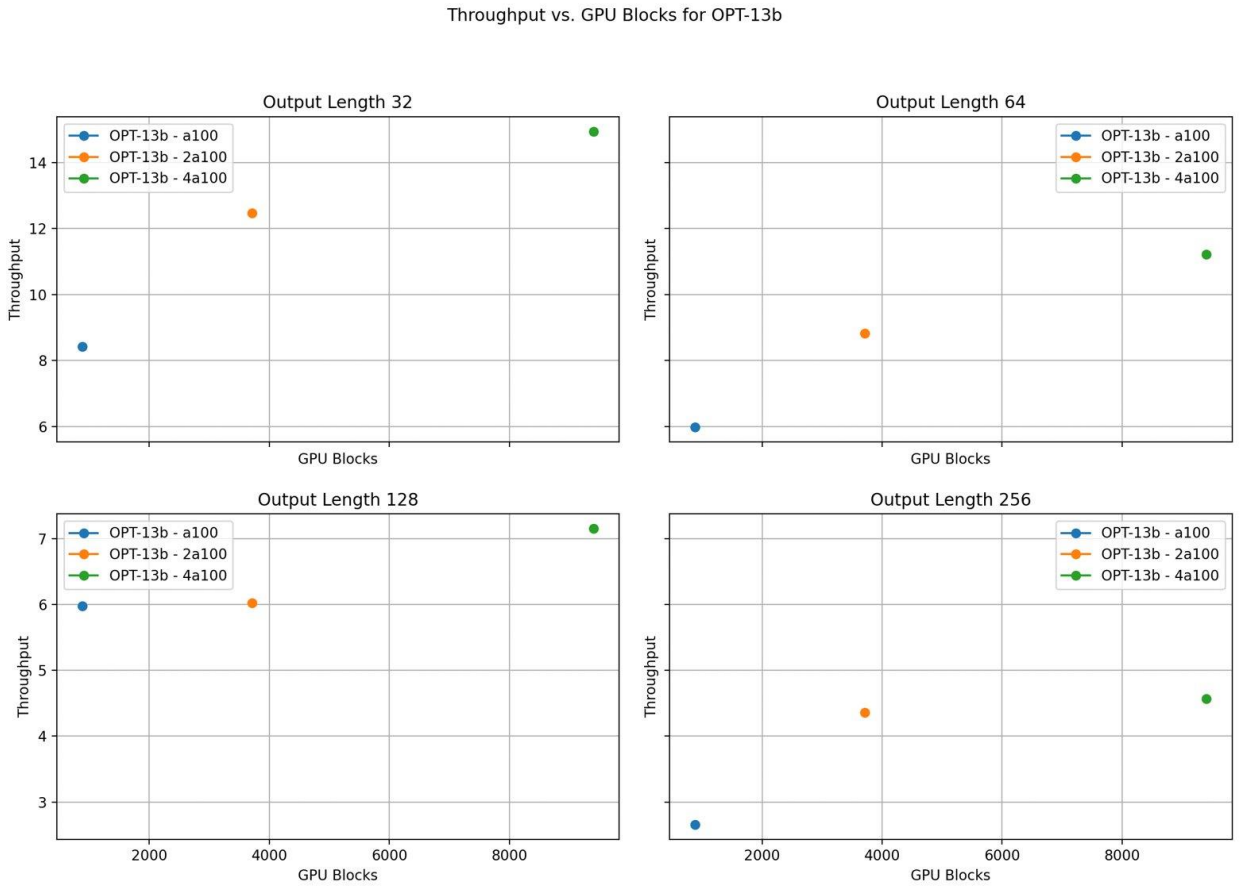
GPU: 1 V100, 1 A100, 2 A100, 4 A100

Throughput / number of GPU blocks

Model parallelism

Trade-off between memory-bandwidth IO bound and compute bound

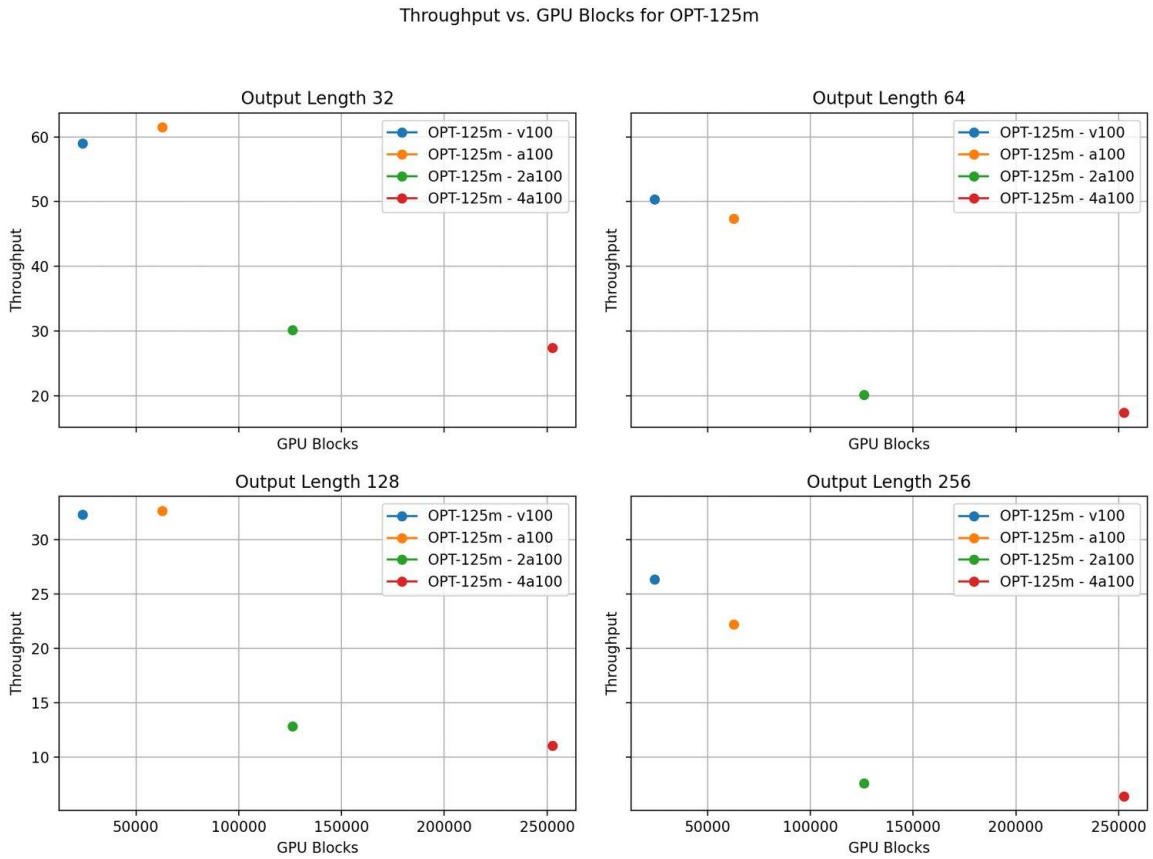
We empirically see that model parallelism is beneficial for large models



Model parallelism

Trade-off between memory-bandwidth IO bound and compute bound

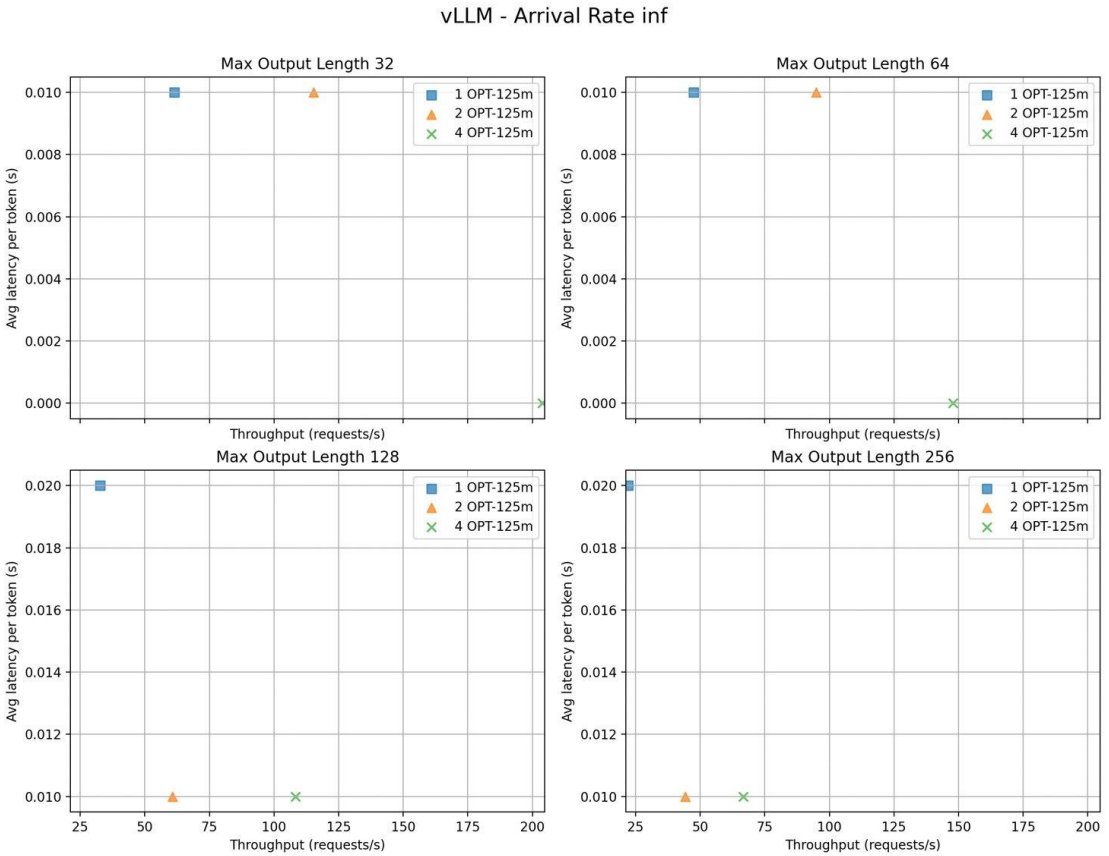
We empirically see that model parallelism is beneficial for large models



Model replication

Hypothesis: For small models are compute-bound in a single GPU

Model replication is more suitable?



UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Challenges and Opportunities on serving LLMs

Pol Garcia Recasens

(w/ Chen Wang and Yue Zhu)

CROMAI group - BSC & UPC

Visiting IBM TJ Watson (Sep'23 – Dec'23)

https://pgarec.github.io/
pol.garcia@bsc.es

IBM TJ Watson