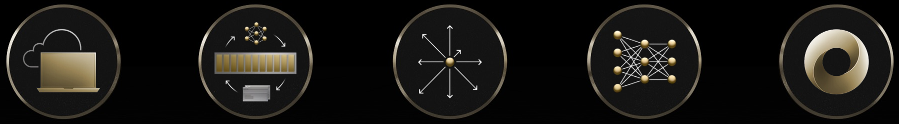
NVIDIA

Megatron-LM

Bryan Catanzaro, VP Applied Deep Learning Research

ACCELERATED COMPUTING:
DO THE COMPUTATIONALLY IMPOSSIBLE



Incredible speed-ups take more than just powerful chips

Full-stack invention: chips, systems, frameworks, compilers, algorithms, apps

Entire stack must be co-optimized

This is mostly software work

The Soul of Megatron-LM

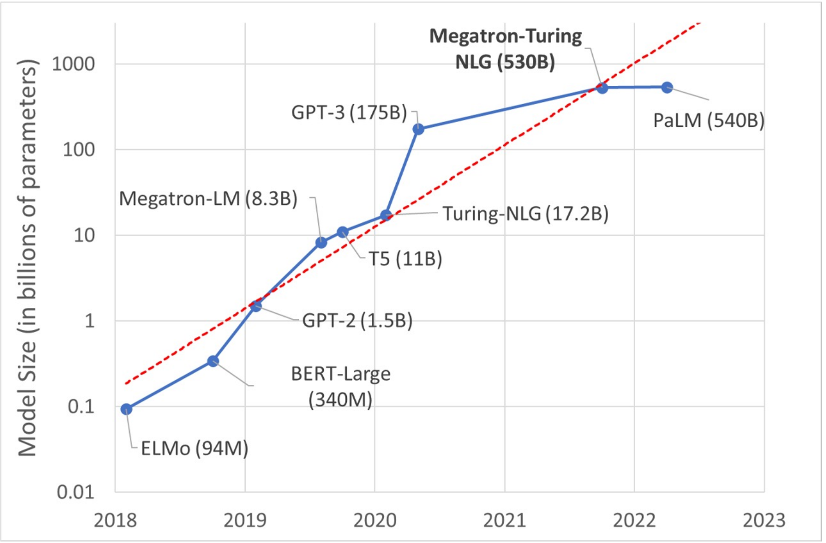
https://github.com/NVIDIA/Megatron-LM

- Today's NLP models require a few million dollars to train so we must have:

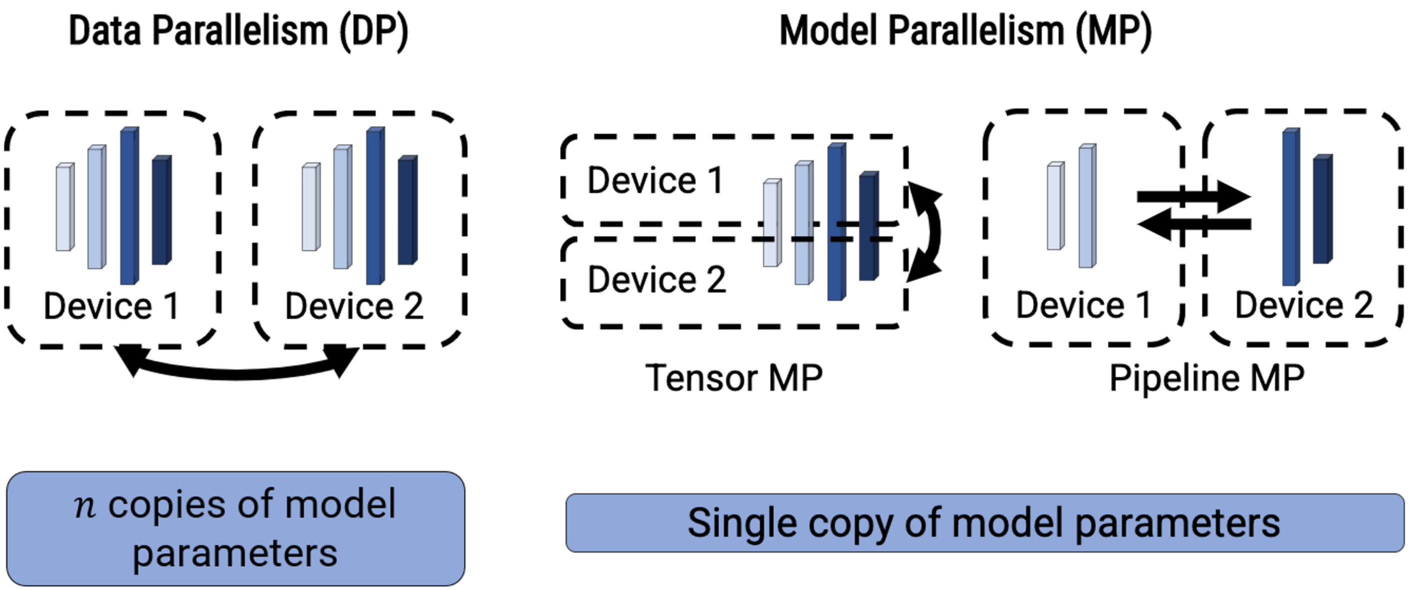
- **Efficiency:** we measure it as the percentage of theoretical peak FLOPs of a processor
  - Best ROI
  - Up to 56% MFU for Megatron-LM

- **Scalability:** Efficient scaling of both model size (weak scaling) and number of GPUs (strong scaling)
  - Biggest model & dataset

- **Simplicity:** Simple yet efficient algorithms mostly in Python, with no fancy compiler
  - Model innovation & agility

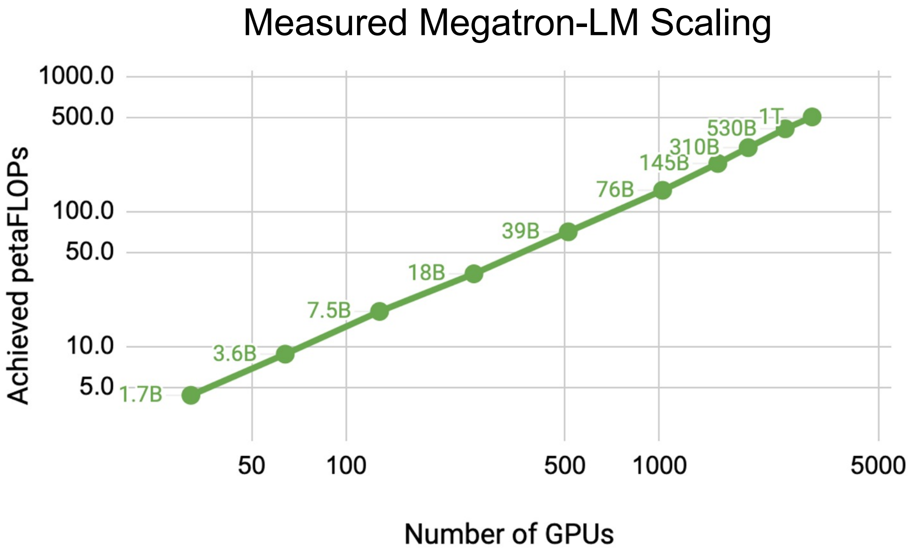


Data and Model Parallelism



Efficiency and Scalability

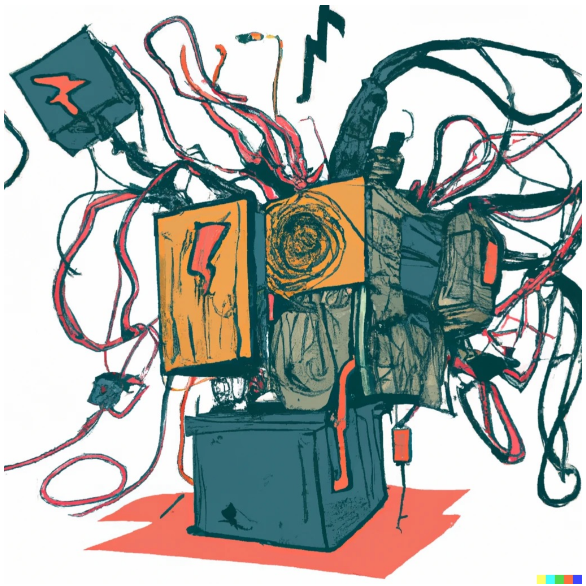
- Achieve scalability using data and model parallelism
  - Model parallelism:
    - Tensor parallelism
    - Sequence parallelism
    - Pipeline parallelism
- Challenge: how to achieve efficiency at scale



Almost linear scaling for models from 1B to 1T parameters (3 orders of magnitude) across 32 to 3K GPUs (2 orders of magnitude)

Simplicity

- The Megatron-LM project is built in PyTorch
- I love compilers! I think the world needs awesome compilers for AI
- But we have an urgent mission:
  - Accelerate Transformers
- Automatic parallel compilers for AI are hard
- We are doing this all by hand
- This shows us Speed-of-light
- Space is moving quickly
  - New ideas all the time



Model Parallel MLP

- MLP:

$$Y = \mathrm{GeLU}(XA)$$
$$Z = \mathrm{Dropout}(YB)$$

- Approach 1: split X column-wise and A row-wise:

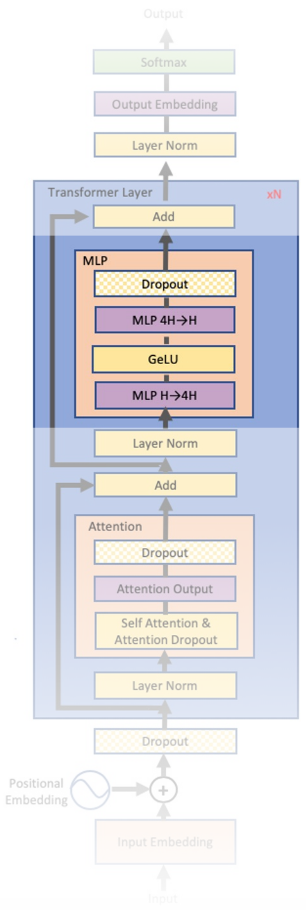
$$X = [X_1, X_2] \qquad A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \Longrightarrow Y = \mathrm{GeLU}(X_1 A_1 + X_2 A_2)$$

  - Before GeLU, we will need a synchronization point

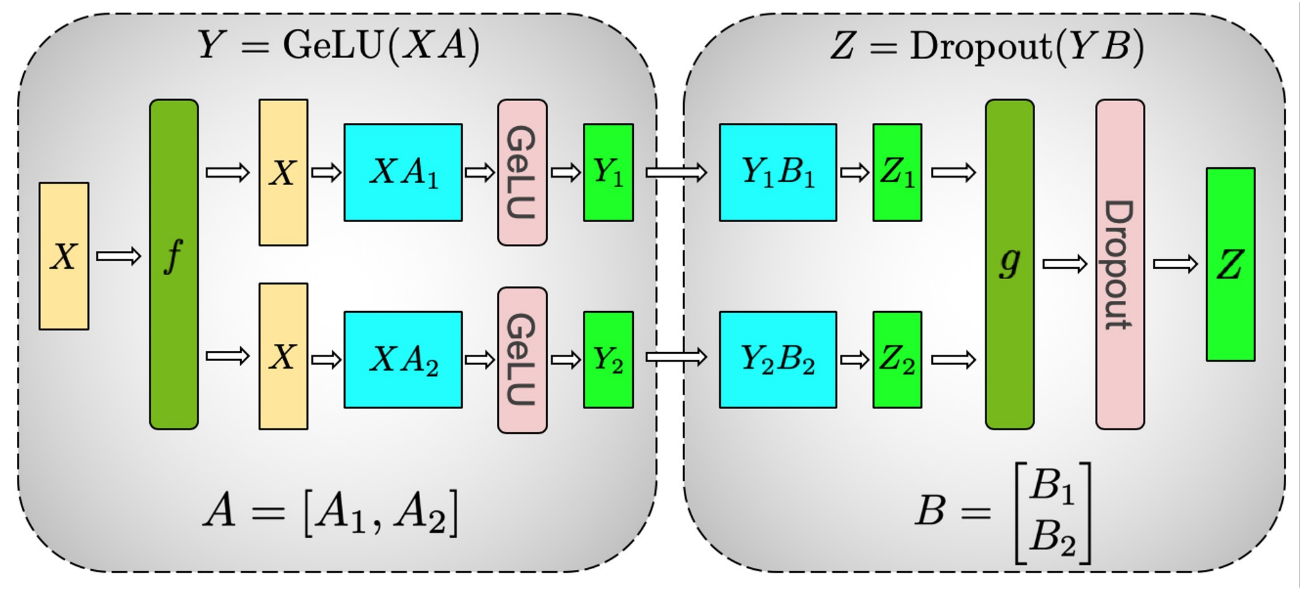
- Approach 2: split A column-wise:

$$A = [A_1, A_2] \Longrightarrow [Y_1, Y_2] = [\mathrm{GeLU}(XA_1), \mathrm{GeLU}(XA_2)]$$

  - no synchronization is required



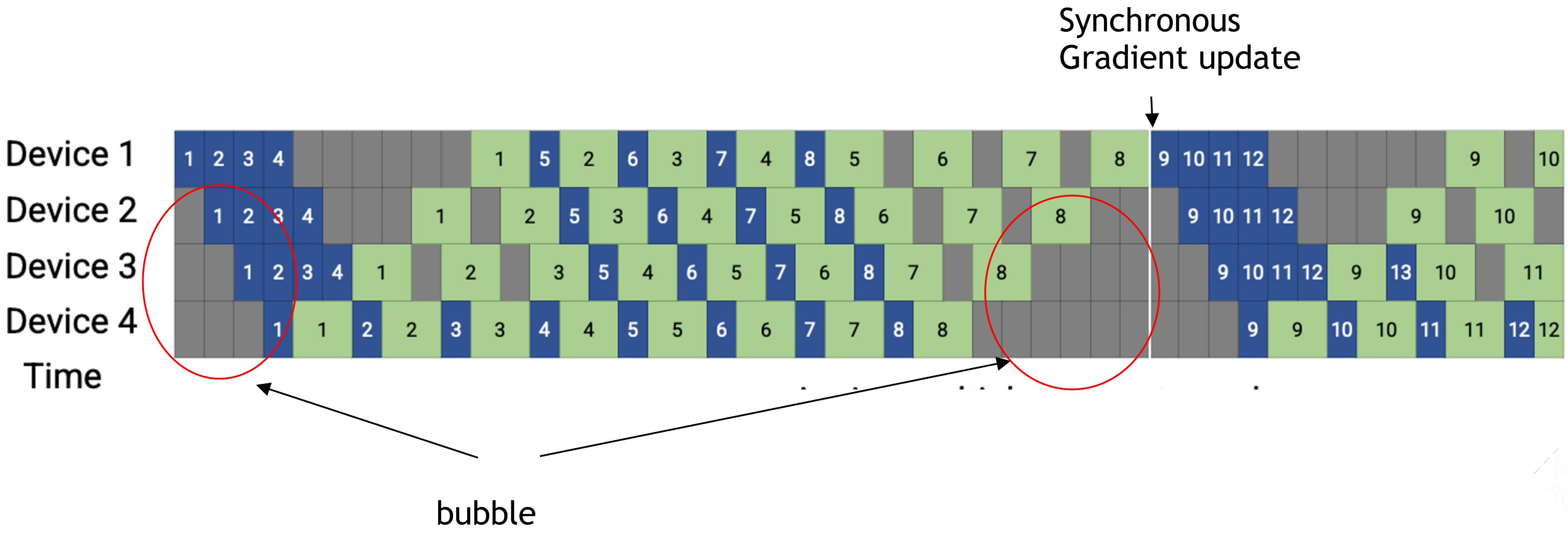
A column-wise, B row-wise: ½ the communication



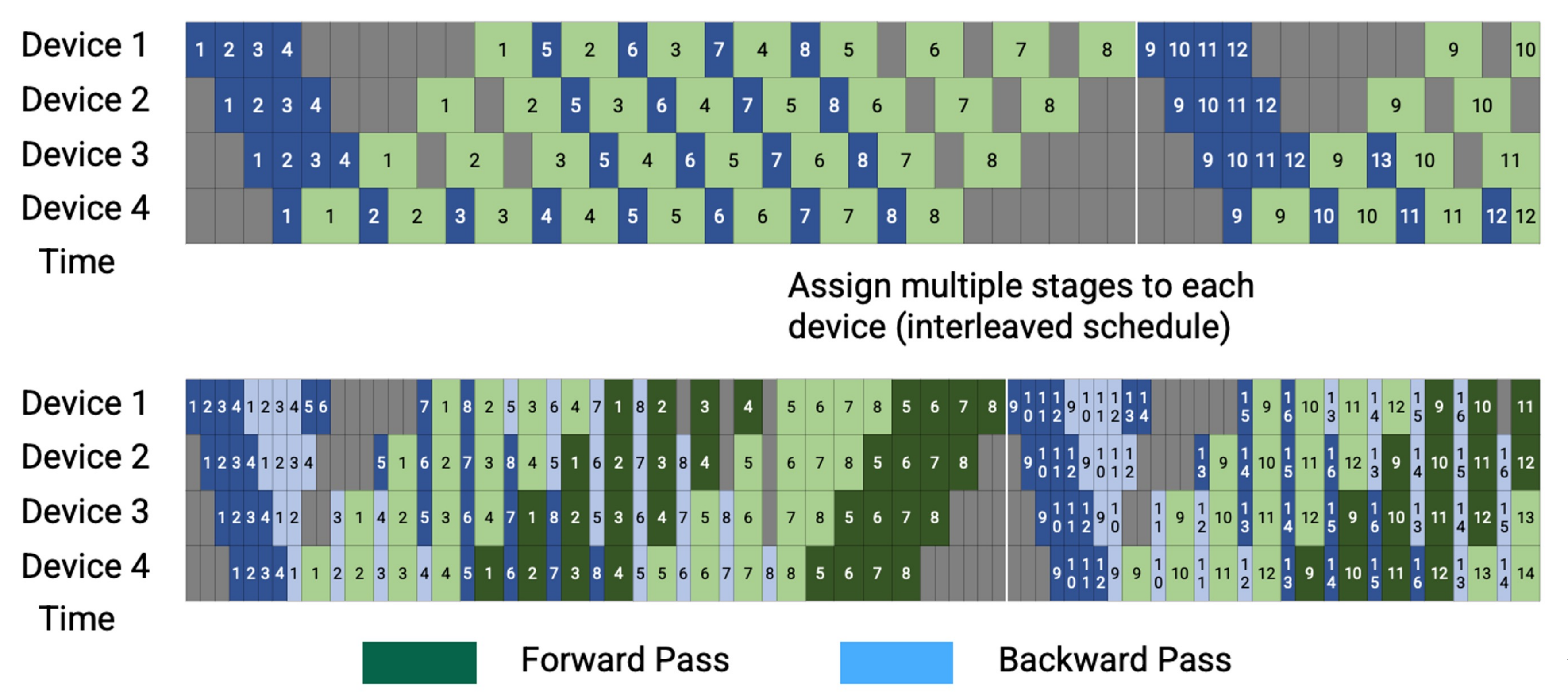
f and ***g*** are **conjugate**, ***f*** is **identity** operator in the forward pass and **all-reduce** in the backward pass while ***g*** is **all-reduce** in forward and **identity** in backward.

Pipeline Parallelism

- Divides a batch size into micro-batches to keep the pipeline pressurized
- However, due to synchronous gradient updates, we have idle times (bubble) at the beginning and end of each iteration

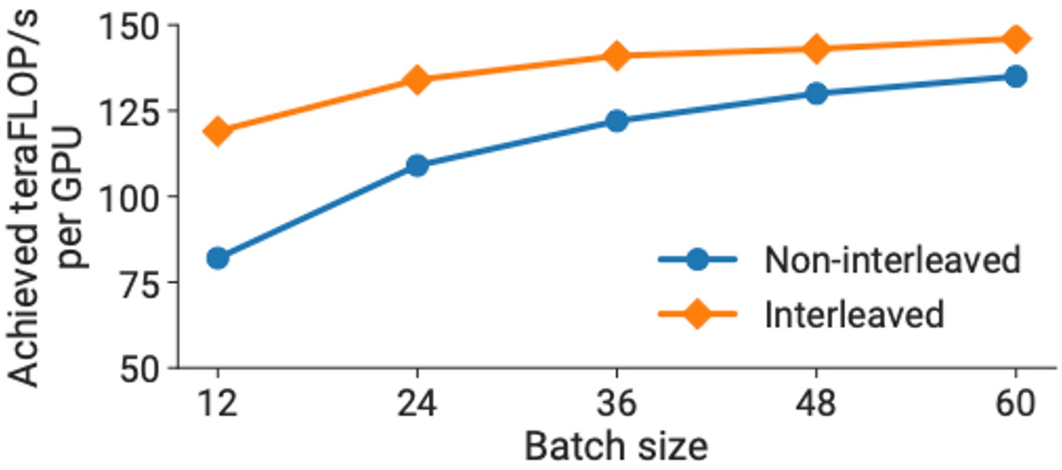


Interleaving Pipeline Schedule



Interleaving Schedule Results

- Interleaving more effective at small batch sizes
- Good for strong scaling

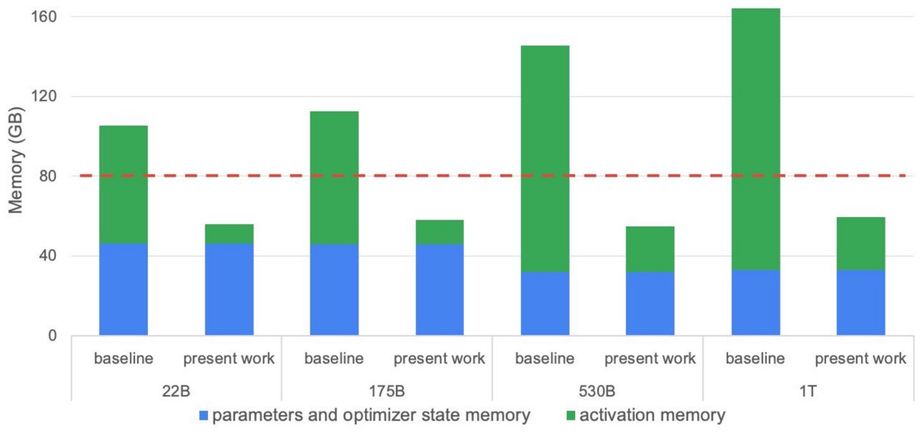


175B GPT-3 model on 96 GPUs
(no data parallelism)

Sequence Parallelism

- Activations require a substantial amount of memory for large models.
- Tensor parallelism can only reduce parts of activations memory (dropout and layernorms are duplicated)
- Standard full activation recomputation introduces **30-40% computational overhead**

Red line shows A100/H100 memory

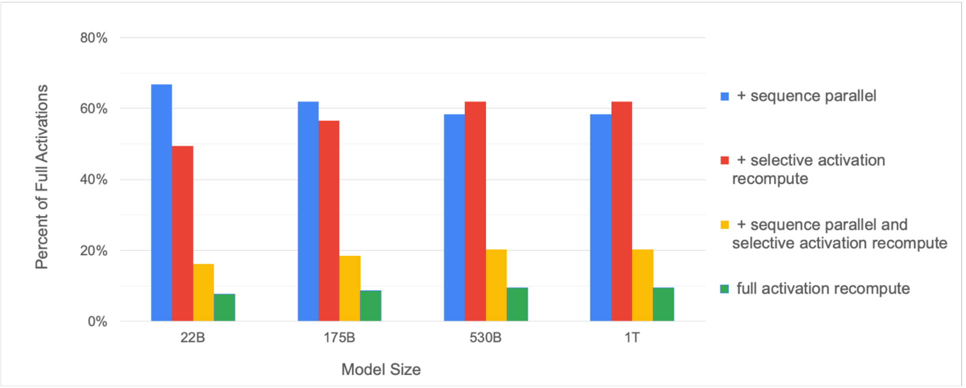


Required memory for
tensor + pipeline parallelism

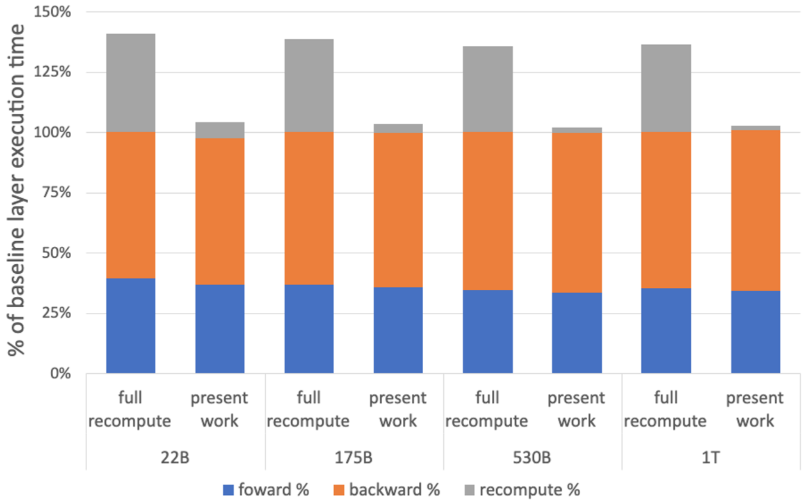
Solution

- Sequence parallelism + Selective activation recomputation

56.3% MFU for 1T parameter model on 512 A100 GPUs

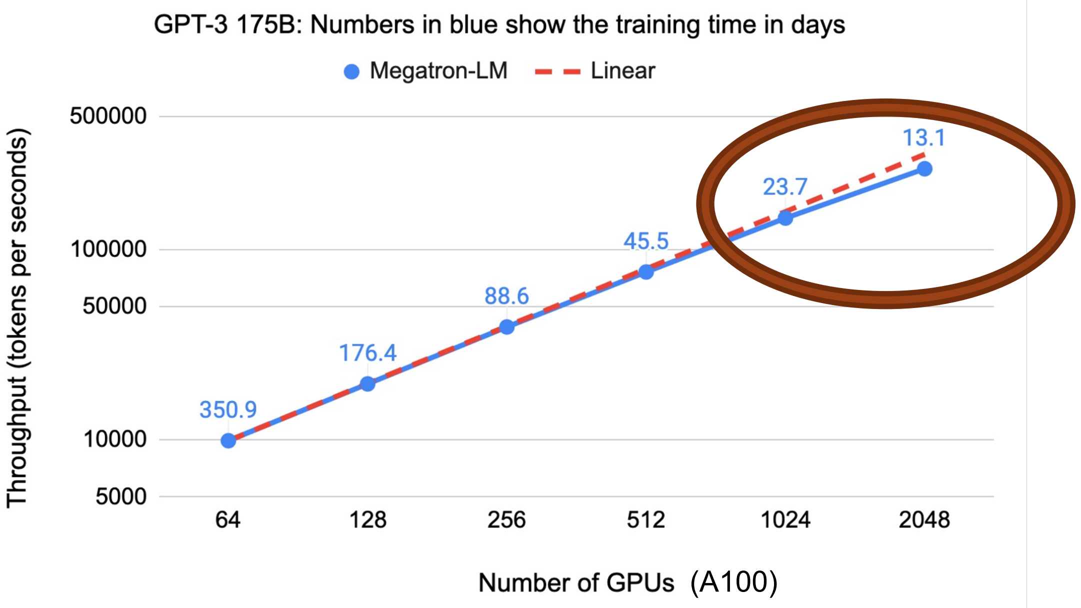


Percentage of required activation memory compared to the tensor+pipeline parallel baseline.



Per-layer breakdown; baseline is the case with no activation recomputation or sequence parallelism

End-to-end Results: Measured Strong Scaling



More work to do here
And Beyond

32x increase in number of GPUs for fixed model size and batch size

Conclusion

- Language models are the biggest compute challenge of our time
- Megatron-LM is a research project for big transformers
- Megatron technologies productized as part of NVIDIA NeMo
- Current work focuses on multimodality and more complex training setups

- A golden age for AI systems: so much more than chips