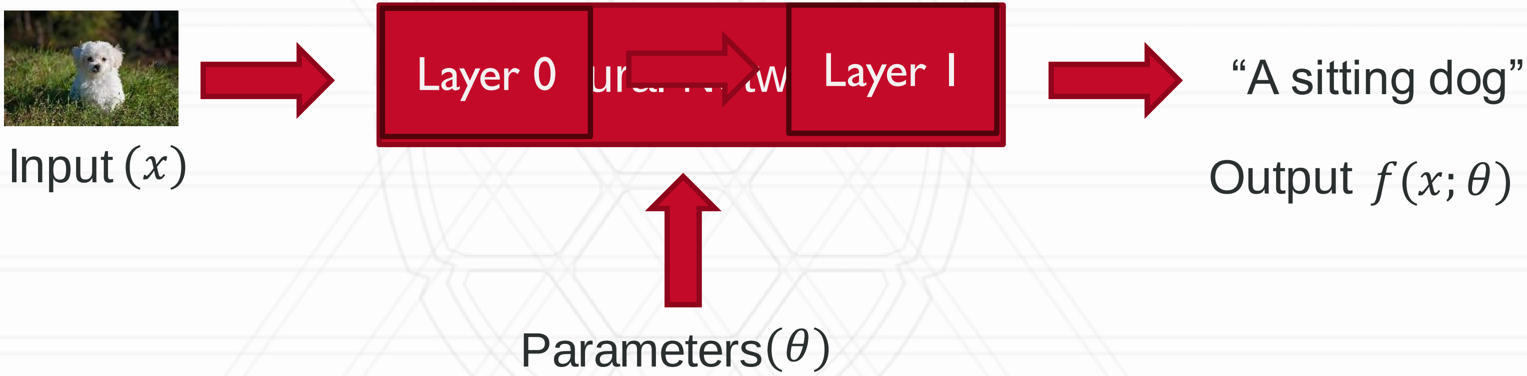
A Hybrid Tensor-Expert-Data Parallelism Approach to Optimize Mixture-of-Experts Training

Siddharth Singh[1], Olatunji Ruwase[2], Ammar Ahmad Awan[2], Samyam Rajbhandari[2], Yuxiong He[2], Abhinav Bhatele[1]
University of Maryland[1], Microsoft, Inc.[2]

UNIVERSITY OF MARYLAND

Neural Networks

- Neural Networks (NNs): ‘Parameterized’ function approximators
- Can work with very high dimensional data.



Organized into Layers

Stochastic Gradient Descent

- Repeat until loss (L) has been minimized sufficiently:
  - Read in a batch of training data
  - Forward Pass : Calculate output $f(x;\theta)$ and the loss (L) on the batch.
  - Backward Pass : Calculate gradients of the loss wrt the parameters ($\frac{\partial L}{\partial \theta}$).
  - Optimizer Step : Use $\frac{\partial L}{\partial \theta}$ to update θ.

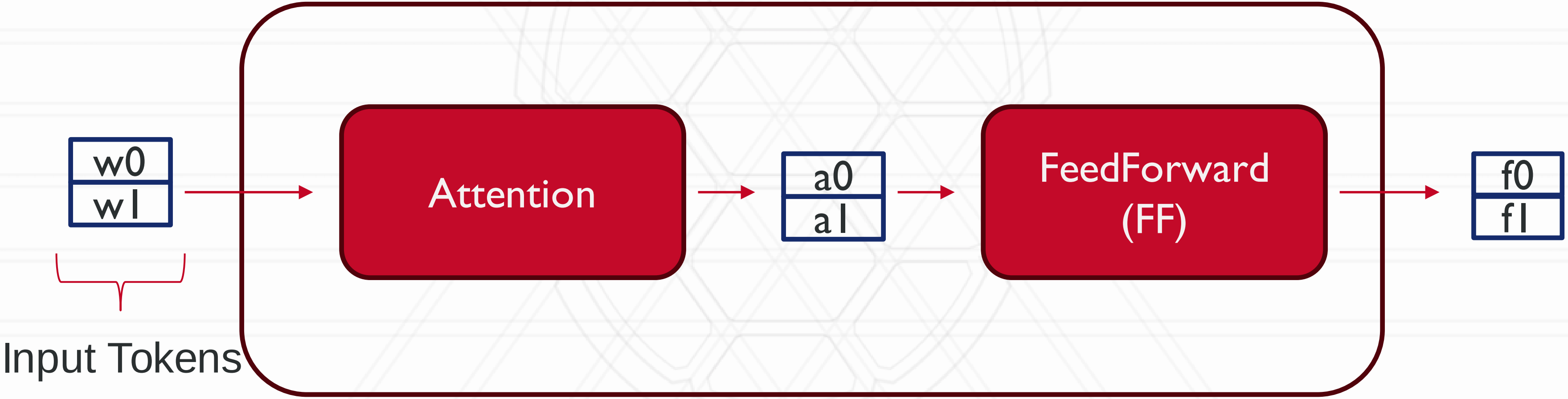
Motivation

- More parameters = More accurate neural networks ☺
- However, more parameters = More FLOPs in training ☹

MoEs can make a given model arbitrarily large <u>without changing its training FLOPs!</u>

Mixture-of-Experts (MoEs)

Step 1: Start with a base model – usually a transformer neural network.



A Transformer Layer (Base Model)

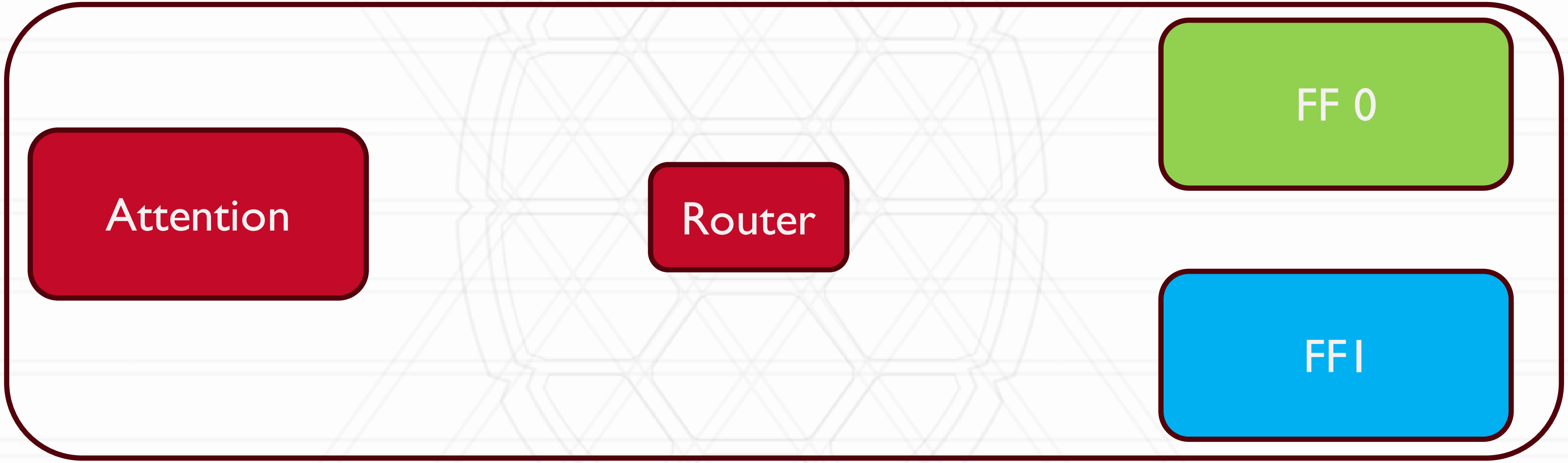
Mixture-of-Experts (MoEs)

Step 2: Introduce multiple FF blocks. Each FF block is an 'expert'.

A Transformer Layer (Base Model) + 2 experts

Mixture-of-Experts (MoEs)

Step 3: Introduce a parameterized router that maps tokens to experts.

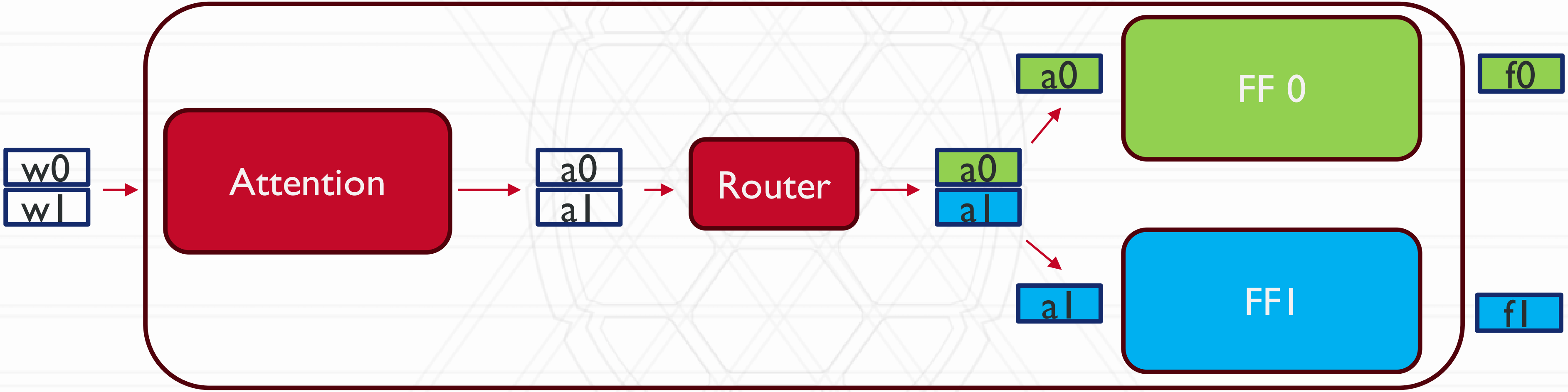


A Transformer Layer (Base Model) + 2 experts

Mixtur

Total FLOPs per token is independent of the number of experts!

Step 3: Introduce a parameterized router that maps tokens to experts.



A Transformer Layer (Base Model) + 2 experts

Caveat

- Diminishing returns beyond 64-128 experts.
- Imperative to increase base model sizes along with expert counts.

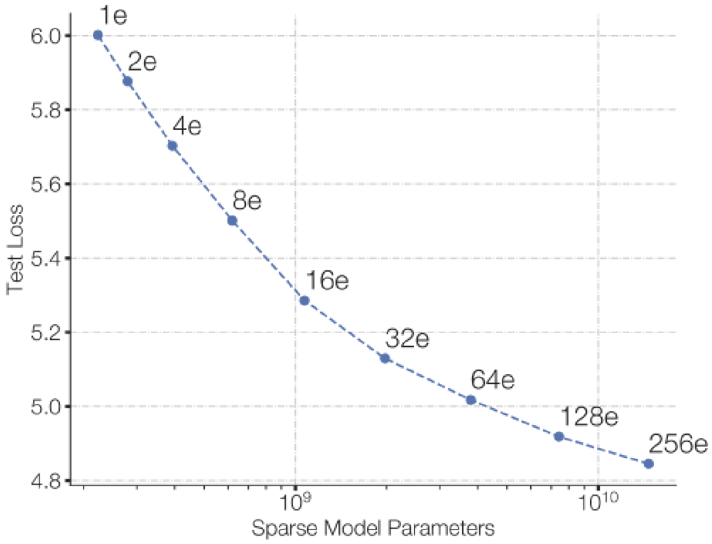


Figure courtesy Fedus et al. [2]

Gaps in Current Work

- Large MoEs are almost always trained in parallel on multiple GPUs.
- However, current parallel frameworks are not suited for MoEs built with large base models.
- They either support base models of limited sizes due to limited dimensions of parallelism.
  - Example – DeepSpeed-MoE [4] which has expert+data parallelism but not model parallelism (tensor/pipeline)
- Or, they use extremely inefficient parallel techniques like out-of-core training or FSDP.
  - Inefficiency occurs due to high communication times

Our work – Deepspeed-TED

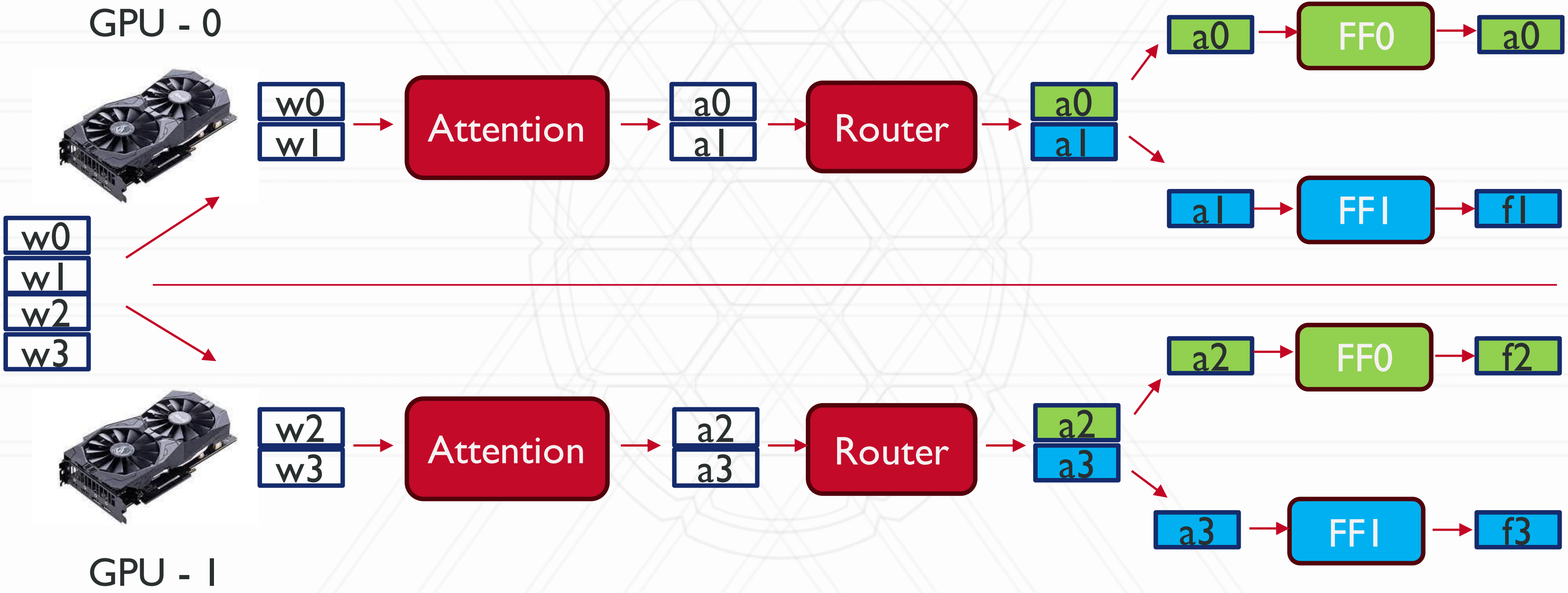
- Goal 1 - Support MoEs with large base models

- Goal 2 - Minimize communication times to maintain efficiency.

- A three-dimensional hybrid of state-of-the-art parallel training algorithms
  - T – Tensor Parallelism (Megatron-LM [3])
  - E – Expert Parallelism (DeepSpeed-MoE [4])
  - D – Sharded Data Parallelism (ZeRO [5])

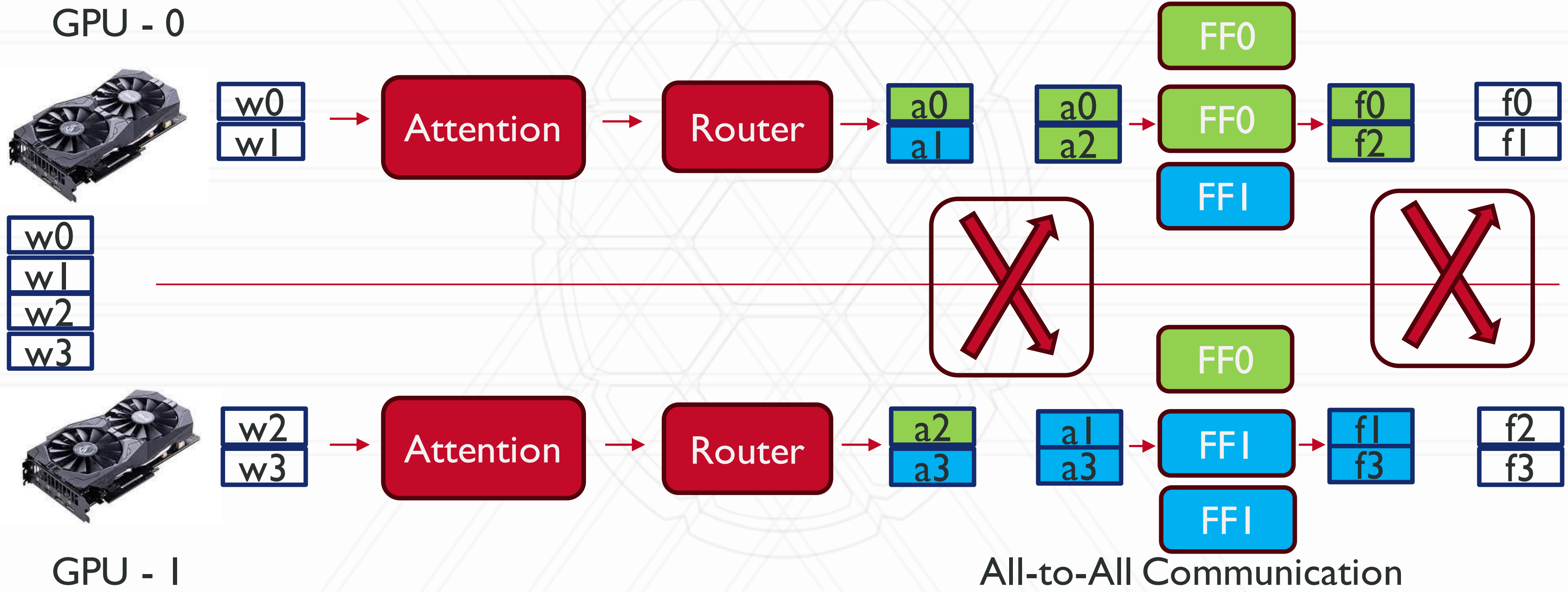
Data Parallelism

Average Gradients



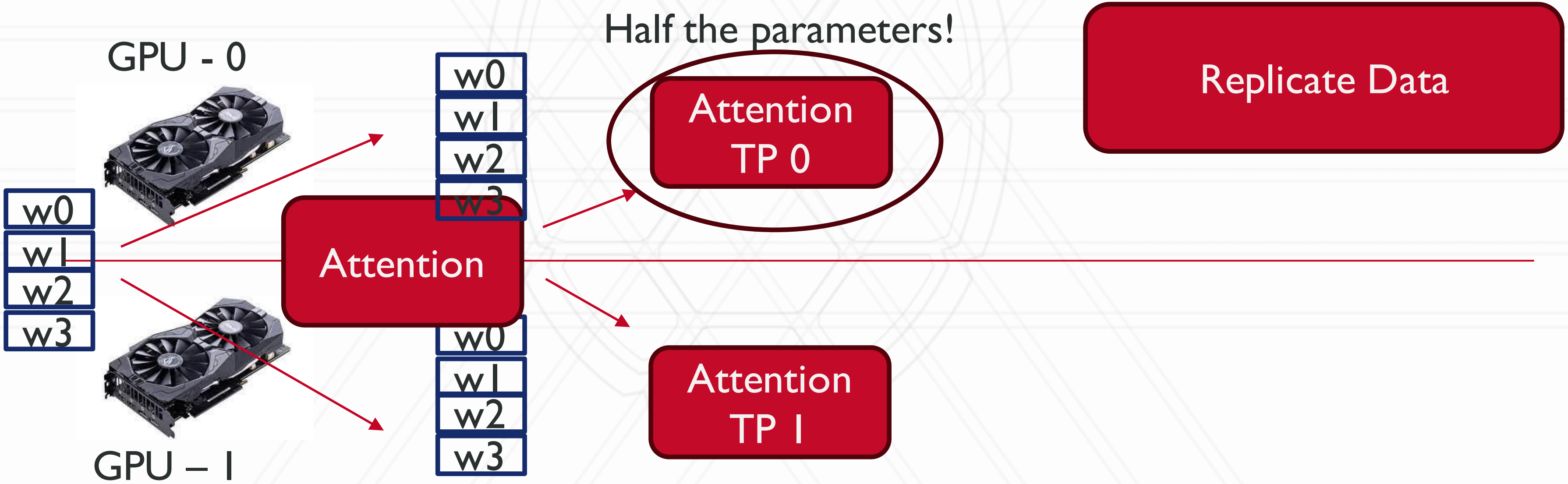
Data + Expert Parallelism

Divide Experts among GPUs



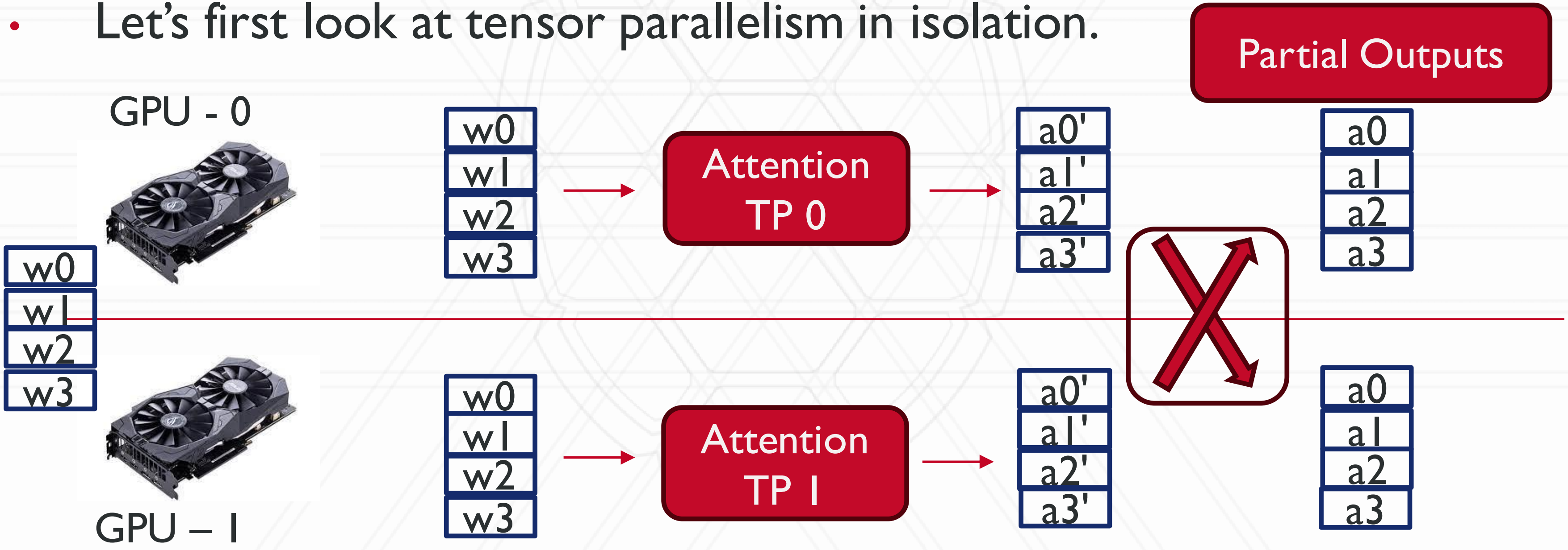
Tensor Parallelism

- Parallelize the matrix multiplications inside Attention and FF.
- Let's first look at tensor parallelism in isolation.



Tensor Parallelism

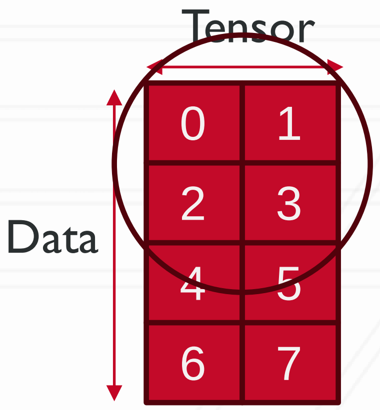
- Parallelize the matrix multiplications inside Attention and FF.
- Let's first look at tensor parallelism in isolation.



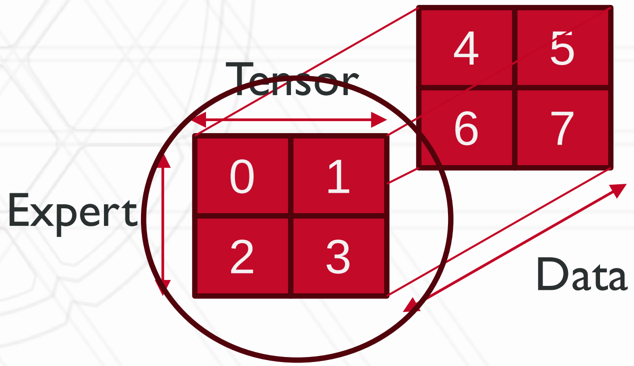
All-Reduce Communication

Data + Expert + Tensor Parallelism

- Now let us look at tensor+expert+data parallelism on 8 GPUs.
- Two virtual topologies for Attention and FF-Blocks.
- For brevity, we will only look at GPUs [0-3]

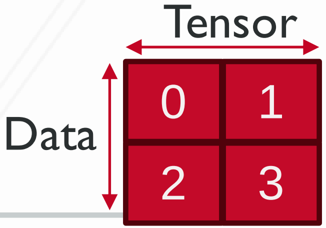


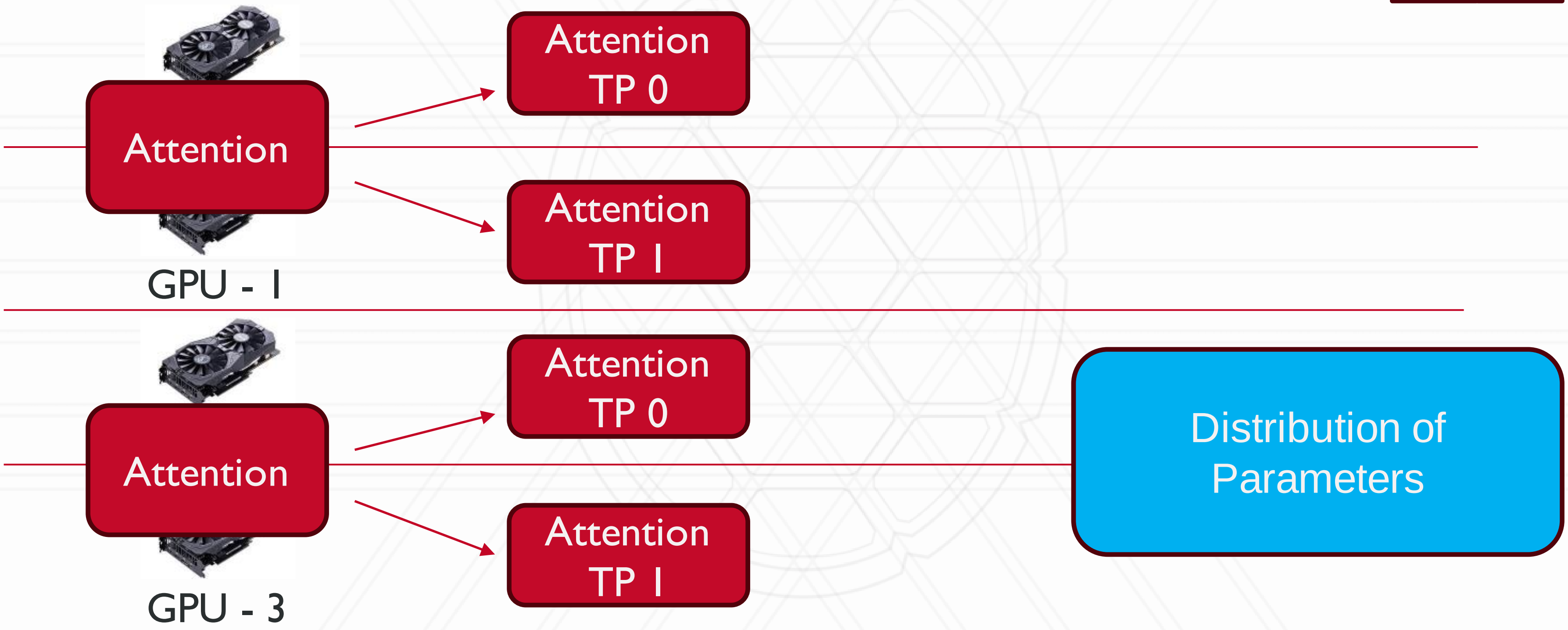
Attention Block (Non-Expert)



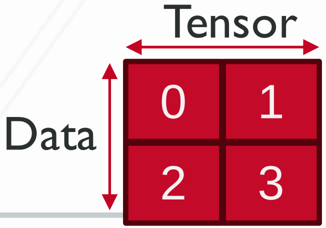
FF Blocks (Expert)

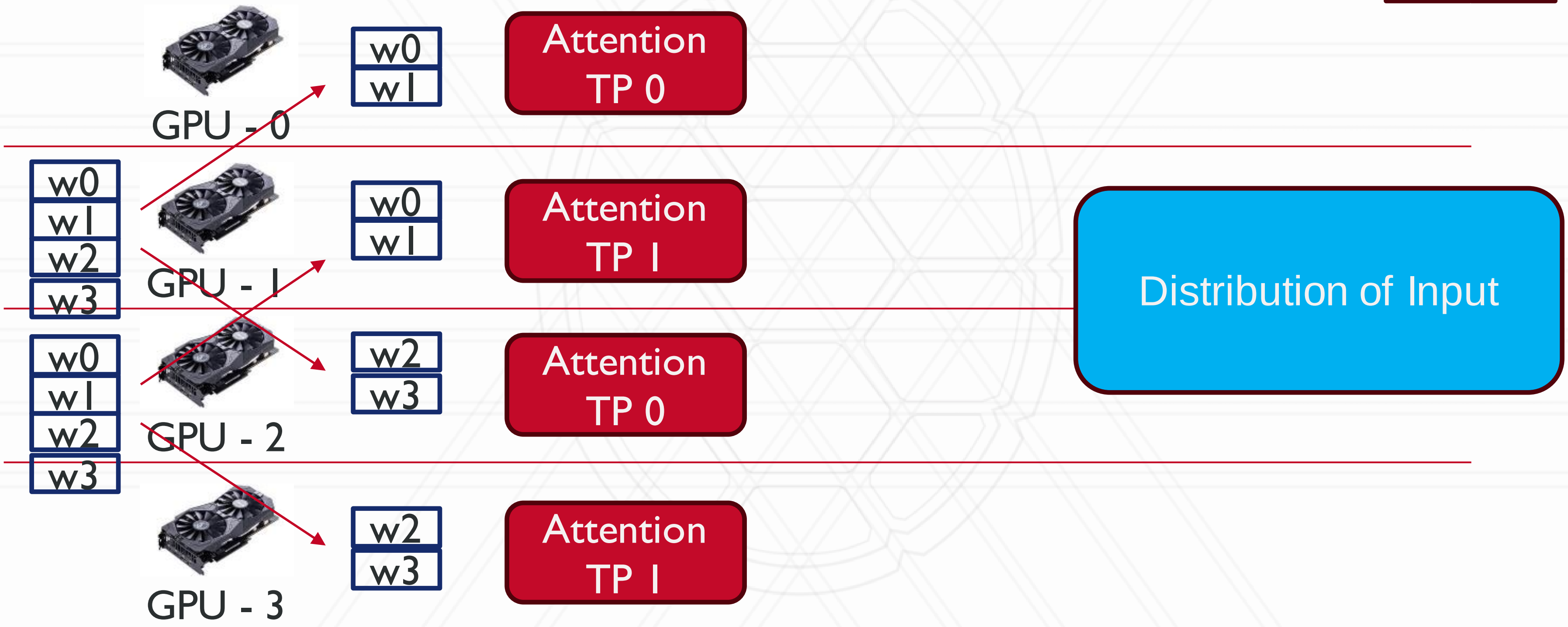
TED for Attention (Non-Experts)



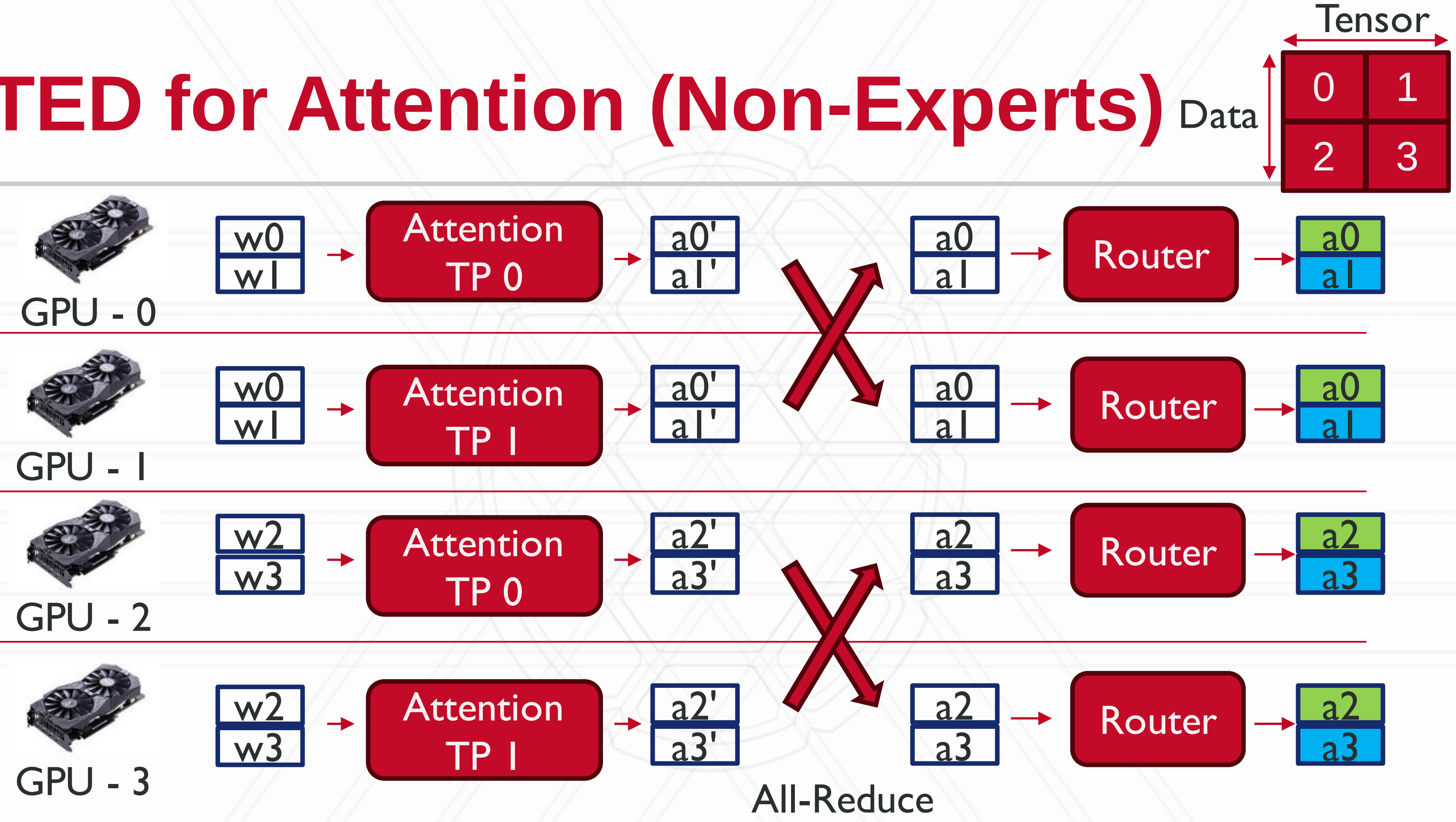


TED for Attention (Non-Experts)

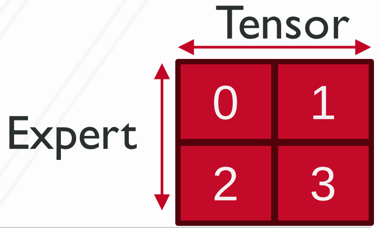


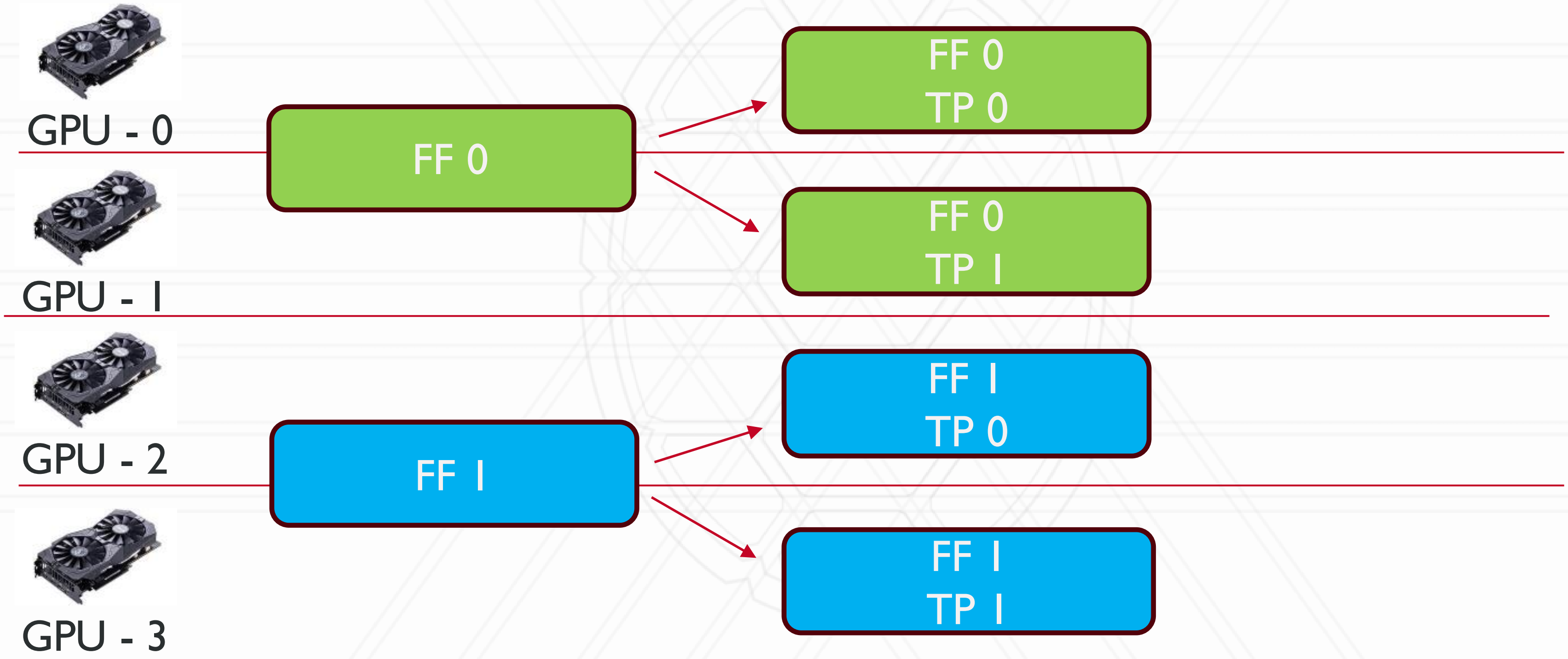


TED for Attention (Non-Experts)

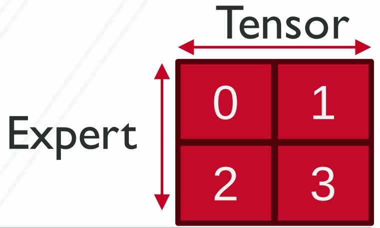


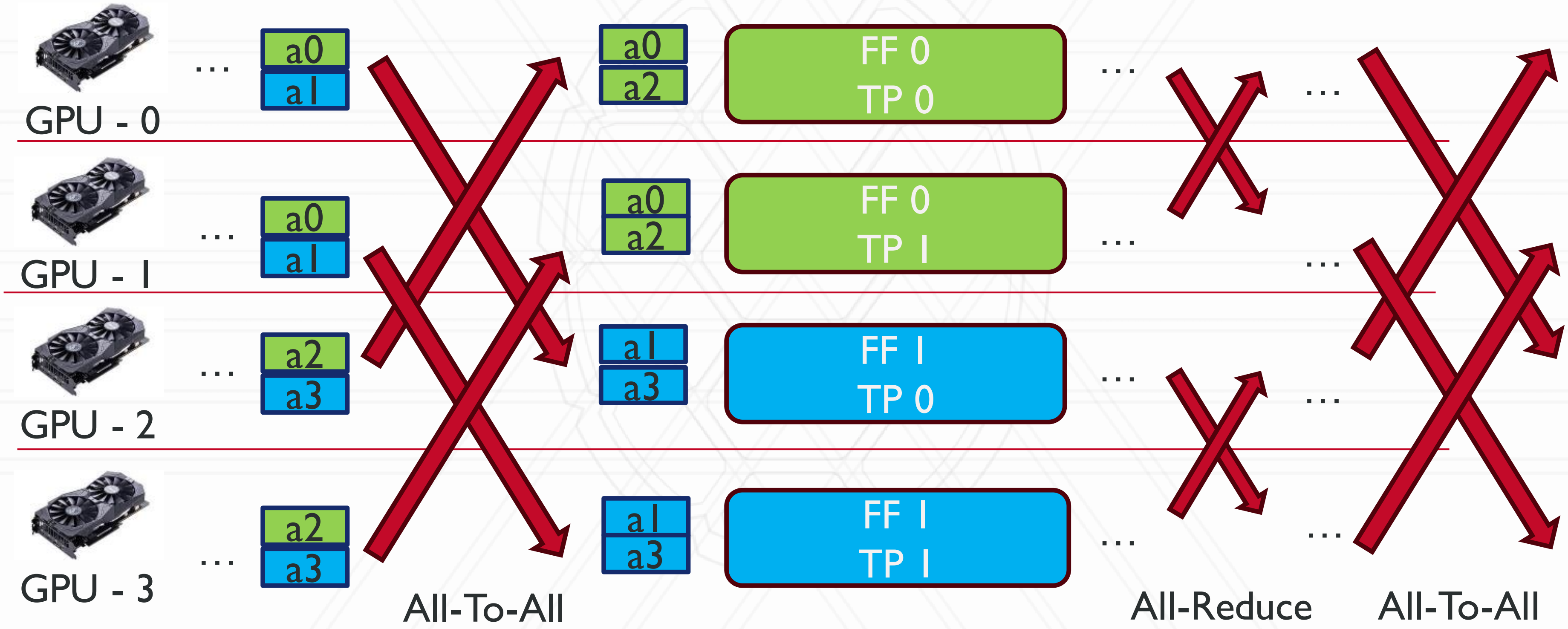
TED for FF (Experts)



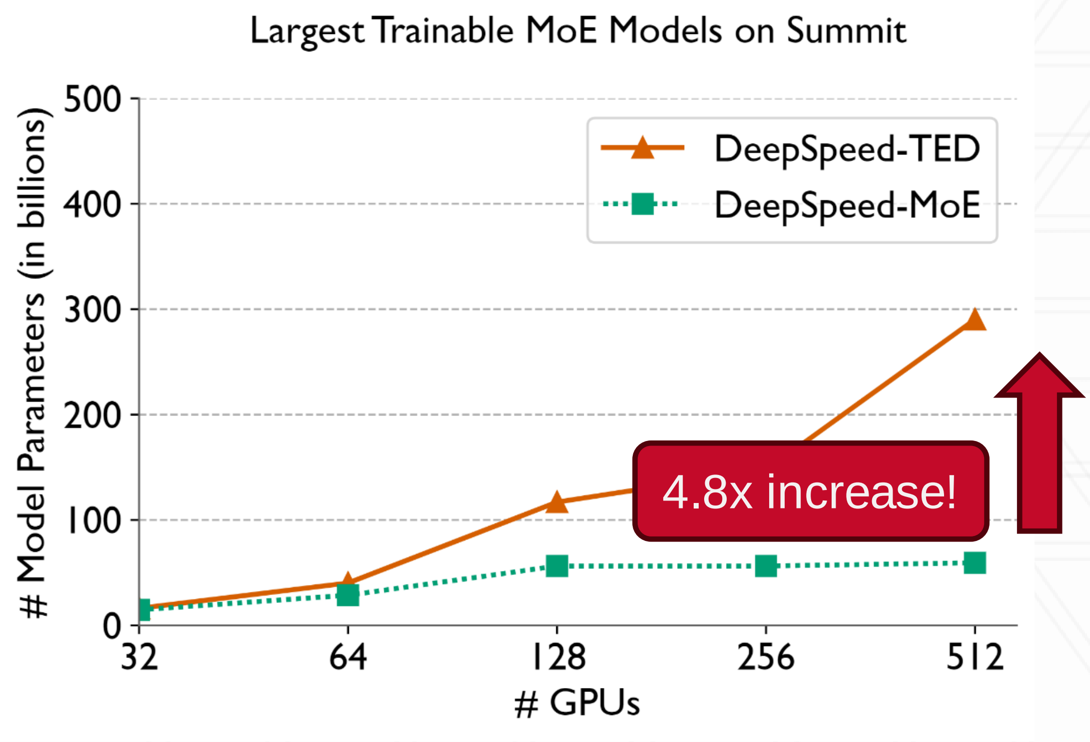


+ TED for FF (Experts)





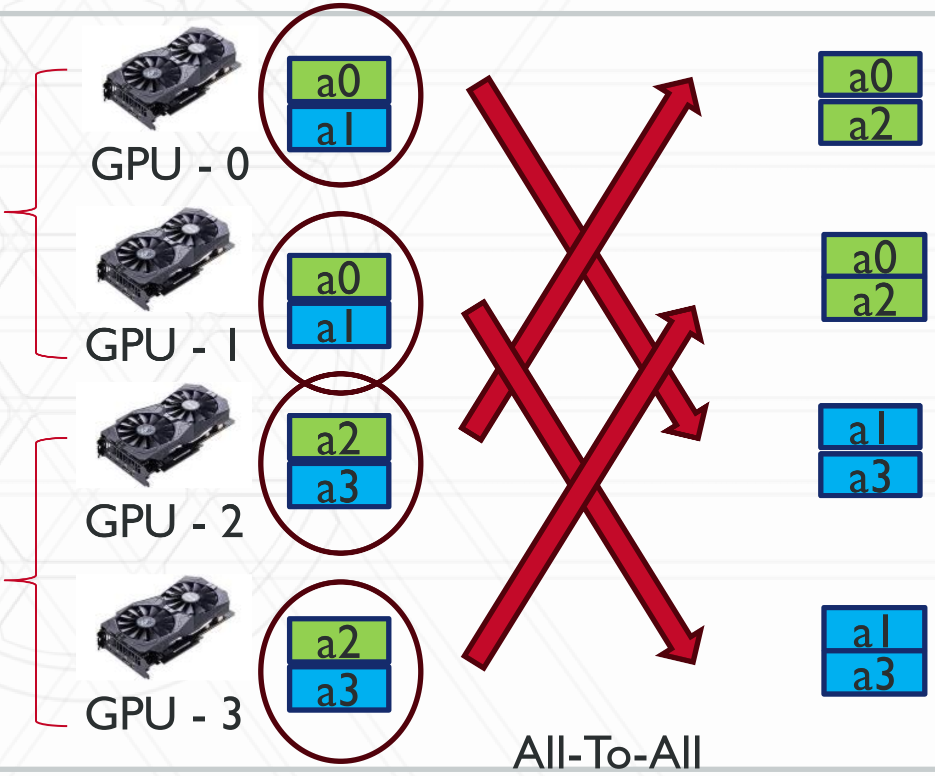
3D parallelism helps up train larger models



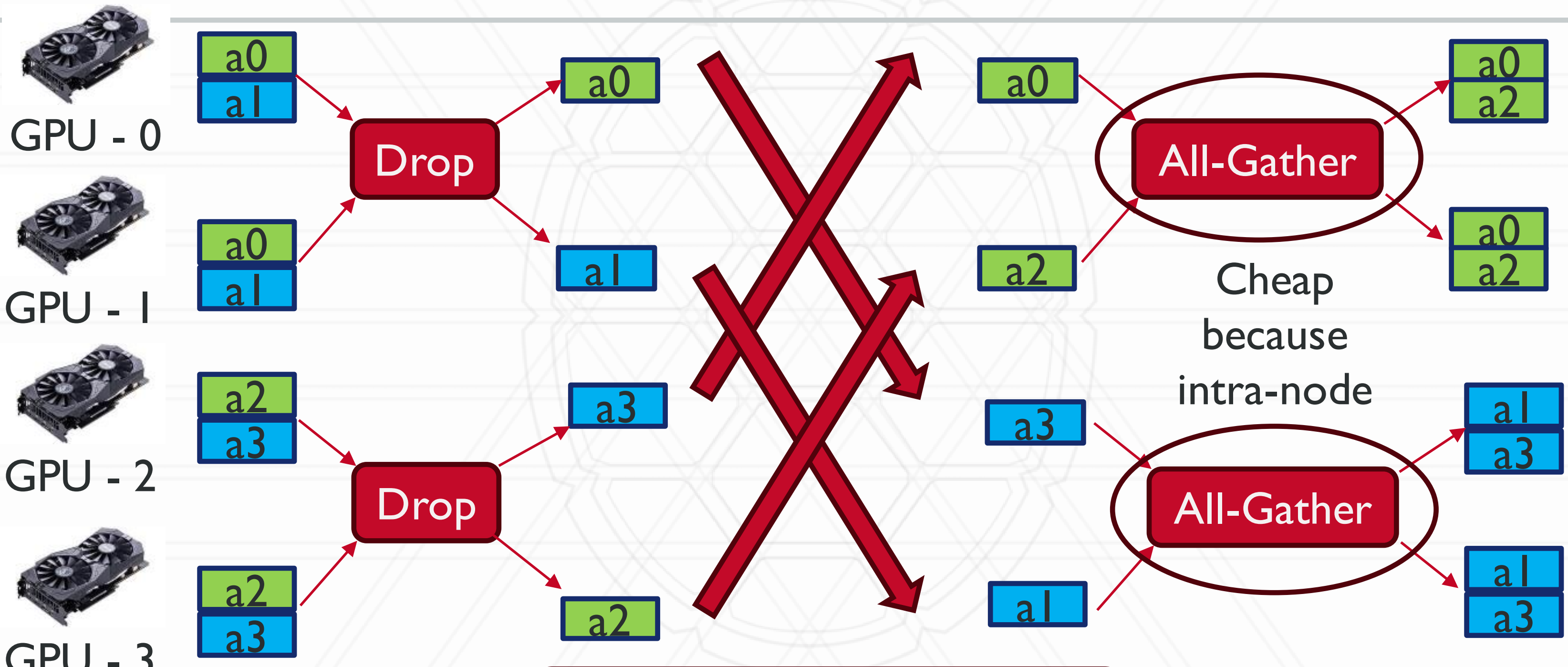
- Limit Number of experts to 128
- Limit tensor parallelism to a node.

Opt #1 Duplicate Token Dropping (DTD)

- Consider the first all-to-all.
- Tensor parallel GPUs communicate duplicate tokens.
- Remove this duplication to decrease All-To-All message sizes.



Opt #1 Duplicate Token Dropping

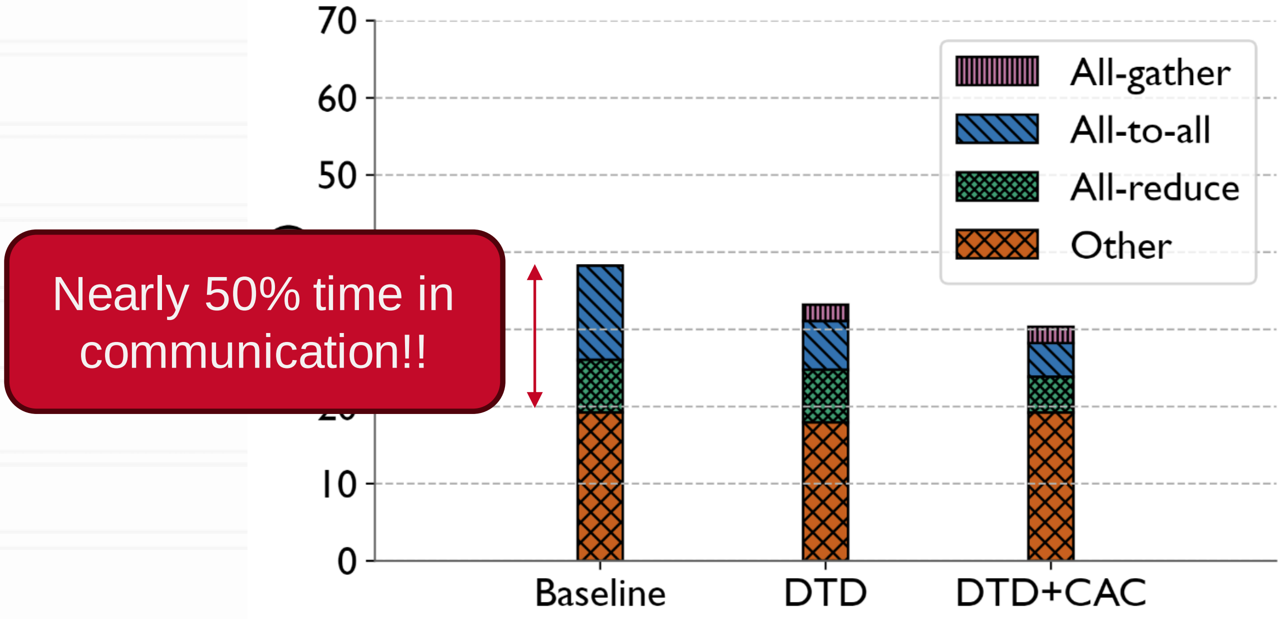


2x reduction in All-to-All volume

Opt #2 Communication-Aware Checkpointing (CAC)

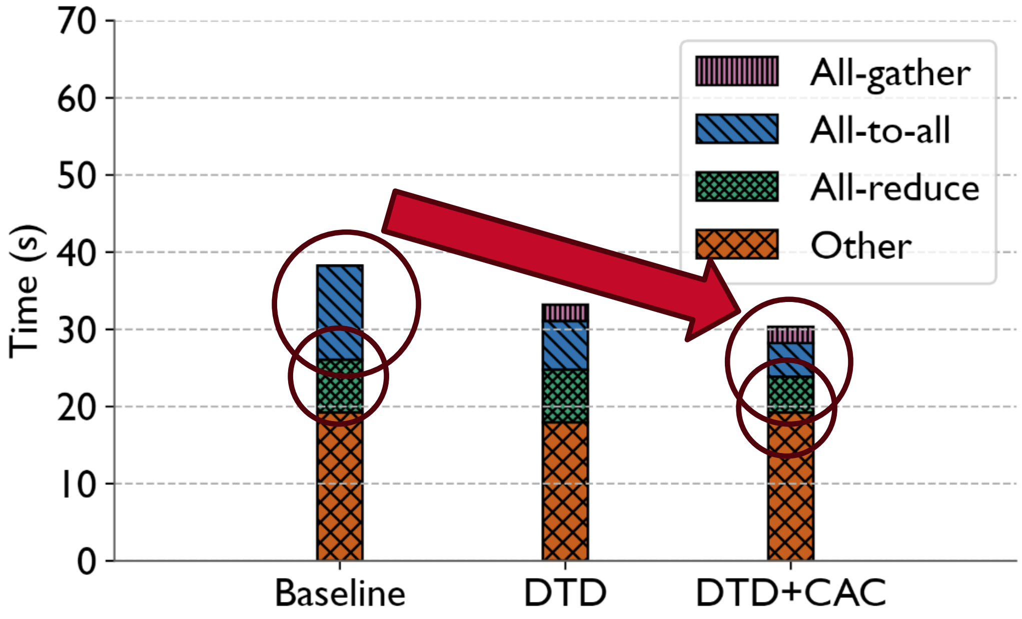
- Reduces number of all-to-all and all-reduce calls by 33 percent by utilizing marginally extra memory.
- More details in paper.

Results



Batch Time Profile of a 6.7B base model + 16 experts on 128 GPUs of Summit

Results

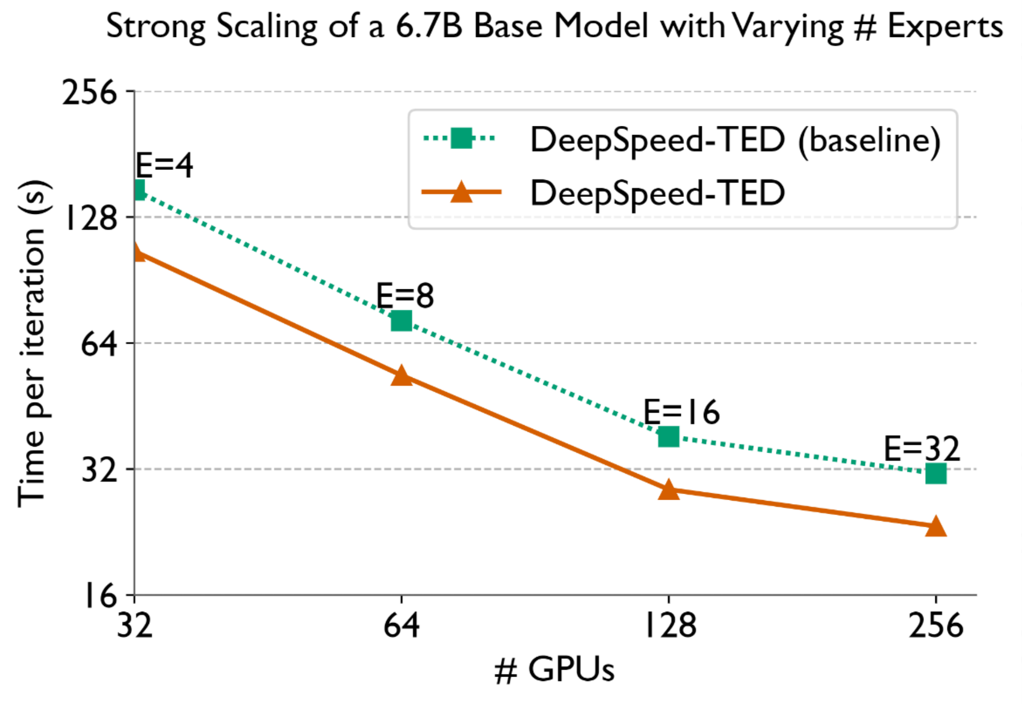


Overall 21% Speedup

Batch Time Profile of a 6.7B base model + 16 experts on 128 GPUs of Summit

Results (Strong Scaling)

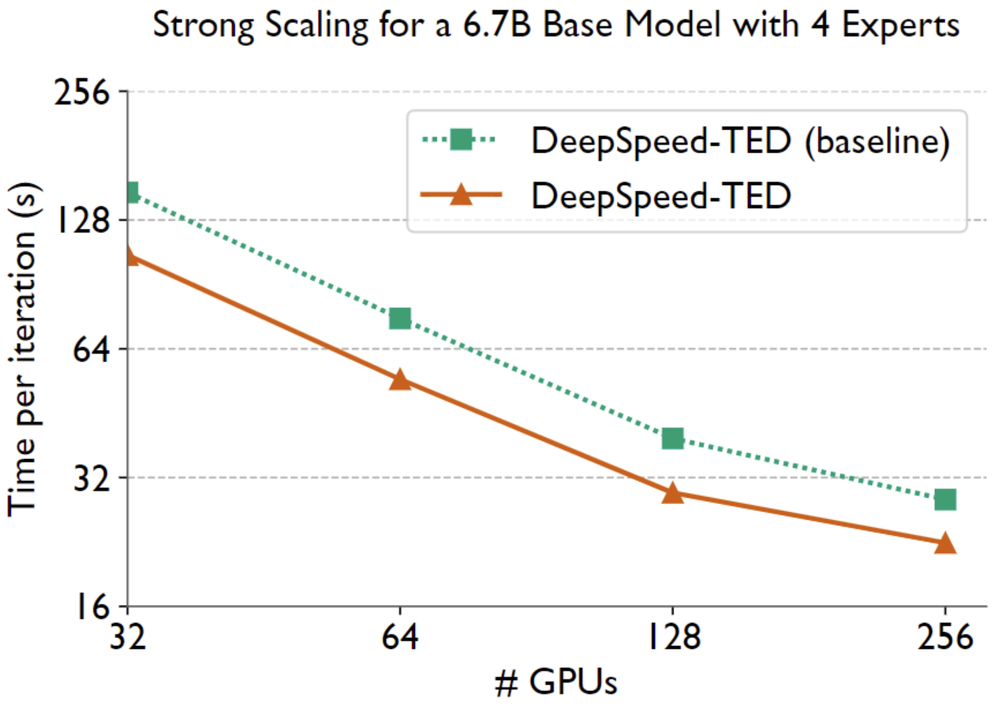
Machine - Summit



22-29% speedups!

Results (Strong Scaling)

Machine - Summit



Conclusion and Future Work

- Developed DeepSpeed-TED, a highly scalable parallel framework for training high quality MoEs with large base models.
- Presented a three-dimensional hybrid parallel method that supports MoEs with 4-8x larger models than the SoTA.
- Introduced communication optimizations that can achieve significant reductions in the collective communication times.
- As future work, we want to explore pipeline parallelism as a fourth dimension to scale to even larger base models.

Code

- Our work is integrated in DeepSpeed, a widely used open-source framework for parallel deep learning.
  - URL - https://github.com/microsoft/DeepSpeed

Bibliography

[1] Using DeepSpeed and Megatron-LM to Train Megatron-LM Turing NLG 530B, A Large-Scale Generative Language Model, Smith et al., https://arxiv.org/abs/2201.11990

[2] Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity, Fedus et al., https://arxiv.org/abs/2101.03961

[3] Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism, Shoeybi et al., https://arxiv.org/abs/1909.08053

[4] DeepSpeed-MoE: Advancing Mixture-of-Experts Inference and Training to Power Next-Generation AI Scale, Rajbhandari et al., https://arxiv.org/abs/2201.05596

[5] ZeRO: Memory Optimizations Toward Training Trillion Parameter Models, Rajbhandari et al., https://arxiv.org/abs/1910.02054

UNIVERSITY OF MARYLAND

Siddharth Singh

ssingh37@umd.edu