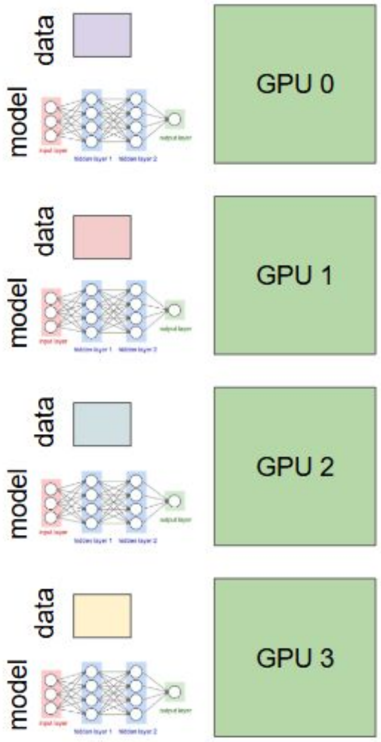
Zero Redundancy Optimizer

Team:
Alissa Amch (aa2739); Wentao Ye(wy335)

Instructor:
Mohamed Abdelfattah

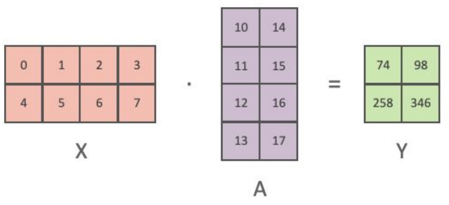
Overview of Distributed Training

Data Parallelism

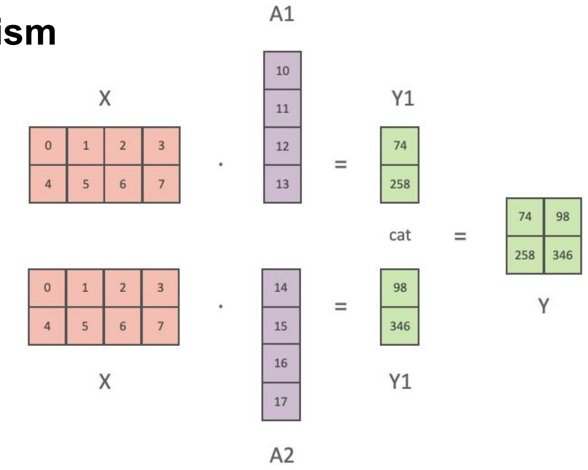


Pipeline Parallelism

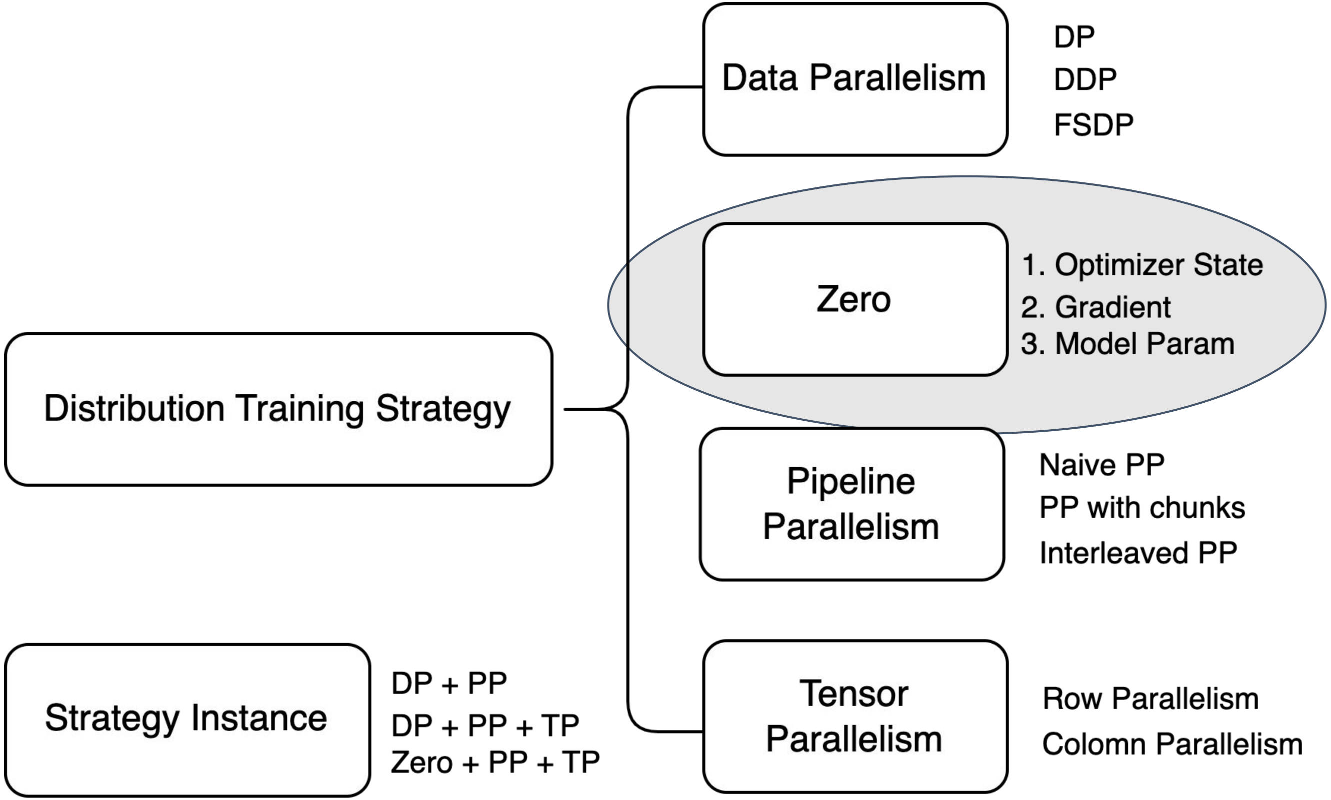
Tensor Parallelism



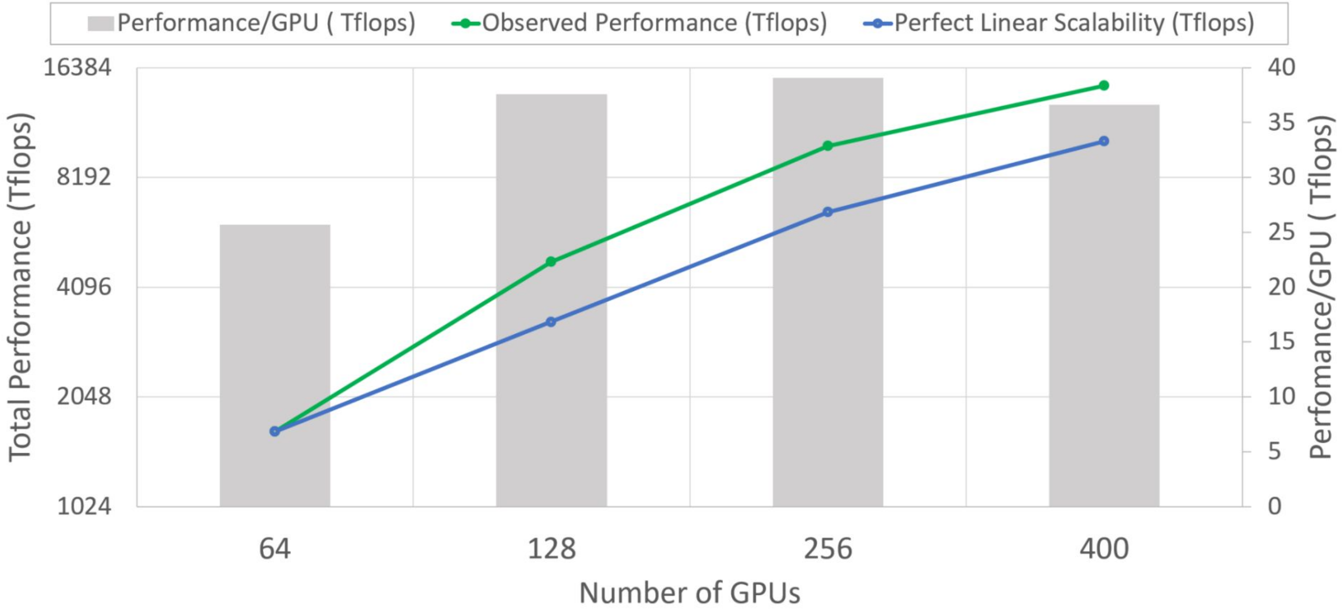
is equal to



Overview of Distributed Training



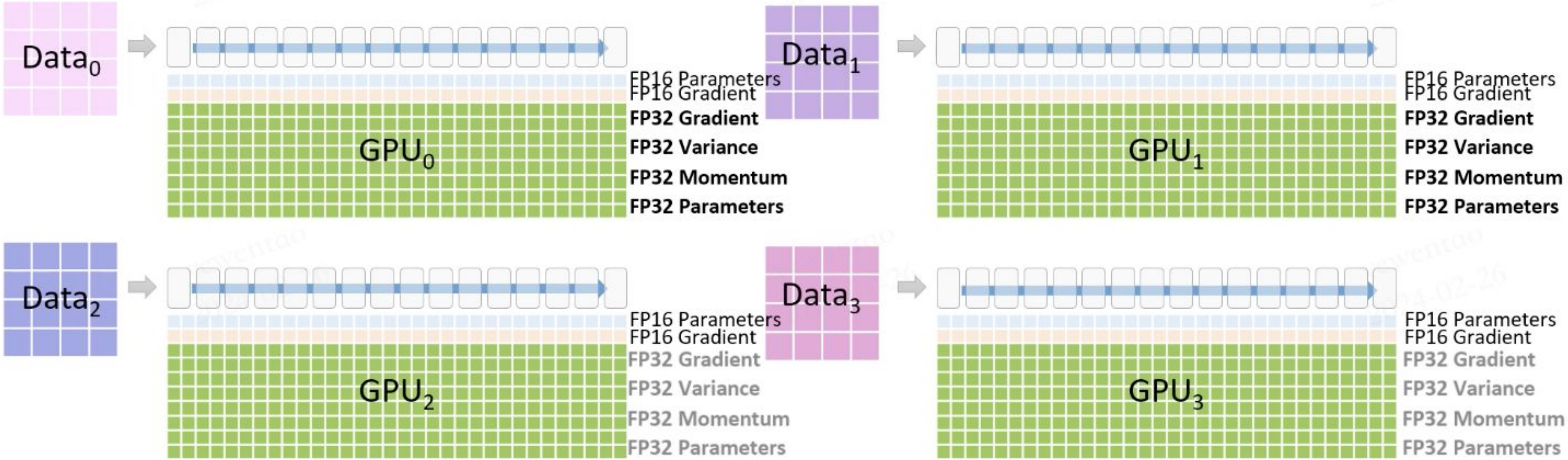
ZeRO Performance



1. Super linear scalability
2. At high complexity (400 GPUs),
   performance per GPU doesn't decrease much

source: https://arxiv.org/pdf/1910.02054

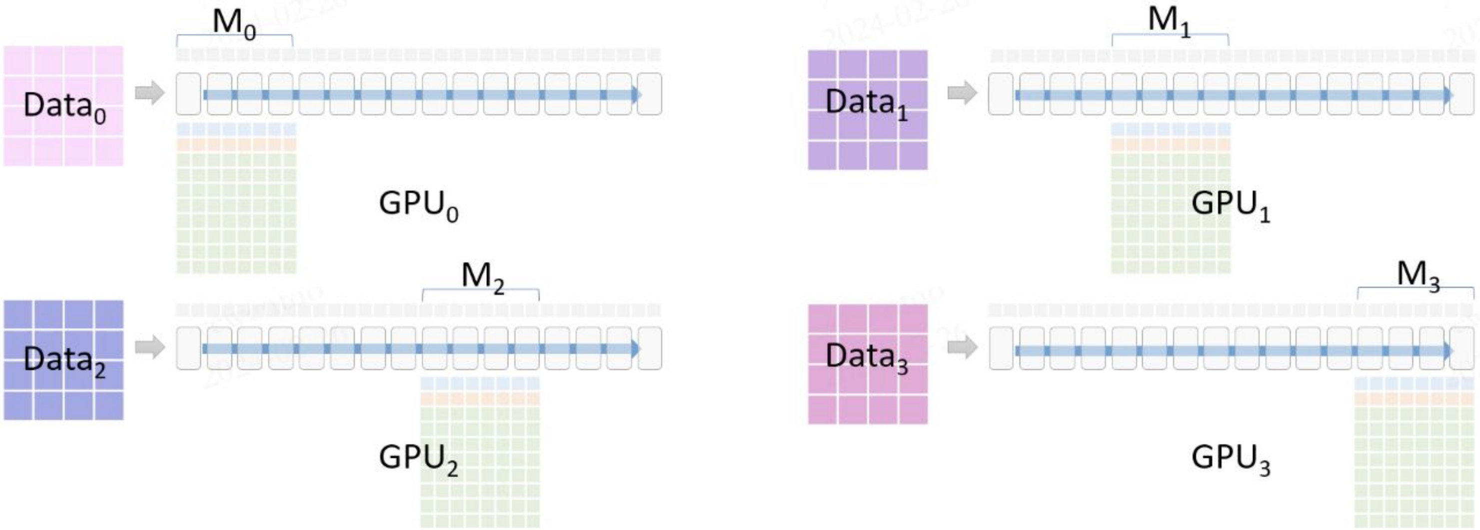
ZeRO Example Forward



Example with 4 GPUs
Model size is represented by the colored squares

source: https://www.microsoft.com/en-us/research/blog/zero-deepspeed-new-system-optimizations-enable-training-models-with-over-100-billion-parameters/

ZeRO Example Forward

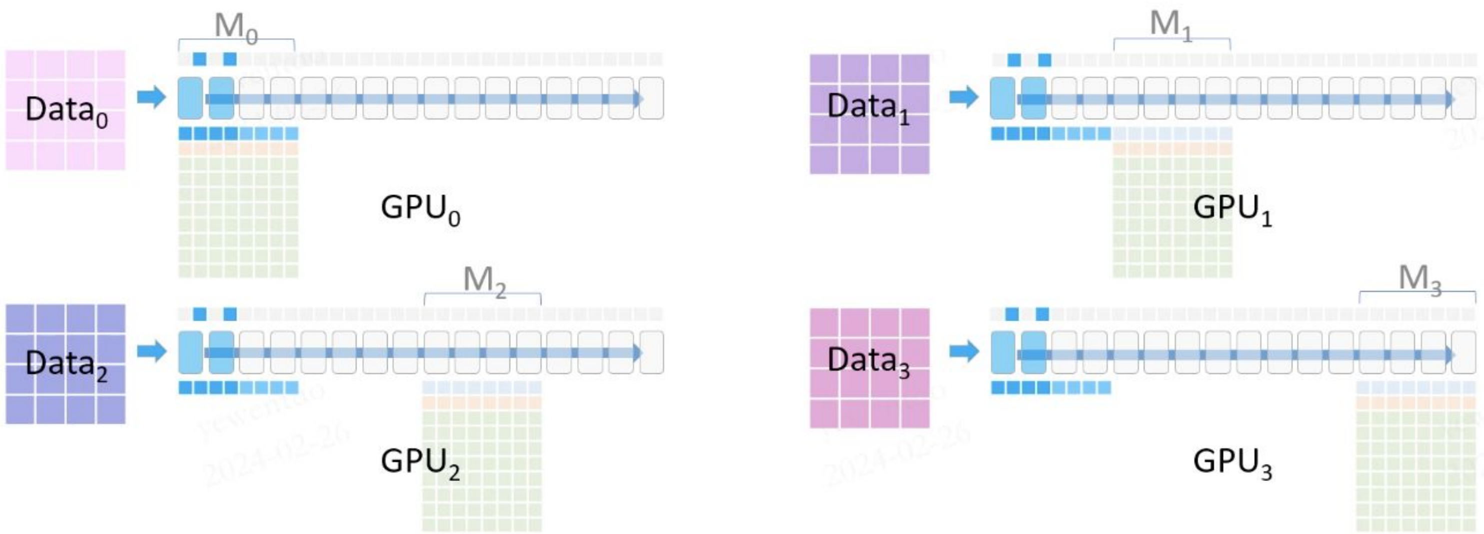


Parameters, **Gradients** and **Optimizer State** split across GPUs

Data is split across GPUs

source: https://www.microsoft.com/en-us/research/blog/zero-deepspeed-new-system-optimizations-enable-training-models-with-over-100-billion-parameters/

ZeRO Example Forward



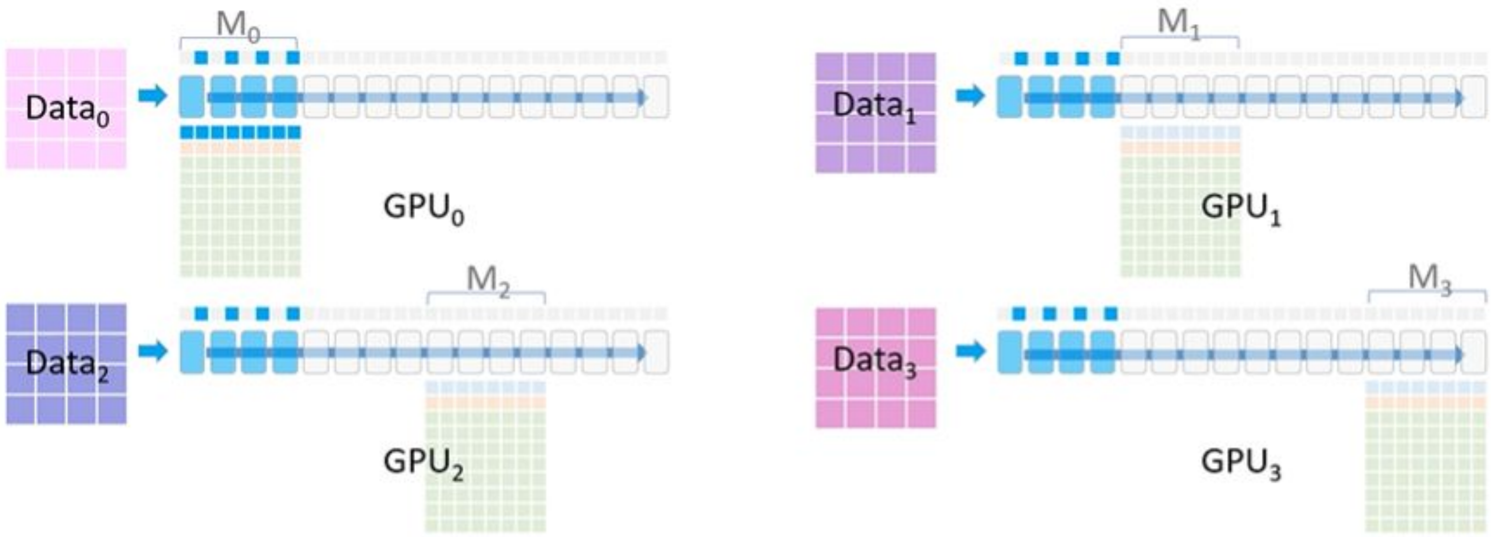
Params **broadcasted** from GPU 0

Each GPU calculates on its own Data

A subset of activations are saved for **checkpointing**

source: *https://www.microsoft.com/en-us/research/blog/zero-deepspeed-new-system-optimizations-enable-training-models-with-over-100-billion-parameters/*

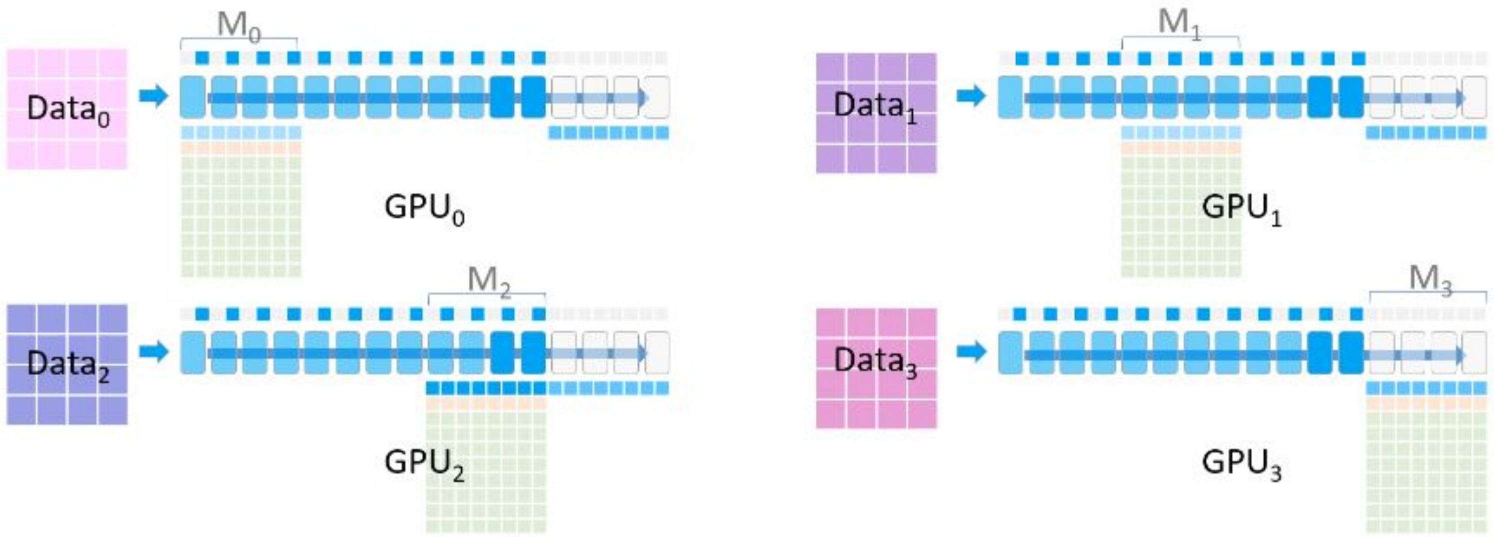
ZeRO Example Forward



Once complete, other GPUs delete the parameters

source: https://www.microsoft.com/en-us/research/blog/zero-deepspeed-new-system-optimizations-enable-training-models-with-over-100-billion-parameters/

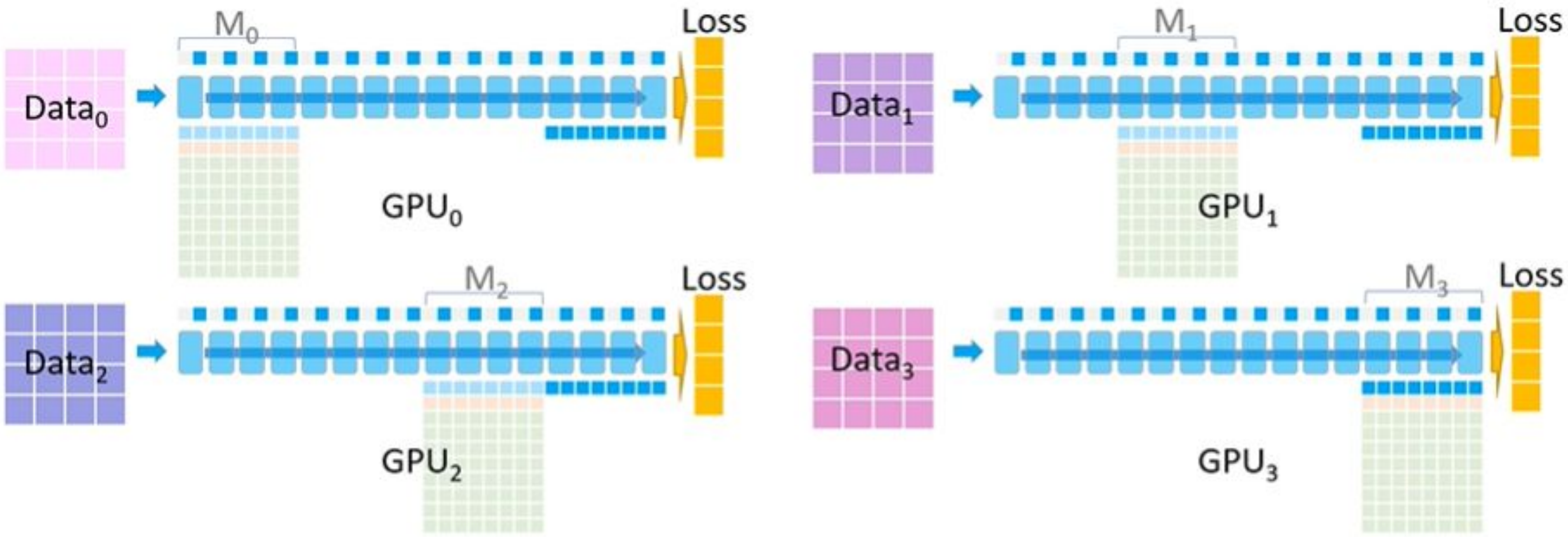
ZeRO Example Forward



Continue until all GPUs are complete

source: https://www.microsoft.com/en-us/research/blog/zero-deepspeed-new-system-optimizations-enable-training-models-with-over-100-billion-parameters/

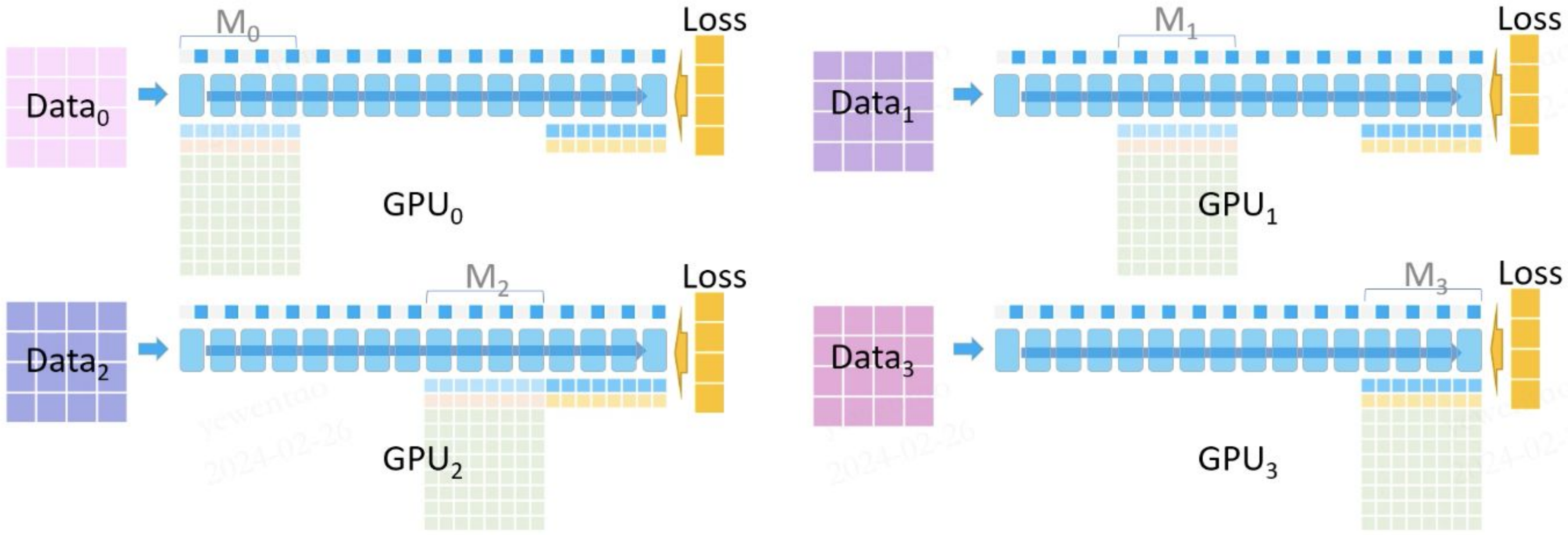
ZeRO Example Forward



All GPUs calculate the **loss**

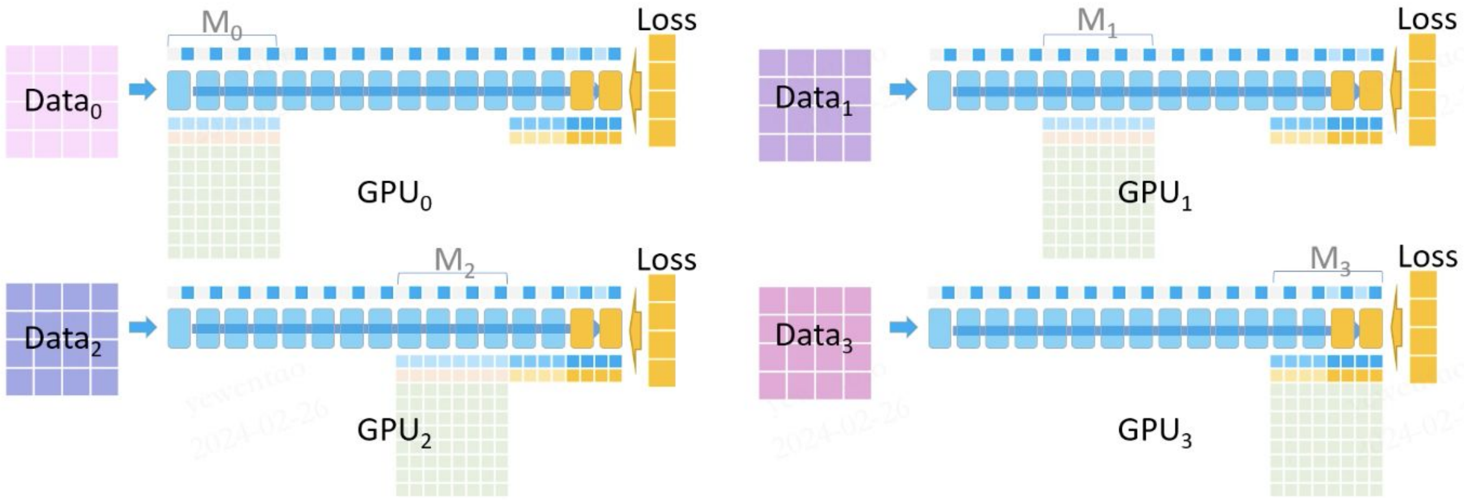
source: *https://www.microsoft.com/en-us/research/blog/zero-deepspeed-new-system-optimizations-enable-training-models-with-over-100-billion-parameters/*

ZeRO Example Backward



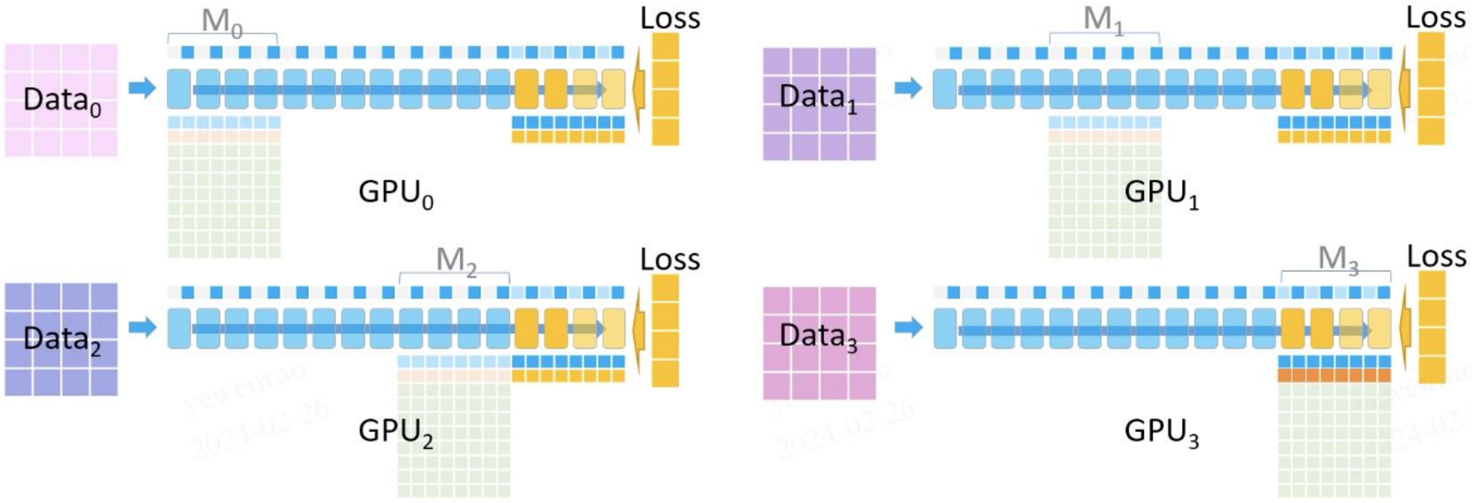
Start backward (Parameters of M3 reused)

ZeRO Example Backward



Params + checkpointing activations -> recompute the activations

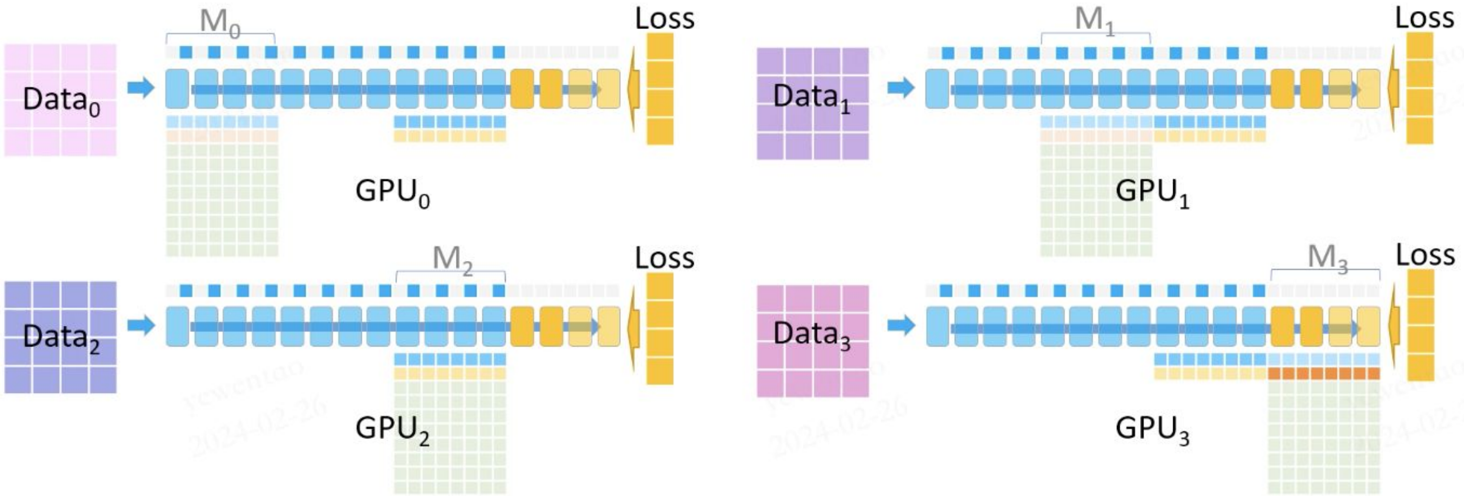
ZeRO Example Backward



Gradients **reduced** to GPU3

Note: This communication is overlapped with calculation as well

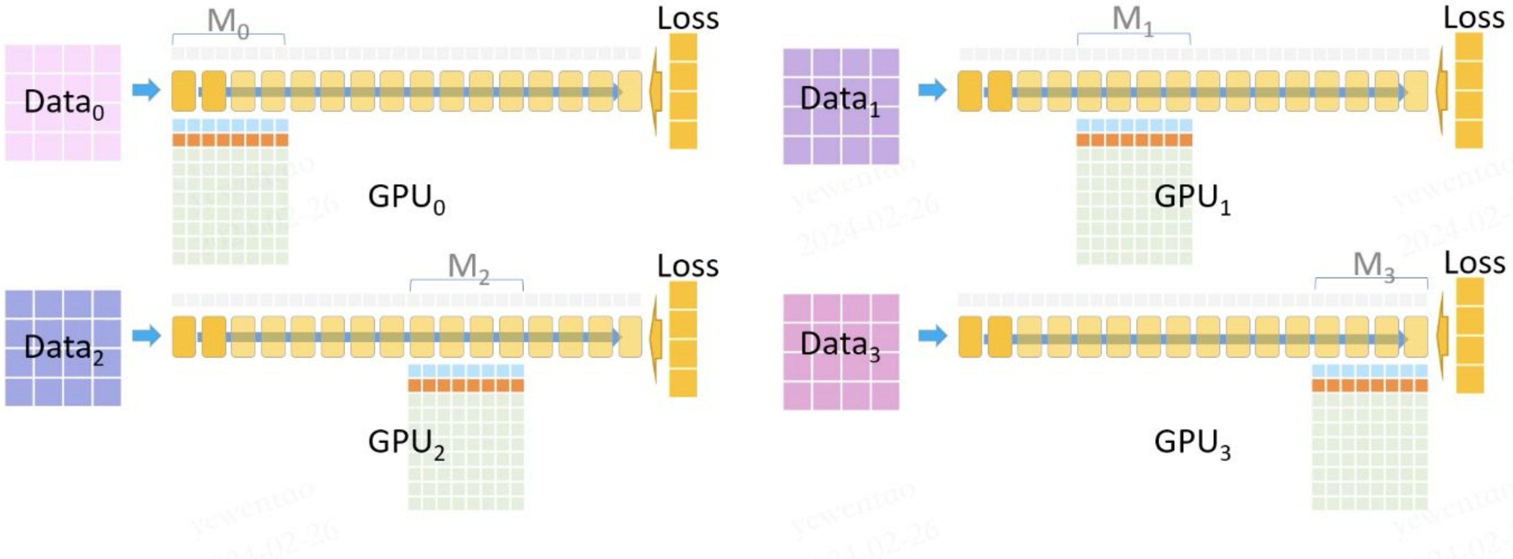
ZeRO Example Backward



Backward pass to the next layer

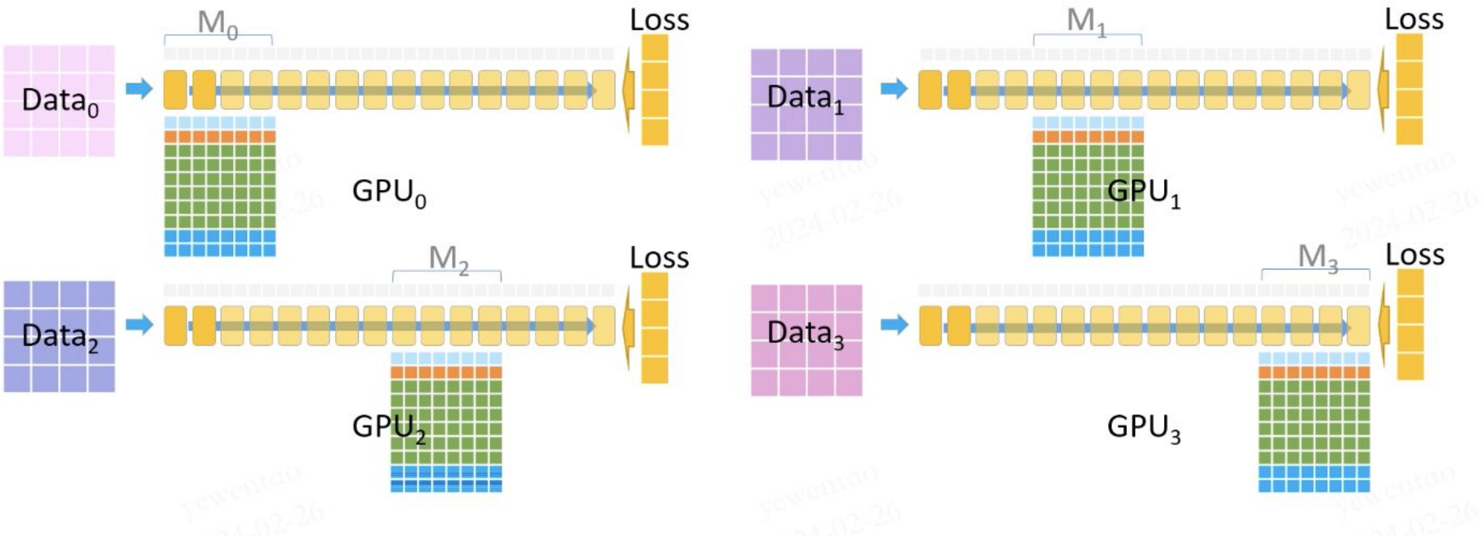
Note: This computation is overlapped with reduced communication

ZeRO Example Backward



All gradients calculated

ZeRO Example Backward

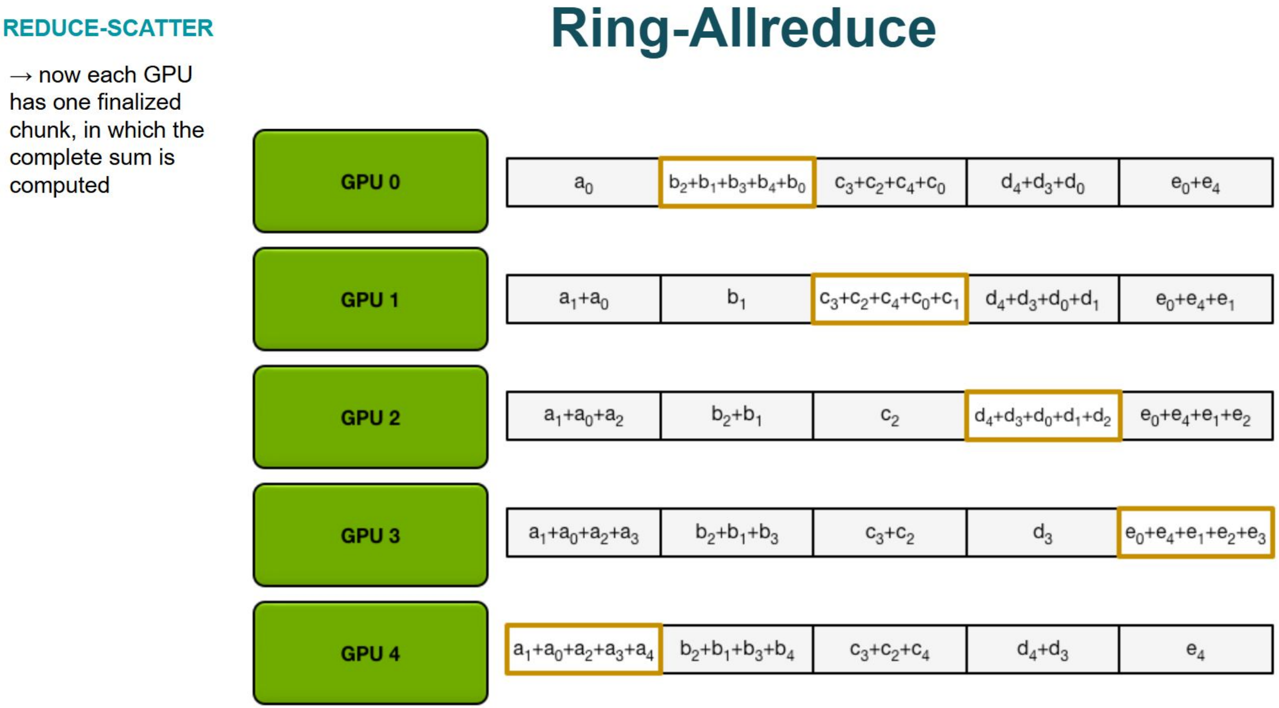


Using the gradients (fp16 casted to fp32) to update the params (fp32)

Question: Communication Volume?

source: https://andrew.gibiansky.com/blog/machine-learning/baidu-allreduce/

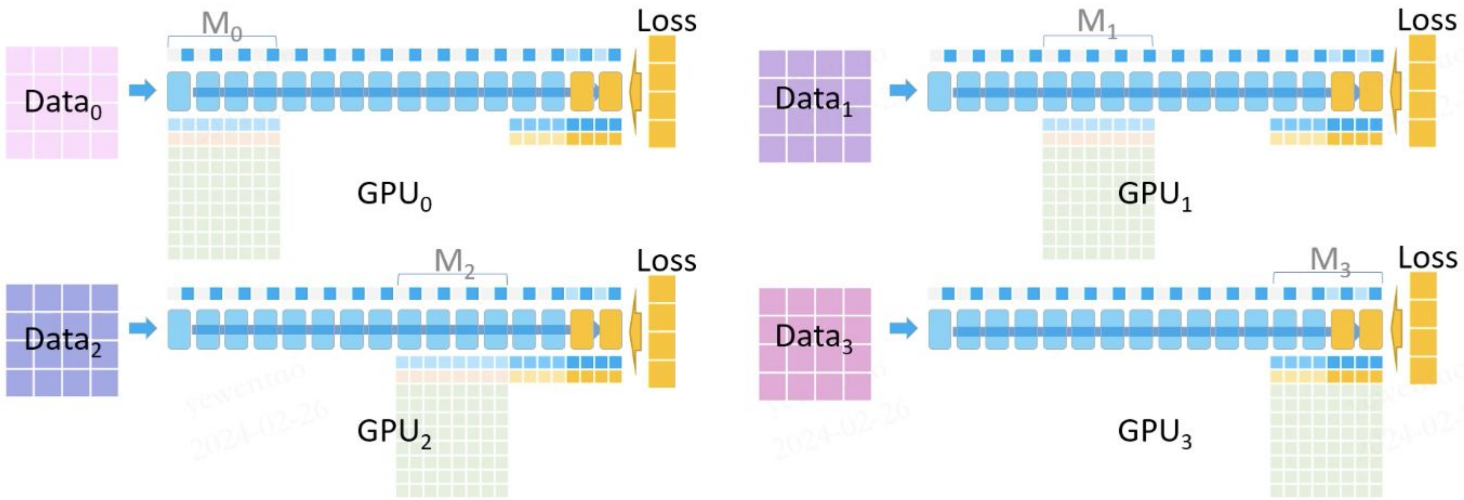
Question: Communication Volume?



Assume we have Ψ parameters, Traditional DP involves a all-reduce for gradients, $2 * (N-1) * \Psi / N \approx 2\Psi$

source: https://andrew.gibiansky.com/blog/machine-learning/baidu-allreduce/

Question: Communication Volume?

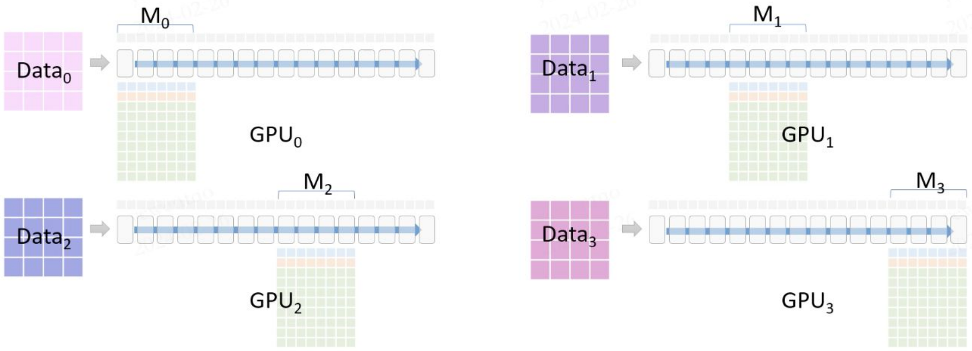


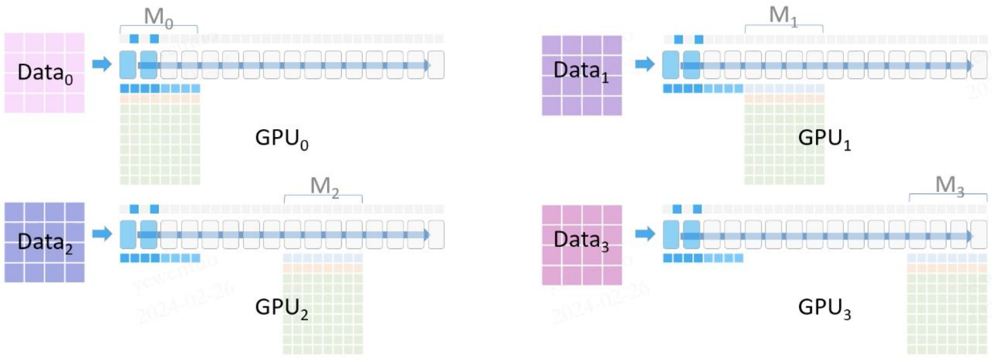
Assume we have Ψ parameters, N GPUs

Hint1: param broadcasted?

Hint2: gradients reduced?

Question: Communication Volume?

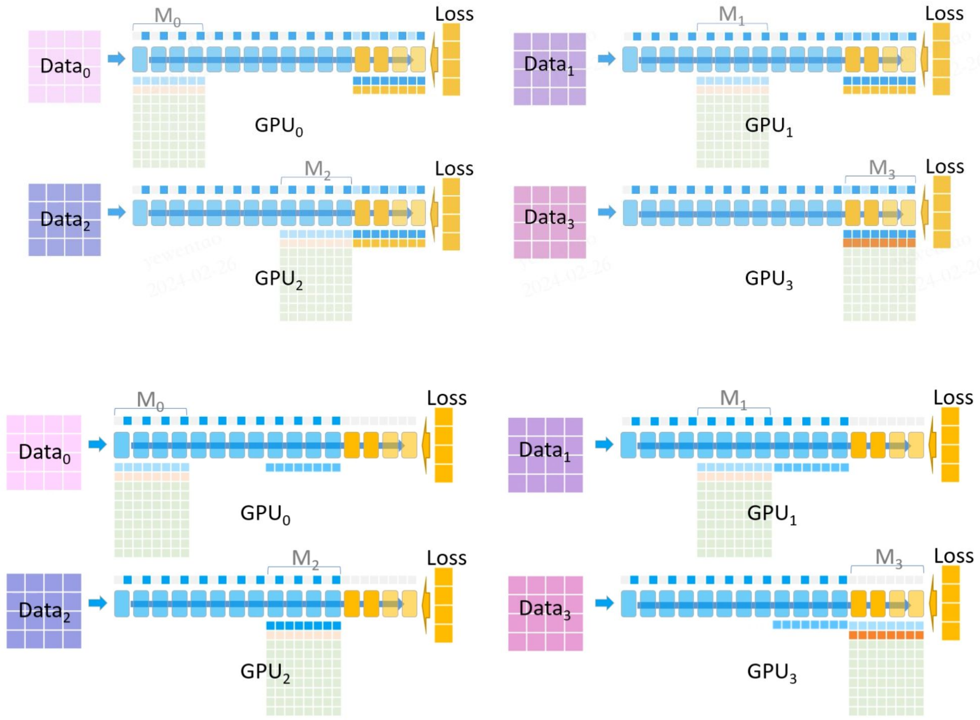




Assume we have Ψ parameters

1. Param broadcasted in forward: $\Psi/N * N = \Psi$

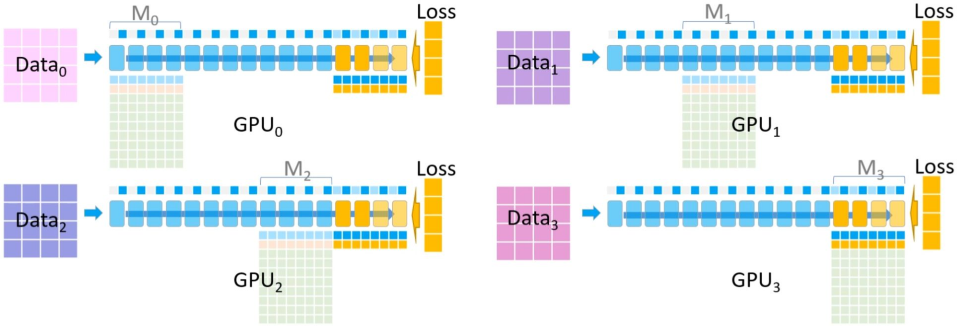
Question: Communication Volume?

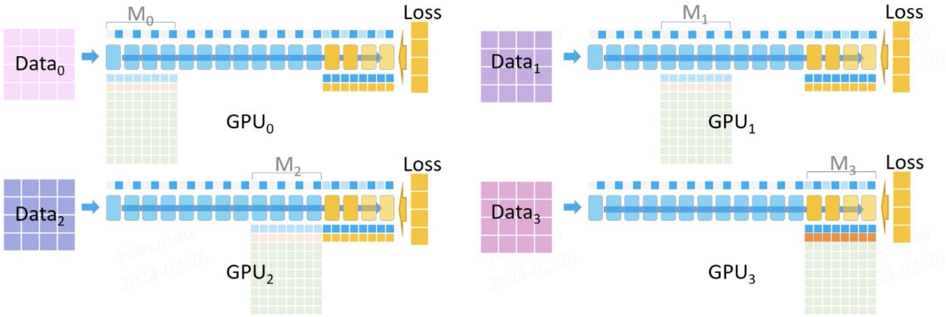


Assume we have Ψ parameters

2. Param broadcasted in backward: Ψ/N * N = Ψ

Question: Communication Volume?





Assume we have Ψ parameters

3. Gradients reduced in backward: $\Psi/N * N = \Psi$

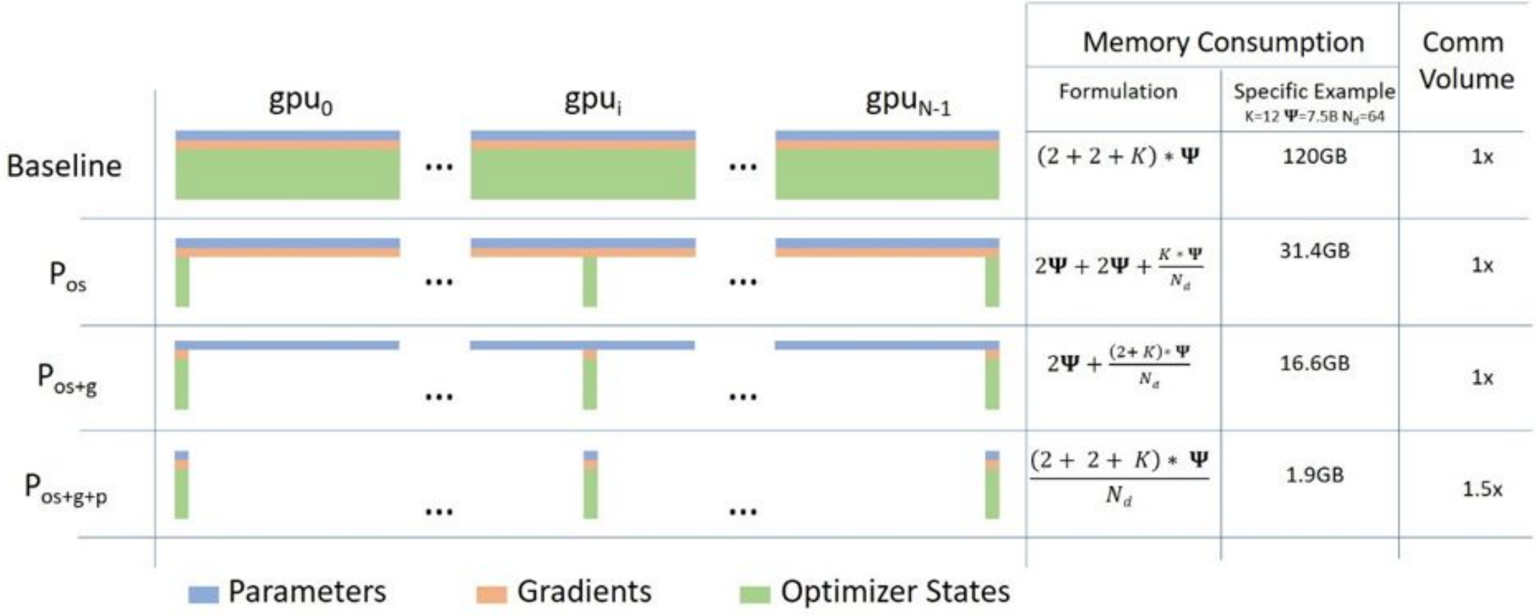
Question: Communication Volume

Assume we have Ψ parameters

1. Param broadcasted in forward: Ψ
2. Param broadcasted in backward: Ψ
3. Gradients reduced in backward: Ψ

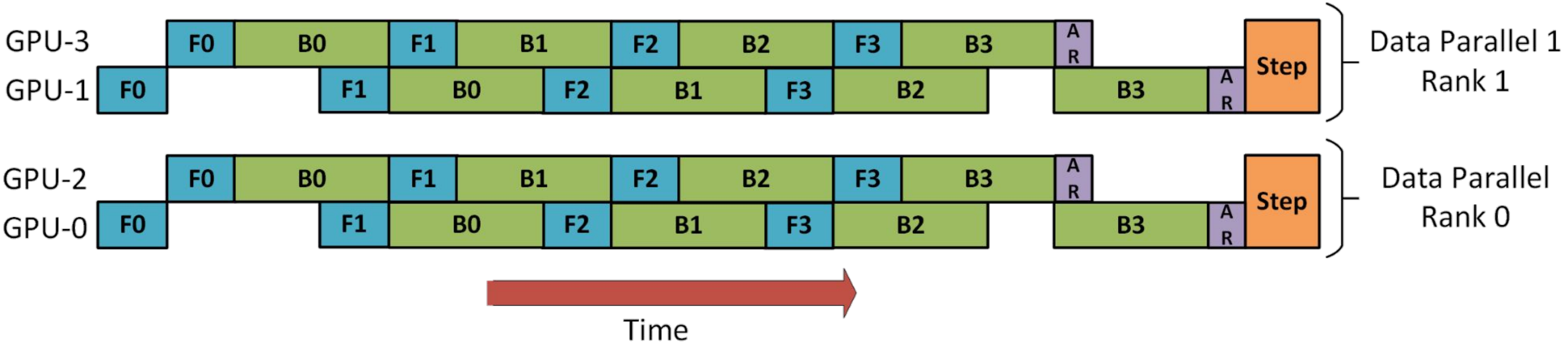
So with stage3: Total volume: $3 * \Psi$

Zero Overview



source: https://arxiv.org/pdf/1910.02054

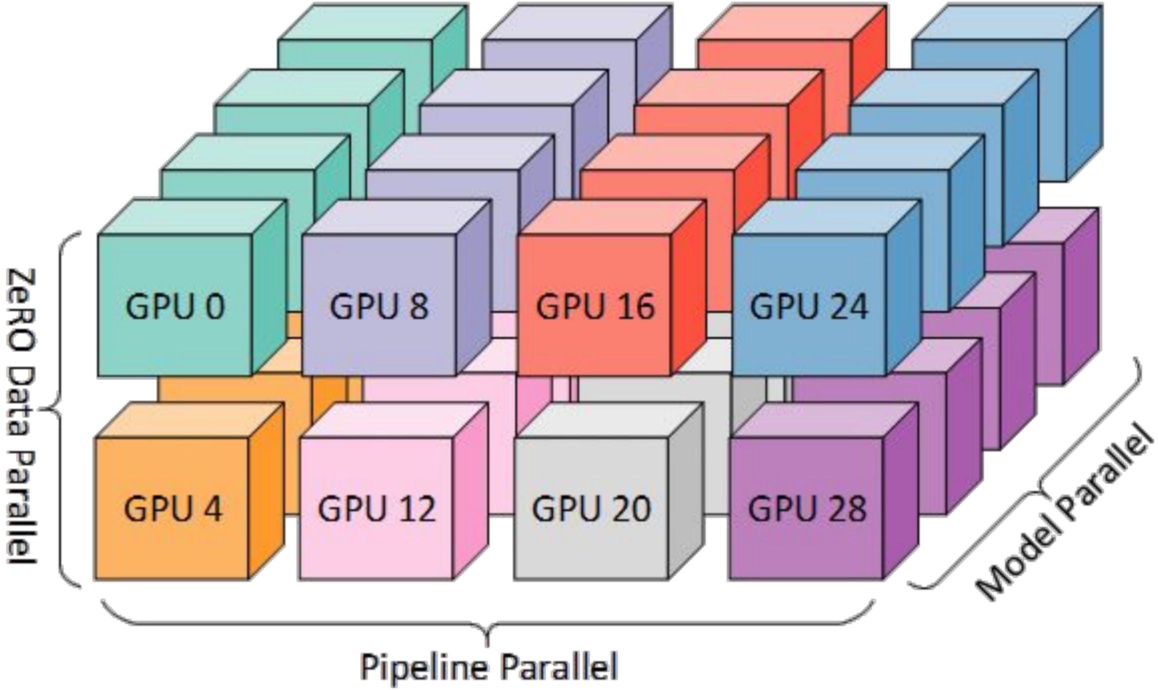
Combining All Together 2D Parallel



DP=2, PP=2

source: https://huggingface.co/transformers/v4.11.3/parallelism.html

Combining All Together 3D Parallel



Example in industry: (MP) TP=16, PP = 8, Zero (stage1) DP= 8, 1024 NPUs to train LLama2 70B

source: https://huggingface.co/transformers/v4.11.3/parallelism.html

Appendix

Team:
Alissa Amch (aa2739); Wentao Ye(wy335)

Instructor:
Mohamed Abdelfattah

Zero-R

1. **Partitioned Activation Checkpointing**
    - The activation chunk is gathered only on-demand during backward pass.
    - **ZeRO-offload** can move these partitioned activations to CPU memory.

2. **CB: Constant-Size Buffers**
    - LLMs fuse tensors into a single massive buffer to improve all-reduce efficiency.
    - ZeRO-R uses **fixed-size buffer**, splitting the work into chunks if necessary.

3. **MD: Memory Defragmentation**
    - ZeRO-R allocates contiguous memory regions for major tensors (long lived).

Zero-DP Performance

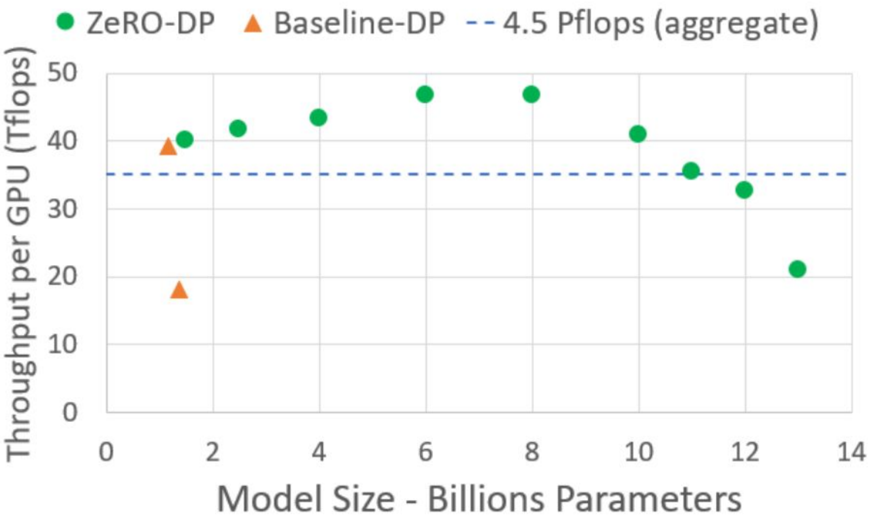


Figure 4: Max model throughput with *ZeRO*-DP.

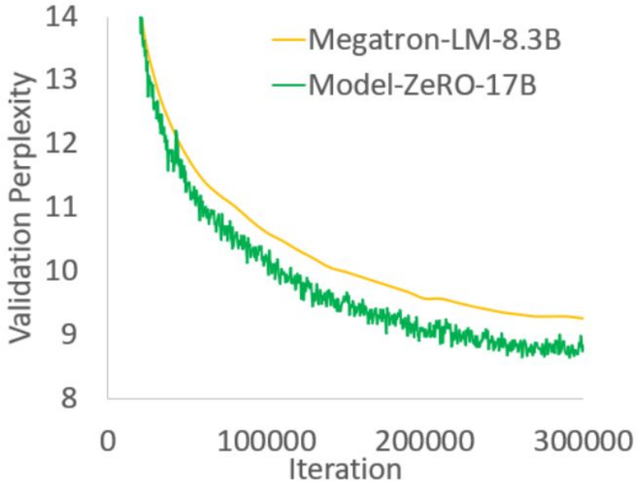


Figure 5: SOTA Turing-NLG enabled by *ZeRO*.

source: https://arxiv.org/pdf/1910.02054

Zero-DP Performance

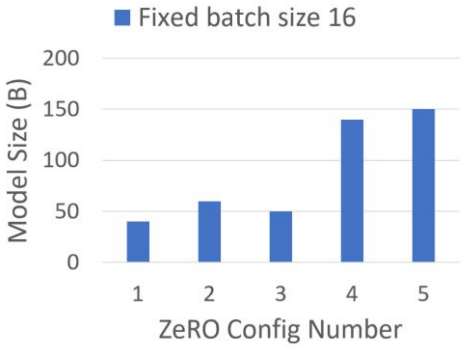


Figure 6: Max model size

.

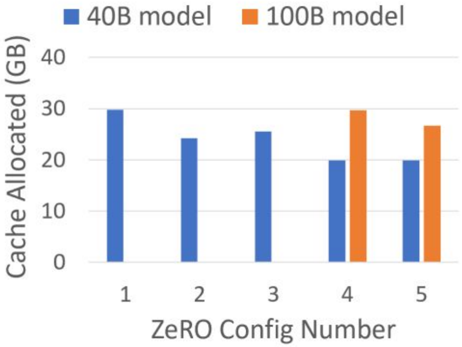


Figure 7: Max cache allocated.

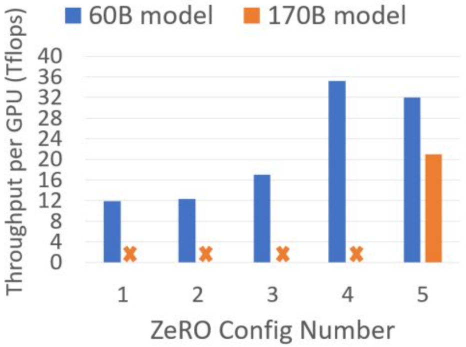


Figure 8: Throughput per GPU.

source: https://arxiv.org/pdf/1910.02054