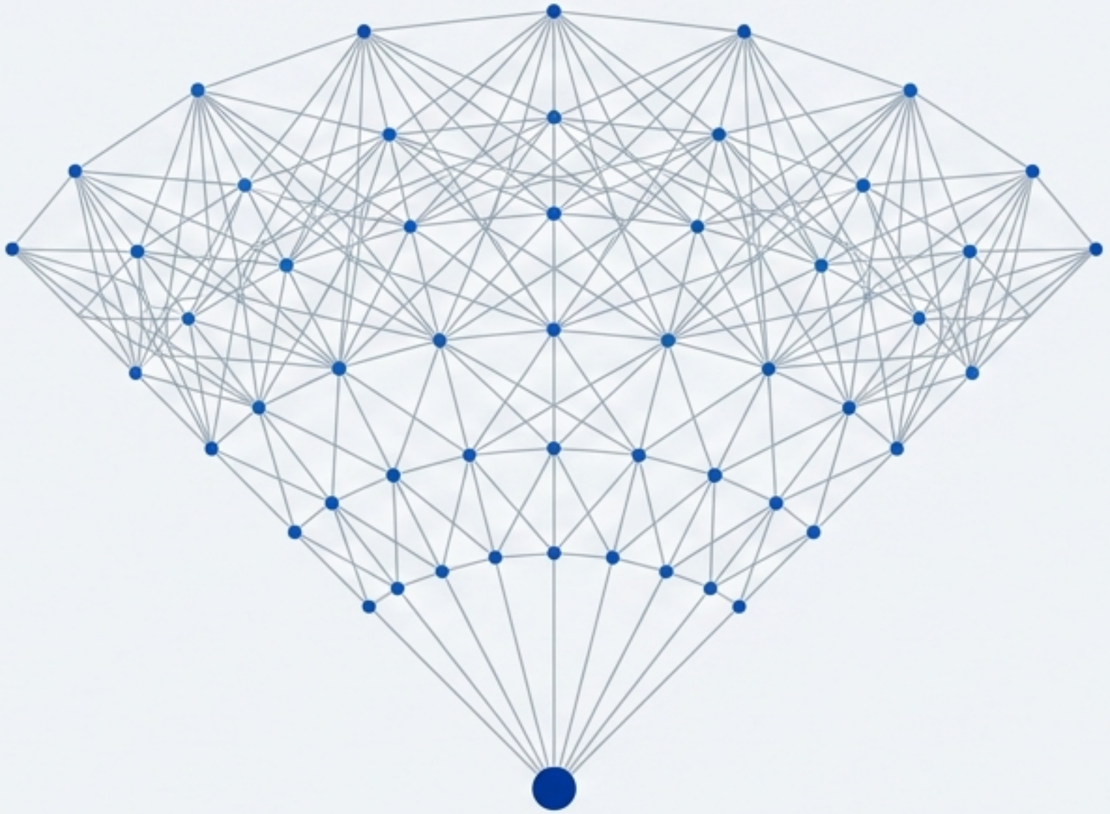
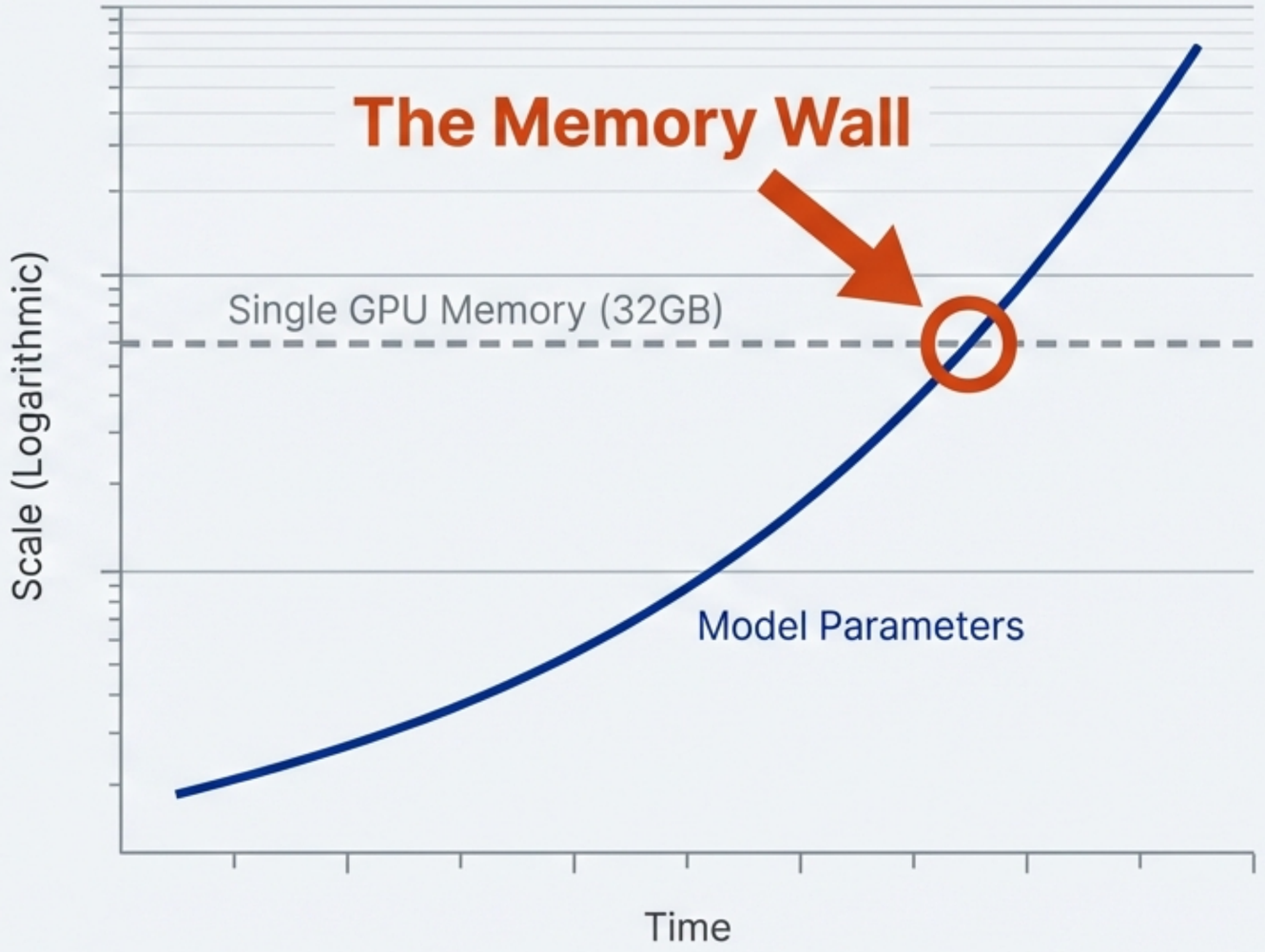
ZeRO: Memory Optimizations Toward Training Trillion Parameter Models



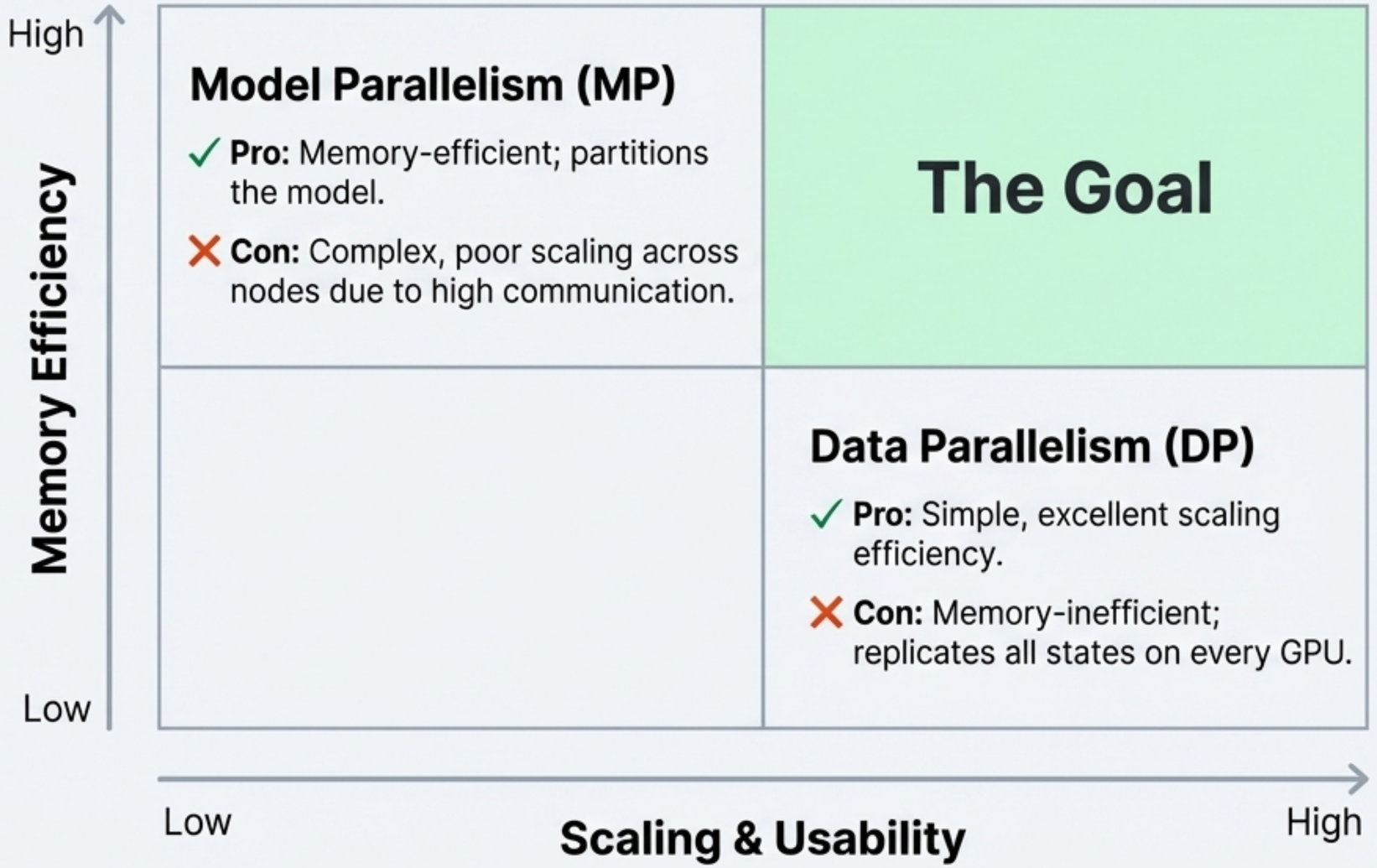
Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, Yuxiong He
Microsoft Research

The Memory Wall: Model Size is Outpacing GPU Capacity

- AI models are growing exponentially, from GPT-2 (1.5B) to T5 (11B), unlocking major accuracy gains.
- However, single GPU memory (e.g., 32GB) is a hard, physical limit. Standard data parallelism fails for models with more than 1.4B parameters.
- The problem is memory bloat: a 1.5B parameter model (3GB for weights) requires over 24GB to train with the Adam optimizer due to optimizer states and gradients.



Existing Parallelism Methods Force a Difficult Trade-off



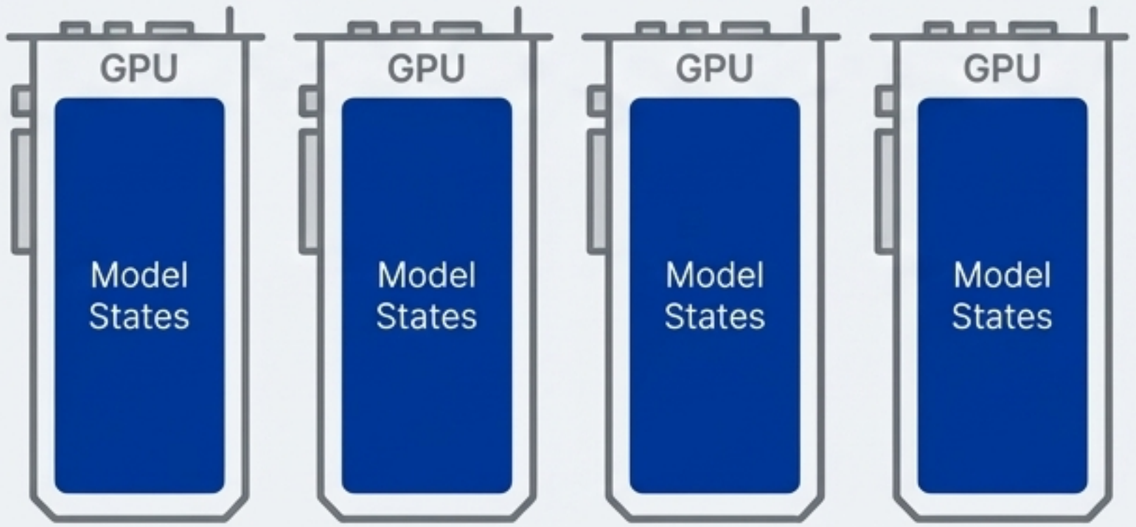
Our Question: Can We Achieve the Best of Both Worlds?

- How can we achieve the **memory efficiency** of Model Parallelism...
- ...while retaining the **usability and scaling efficiency** of Data Parallelism?
- **The Goal:** Scale trainable model size proportionally to the number of devices, without sacrificing speed or requiring complex model refactoring.

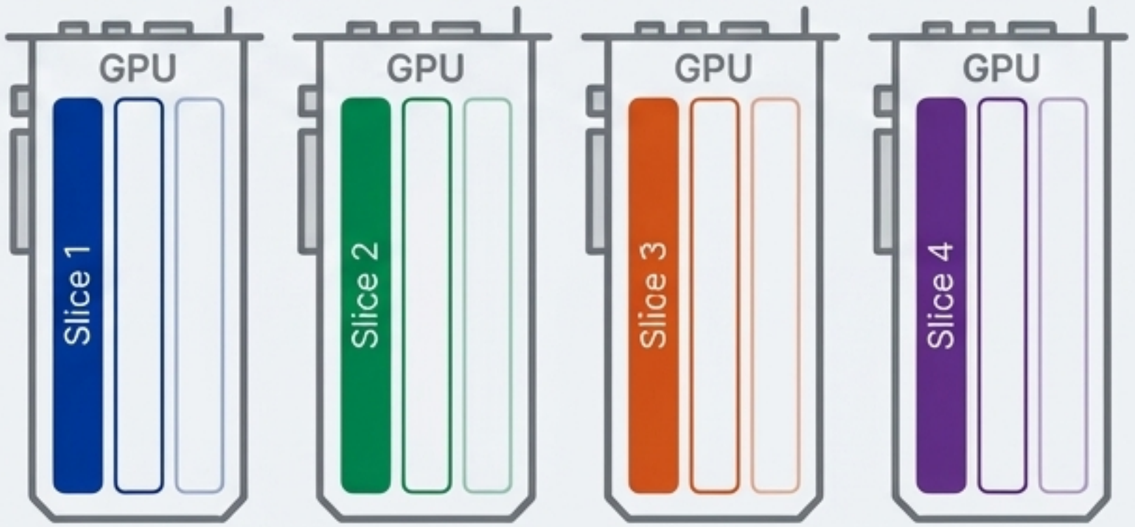
The Breakthrough: Partition Model States, Don't Replicate Them

- **The Problem:** Standard Data Parallelism redundantly replicates model states on all GPUs.
- **ZeRO's Insight:** Eliminate this redundancy. Instead of replicating, partition the model states (optimizer states, gradients, parameters) across the data parallel GPUs.
- Each GPU stores and updates only its slice of the state, making the aggregate memory of the cluster available.

Before: Standard DP

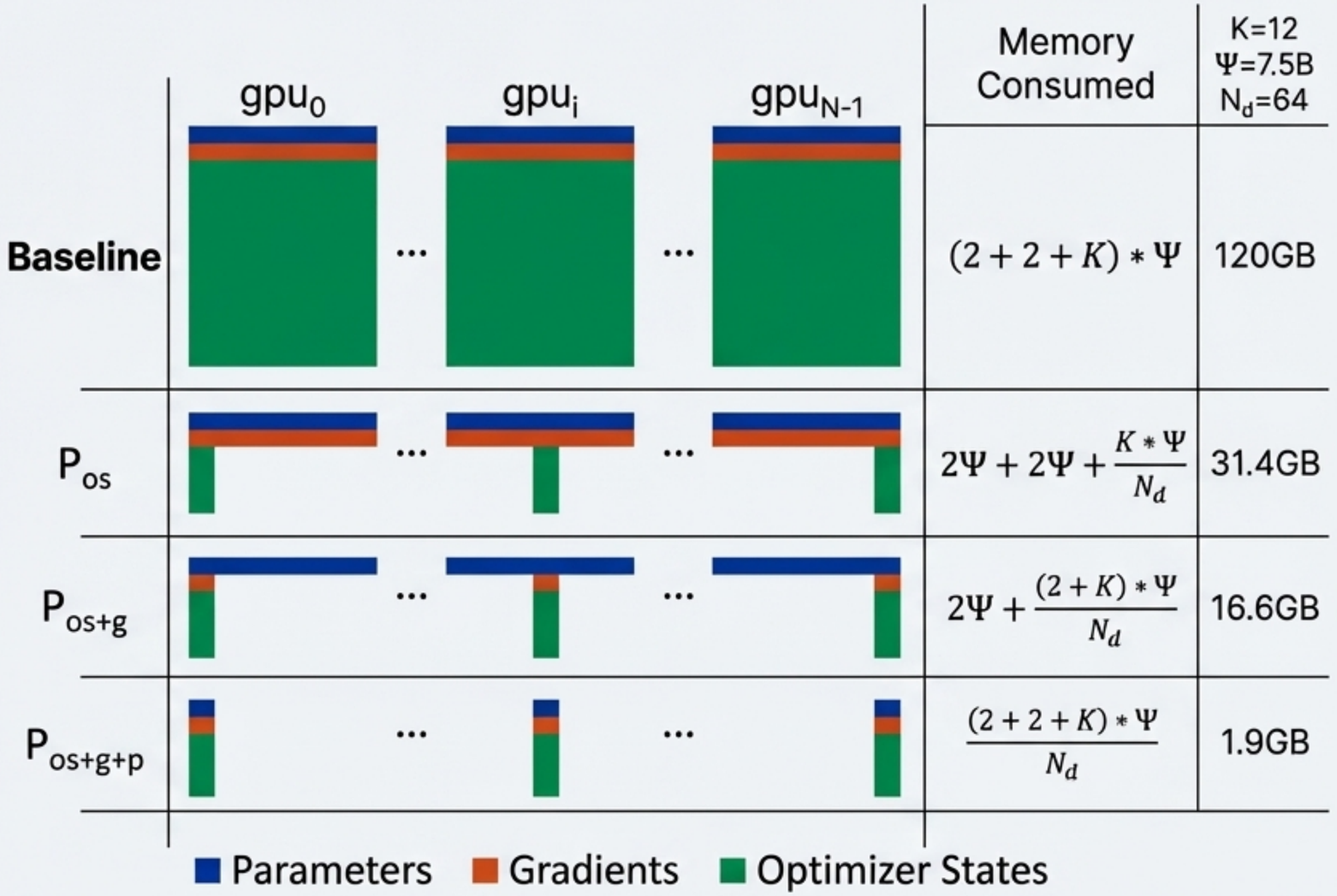


After: ZeRO-DP



ZeRO-DP Progressively Partitions States in Three Stages

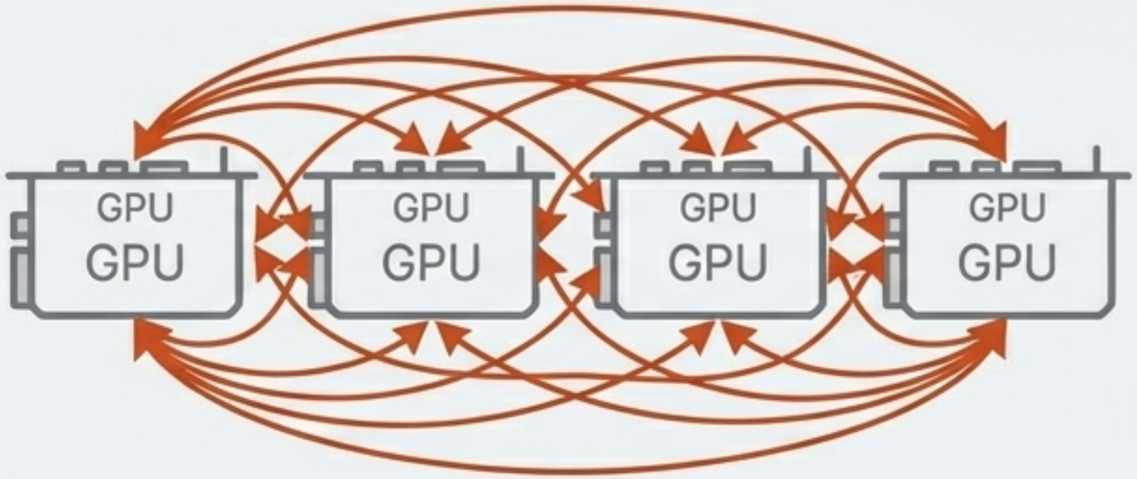
- Based on a 7.5B parameter model (64 GPUs):
  - → Stage 1 (P_{os}): Partition Optimizer States. 4x memory reduction. (120GB → 31.4GB)
  - → Stage 2 (P_{os+g}): Also partition Gradients. 8x memory reduction. (31.4GB → 16.6GB)
  - → Stage 3 (P_{os+g+p}): Also partition Parameters. Memory scales with GPU count (N_d). (16.6GB → 1.9GB)



ZeRO Retains High Efficiency with Low Communication Overhead

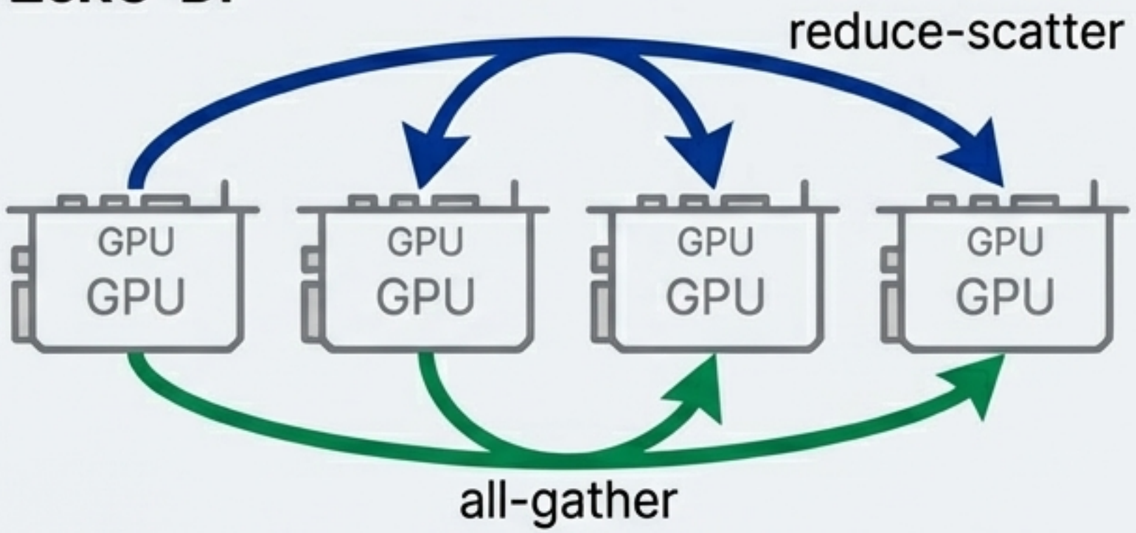
- ZeRO avoids the fine-grained communication that cripples Model Parallelism across nodes.
- It uses a dynamic communication schedule with highly optimized collective operations.
- **Communication Volume vs. Standard DP:**
  - Stages 1 & 2: Same communication volume.
  - Stage 3: Only a modest 1.5x increase in communication volume.

Model Parallelism



High Volume, Fine-Grained

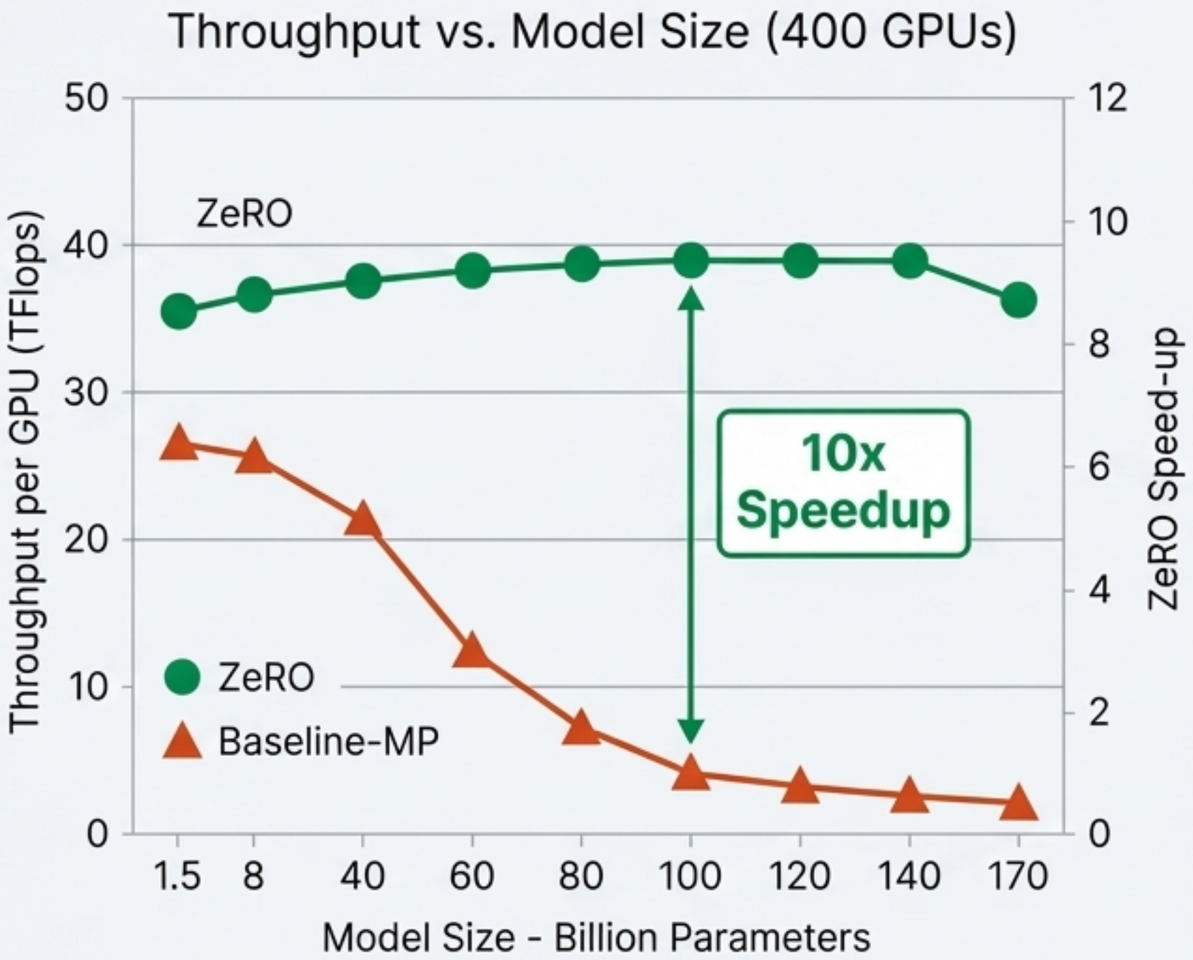
ZeRO-DP



Efficient, Coarse-Grained

ZeRO Trains 100B+ Models with a 10x Throughput Advantage

- On a 400 GPU cluster, ZeRO achieves over **38 TFlops/GPU** on a 100B parameter model, with aggregate performance surpassing **15 Petaflops**.
- The SOTA baseline (Megatron-LM) performance degrades rapidly on models larger than 40B parameters when scaling across nodes.
- At 100B parameters, ZeRO provides a **10x speedup**, enabling training for models 8x larger than previously possible with high efficiency.



Super-Linear Speedup: Adding More GPUs Makes Each One Faster

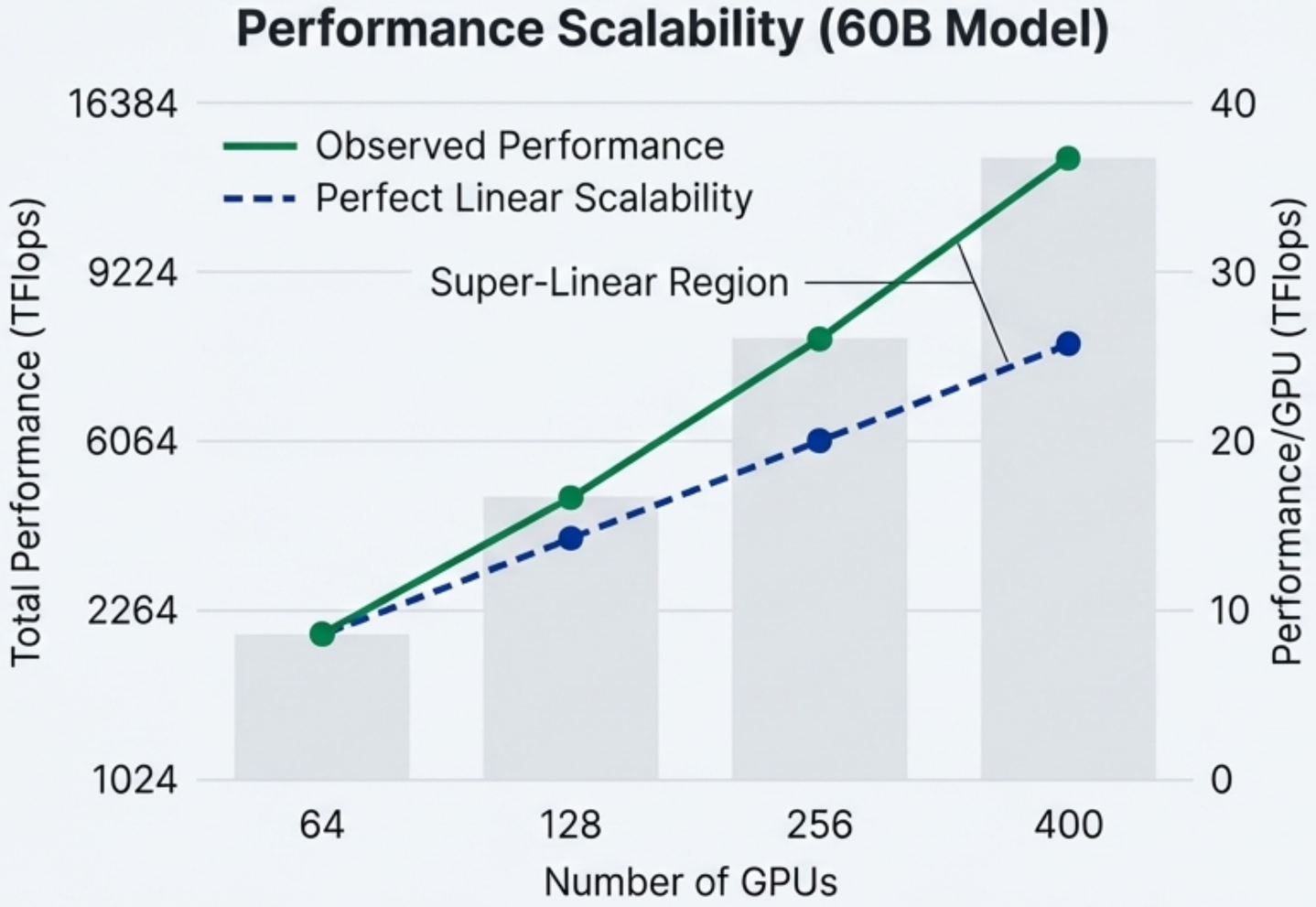
The Virtuous Cycle

1. More GPUs
→ Higher data parallel degree.

2. Higher DP degree
→ Less memory used per GPU.

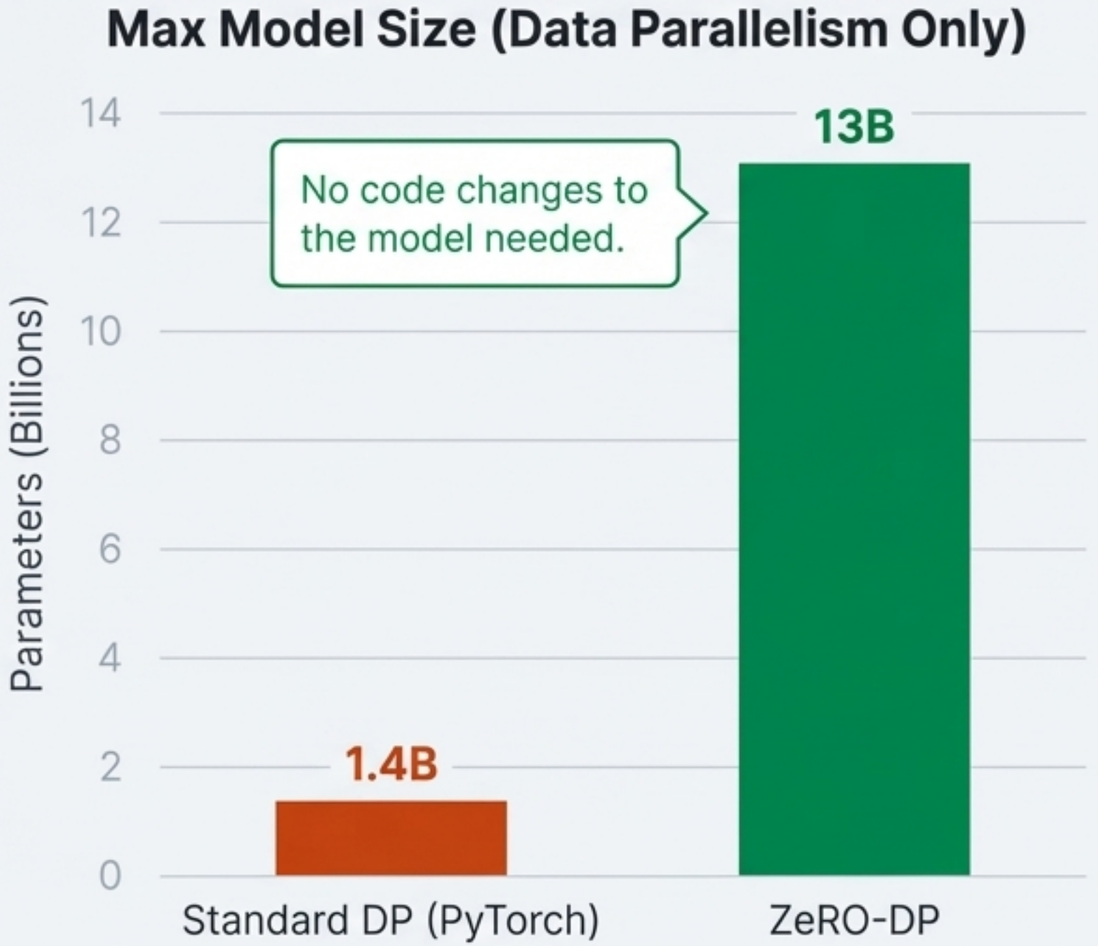
3. Less memory used
→ Larger batch size can be used per GPU.

4. Larger batch size
→ Higher computational efficiency.



ZeRO Democratizes Large Model Training

- ZeRO empowers scientists to train huge models using simple data parallelism, with **no model refactoring required**.
- It can train a **13 Billion** parameter model using only data parallelism, larger than T5 (11B) and Megatron-LM (8.3B).
- In comparison, standard systems like PyTorch DDP run out of memory with just 1.4B parameters.
- Powered the creation of **Turing-NLG (17B)**, a record-breaking language model.



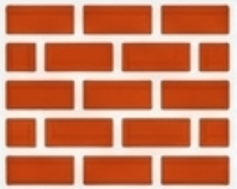
ZeRO Paves a Clear Path to Trillion-Parameter Models

- ZeRO fundamentally solves the memory capacity bottleneck for model states.
- With Stage 3, a **1 Trillion parameter model** (requiring ~16TB of state) fits on today's hardware:

$$\frac{\textbf{16}\ \text{TB Total State}}{1024\ \text{GPUs (32GB each)}} = \textbf{16 GB / GPU}$$

- The primary bottleneck for training now shifts from **memory capacity** to **compute time**.

A New Paradigm for Training AI at Scale



The Problem

The GPU memory wall was blocking progress in AI model scale.

The Insight

Eliminate memory redundancy by **partitioning** model states instead of replicating them.

The Proof

An order-of-magnitude leap in model size and speed, with unprecedented super-linear scaling.

The Future

Democratizes large model training and provides the foundation for the trillion-parameter era.

Available in: Microsoft DeepSpeed (github.com/microsoft/deepspeed)