Centauri: Enabling Efficient Scheduling for Communication-Computation Overlap in Large Model Training via Communication Partitioning

Chang Chen, Xiuhong Li, Qianchao Zhu, Jiangfei Duan, Peng Sun, Xingcheng Zhang, Chao Yang

ASPLOS '24

Presenter: Muhan Zhang

Background

- Large Language Models (LLMs) demand massive GPU resources and intensive communication.
- Communication overhead severely limits scalability and efficiency.
- Existing overlap strategies either too coarse or too fine-grained, failing to fully utilize resources.

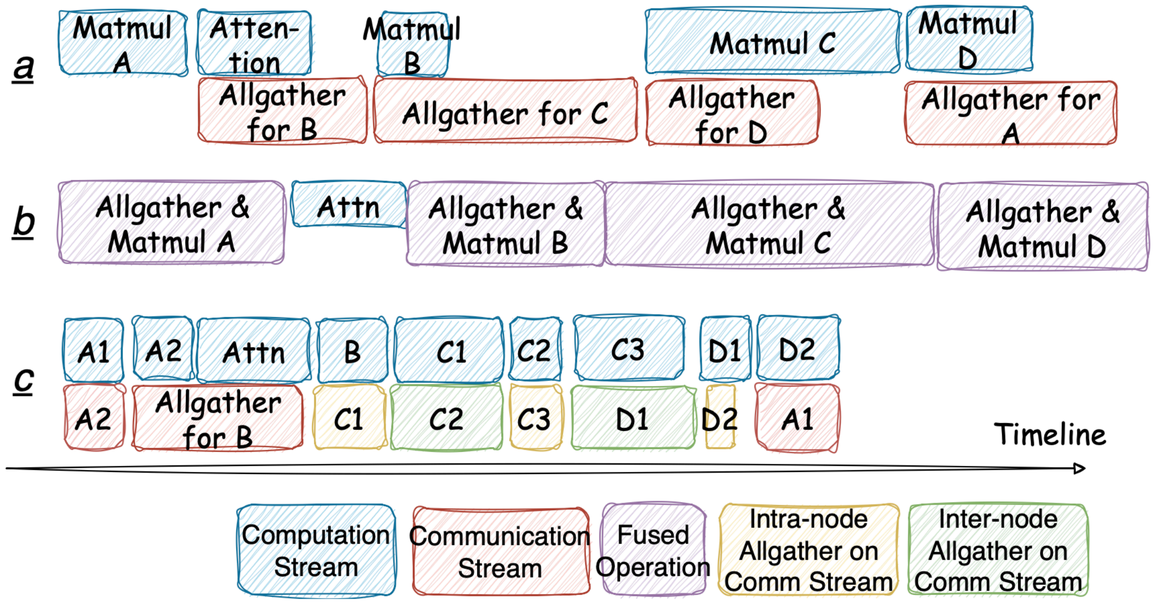


Figure 1. Different overlapping strategies on FSDP training of a simplified Transformer structure:
a is direct schedul-ing that weights are gathered before MatMuls;
b is MatMul and Allgather kernel fusion;
c is Centauri scheduling that communication is partitioned from group and workload dimensions for better overlapping

Key Contributions

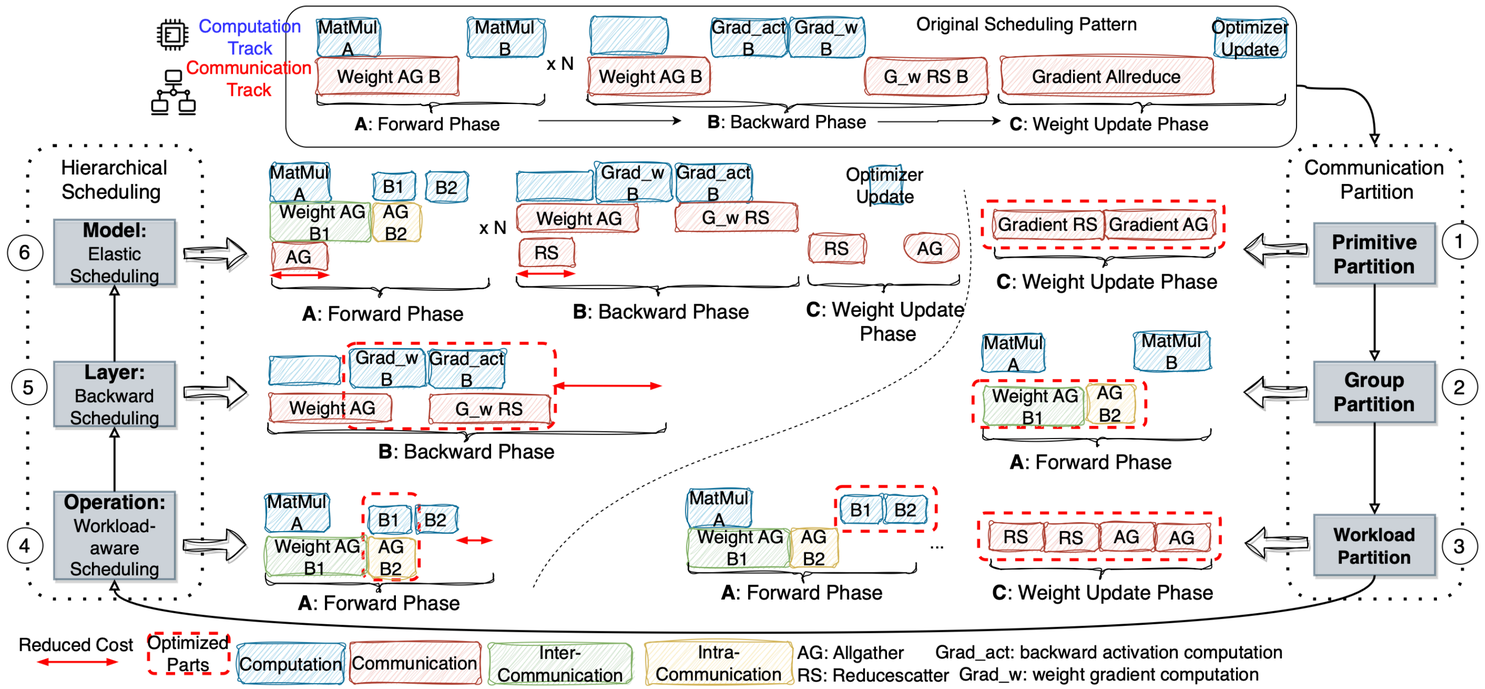
Centauri's Innovations

- Proposed a novel three-dimensional communication partition space:
  - Primitive Substitution
  - Topology-aware Group Partitioning
  - Workload Partitioning
- Designed hierarchical scheduling schemes:
  - Operation-level Scheduling
  - Layer-level Scheduling
  - Model-level Scheduling
- Implemented Centauri in Megatron-LM, demonstrating significant performance improvements.

Overview of Centauri

How Centauri Works

- Communication Partitioning decomposes communication into finer granularity.
- Hierarchical Scheduling optimizes overlap at three levels: operation, layer, and model.



Centauri workflow overview for a hybrid parallel training example of DP and FSDP.

Centauri-Primitive Partition

Optimizing Communication Patterns

- Key Idea: Break complex communication operations into simpler sub-primitives.
  - Example: AllReduce → Reduce-Scatter + AllGather
- Enables finer overlap granularity and reduces sequential dependencies.

Primitive	Sub-primitives	Scalability
AllReduce	Reduce + Broadcast	x
	Reduce-scatter + AllGather	✓
Reduce-Scatter	Reduce + Scatter	x
	Reduces of distinct roots	✓
AllGather	Gather + Broadcast	x
	Broadcasts of distinct roots	✓

Primitive Substitution List

Topology-aware Group Partitioning

Efficient Use of Network Topology

- Partition communication groups based on GPU network topology:
- High-bandwidth intra-node groups
- Low-bandwidth inter-node groups
- Maximizes communication efficiency by exploiting hierarchical topology.

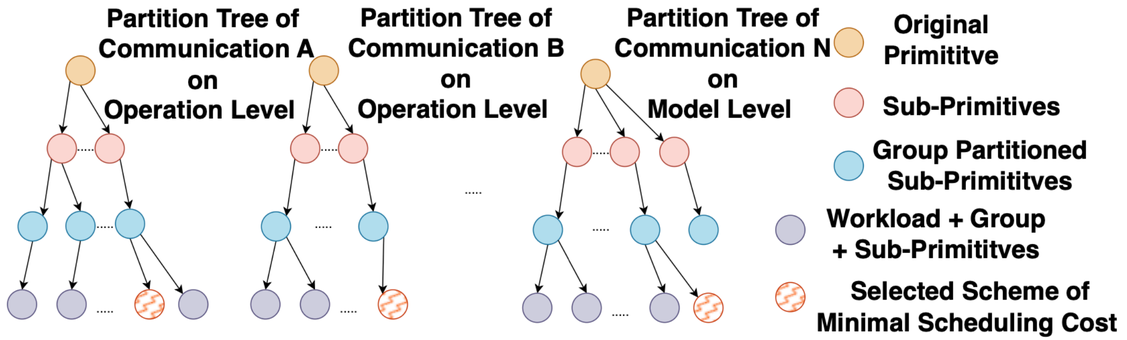


Figure 4. Communication partitioning workflow for a hybrid training task containing N communication operations.

Centauri-Workload Partition

Operation	Workload	Dims Types	Partition Dims
MatMul	$[A, B]x[B, C]$	OD, CD, OD	A, B, C
+, −, *, /, Dropout, ReLU, GeLU	$[A, B]$	OD, OD	A, B
SwiGLU, Softmax, LayerNorm	$[A, B]$	OD, ND	A

Computation Dimension Types

Principles:

- Partition workloads for fine-grained overlapping with computation.
- Consider compatibility of partition dimensions to avoid unnecessary overhead.

- Contraction dimension (CD): Contraction dimensions of operations like MatMuls and Einsums.
- None-split dimension (ND): Coupled computation along this dimension and is not preferable for splitting. It includes reduced dimensions of normalization functions.
- Other dimension (OD): This type encompasses the remaining workload dimensions, such as the batch dimension and none-contraction dimensions of MatMuls.

Hierarchical Scheduling - Overview

Why hierarchical?

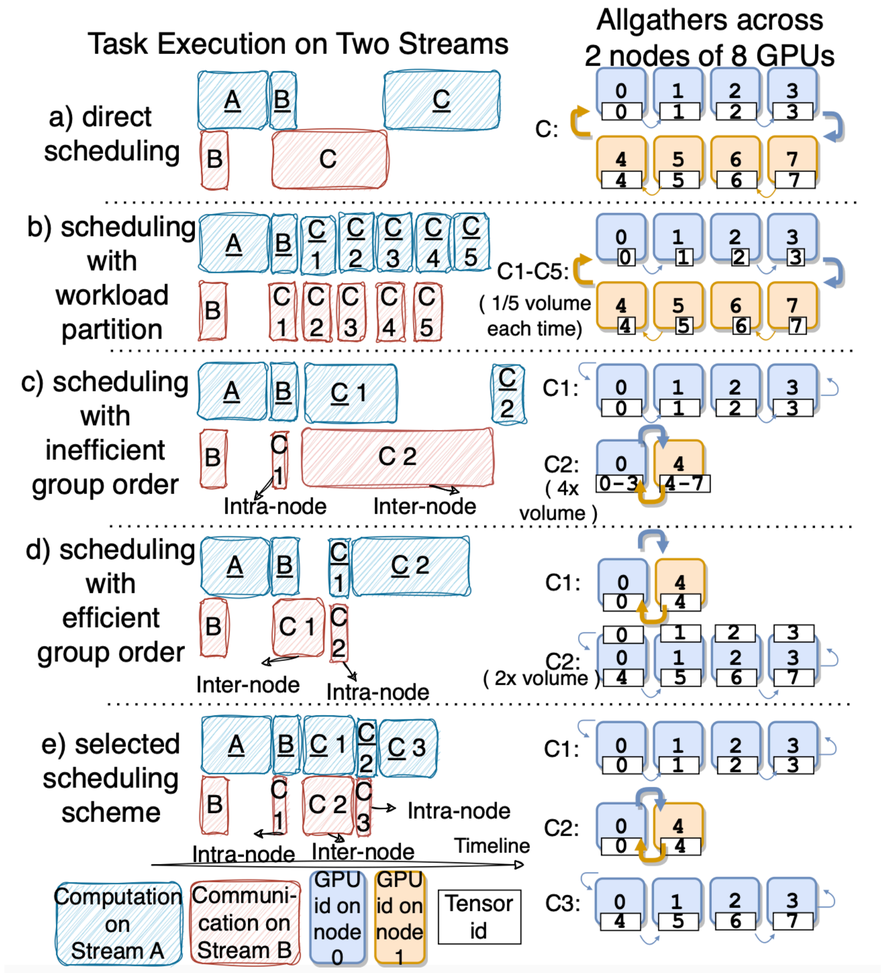
- Directly optimizing complex hybrid parallel communications is challenging.
- Hierarchical approach disentangles scheduling into clear, manageable levels.

Levels of Scheduling:

- Operation-level: Optimize overlap within each computation layer.
- Layer-level: Determine optimal execution order in backward propagation.
- Model-level: Dynamically schedule tasks across model phases for maximum overlap.

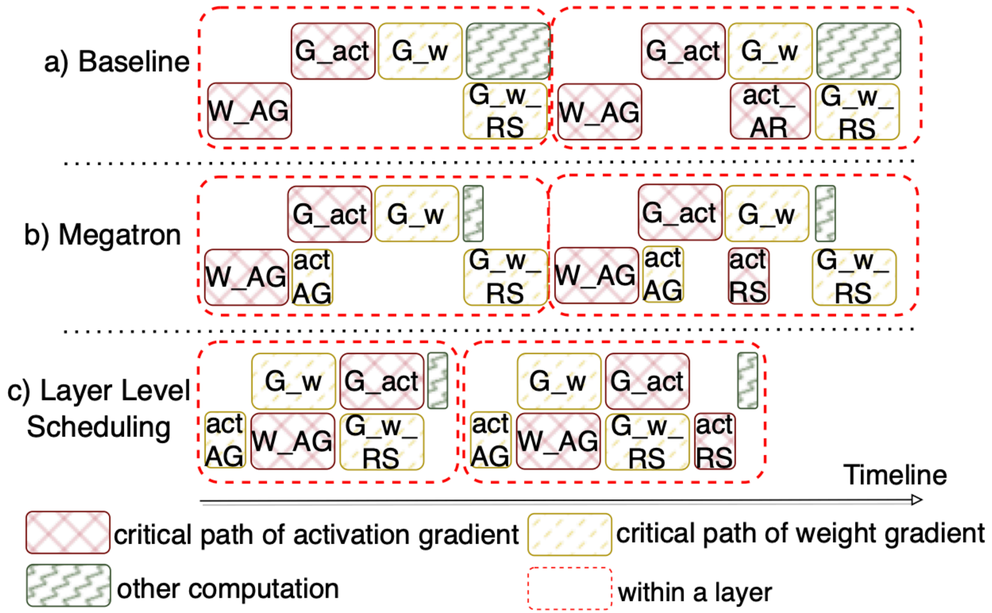
Centauri-Operation Level

- a) scheduling with coarse granularity results in a large idle gap
- b) fine granularity scheduling introduces large overhead due to excessive workload partition
- c) scheduling with a group order of large inter-node communication overheads
- d) group partition with a group order of small inter-node and acceptable intra-node communication over-heads
- e) the selected partition and scheduling scheme with no idle gaps



Operation Level Scheduling

Centauri-Layer Level



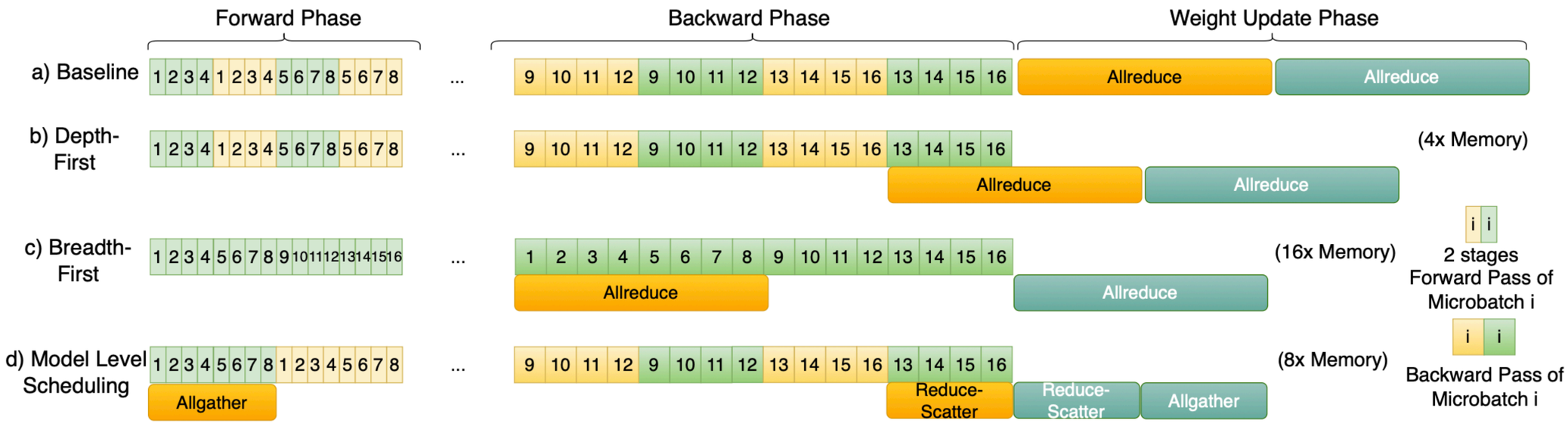
Layer Level Scheduling

- backward phase has a natural scheduling space
- a) Out-of-order propagation that activation computation is scheduled with higher priority.
- b) Megatron- LM sequence parallel method that aims at overlapping activation communication.
- c) Centauri schedules critical path to maximize overlapping.

T1 = T(W_AG) + max(T(G_act), T(act_AG))+max(T(G_w), T(act_RS)) + max(T(G_w_RS), T(others))

T2 = T(act_AG) + max(T(G_w), T(W_AG))+max(T(G_act), T(G_w_RS)) + T(act_RS) + T(others)

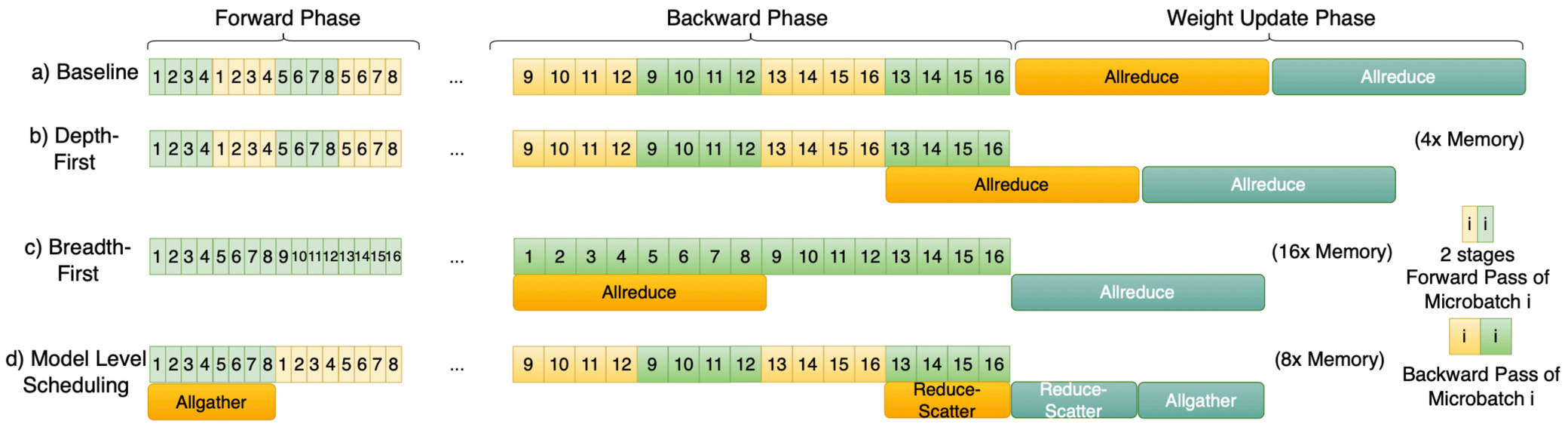
Centauri-Model Level



Model Level Scheduling

- a) An interleaved pipeline of 2 stages, 16 micro-batches per batch, and a depth of 4.
- b) Direct overlap allreduce of the second stage with backward of the first stage, with minimal memory cost of 4x activation.
- c) All micro-batches (16) launched together for maximal overlapping, 16x activation.
- d) Minimal number (8) of micro-batches launched together for well overlapping, with medium memory cost of 8x activation.

Centauri-Model Level



Model Level Scheduling

$$\min \; T = \max\{C\_AG - L\_1 * C\_fw, 0\} + \max\{C\_RS - L\_2 * C\_bw, 0\}$$

$$\text{s.t.} \;\; L\_1 * m \le M, \; L\_2 * m \le M, \; L\_1 + L\_2 \le mb$$

Evaluation Setup

- Testbeds: Two GPU clusters with different network characteristics
  - Cluster A: 8 NVIDIA A100-80G GPUs per node with NVSwitch (600 GB/s)
    - Limited cross-node bandwidth: 25 Gb/s per device
  - Cluster B: Higher cross-node bandwidth (100 Gb/s per device)
- Models: LLaMA in 4 size configurations (Small, Medium, Big, Large)
- Metric: TFLOPS/GPU (attention and MLP computation FLOP ÷ end-to-end time)

Single Parallel Performance - FSDP

- FSDP training shows consistent performance improvement across models
- Batch size significantly impacts overlapping efficiency:
  - Larger batch sizes → better speedup (up to 1.46× improvement)
- Centauri's partitioning strategy:
  - Group partition: intra-node and inter-node workload scheduling
  - Workload partition: optimizes granularity to balance overheads

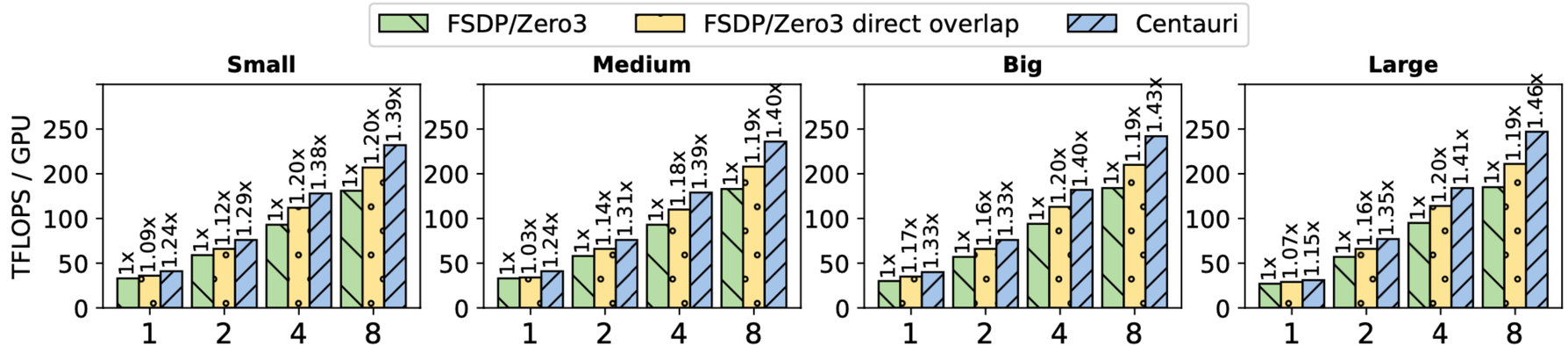


Figure 8. Performance of FSDP/Zero3 tasks on 2 nodes of Cluster A, with FSDP group size 16.

Single Parallel Performance - DP

- Performance gains increase with batch size
- Up to 1.45× speedup over Zero1+2 baseline

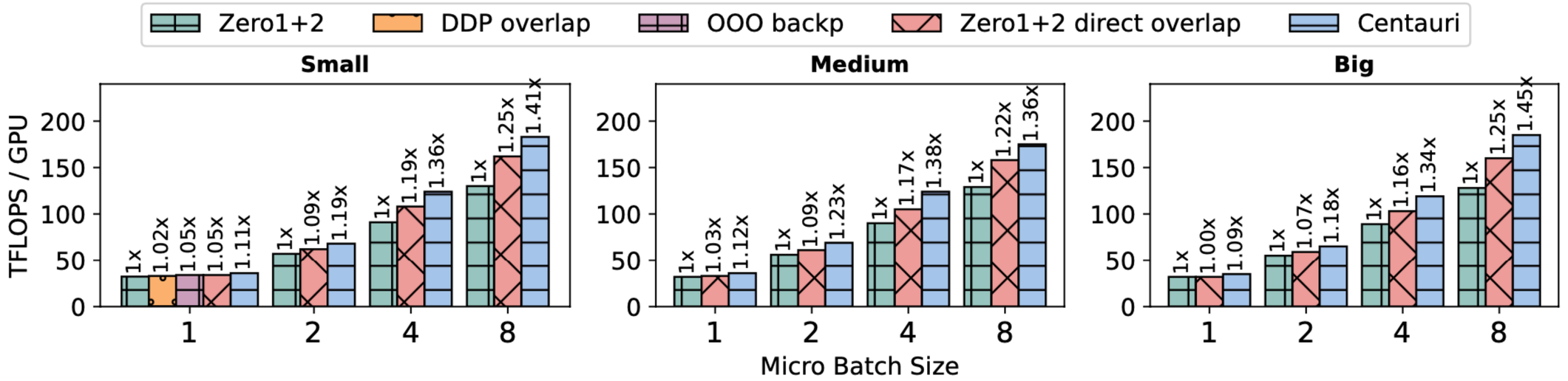


Figure 9. Performance of DP tasks on 2 nodes of Cluster A, with DP group size of 16.

Hybrid Parallelism Performance

Robust Performance in Hybrid Configurations

- FSDP+DP: Centauri efficiently overlaps inter-node and intra-node communication.
- TP+PP+DP: Dynamically adjusts micro-batches for maximal overlap and optimal memory usage.
- Up to 1.49× speedup over current hybrid methods.

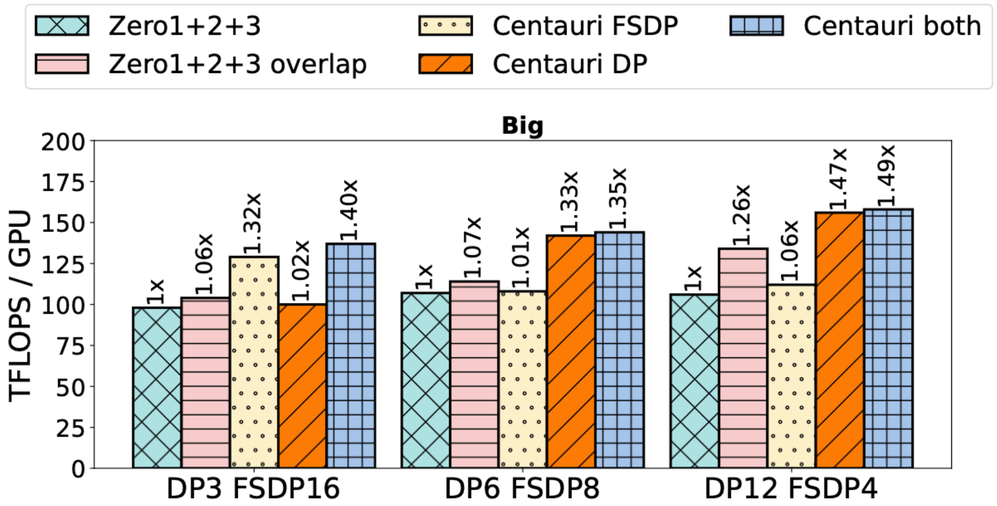


Figure 10. Performance of FSDP+DP tasks on 6 nodes of Cluster A, with a total group size of 48.

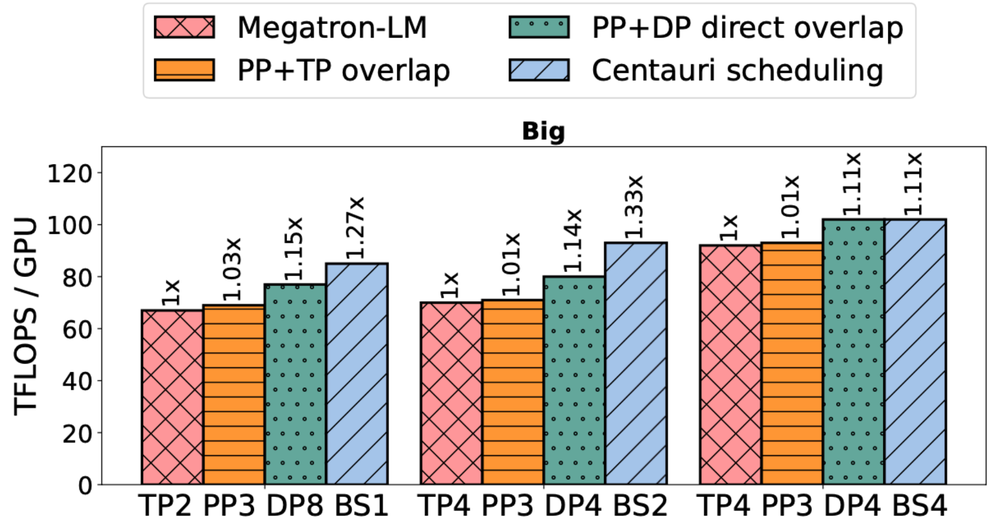


Figure 11. Performance of TP+PP+DP tasks on 6 nodes of Cluster A, with a total group size of 48 and total gradient accumulation steps of 6.

Scalability Analysis

Excellent Scalability Across Diverse Clusters

- Tested on two GPU clusters with different inter-node bandwidth (Clusters A & B).
- Centauri consistently achieves higher throughput and maintains strong scalability.

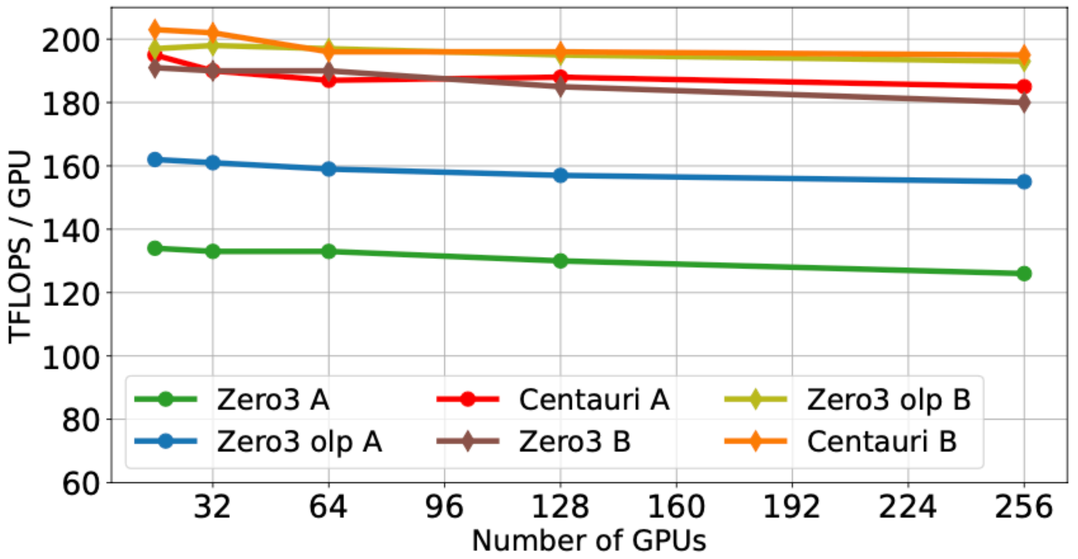


Figure 13. Scalability of FSDP/Zero3 on Cluster A&B

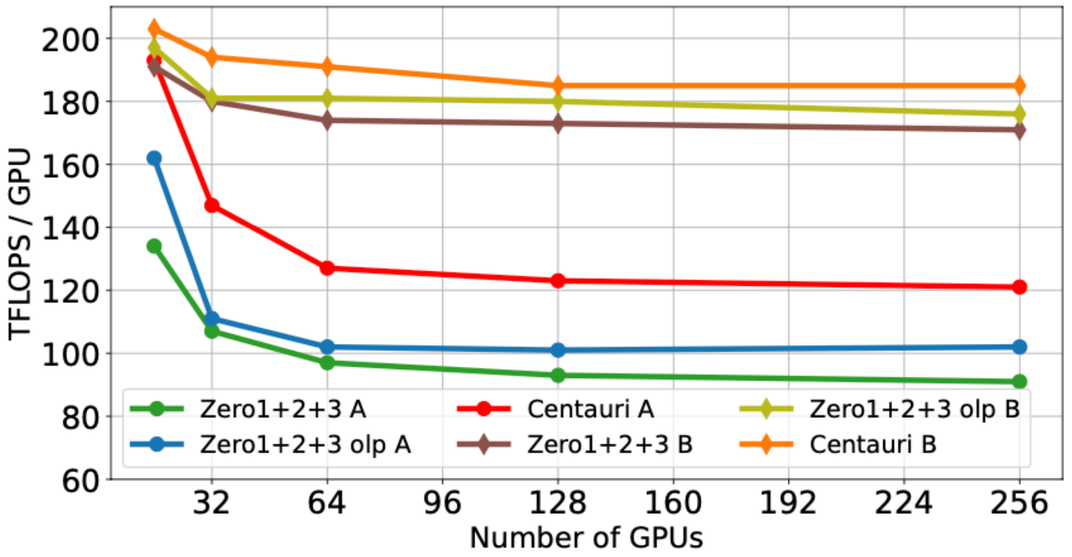


Figure 14. Scalability of FSDP16+DP on Cluster A&B

Comparison with Related Work

Centauri vs. Existing Solutions

- Previous works either too coarse or too fine-grained.
- Centauri uniquely integrates comprehensive partitioning and hierarchical scheduling.
- Enables broader exploration of overlap optimization space.

- Future Directions:
  - Explore communication optimization in parallel inference tasks.