FASOP: Fast yet Accurate Automated Search for Optimal Parallelization of Transformers on Heterogeneous GPU Clusters

Sunyeol Hwang, Eungyeong Lee, Hongseok Oh, Youngmin Yi
University of Seoul

Presenter
Dipak Acharya
University of North Texas
Denton TX

Data Parallelism (DP) and ZeRO optimizer

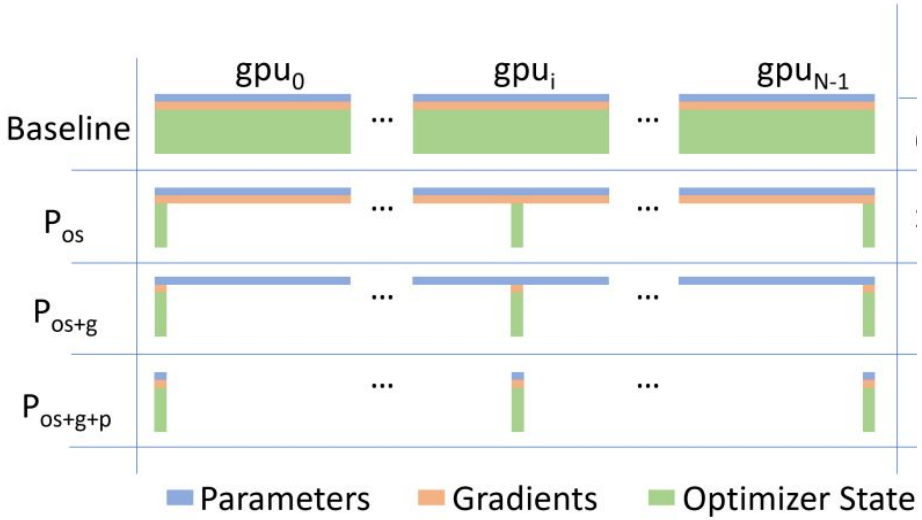
Data parallelism replicates the model among all available GPUs

Using different ZeRO stages, the amount of replication can be reduced

ZeRO 1 - Remove replication of Optimizer State

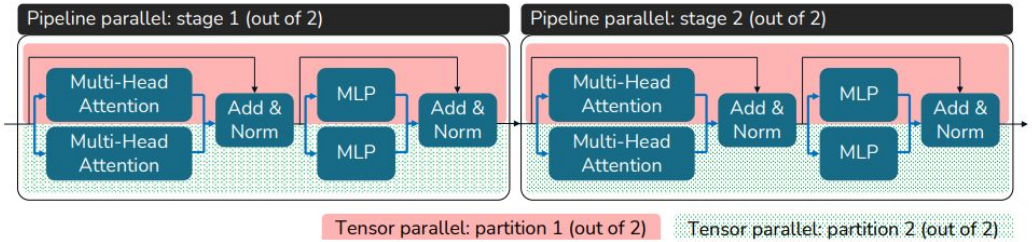
ZeRO 2 - Remove replication of the Gradients

ZeRO 3 - Remove replication of Parameters



FASOP: Fast yet Accurate Automated Search for Optimal Parallelization of Transformers on Heterogeneous GPU Clusters

Model Parallelism



Tensor Parallelism(TP)

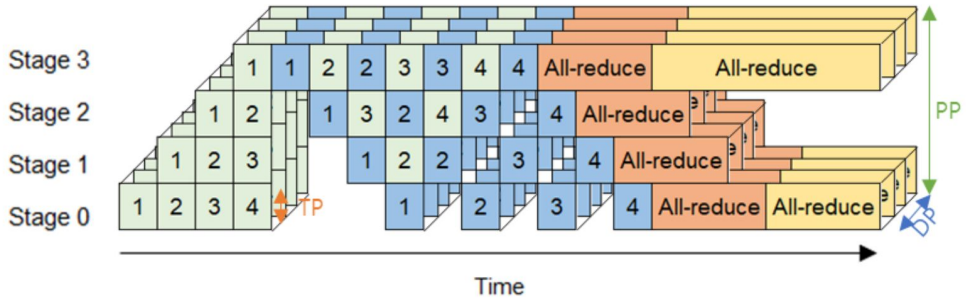
- Divides the model weights into multiple sub-matrices and performs the computation in different GPUs
- It requires much larger communication between each sub-layer operations and thus requires high bandwidth interconnect.

Pipeline Parallelism(PP)

- Divides the model into series of sub-models
- Each submodel receives the output from the previous submodel in forward pass
- Each submodel receives the output of next submodel in backward pass

FASOP: Fast yet Accurate Automated Search for Optimal Parallelization of Transformers on Heterogeneous GPU Clusters

3D parallelism



- 3D parallelism combines 3 types of parallelism (Data, Tensor and Pipeline)
- This can be observed as different types of parallelisms across 3 axes.

FASOP: Fast yet Accurate Automated Search for Optimal Parallelization of Transformers on Heterogeneous GPU Clusters

Parallel Training

Device Placement

- How to map the available GPUs across multiple nodes to the TP, DP and PP
- A rule of thumb is to map so that latency is minimized. Use high bandwidth communication for TP and DP and low bandwidth for PP

Automatic Parallelization Strategy Search

- This includes model partition for PP, the degrees of TP and DP, device assignment etc.
- Most common techniques include Dynamic programming and Integer Linear Programming (ILP)
- Eg. Pipedream (PP), Dapple (PP and DP), Piper(TP, DP, PP), Alpa (PP and TP)

FASOP: Fast yet Accurate Automated Search for Optimal Parallelization of Transformers on Heterogeneous GPU Clusters

Challenge

- How to find the optimal degrees for each parallelism for a given model and cluster configuration in 3D parallelism?
  - Each axes in 3D parallelism have different characteristics and trade-offs in terms of computation and communication
- Once PP degree is determined, how to determine model partitioning?
  - Model partitioning and mapping is NP-hard problem.
- How to enable optimal training configurations in heterogeneous device cluster configurations?
  - Heterogenous device cluster increase the search space for configurations to be very large and challenging for optimization

FASOP: Fast yet Accurate Automated Search for Optimal Parallelization of Transformers on Heterogeneous GPU Clusters

FASOP: Problem

Transformer model, Layers, GPUs	t, L, R
Global Batch size	β
Micro Batch sizes	B
Possible ways to partition model	P
Parallelism degrees	pp, tp, dp
Possible combs. of pp, dp, tp	D
Cluster configurations	C
Parallel strategy	S
Num. of Nodes, GPUs per nodes	N, M

Constraints

$tp \,.\, dp \,.\, pp = M.N$

$S = B \times P \times D \times C$

Cost functions

$f()$ - training time cost fun.

$g()$ - training cost($) fun.

$h()$ - Peak memory cost fun.

$\rho()$ - Memory of each GPU

Optimization Objective

$$\arg\min_{s \,\epsilon\, S} f(s; t) \quad s.t. \quad w \le u \quad s.t. \quad h(s, r; t) \le \rho(r) \text{ for } r \,\epsilon\, R$$

where

cluster usage cost: $w = f(s; t) \,.\, g(c)$

budget: u

FASOP: Fast yet Accurate Automated Search for Optimal Parallelization of Transformers on Heterogeneous GPU Clusters

FASOP: Search Space

For exhaustive search, the complexity is $O(N^{a+1} \cdot N! \cdot \sqrt{\beta} \cdot L^4 \cdot M)$

Reducing search space

- Limit device configuration to use 2 types of GPUs
- Apply heterogenous device placement heuristic that groups similar nodes and applies circular permutations
- Min-max stage algorithm for model partitioning

New complexity becomes: $O(N^4 \cdot \sqrt{\beta} \cdot L^2 \cdot M)$

FASOP: Fast yet Accurate Automated Search for Optimal Parallelization of Transformers on Heterogeneous GPU Clusters

Heterogeneous Device Placement Heuristic

- GPUs within a node are homogeneous
  - If not, performance may be limited by the slower GPU in the node

- Nodes with same GPUs must be placed together
  - If not, DP groups or TP groups formed by heterogeneous GPUs will result in performance bottleneck
  - The orders of node placement also leads to circular permutation

- Limit the TP degree to be smaller than or equal to number of GPUs per node
  - TP requires high communication; communication across nodes is inefficient

FASOP: Fast yet Accurate Automated Search for Optimal Parallelization of Transformers on Heterogeneous GPU Clusters

Model partitioning Heuristic

Min-max stage:

First partition the model evenly, then iteratively move layer from the stage with maximum latency to stage with the minimum latency.

The computation and communication times are calculated based on profiling results of the corresponding GPU.

Training Time Estimation

Simulation-based estimation:

Training time of 1F1B pipeline is simulated with two double-nested loops whose complexity is: $O(mb \times (pp + pp))$, where mb is the number of micro-batches.

All-reduce estimation with bandwidth sharing:

All-reduce communication is estimated based on ring topology.

Based on the degrees of TP and DP, multiple rings can be formed within a node, sharing the total effective bandwidth.

FASOP takes bandwidth sharing into account to estimate the all-reduce time.

Reducing input and output embedding layers

In PP, the first and last stage has embedding layer, the gradient values calculated need to be synchronized. This causes inter-node communication which takes significant time and needs to be considered in training time estimation.

Evaluation: Experimental Setup

Cluster Configurations:

Cluster configurations includes both homogeneous and heterogeneous clusters in small and large variety

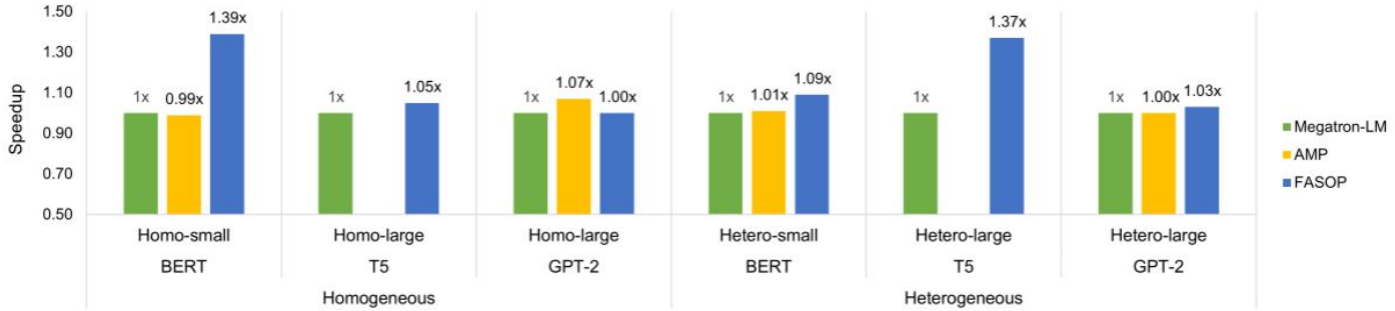
Cluster-ID	Cluster Configuration	#GPUs
Homo-small	4 × nodes (4 × A10 GPU each)	16
Homo-large	8 ×nodes (4 × A10 GPU each)	32
Hetero-small	1 × nodes (4 × A100 GPU each), 3 × nodes (4 × A10 GPU each)	16
Hetero-large	1 × nodes (4 × A100 GPU each), 7 × nodes (4 × A10 GPU each)	32

Baselines:
AMP: State of the art automated search for heterogeneous clusters

Megatron - LM: Large scale training frameworks for LLMs

Models:
BERT 345M (L = 24, H = 1024)
T5 1.3B (L = 48, H = 1024)
GPT-2 (L = 48, H = 1600)

Evaluation: Training Throughput



In the homogeneous cluster, FASOP outperformed Megatron-LM by 1.05× to 1.39× for BERT and T5 models

In the heterogeneous cluster, FASOP delivered a speed increase of between 1.03× to 1.37× over Megatron-LM

The search time in AMP for a heterogeneous cluster was 38s and 19,086s for BERT and GPT-2, respectively. The search time in FASOP for a heterogeneous cluster was 3.4, 156, and 13 seconds

FASOP: Fast yet Accurate Automated Search for Optimal Parallelization of Transformers on Heterogeneous GPU Clusters

Evaluation: Prediction Error

The overall estimation error (MAE) shows FASOP is slightly better than AMP

MAE	FASOP	AMP
BERT	28.6%	**27.9%**
T5	**29.6%**	-
GPT-2	**8.7%**	20.3%

When comparing detailed estimation error for pipeline latency, DP all-reduce latency and embedding all-reduce latency, we can see FASOP is a lot better than AMP

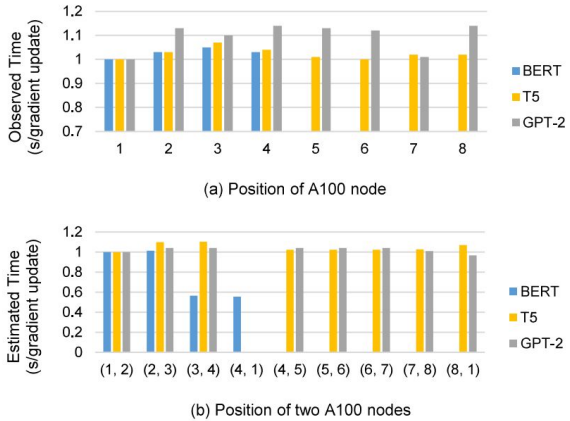
Model	**Cluster-ID**	**Method**	**PP**	**DP AR**	**Emb.**
GPT-2 1.5B	Hetero-large	FASOP	7.60%	12.35%	11.40%
		AMP	14.24%	45.69%	-
	Homo-large	FASOP	14.36%	5.26%	2.81%
		AMP	38.65%	6.55%	-
BERT 345M	Hetero-small	FASOP	29.5%	12.39%	11.38%
		AMP	33.72%	30.11%	-
	Homo-small	FASOP	21.17%	11.61%	3.16%
		AMP	27.10%	12.09%	-

Evaluation: Bandwidth Sharing

Model	Cluster-ID	(tp, dp, pp)	r	FASOP	AMP
GPT-2	Homo-large	(2, 2, 8)	2	3.50%	57.41%
		(4, 2, 4)	4	0.24%	51.81%
BERT	Homo-small	(2, 2, 4)	2	8.08%	57.73%
		(2, 4, 2)	2	11.49%	379.40%

For the four configurations where bandwidth sharing occurs, AMP had an average error of **137%** while FASOP had an average error of only **5.8%**

Evaluation: Device Placement and Model Partitioning



Model	(tp, dp, pp)	Minmax-Stage algorithm	Dynamic Programming
BERT	(1, 4, 4)	[14, 5, 5, 2]	[14, 5, 5, 2]
T5	(1, 4, 8)	[15, 10, 5, 4, 4, 4, 4, 4]	[15, 10, 5, 4, 4, 4, 4, 4]
GPT-2	(1, 2, 16)	[5, 4, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]	[5, 4, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]

Result with different device placement support FASOP's heuristic of placing faster nodes in the beginning or end

Comparing the model partitioning determined by min-max stage algorithm and dynamic programming are identical

FASOP: Fast yet Accurate Automated Search for Optimal Parallelization of Transformers on Heterogeneous GPU Clusters

Evaluation: Optimal device configurations

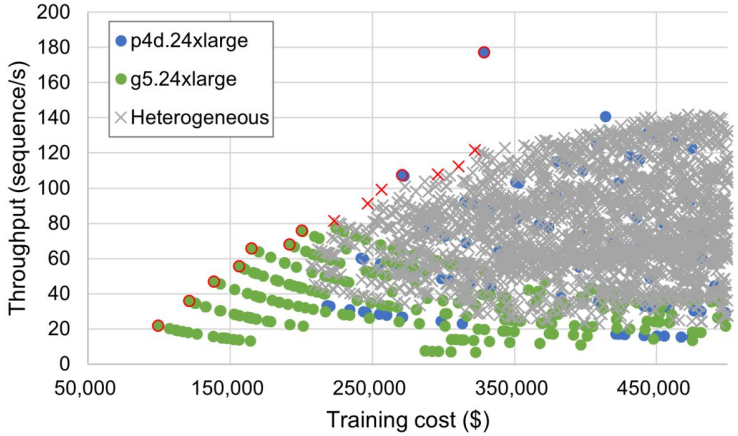
Evaluation with AWS **p4d.24xlarge** (8 x A100) or AWS **g5.24xlarge** (8 x A10) instances

The best throughput of 177.2 sequences/s was achieved by using 8 p4d.24xlarge nodes with DP only strategy

- This is because, the model fits in 40GB memory of A100s

The second to fourth pareto-optimal models are all heterogenous

- If model cannot fit into single GPU, heterogenous clusters will be pareto optimal



Cluster information	(tp, dp, pp)	Throughput (sequence/s)	Pre-train cost ($)
8 p4d	(1, 64, 1)	177.2	328,762
3 p4d, 8 g5	(1, 4, 22)	121.7	322,379
3 p4d, 6 g5	(1, 8, 9)	112.1	311,055
2 p4d, 8 g5	(1, 4, 20)	107.8	296,188
4 p4d	(1, 32, 1)	107.5	270,889

Thank You

Questions