Thorough Characterization and Analysis of Large Transformer Model Training At-Scale

SCOTT CHENG, The Pennsylvania State University, USA
JUN-LIANG LIN, The Pennsylvania State University, USA
MURALI EMANI, Argonne National Laboratory, USA
SIDDHISANKET RASKAR, Argonne National Laboratory, USA
SAM FOREMAN, Argonne National Laboratory, USA
ZHEN XIE, Binghamton University, USA
VENKATRAM VISHWANATH, Argonne National Laboratory, USA
MAHMUT T. KANDEMIR, The Pennsylvania State University, USA

Proc. ACM Meas. Anal. Comput. Syst., Vol. 8, No. 1, Article 8.
Publication date: March 2024.

Presenter
Dipak Acharya
University of North Texas
Denton TX

Transformer Model

Model parameters: $\mathcal{P}$

Batch size: b

Hidden dimensions: d

Layers: l

Attention heads: a

Input sequence length: s

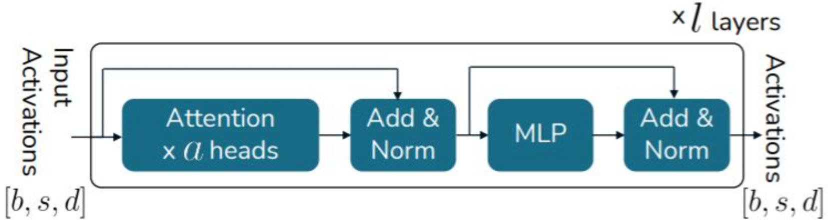
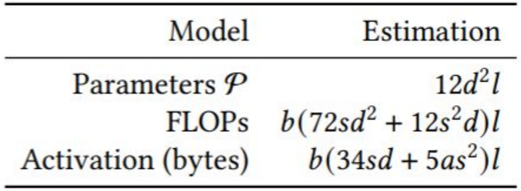


Fig. 4. Components of a transformer layer.

There might be small variations based on architecture

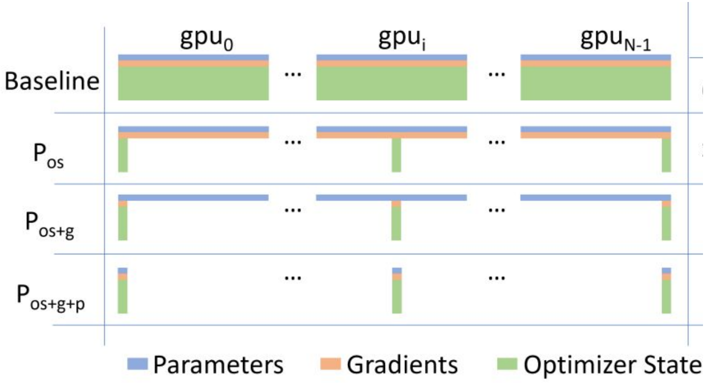
For eg. GPT employs masked attention, whereas BERT enables visibility to the whole input sequence

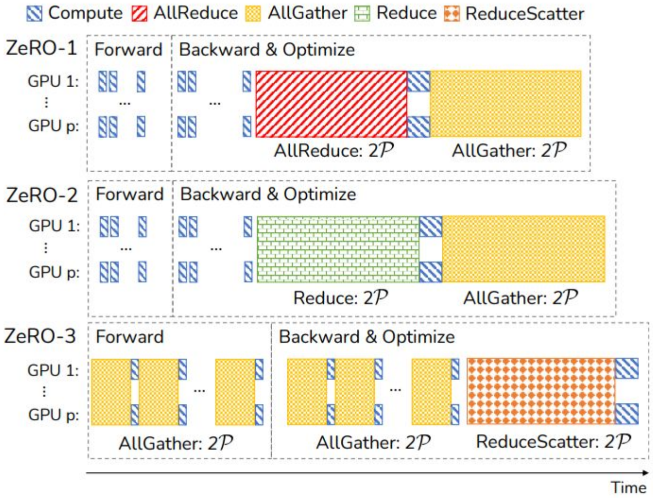


Model	Estimation
Parameters $\mathcal{P}$	$12d^2l$
FLOPs	$b(72sd^2 + 12s^2d)l$
Activation (bytes)	$b(34sd + 5as^2)l$

Estimation for transformer attributes

Data Parallelism (ZeRO Optimizer)





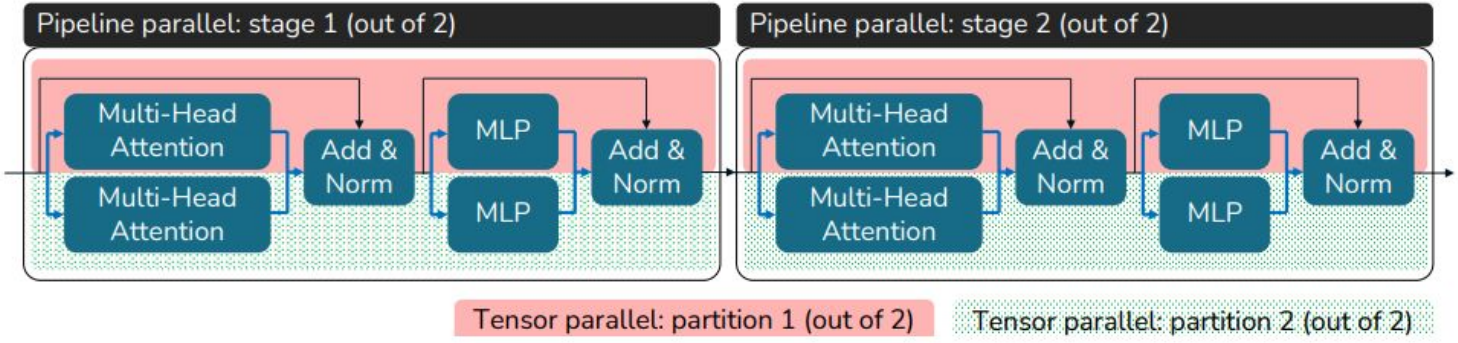
ZeRO 1 - Remove replication of Optimizer State

ZeRO 2 - Remove replication of the Gradients

ZeRO 3 - Remove replication of Parameters

Compute and communication patterns in different ZeRO stages

Model Parallelism


Pipeline parallel: stage 1 (out of 2)
Pipeline parallel: stage 2 (out of 2)
Multi-Head Attention
Multi-Head Attention
Add & Norm
MLP
MLP
Add & Norm
Multi-Head Attention
Multi-Head Attention
Add & Norm
MLP
MLP
Add & Norm
Tensor parallel: partition 1 (out of 2)
Tensor parallel: partition 2 (out of 2)

In model parallelism, the model is distributed on multiple devices for simultaneous computation.

Pipeline parallelism(PP) and Tensor parallelism(TP)

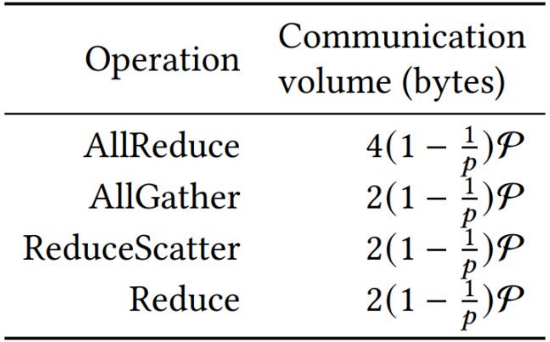
Collective communication

Various collective communication primitives require different volume of data communication

Eg. All reduce for p devices on $2\mathcal{P}$ parameters require $4(1- 1/p)\mathcal{P}$ communication per device

The actual implementation of the communication algorithm may affect the actual volume

AllReduce requires double the volume compared to AllGather and ReduceScatter as AllReduce can be decomposed into an AllGather and ReduceScatter operation



Operation	Communication volume (bytes)
AllReduce	$4(1-\frac{1}{p})\mathcal{P}$
AllGather	$2(1-\frac{1}{p})\mathcal{P}$
ReduceScatter	$2(1-\frac{1}{p})\mathcal{P}$
Reduce	$2(1-\frac{1}{p})\mathcal{P}$

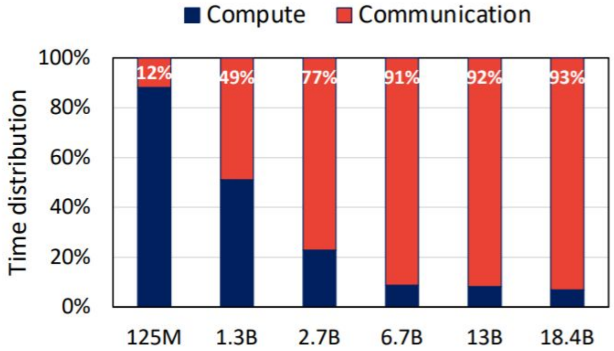
Estimated communication volume per device when communicating $\mathcal{P}$ bytes on p devices

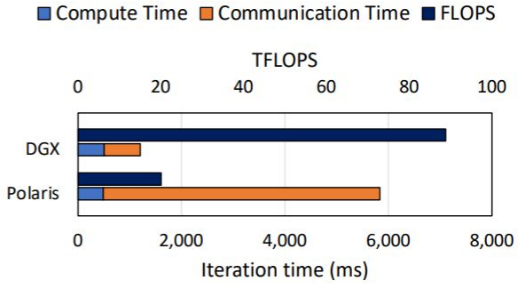
Distributed Training of Transformers

Parallel training requires the weights and activations to be communicated throughout the system, so the system network bandwidth is very important

Based on the system bandwidth, the communication time will be significantly impacted and consequently the overall iteration time

Widely adopted metrics such as FLOPS do not fully reflect the actual training throughput of the system.

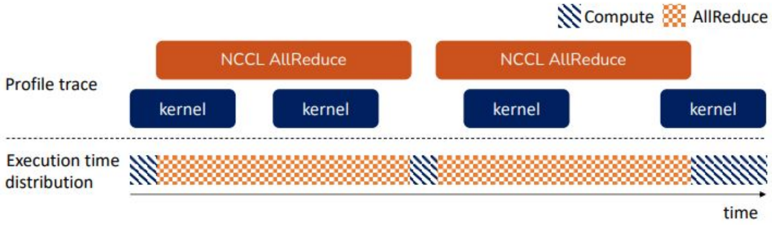




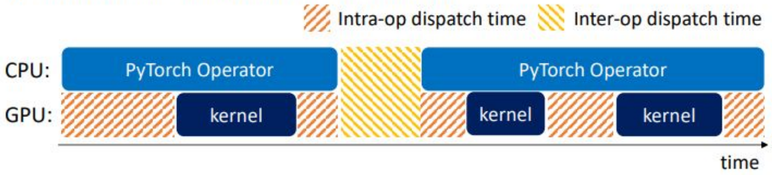
Evaluation Methodology

In case of overlapping compute and communication kernels, communication kernels are represented first and then the compute kernel.

Compute time is classified into intra-operator and inter-operator dispatch time, and kernel execution time.



(a) Converting a profile trace with overlapping compute and communication kernels into the execution time distribution.



(b) Dissect compute time into kernel execution, intra-op, and inter-op dispatch time.

Scalability analysis

Speedup formulation

$$\text{speedup}_{8\to 16} = \frac{t_{\text{iter.}}^{(8)}}{t_{\text{iter.}}^{(16)}} = \frac{t_{\text{compute}}^{(8)} + t_{\text{comm.}}^{(8)}}{t_{\text{compute}}^{(16)} + t_{\text{comm.}}^{(16)}}$$

Multi-node scaling efficiency:

$$\text{eff}_{g\to n} = \frac{t_{\text{comm.}}^{(g)}}{t_{\text{comm.}}^{(n)}}$$

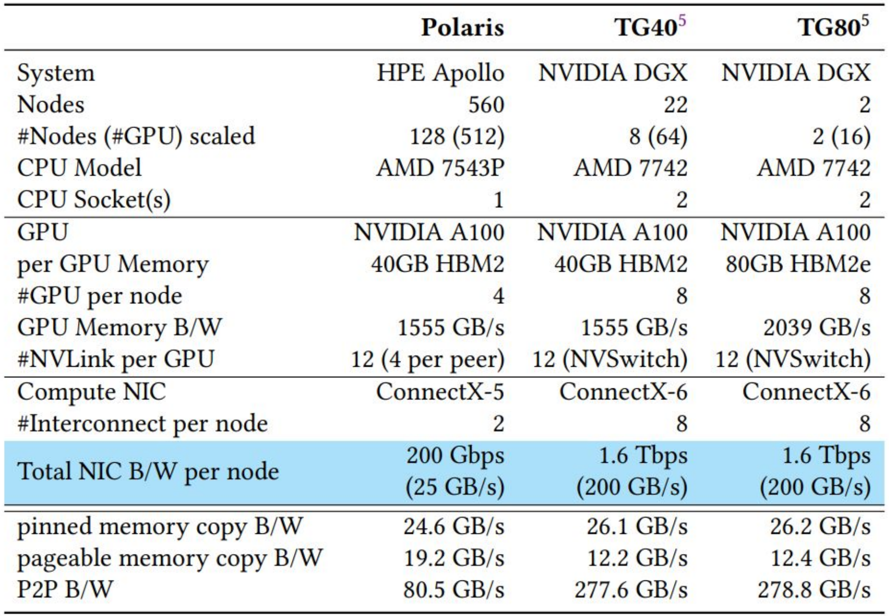
End-to-end speedup from g to n GPUs

$$\text{speedup}_{g\to n} = \frac{t_{\text{iter.}}^{(g)}}{t_{\text{iter.}}^{(n)}} = \frac{t_{\text{compute}}^{(g)} + t_{\text{comm.}}^{(g)}}{t_{\text{compute}}^{(n)} + t_{\text{comm.}}^{(n)}} = \frac{r_{\text{compute}}^{(g)} + r_{\text{comm.}}^{(g)}}{r_{\text{compute}}^{(g)} + r_{\text{comm.}}^{(g)}/\text{eff}_{g\to n}}$$

$$r_{\text{compute}}^{(n)} = t_{\text{compute}}^{(n)}/t_{\text{iter}}^{(n)} \qquad r_{\text{comm.}}^{(n)} = t_{\text{comm.}}^{(n)}/t_{\text{iter}}^{(n)}$$

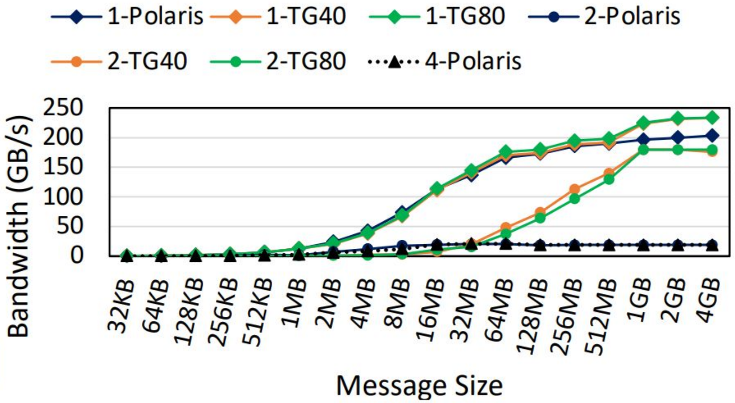
The **compute and communication time distribution** per iteration and the **network efficiency** is enough for scalability analysis

Experimental Setup



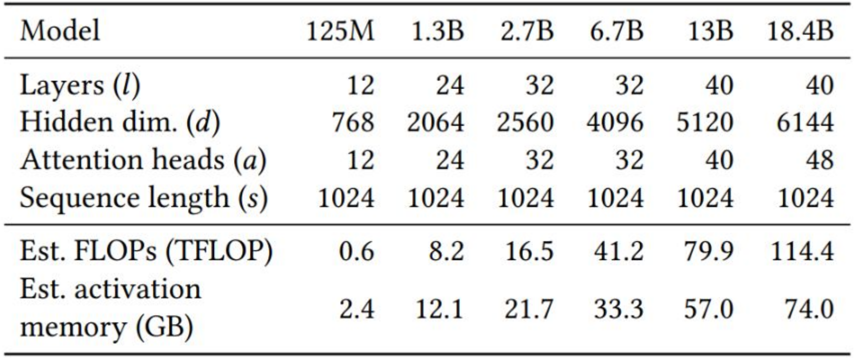
	Polaris	TG40[5]	TG80[5]
System	HPE Apollo	NVIDIA DGX	NVIDIA DGX
Nodes	560	22	2
#Nodes (#GPU) scaled	128 (512)	8 (64)	2 (16)
CPU Model	AMD 7543P	AMD 7742	AMD 7742
CPU Socket(s)	1	2	2
GPU	NVIDIA A100	NVIDIA A100	NVIDIA A100
per GPU Memory	40GB HBM2	40GB HBM2	80GB HBM2e
#GPU per node	4	8	8
GPU Memory B/W	1555 GB/s	1555 GB/s	2039 GB/s
#NVLink per GPU	12 (4 per peer)	12 (NVSwitch)	12 (NVSwitch)
Compute NIC	ConnectX-5	ConnectX-6	ConnectX-6
#Interconnect per node	2	8	8
Total NIC B/W per node	200 Gbps (25 GB/s)	1.6 Tbps (200 GB/s)	1.6 Tbps (200 GB/s)
pinned memory copy B/W	24.6 GB/s	26.1 GB/s	26.2 GB/s
pageable memory copy B/W	19.2 GB/s	12.2 GB/s	12.4 GB/s
P2P B/W	80.5 GB/s	277.6 GB/s	278.8 GB/s

Table 4. The GPU supercomputing systems evaluated in this study.



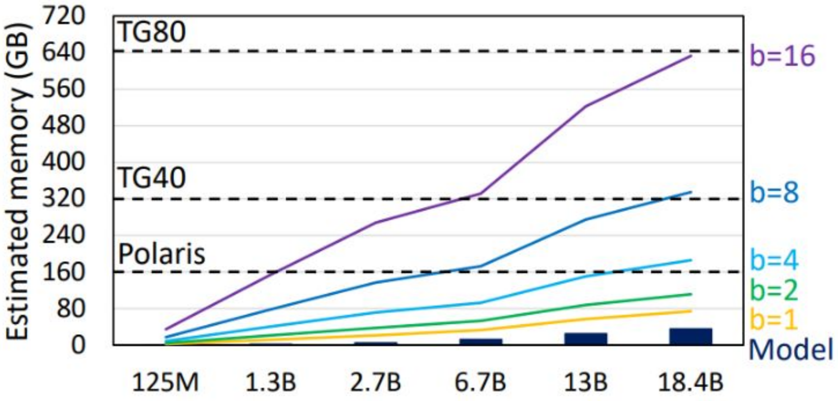
All reduce scales poorly with number of nodes for Polaris while the DGX systems maintain high bandwidth.

Model Setup



Model	125M	1.3B	2.7B	6.7B	13B	18.4B
Layers (l)	12	24	32	32	40	40
Hidden dim. (d)	768	2064	2560	4096	5120	6144
Attention heads (a)	12	24	32	32	40	48
Sequence length (s)	1024	1024	1024	1024	1024	1024
Est. FLOPs (TFLOP)	0.6	8.2	16.5	41.2	79.9	114.4
Est. activation memory (GB)	2.4	12.1	21.7	33.3	57.0	74.0

These model configurations of the GPT models were used for the evaluation



Projected memory consumption of the training under different batch sizes (b)

Transform characterization: Computation

- For small models, dispatch time dominates about 95% of compute time
- As model size grows, the proportion of compute kernel increases;
- Intra-op dispatch decreases from 78% to 14%

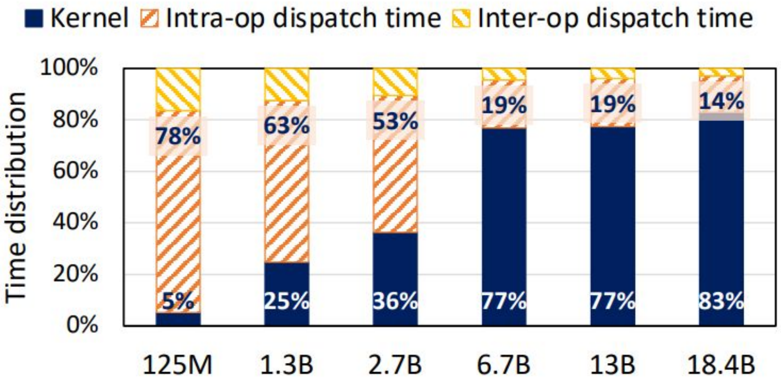


Fig. 10. Dissection of compute time per iteration.

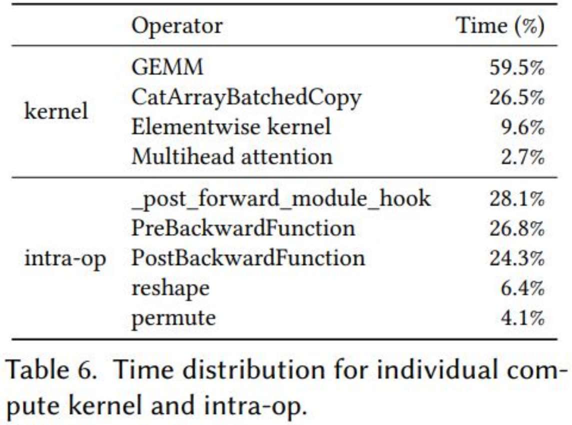
Transform characterization: Computation

- Computation is mostly dominated by GEMM and tensor concatenation
- Intra-op involves model parameter sharding, data transformation, reshaping, permuting etc.

- Based on this compute time can be estimated

  k_1 = distribution of compute time (83% on previous example)

  k_2 = transformer kernel distribution within kernel time (1 - 26.5% on example)



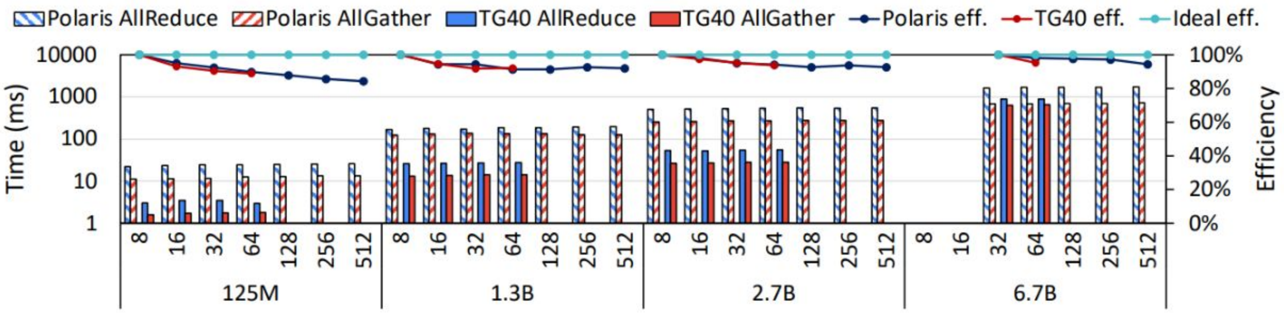
	Operator	Time (%)
kernel	GEMM	59.5%
	CatArrayBatchedCopy	26.5%
	Elementwise kernel	9.6%
	Multihead attention	2.7%
intra-op	_post_forward_module_hook	28.1%
	PreBackwardFunction	26.8%
	PostBackwardFunction	24.3%
	reshape	6.4%
	permute	4.1%

Table 6. Time distribution for individual compute kernel and intra-op.

$$t_{\text{compute-transformer}} = \frac{\text{Model FLOPs}}{\text{FLOPS}} = \frac{b(72sd^2 + 12s^2d)l}{\text{FLOPS}}$$

$$t_{\text{compute}} = \frac{t_{\text{compute-transformer}}}{k_1 \times k_2}$$

Communication: ZeRO-1



(a) AllReduce and AllGather operations in ZeRO-1.

For ZeRO-1 $\mathcal{P}$ Parameters results in $4\mathcal{P}$ bytes of communication per GPU.

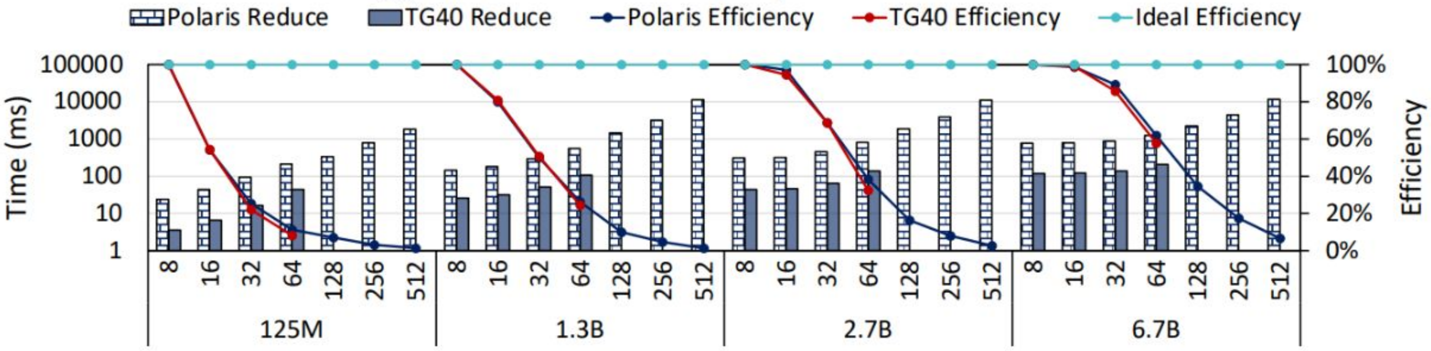
$$t^{(p)}_{\text{AllReduce}} = 4\mathcal{P} \ / \ (BW^{(8)}\text{eff}_{8\to p}) \approx 4 \times 12d^2 l \ / \ (BW^{(8)}\text{eff}_{8\to p}),$$

AllGather takes half of AllReduce time.

Communication time on TG40 is lower than Polaris (8x higher bandwidth)

As the number of GPU increases, the efficiency drops to ~89%

Communication: ZeRO-2



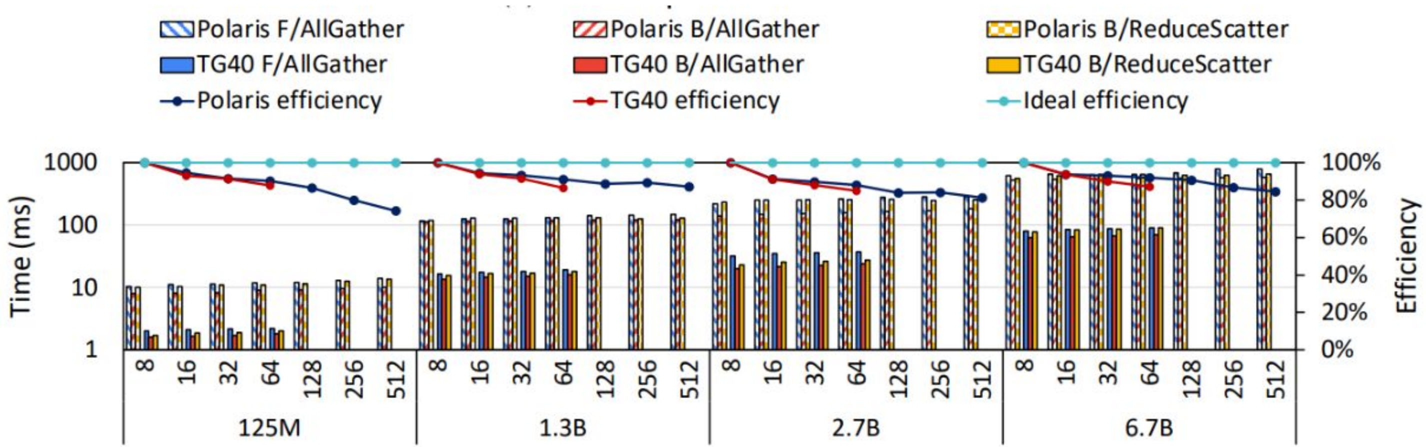
ZeRO-2 requires ReduceScatter in backward pass

ReduceScatter is decomposed into multiple reduce operations

Communication volume of Reduce operation is $2\mathcal{P}$ bytes and involves peer to peer communication

Hence there is sharp drop in efficiency as the number of nodes increases

Communication: ZeRO-3



(c) AllGather and ReduceScatter operations in ZeRO-3.

AllGather in forward and backward passes, ReduceScatter in backward passes all require $2\mathcal{P}$ bytes

As the number of nodes increase, the communication efficiency drop to ~80%

DP: ZeRO-1

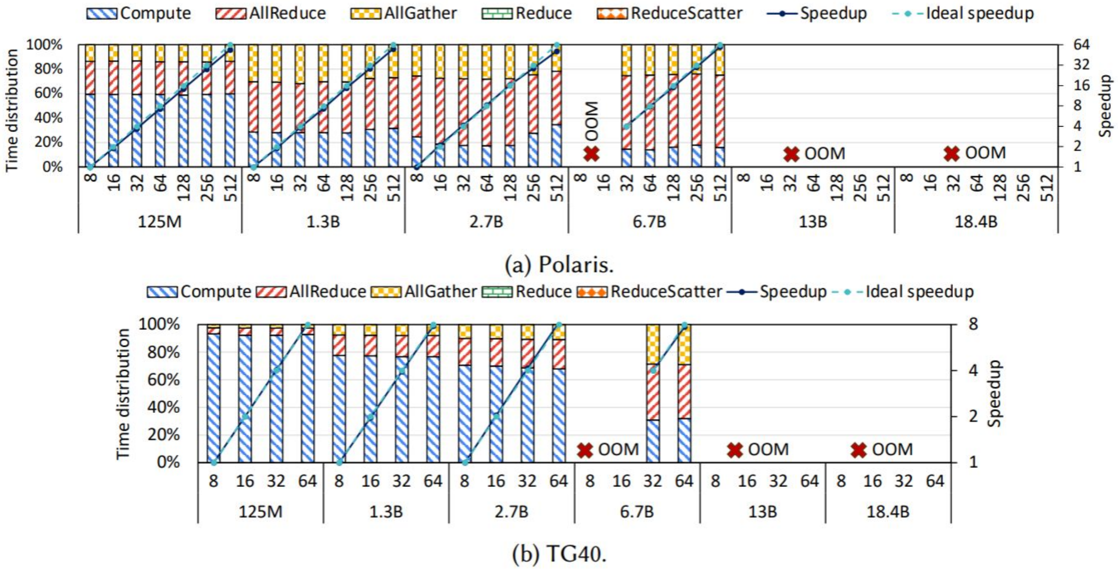
The distribution of compute and communication remains at a fixed ratio under the same model size as the number of GPUs increases

AllReduce and AllGather communications exhibit near-linear scalability, with efficiency dropping slightly to 89% when scaling to 512 GPU

ZeRO-1 can achieve a near-ideal speedup in the evaluation

Communication dominates as model size increases

For constant model size communication is very low in high bandwidth systems



(a) Polaris.

(b) TG40.

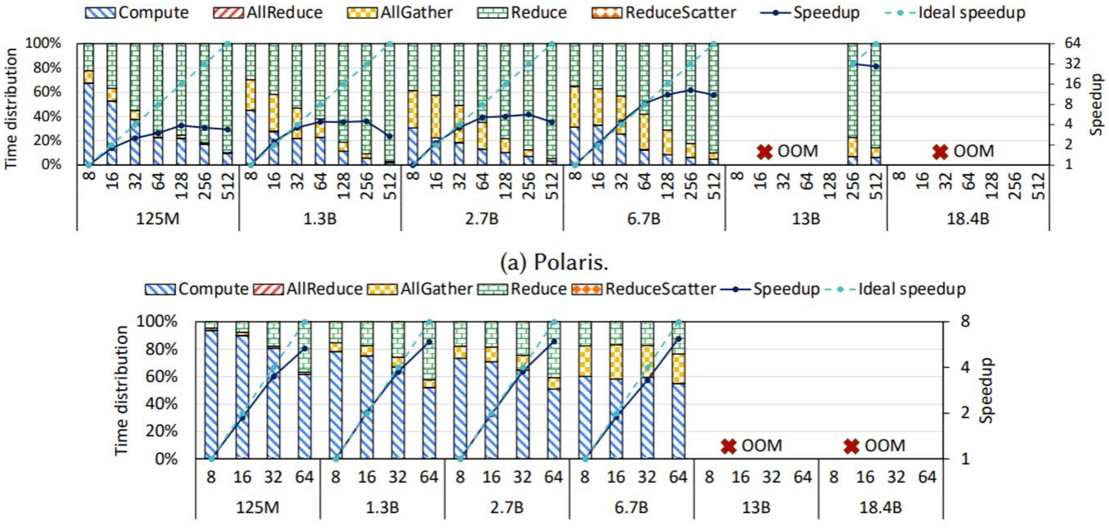
Fig. 12. End-to-end per-iteration training time distribution and speedup for different number of GPUs, model parameters using ZeRO-1 stage.

DP: ZeRO-2

Reduce communication does not scale well across multiple GPUs on Polaris whereas scales better in TG40 even if Reduce efficiency is same in both

Compared to Polaris Reduce communication time is 7.16 times shorter in TG40 (8x bandwidth)

As a result, the communication contributes less to the overall training, and thus, the TG40 iteration time is 62.29% faster than Polaris, on average



(a) Polaris.

(b) TG40.

Fig. 13. End-to-end per-iteration training time distribution and speedup for different number of GPUs, model parameters using ZeRO-2 stage.

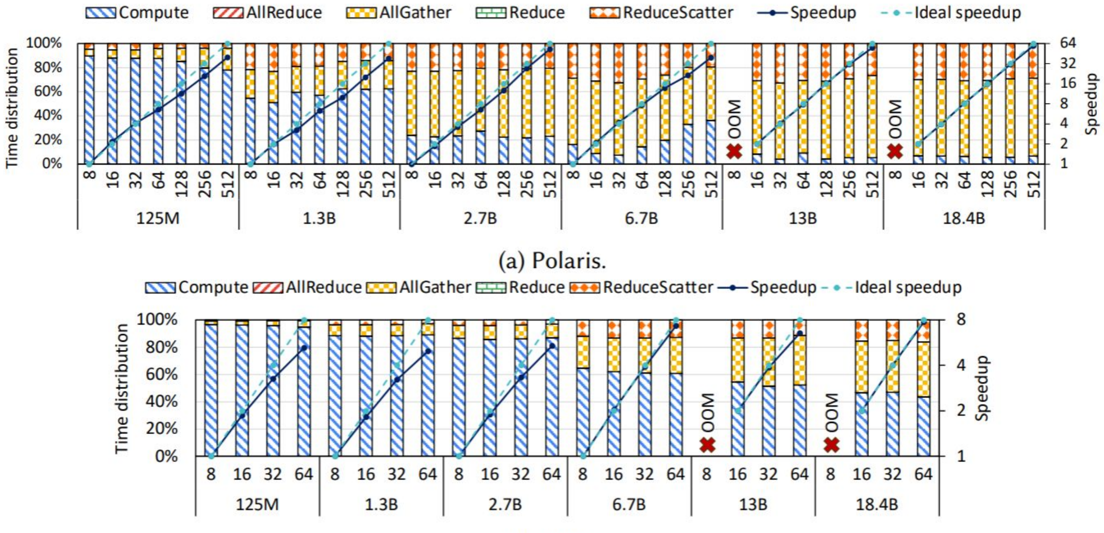
DP: ZeRO-3

ReduceScatter and AllGather also scale almost linearly, leading to a near-linear training scalability

Communication time is significantly reduced in high bandwidth systems compared to low bandwidth system.

Thus, high system bandwidth provides significant edge in communication scaling.

But linear scaling is possible regardless the bandwidth is high or low



(a) Polaris.

(b) TG40.

Fig. 14. End-to-end per-iteration training time distribution and speedup for different number of GPUs, model parameters using ZeRO-3 stage.

ZeRO Insights

- As model size increases, communication volume increases
- Higher communication volume leads to higher communication distribution in training time. This reduces the non-overlapping computation time
- Hence **Large model** makes the overall training throughput further bottlenecked by **network bandwidth**

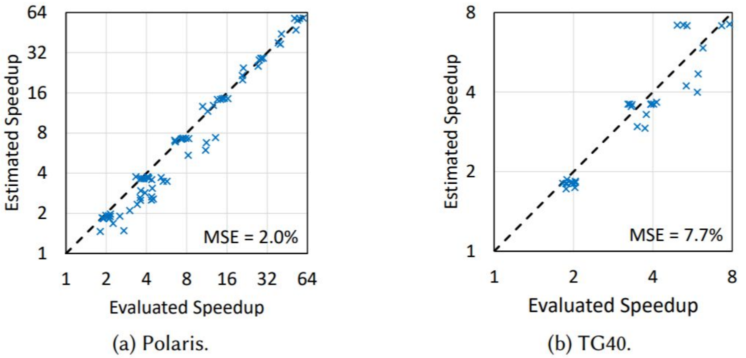
Increasing the ZeRO stages while keeping the same number of GPUs and the same model size, more data will be sharded across multiple GPUs. Therefore, with a higher ZeRO stage, the system is able to accommodate a larger model, such as the 13B and 18.4B models, without encountering any OOM issues

Speedup Analysis

Figure shows the MSE between evaluated speedup and estimated speedup based on the time evaluated by the analytical model.

The training time includes both computation and communication time.

Figure shows that the result aligns with the evaluations and can predict precise per-iteration time.



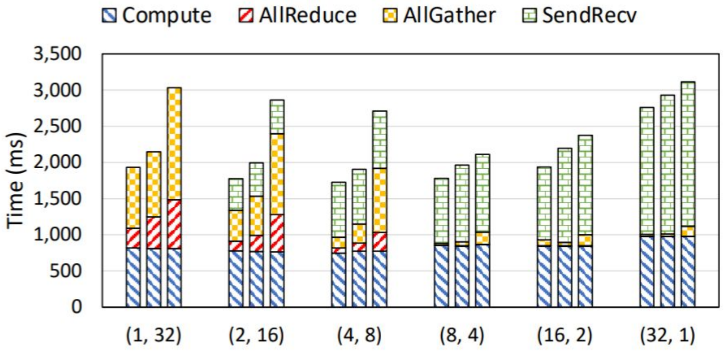
(a) Polaris.

(b) TG40.

Fig. 15. Estimated speedup compared to evaluated speedup.

	TG40	Polaris
Training Time	1x	2.45x
Communication time	1x	7.35x
Communication time distribution	26.7%	69.02%

Model Parallelism



Order:
P2 - 25 GB/s
P1 - 12.5 GB/s
P0 - 5 GB/s

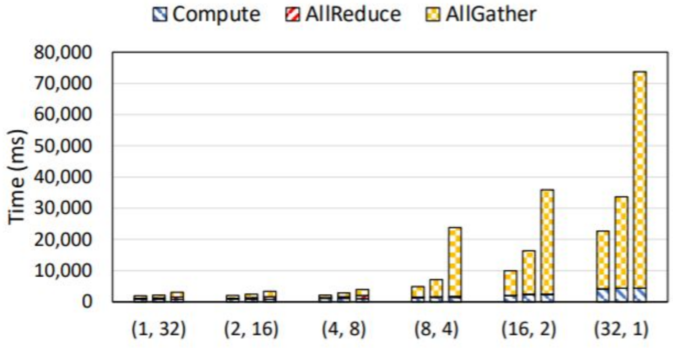
X-axis (#PP, #DP)

As #PP increases and #DP decreases, communication time first reduces and then increases as AllGather time decreases and SendRecv increases

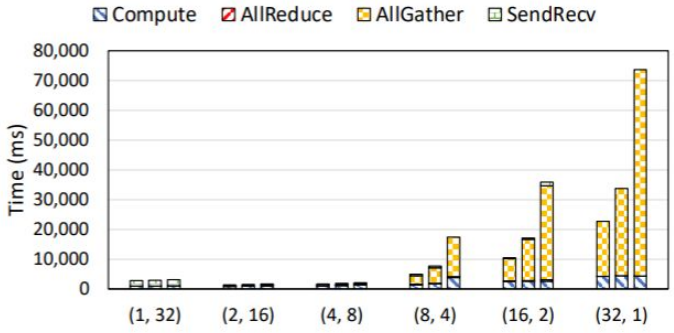
Setting #PP >= 8 will have DP done within single node, utilizing high bandwidth and significantly reducing the AllGather time

In low bandwidth(P0), when #PP increases from 4 to 8, AllGather time increases drastically as it requires inter-node AllGather.

Model Parallelism



(a) (#TP, #DP).



(b) (#TP, #PP).

Order:
P2 - 25 GB/s
P1 - 12.5 GB/s
P0 - 5 GB/s

As we increase #TP, the communication time increases, especially from 4 to 8 (Inter-node TP).

Once #TP exceeds the number of GPUs per node (e.g., 4), the communication time will increase rapidly, given that the inter-node Infiniband bandwidth is much lower than the intra-node NVLink bandwidth

Best speedup achieved when #TP is 4 and #PP is 8.

Set #TP to match number of GPUs per node and #PP smaller than the number of nodes.

Discussions

Bottom up analysis

- Provides information on inter-op and intra-op dispatch times which may take up to 95% of compute time in smaller models
- Provides impact of each parallel strategy on each component of training time as opposed to FLOPS which is used more commonly
- Provides the scaling effect and accurately predict training throughput in new hardware

Discussions

Network efficiency and throughput:

- Evaluations show the significant effect on scaling of the system based on network efficiency
- Having information about network efficiency can help make accurate scaling estimates

Data and Model Parallel Strategies

- DP scales better than model parallelism
- Degree of PP should be larger than number of nodes, while degree of TP should not exceed the number of GPU per node

Discussions

GuideLines for future designs

- It is crucial to focus on network bandwidth
- Systems with same FLOPS can have interconnect bandwidth varying by upto 8x. So upgrading the interconnection should be primary consideration.
- There are several ways to improve training throughput, eg. reduce the dispatch time or optimize the communication time

Limitations

- Evaluations only include transformer based LLMs
- Study presents high level classification of communication primitives, but lacks detailed report on each communication operation such as how many bytes of data communicated, how many individual communication operations etc.
- The study perform the analysis of computation involved in distributed transformer training but it doesn't consider any optimizations that can be done in operator level.

Thank You

Questions