Ladder-Residual: Parallelism-Aware Architecture for Accelerating Large Model Inference with Communication Overlapping

Muru Zhang*, Mayank Mishra*, Zhongzhu Zhou, William Brandon, Jue Wang, Yoon Kim, Jonathan Ragan-Kelley, Shuaiwen Leon Song, Ben Athiwaratkun, Tri Dao

Paper

Code

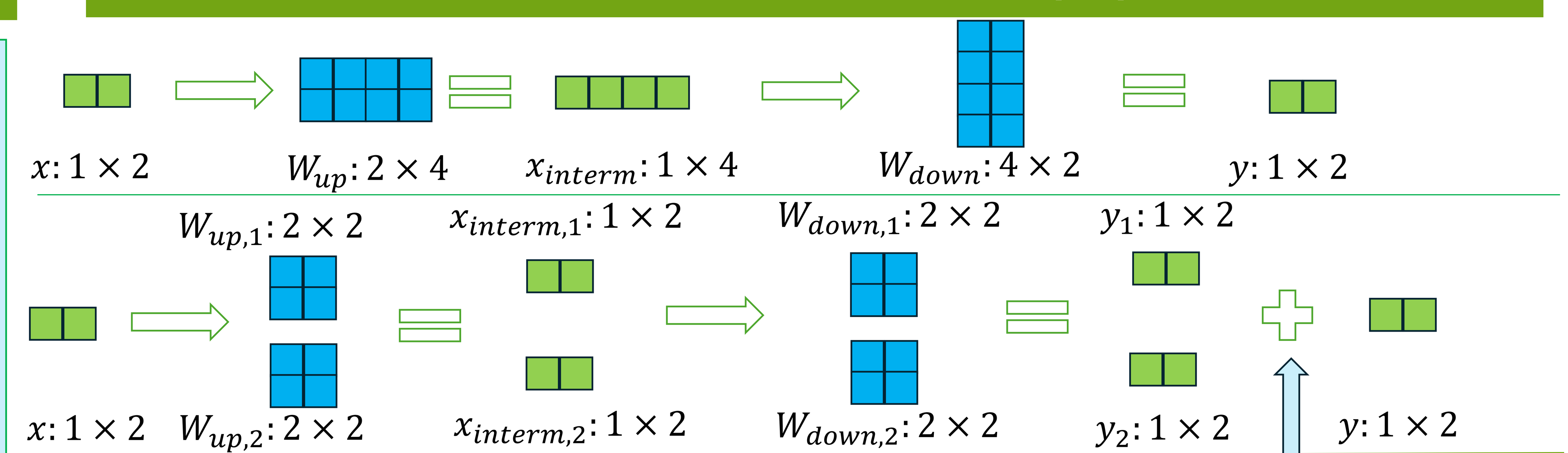
Overview

Background: Modern models are large, memory-intensive, and running them is slow.

Common practice: running models on multiple GPUs, with Tensor Paralellism (TP) being the most flexible/popular approach.

Challenge: Multi-GPU inference requires synchronization between devices, which can account for 38% of the latency for a 70B model running on 8 GPUs with TP.

What's Tensor Parallelism (TP)



Above diagram illustrates how TP parallelizes a sequence of two matrix multiplication onto two GPUs; the final summation requires an all-reduce communication.

$$x_i^* = h_i(x_{i-1})$$
$$x_i = \mathtt{AllReduce}(x_i^*) + x_{i-1}$$
$$x_{i+1}^* = h_{i+1}(x_i)$$
$$x_{i+1} = \mathtt{AllReduce}(x_{i+1}^*) + x_i$$

How does Ladder-Residual accelerate TP

Motivation: activation changes slowly within the model, modules aren't strongly sequentially dependent on each other.

Idea: Decouple the communication of x_i with the computation of h_{i+1} to overlap them.

$$x_i^* = h_i(x_{i-2})$$
$$x_i = \mathtt{AllReduce}(x_i^*) + x_{i-1}$$
$$x_{i+1}^* = h_{i+1}(x_{i-1})$$

Can overlap!

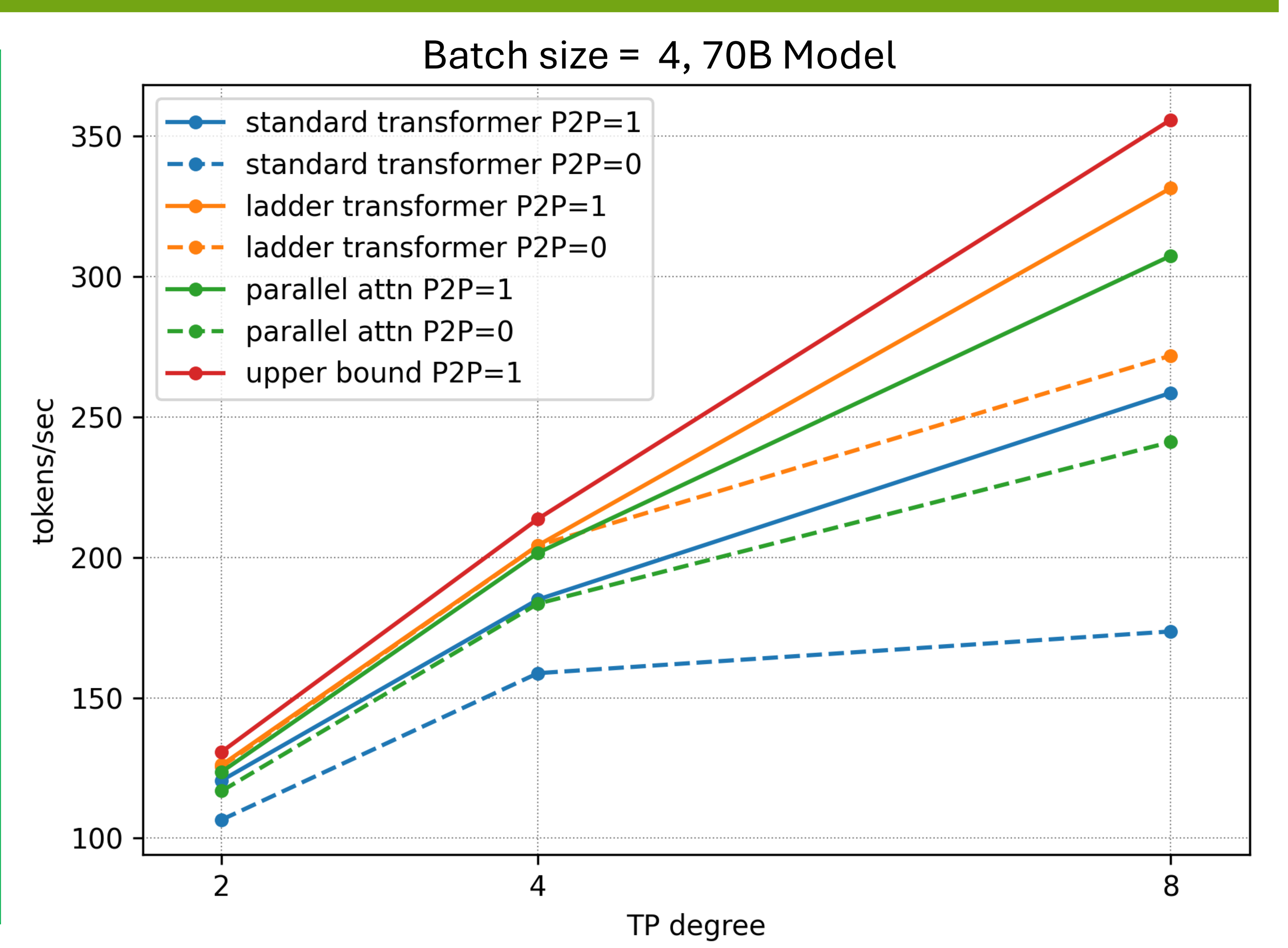
$$x_{i+1} = \mathtt{AllReduce}(x_{i+1}^*) + x_i$$

How Much Speedup does Ladder-Residual Offer

Benchmarking setup: H100 Cluster with NVLink interconnect (P2P=1); Llama architecture, 1024 prompt tokens, 512 generated tokens

Manually disabled P2P (P2P=0) communication to simulate case with no NVLink access

Parallel attn: parallelize attention and mlp within the same layer, effectively cut half of the communication as an alternative



Speedup vs. bsize	1	4	16	64
Standard	77.39	258.56	843.15	1940.99
Ladder	1.308x	1.282x	1.190x	1.155x
P2P Disabled				
Standard	51.66	173.62	546.68	1454.42
Ladder	1.599x	1.566x	1.351x	1.282x

Diminishing but consistent speedup when increasing the batch size

Model size	P2P disabled	P2P enabled
1B	1.39x	1.56x
3B	1.50x	1.57x
8B	1.40x	1.46x
34B	1.47x	1.44x
70B	1.59x	1.29x
176B	1.54x	1.35x
405B	1.57x	1.31x

Bsize=4, >=30% speedup across model sizes

Pre-training Ladder-Residual models from Scratch

Table 3. Performance of three architectures under two sizes, trained on FineWeb-edu for 100B tokens.

Model	ARC-C	ARC-E	HellaSwag	PIQA	SciQ	Winogrande	Average	Wikitext PPL
Standard-Transformer-1.2B	34.22	70.33	41.10	71.49	87.30	55.41	59.98	18.54
Parallel-Transformer-1.2B	30.46	67.97	40.35	71.16	87.40	55.17	58.75	18.95
Ladder-Transformer-1.2B	31.31	67.76	41.18	71.49	86.60	55.17	58.92	18.42
Standard-Transformer-3.5B	38.99	74.12	46.48	74.59	92.00	58.48	64.11	14.48
Parallel-Transformer-3.5B	38.48	73.02	45.55	73.67	90.00	57.46	63.03	14.96
Ladder-Transformer-3.5B	36.77	72.43	45.66	73.72	89.90	58.96	62.91	14.90

Model	ARC-C	ARC-E	HellaSwag	PIQA	SciQ	Winogrande	Average	Wikitext PPL	Tokens/sec
Standard-Transformer-1.2B	34.22	70.33	41.10	71.49	87.30	55.41	59.98	18.54	1008.29
Ladder-Transformer-1.5B	33.96	70.16	42.58	71.98	87.90	55.41	60.33	17.47	1277.66
Standard-Transformer-3.5B	38.99	74.12	46.48	74.59	92.00	58.48	64.11	14.48	949.6
Ladder-Transformer-4.5B	40.96	75.00	46.81	73.99	90.80	57.70	64.21	14.05	1217.71

Ladder-Transformer can achieve higher throughput with better performance compare with the baseline

Adapting a Pre-trained Model into Ladder-Residual

Model	MMLU	ARC-C	OBQA	HS	TQ	GSM	HE+	IE	AE	Average
Llama-3.1-8B-Instruct	68.14	60.32	43.00	80.04	36.84	84.99	60.40	52.57	18.69	56.11
Hybrid-Ladder-8B-16L-zeroshot	63.19	56.57	42.60	77.70	35.50	10.54	30.50	46.25	11.99	41.65
Hybrid-Ladder-8B-16L-retrained	67.33	59.98	45.00	79.05	37.58	86.81	60.51	59.76	22.43	57.61
Hybrid-Ladder-8B-20L-retrained	62.31	59.90	42.60	77.49	36.72	76.19	48.80	59.05	21.72	53.86

OBQA: OpenBookQA, HS: HellaSwag, TQ: TruthfulQA, HE+: HumanEval+, IE: IFEval, AE: AlpacaEval 2.0

We took Llama-3.1-8B-Instruct, adapted x of its layers (denoted as xL) into Ladder-Residual architecture, then fine-tune with 1.6B tokens to heal the distribution shift. The result model has < 1 point of accuracy gap on every task.