parallel pipeline vs data parallel

Experiment	Dataset	Backend	GPUs	Batch size	Learning rate	Top-1 acc (%)	Throughput	Speed up
Pipeline-2gpu	CIFAR10	MobilenetV2	2	64(4 chunks)	0.005	95.89±0.07	228.57/s	0.607×
Dataparallel-2gpu	CIFAR10	MobilenetV2	2	64	0.005	95.83±0.04	376.47/s	1×
Pipeline-4gpu	CIFAR10	MobilenetV2	4	256(4 chunks)	0.02	96.03±0.14	400.30/s	1.07×
Pipeline-4gpu	CIFAR10	MobilenetV2	4	256(8 chunks)	0.02	96.07±0.05	397.30/s	1.06×
Dataparallel-4gpu	CIFAR10	MobilenetV2	4	256	0.02	95.94±0.09	627.22/s	1.66×
Pipeline-2gpu	RTE	Roberta	2	32(4 chunks)	2e-5	78.33±0.31	61.53/s	0.80×
Dataparallel-2gpu	RTE	Roberta	2	32	2e-5	79.4±0.57	76.19/s	1×
Pipeline-4gpu	RTE	Roberta	4	64(4 chunks)	4e-5	78.17±0.44	106.40/s	1.40×
Pipeline-4gpu	RTE	Roberta	4	64(2 chunks)	4e-5	78.15±0.22	96.40/s	1.01×
Dataparallel-4gpu	RTE	Roberta	4	64	4e-5	79.4±0.41	95.53/s	1.25×

I could use bigger batch size for nlp tasks, but this will hurt the performance of the model.

And all separations are separated by the rule of server and client, which slow down the model parallel method.

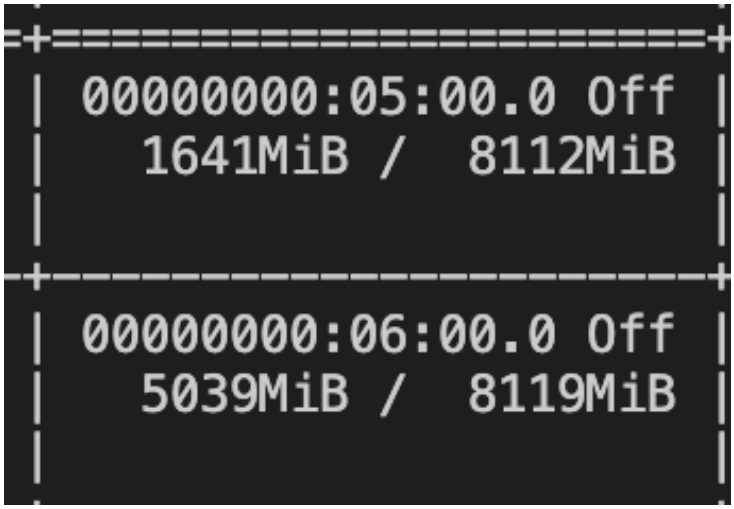
Pipeline parallel could hardly get the same speed as data parallel!

Experiment	GPUs	Batch size	Learning rate	Top-1 error (%)	Throughput	Speed up
reference-256 ([paper][])	8	256	0.1	22.08±0.06	N/A	N/A
reference-8k ([paper][])	256	8K	3.2	22.36±0.09	N/A	N/A
dataparallel-256	2	256	0.1	22.02±0.11	180.344/s	1×
dataparallel-1k	8	1K	0.4	22.04±0.24	606.916/s	3.365×
dataparallel-4k	8	4K	1.6	Out of memory	N/A	N/A
pipeline-256	2	256	0.1	21.99±0.13	117.432/s	0.651×
pipeline-1k	8	1K	0.4	22.24±0.19	294.739/s	1.634×
pipeline-4k	8	4K	1.6	22.13±0.09	378.746/s	2.100×

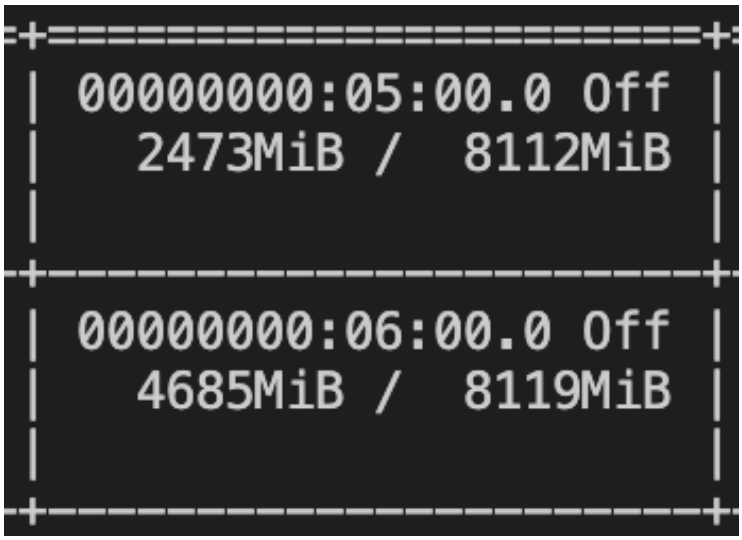
It is always the half speed of data-parallel with the same device count.

why does mobilenetv2 perform badly at parallel pipeline?

you could see that even if I put the first 2 and last 2 layers at client GPU. The memory partition is shown below.



But at Roberta, if I put the first 2 and last 2 layers at client GPU.



I could prove this by separately spreading the MobileNetV2. BUt it violates the principles of server and client

The result shows below.

If I put if I put the first 4 and last 3 layer at the client GPU, the result is

Experiment	Dataset	Backend	GPUs	Batch size	Learning rate	Throughput	Speed up
Pipeline-2gpu	CIFAR10	MobilenetV2	2	64(4 chunks)	0.005	228.57/s	0.851×

Altogether Ablation Study

Dataset	Backend	Batchsize	activation memory size(al together)	Compression method	compression ratio	Validation acc(in cola is Matthew)	Bandwidth
CIFAR10	MobileNetV2	256(8 chunks)	[256,32,112,112] [256,1280,7,7]f1l1	Sort Quantization 16bits	0.5	96.0%±0.13%	160.73G/s 25.94G/s
CIFAR10	MobileNetV2	256(8 chunks)	[256,32,112,112] [256,1280,7,7]	Sort Quantization 12bits	0.375	95.9%±0.14%	131.41G/s 17.97G/s
CIFAR10	MobileNetV2	256(8 chunks)	[256,32,112,112] [256,1280,7,7]	Sort Quantization 8bits	0.25	95.7%±0.03%	89.51G/s 13.03G/s
CIFAR10	MobileNetV2	256(8 chunks)	[256,32,112,112] [256,1280,7,7]	Sort Quantization 4bits	0.125	87.1%	37.13G/s 6.51G/s
RTE	Roberta	32(4 chunks)	[32,128,768], [32,128,768]f2l2	Sort Quantization 16bits	0.5	79.6%±0.18%	11.04G/s
RTE	Roberta	32	[32,128,768],[32,128,768]	Sort Quantization 12bits	0.375	79.6%±0.20%	8.19G/s
RTE	Roberta	32	[32,128,768],[32,128,768]	Sort Quantization 8bits	0.25	79.4%±0.21%	5.37GB/s
RTE	Roberta	32	[32,128,768],[32,128,768]	Sort Quantization 4bits	0.125	52.2%	2.774G/s
Cola	Roberta	32	[32,128,768], [32,128,768]f2l2	Sort Quantization 16bits	0.5	64.5±0.48	11.33G/s
Cola	Roberta	32	[32,128,768],[32,128,768]	Sort Quantization 12bits	0.375	63.93±0.22	7.96G/s
Cola	Roberta	32	[32,128,768],[32,128,768]	Sort Quantization 8bits	0.25	63.20±0.12	5.91GN/s
Cola	Roberta	32	[32,128,768],[32,128,768]	Sort Quantization 4bits	0.125	0	2.65G/s

Bandiwidth is calculated by recv_bytes / recv_time

Also, bandwidth has a linear relationship with recv size.