The Alan Turing Institute

Benchmarking AI applications on GPUs

Tomas Lazauskas, David Llewellyn-Jones

Overview

HPC Benchmarking

1. There are existing benchmarks such as MLPerf
   - Mattson *et al.*, "MLPerf Training Benchmark," *Proceedings of the 3rd MLSys Conference*, Austin, TX, USA, 2020
2. We wanted to explore real-world training performance
3. Focus on GPT-2 (minGPT)
4. Use PyTorch Lightning for multi-GPU strategies
   - `https://github.com/Lightning-Universe/lightning-GPT`

HPC Systems

Service	Name	Type	Accelerator	Mem (GB)	Interface	Launched
JADE 2	J-V100-32	GPU	Nvidia V100	32	SXM2	06-2017
Baskerville	B-A100-40	GPU	Nvidia A100	40	SXM4	06-2020
Baskerville	B-A100-80	GPU	Nvidia A100	80	SXM4	06-2020
Stanage	S-H100-80	GPU	Nvidia H100	80	PCIe 4.0	03-2023
COSMA8	C-MI100-32	GPU	AMD MI100	32	PCIe 4.0	11-2020
COSMA8	C-MI210-64	GPU	AMD MI210	64	PCIe 4.0	03-2022
Graphcore	IPU-POD 16	IPU	IPU-M2000	14.4	RoCEv2	03-2021
Dawn	D-M1550-128	GPU	Intel Max 1550	128	PCIe 5.0	03-2023


Lenovo
UNIVERSITY OF BIRMINGHAM
BASKERVILLE


COSMA
DELL EMC

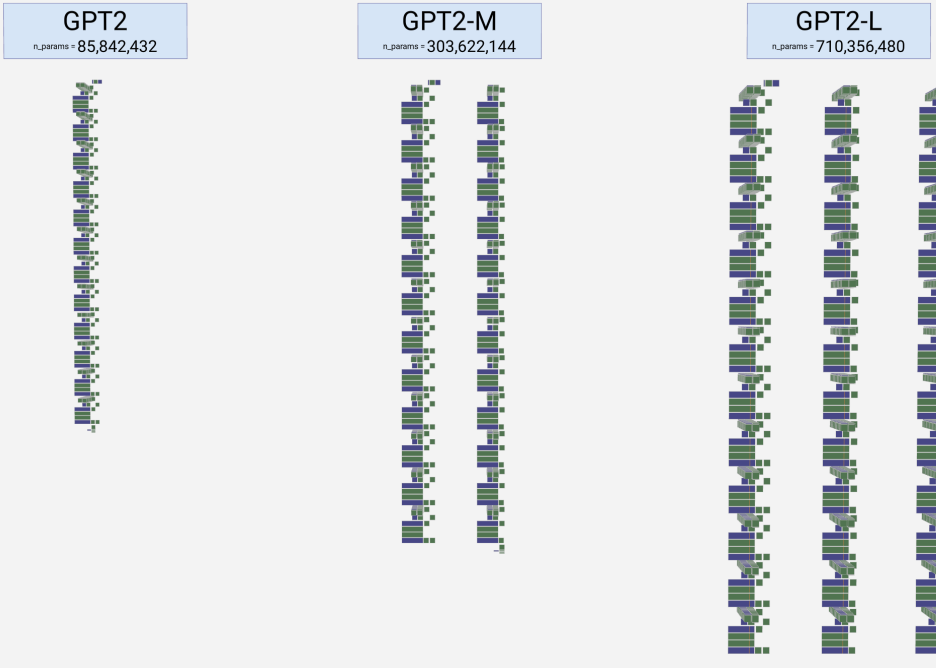
HPC Peak Performance on paper (TFLOPs)

Service	Name	FP16	BFLOAT16	FP32	FP64
JADE 2	J-V100-32	31.33	–	15.7	7.8
Baskerville	B-A100-40	77.97	312	19.5	9.7
Baskerville	B-A100-80	77.97	312	19.5	9.7
Stanage	S-H100-80	204.9	1513	51	26
COSMA8	C-MI100-32	184.6	92.3	23.1	11.5
COSMA8	C-MI210-64	181	181	22.6	22.6
Graphcore	IPU-POD 16	3994	-	998	–
Dawn	D-M1550-128	52.43	832	52.43	52.43

Training

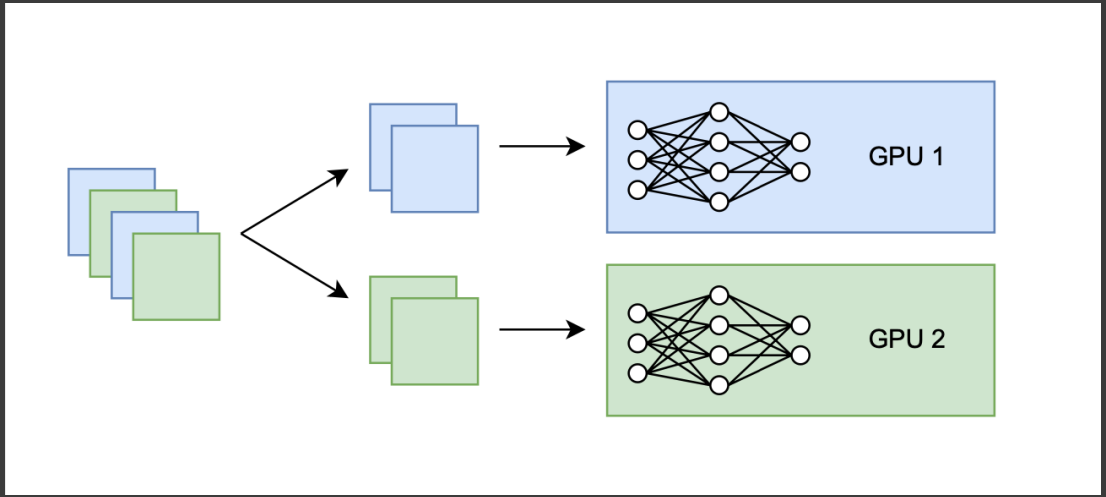
GPT-2 model sizes used for training

Model	Hidden layers	Attention heads	Embedding dim	Parameters (M)	16 bit Size (MB)
GPT2	12	12	768	85.21	170.51
GPT2-M	24	16	1024	302.51	605.16
GPT2-L	36	20	1280	708.64	1417.45
GPT2-XL	48	25	1600	1475.87	2951.96
GPT2-XXL	60	30	1920	2656.08	5312.43
GPT2-XXXL	84	40	2560	6609.33	12219.00

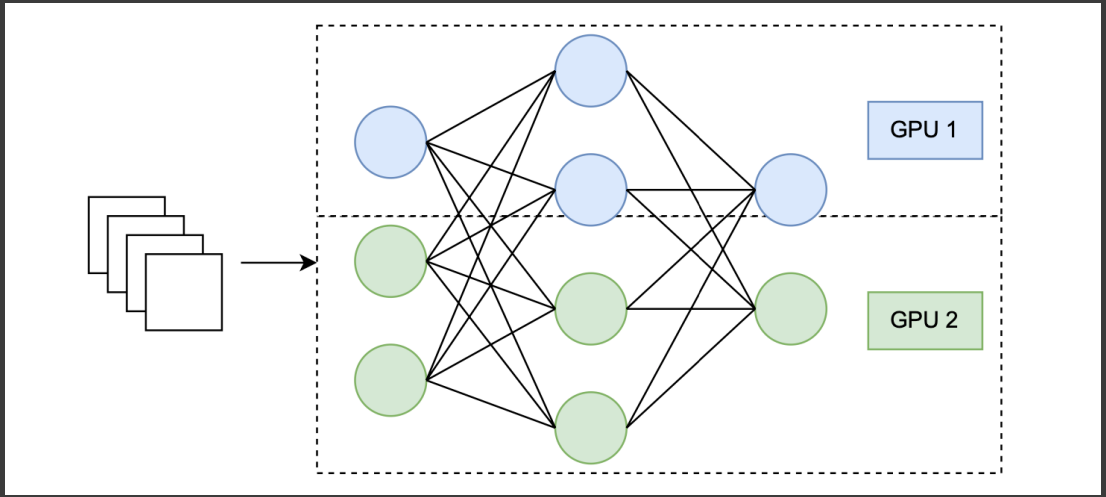

GPT2
n_params = 85,842,432
GPT2-M
n_params = 303,622,144
GPT2-L
n_params = 710,356,480

Strategies

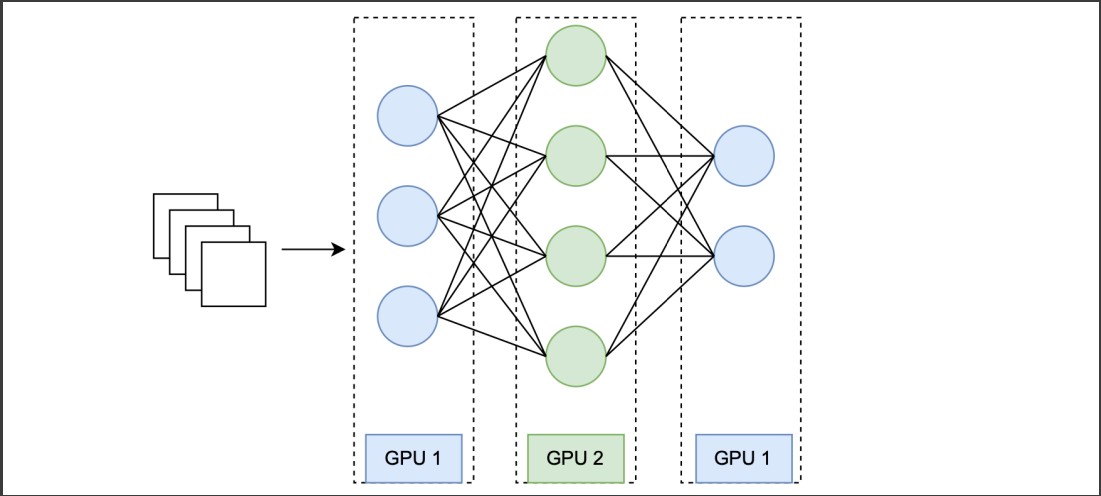
Distributed Data Parallel



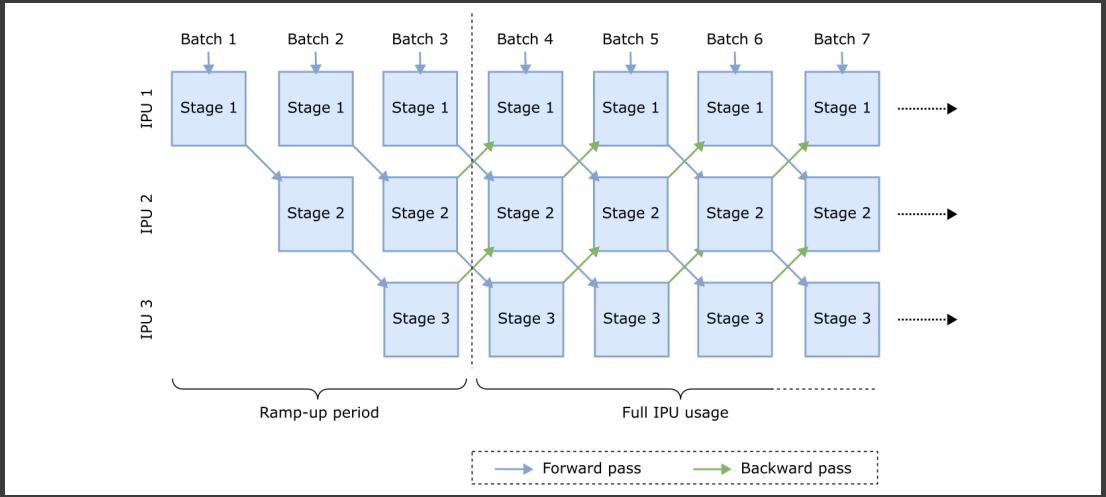
DeepSpeed ZeRO



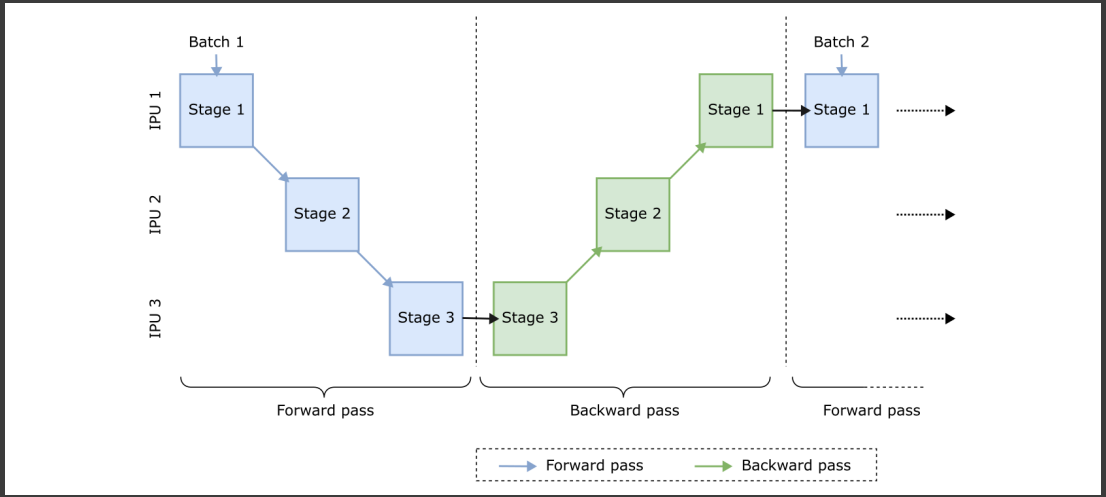
Fully Sharded Data Parallel



IPU Pipelined Execution

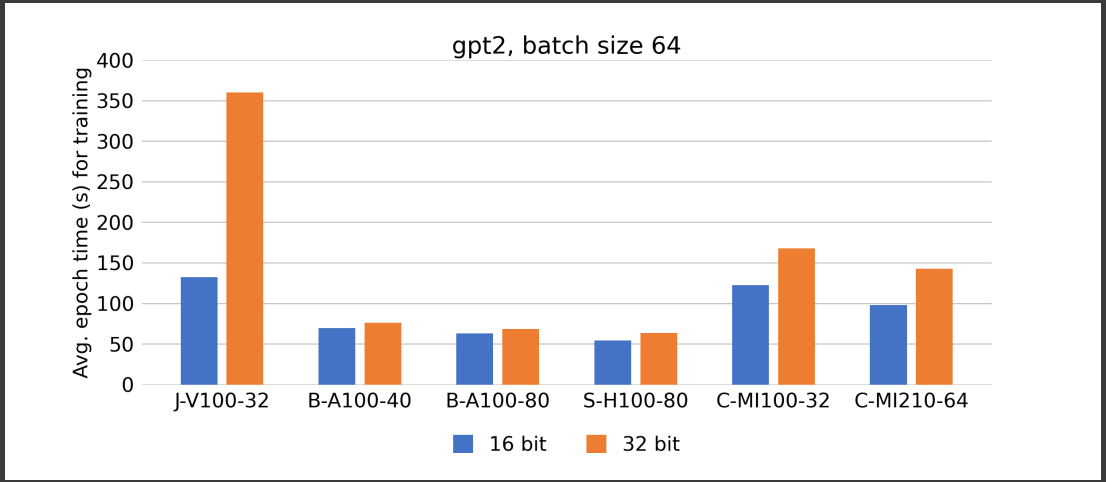


IPU Sharded Execution

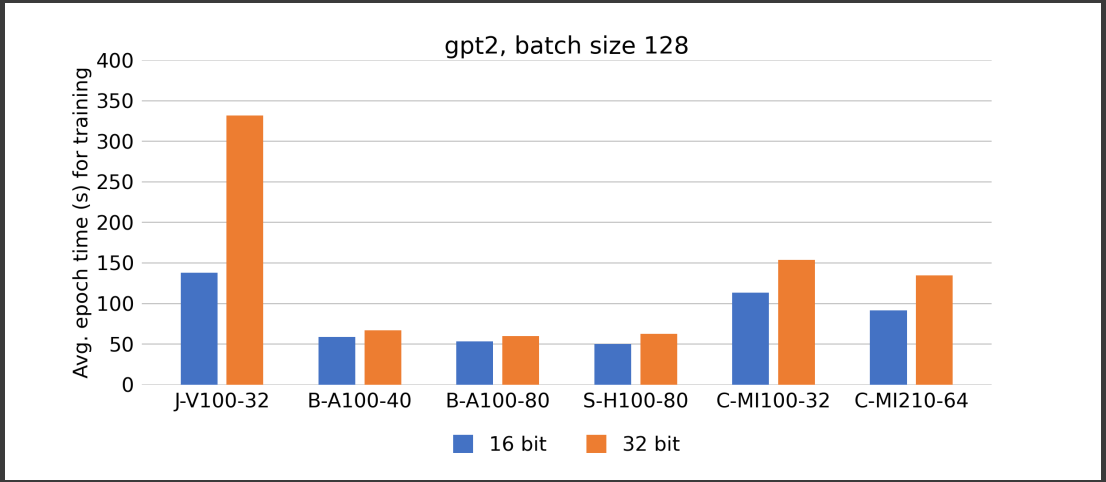


Single Accelerator Comparison

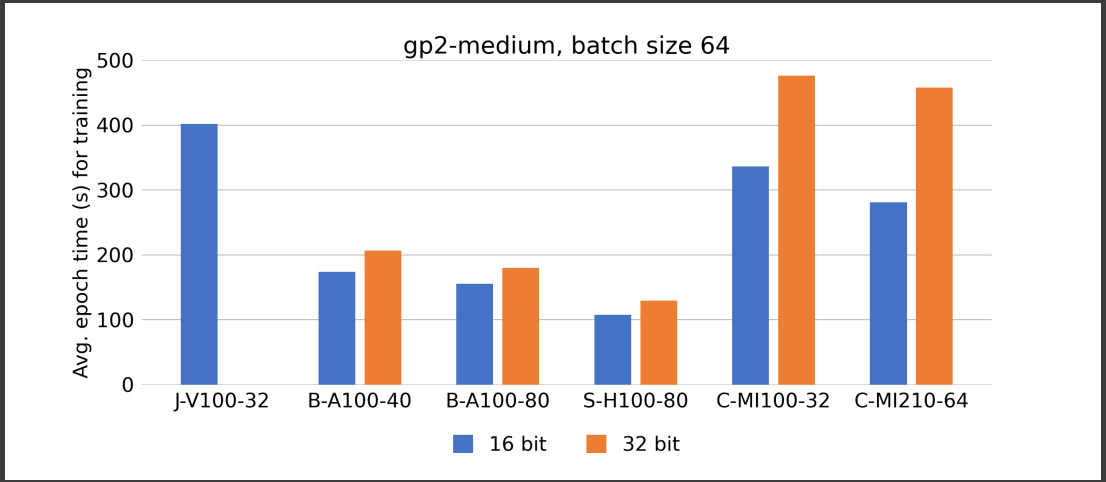
Single Accelerator Comparison



Single Accelerator Comparison



Single Accelerator Comparison

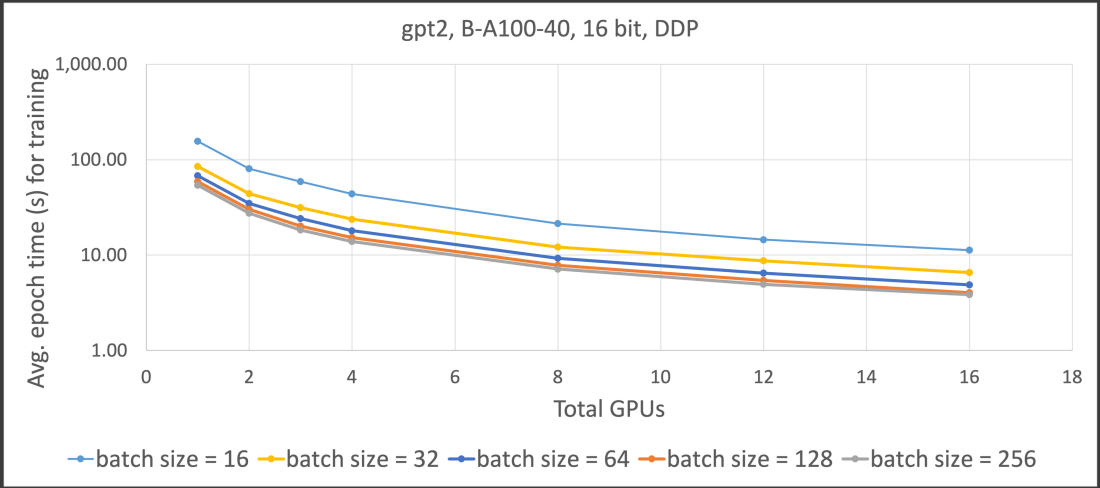


Single Accelerator Comparison - Observations

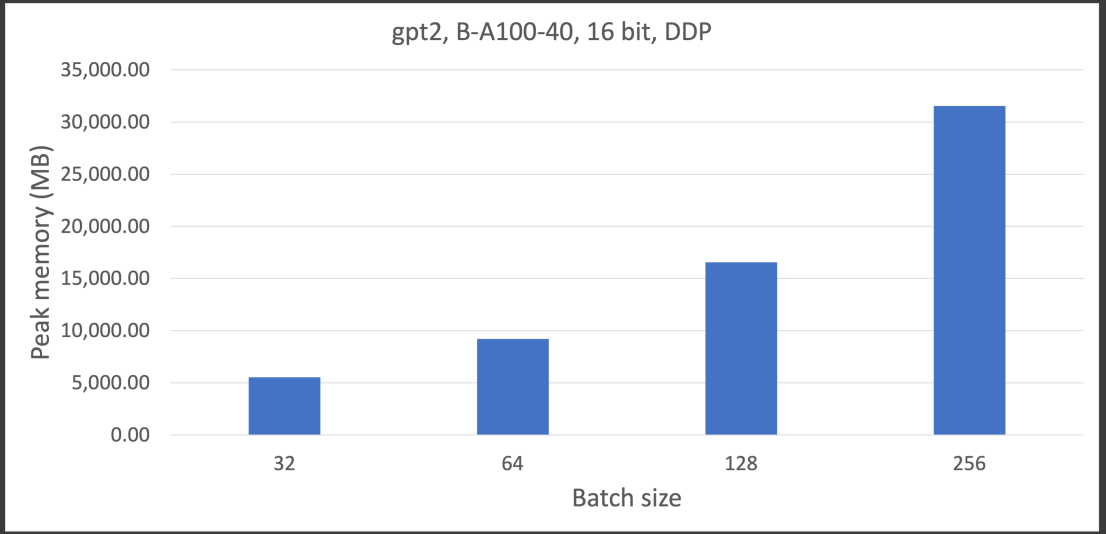
1. S-H100-80 is the fastest GPU in terms of theoretical and actual performance.
2. Although the S-H100-80 give the best performance, the performance gap between this and the B-A100s was not as large as we'd expected.
3. Performance largely depends on peak performance and precision used.
4. Reported peak FP16 and FP32 performances don't reflect our observations.
5. Difference between 16 bit and 32 bit precision, except for J-V100-32, is less significant when training smaller models.
6. Increasing from GPT2 to GPT2-M results in a significant increase in training time.
7. Doubling the batch size from 64 to 128 does not significantly improve training time.
8. GPT2-M and a batch size of 128 wouldn't fit into 40 GB.

Scaling Up and Out with DDP

Scaling Up and Out with DDP



Scaling Up and Out with DDP

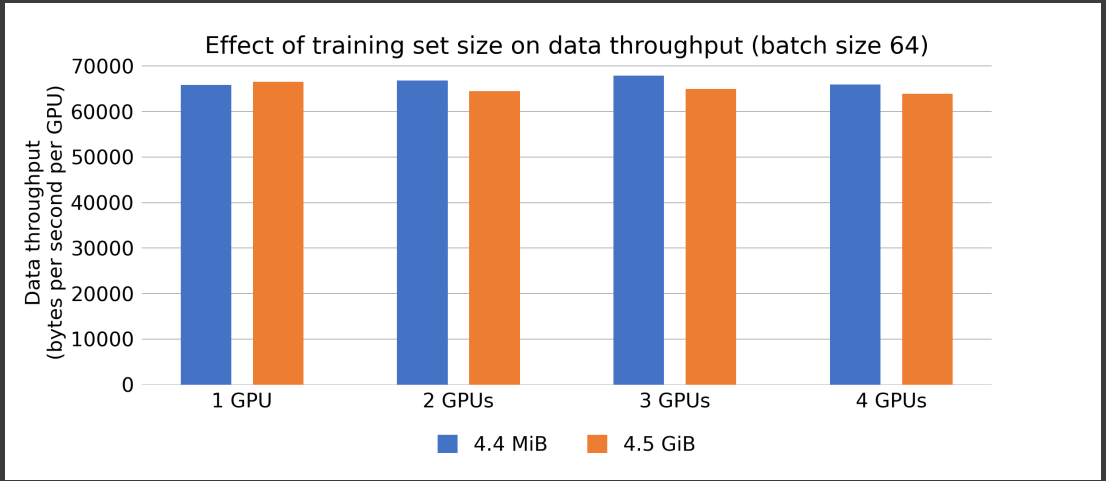


Scaling Up and Out with DDP - Observations

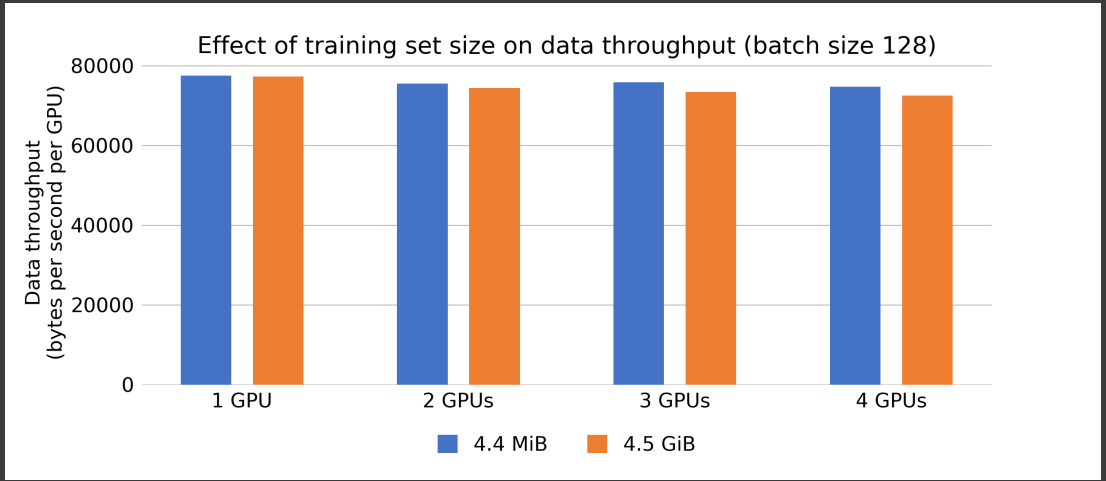
1. The scaling between 1 and 16 GPUs is almost linear.
2. Doubling batch size from 64 to 128 training time decreases by 15%.
3. Quadrupling from 64 to 256 training time reduces by 22%.
4. Halving batch size from 64 to 32 training time increases by 31%.
5. Quartering from 64 to 16 training time increases by 137%.
6. For fixed model size the limiting performance factor is the batch size.
7. Batch size is limited by GPU memory.
8. Doubling batch size increases the peak memory usage by a factor of 1.5.
9. Peak memory usage did not significantly change between 1 to 16 GPUs.

Larger Datasets

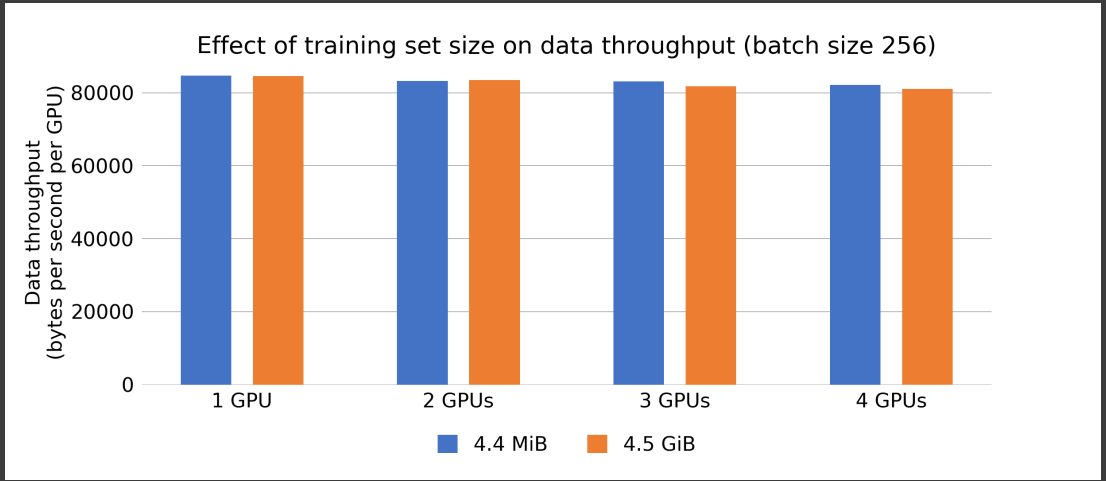
Larger Datasets



Larger Datasets



Larger Datasets

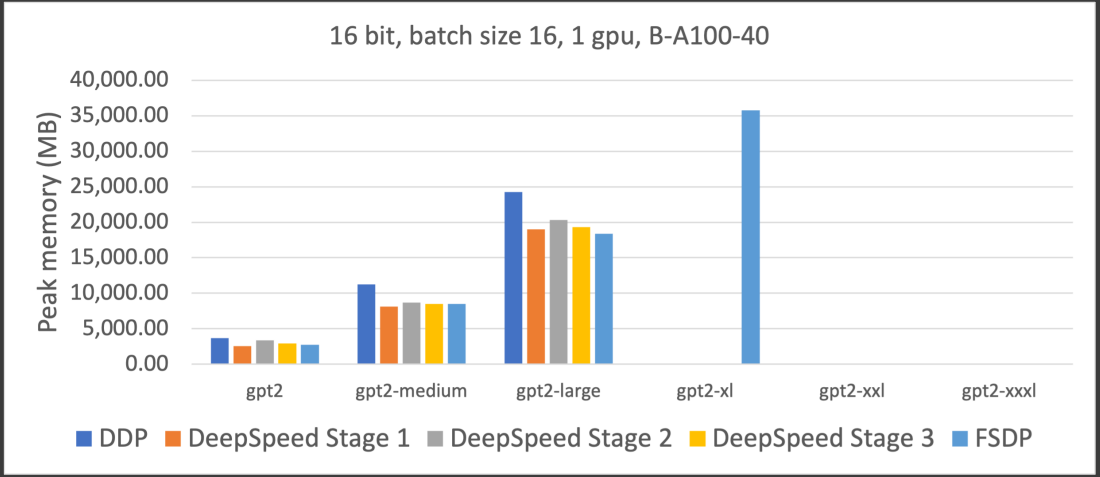


Larger Datasets - Observations

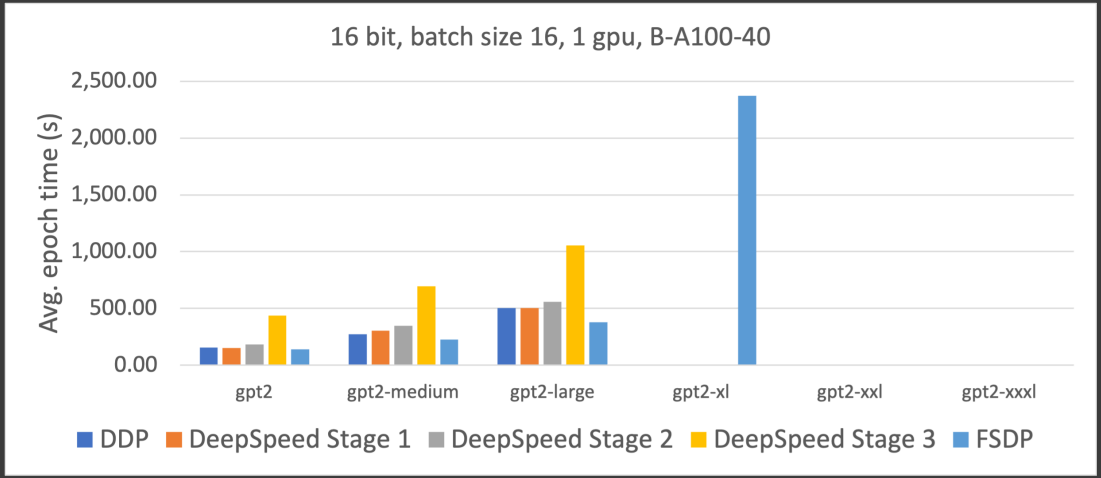
1. Average epoch time scales only slightly more than linearly with training dataset size.
2. Batch size has a larger impact on data throughput than dataset size.
3. Throughput reducing slightly as the number of GPUs increases.
4. PCI data sent and received remains broadly the same for both the small and large training datasets.
5. Rule of thumb: training time increases linearly with training data size.

Model Parallelism

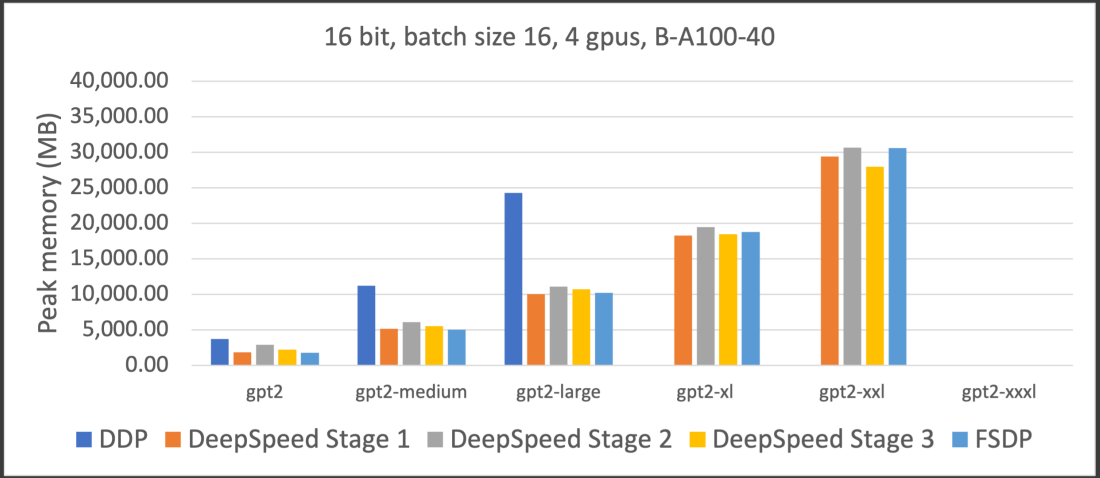
Model Parallelism



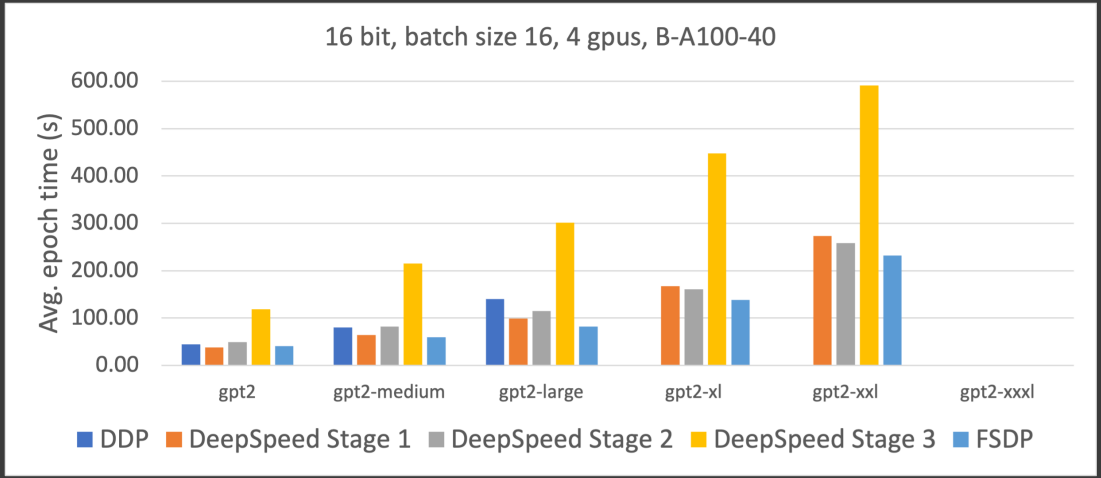
Model Parallelism



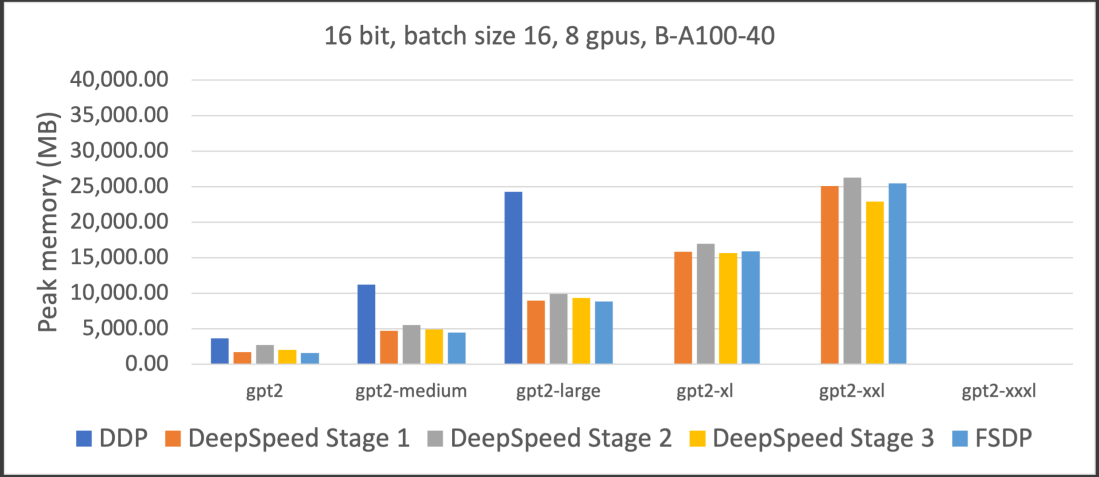
Model Parallelism



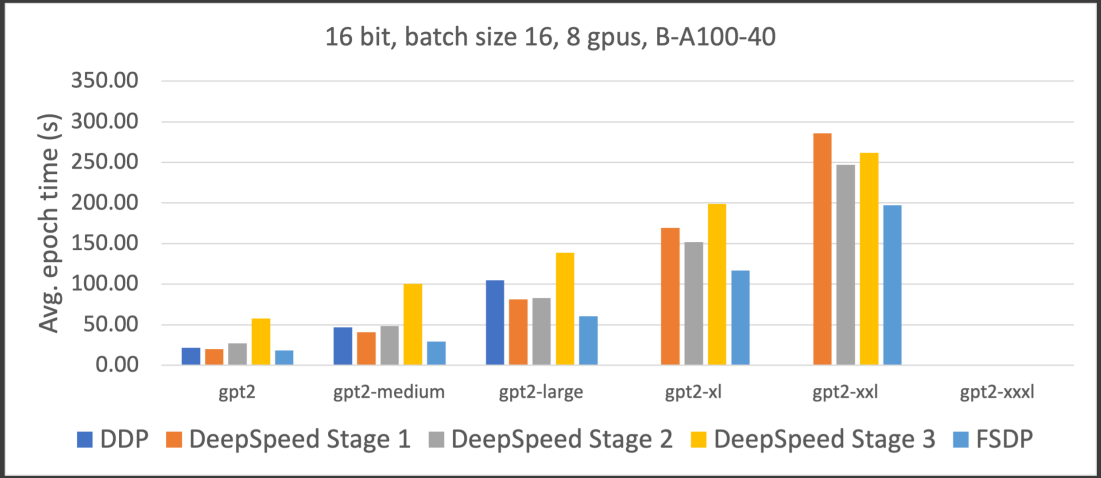
Model Parallelism



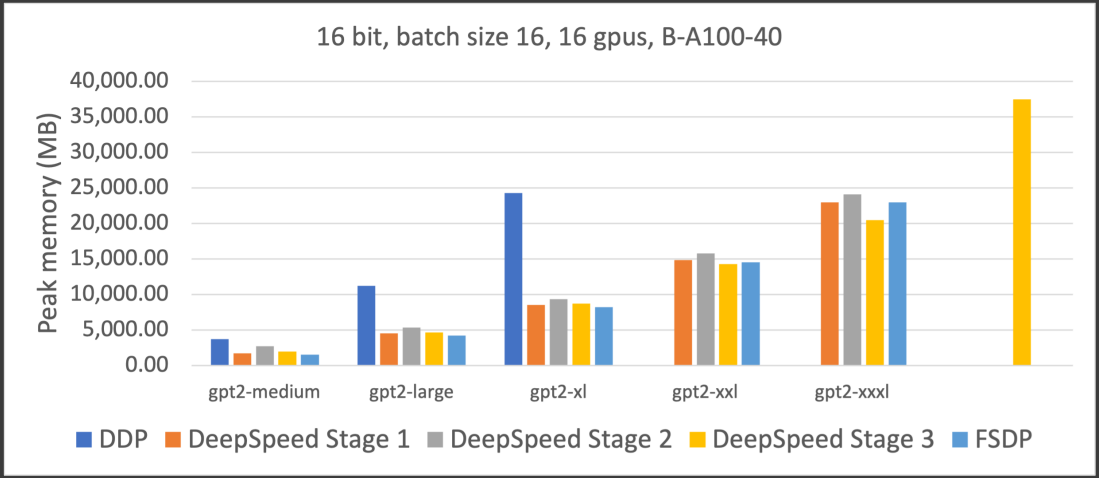
Model Parallelism



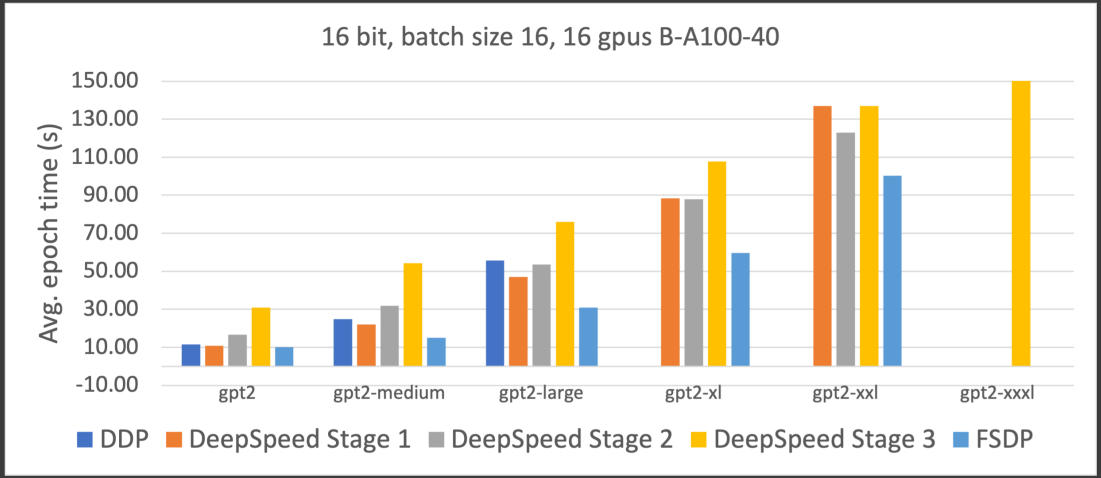
Model Parallelism



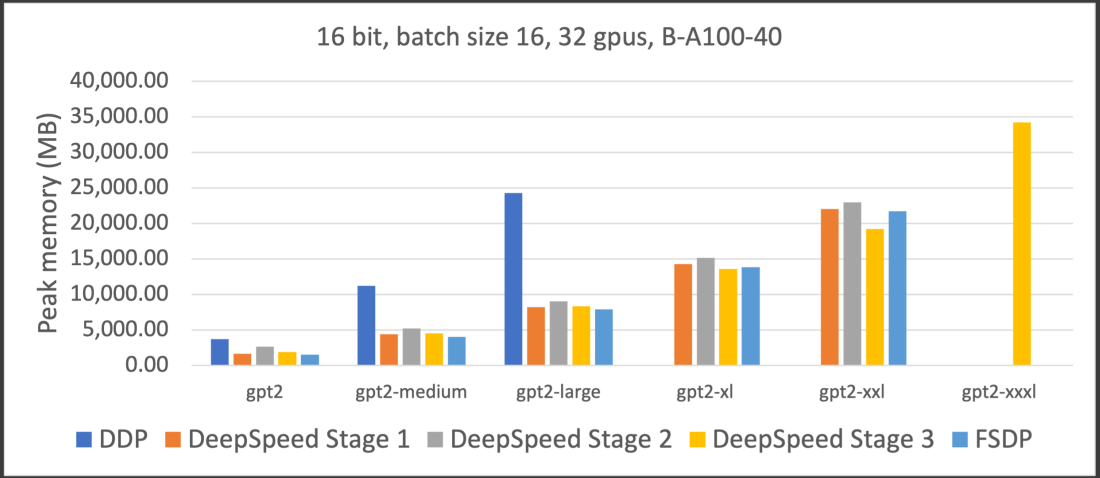
Model Parallelism



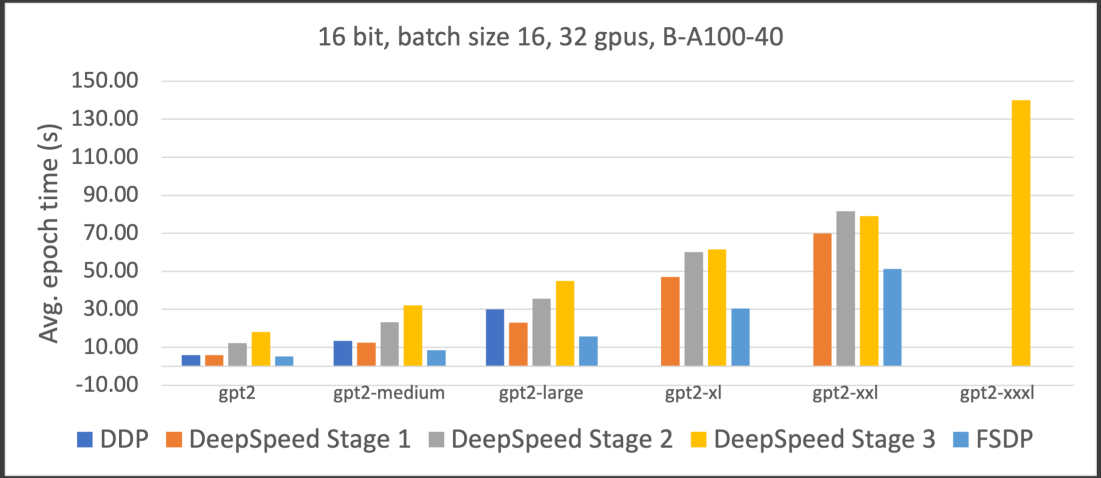
Model Parallelism



Model Parallelism



Model Parallelism

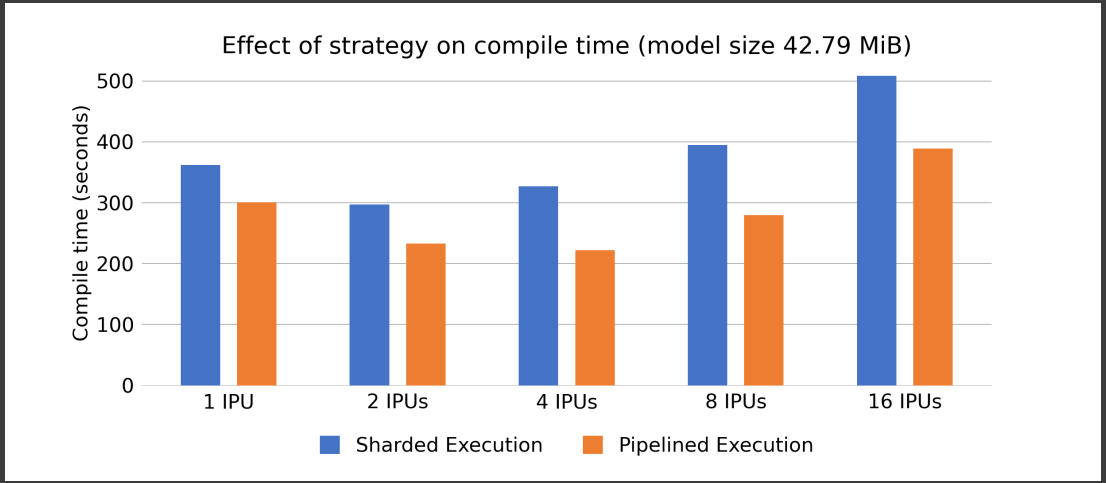


Model Parallelism - Observations

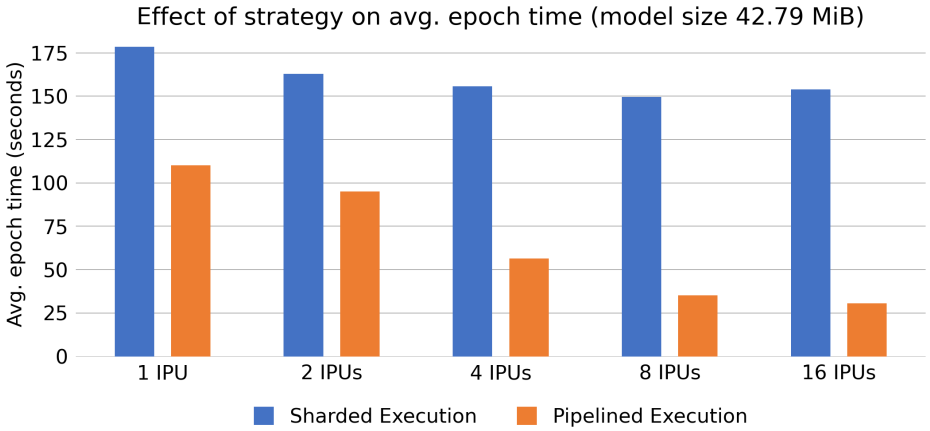
1. Utilising DeepSpeed Stage 3 enables training of the largest model size with a minimum of 16 GPUs.
2. FSDP facilitates training GPT2-XXL when at least 4 GPUs are utilised.
3. With 1 GPU, DeepSpeed and FSDP used less memory than DDP.
4. With 4 GPUs DeepSpeed and FSDP used 50% less memory than DDP.
5. Similar trends were found for 8, 16, and 32 GPUs.
6. With more than 8 GPUs, DeepSpeed Stage 3 exhibited lowest average peak memory for model sizes of GPT2-XL or larger.

Graphcore IPU-IPOD 16

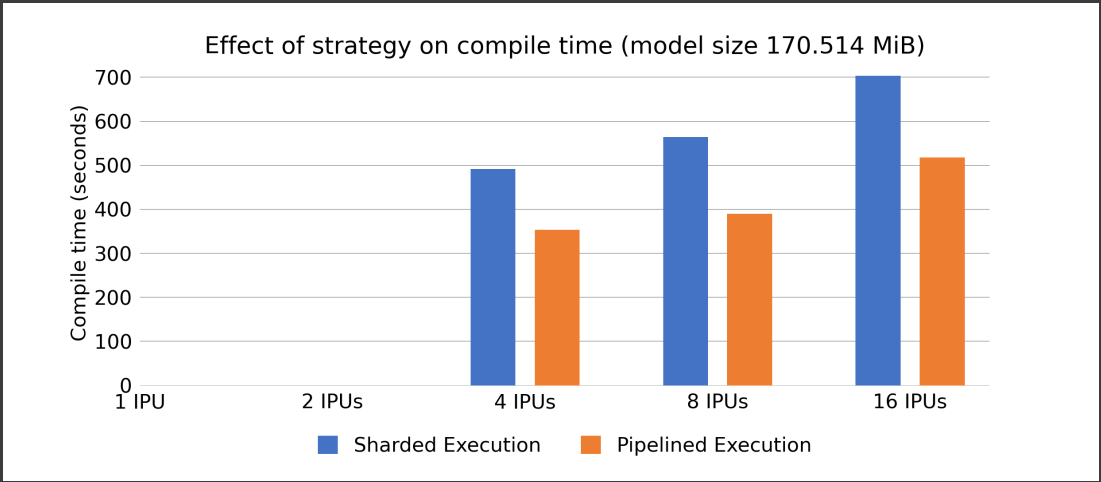
Graphcore IPU-IPOD 16



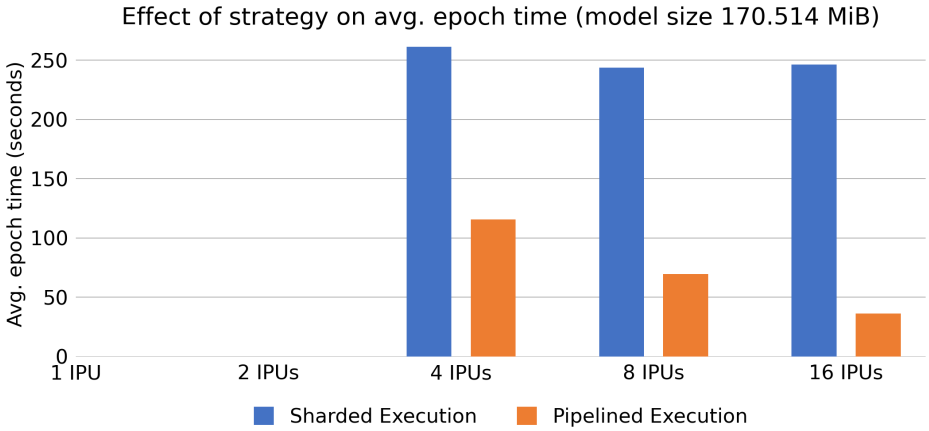
Graphcore IPU-IPOD 16



Graphcore IPU-IPOD 16



Graphcore IPU-IPOD 16



Graphcore IPU-IPOD 16 - Observations

1. Compilation time increases logarithmically with IPUs for Sharded and Pipelined.
2. Parallel Phased Execution ran on average 50 times slower than Pipelined Execution.
3. Serial Phased Execution failed due to execution phases needing to be data-dependent.
4. Increasing model size also causes the compilation time to increase.
5. Epoch time decreases with the number of IPUs in use.
6. Compilation time was nearly double the time taken to run a single training epoch.
7. Pipelined Execution is the superior strategy.

Conclusions

Conclusions - Speed

1. BFLOAT16 peak performance is a better indicator for AI workloads than FP16 or FP32.
2. Tensor Core on B-A100 enables mixed-precision training, better for AI workloads.
3. C-MI100-32 and C-MI210-64 potentially more suitable for traditional HPC tasks.
4. FSDP is faster than DeepSpeed, but DeepSpeed Stage 3 is more memory-efficient for the largest models.

Conclusions - Memory

1. DDP allows only data parallelism so model size is limited by single GPU memory.
2. Largest model on a single B-A100-80 using DDP is GPT2-XXL; for larger models, model or pipeline parallelism is required.
3. DeepSpeed and FSDP showed improvements in peak memory and average epoch time compared to DDP.
4. Incrementing the GPU count consistently demonstrated diminishing average peak memory usage for DeepSpeed and FSDP.
5. The largest model sizes that can be trained using DeepSpeed and FSDP strategies are GPT2-XXXL.
6. A balanced consideration of both memory and time efficiency is needed especially for larger models.

Acknowledgements

1. With thanks to Edwin Brown, Sheffield and Turing.
2. Brenden Bycroft, LLM visualisation code.
3. This work was funded by The Alan Turing Institute under the EPSRC grant EP/N510129/1.
4. It was partially supported by Baskerville, a national accelerated compute resource under the EPSRC Grant EP/T022221/1.
5. It was partially supported by JADE: Joint Academic Data Science Endeavour - 2 under the EPSRC Grant EP/T022205/1, and The Exascale Computing: Algorithms and Infrastructures Benefiting UK Research (ExCALIBUR) program, which is funded under Wave 2 of the Strategic Priorities Fund (SPF).