UC San Diego
JACOBS SCHOOL OF ENGINEERING
Computer Science and Engineering

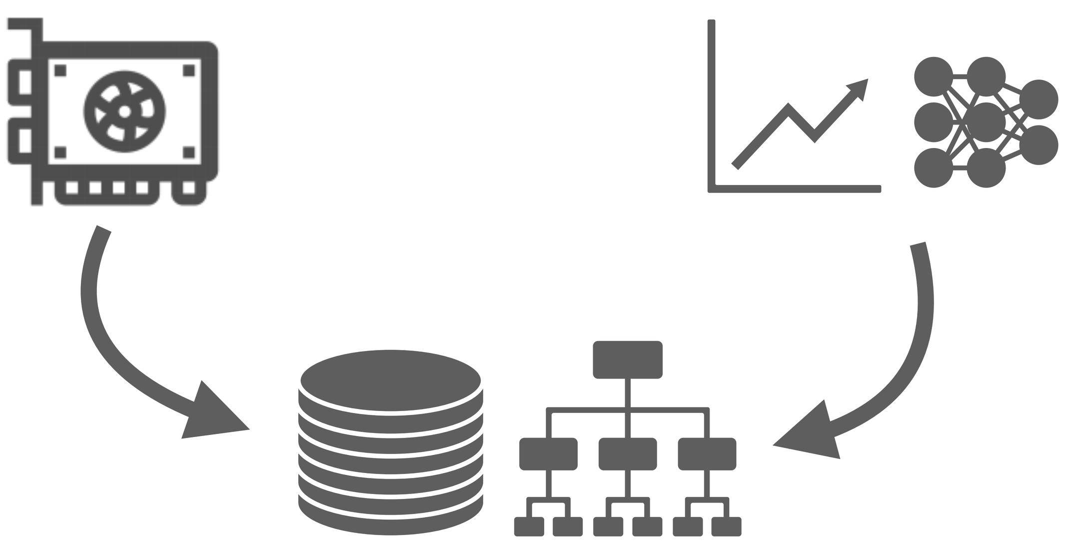
Hydra: Efficient Training for Larger-Than-Memory Deep Learning Models

Kabir Nagrecha | Advised by Arun Kumar

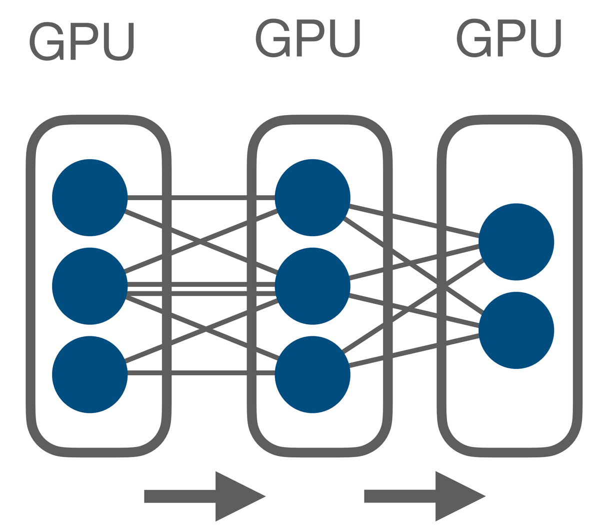
Problem

GPU memory is limited…

Deep Learning models are growing rapidly!



What's Needed: An efficient platform for distributed training of large models



Larger Models

Model Parallelism

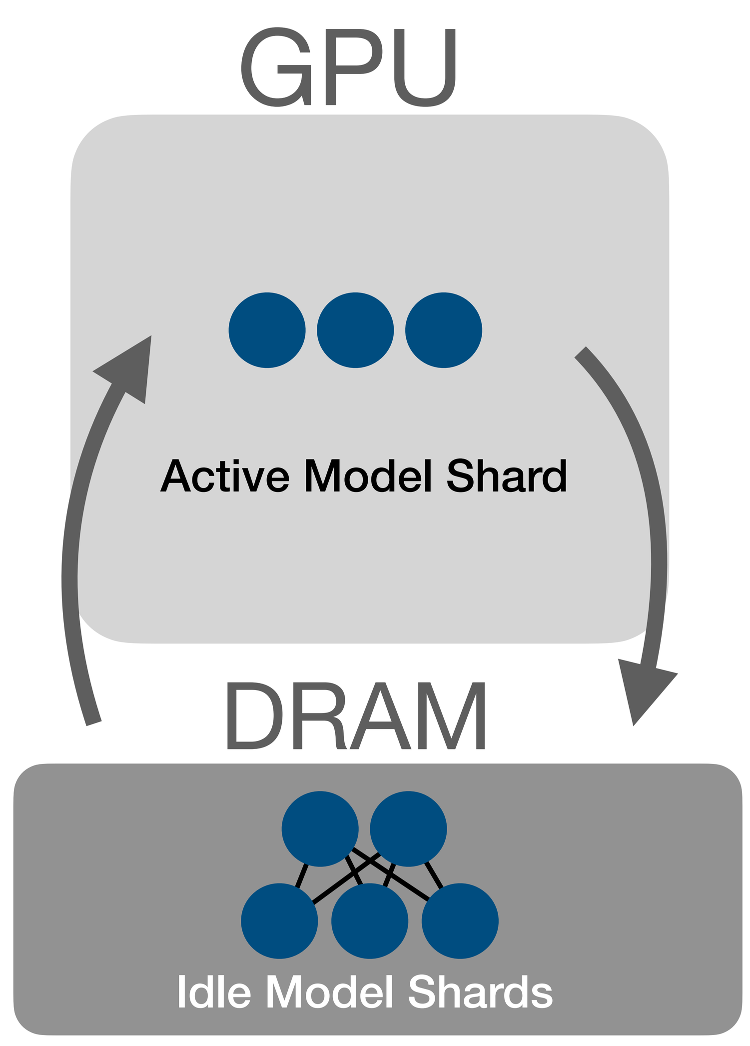
AN IMPOSSIBLE CHOICE!

Faster Training

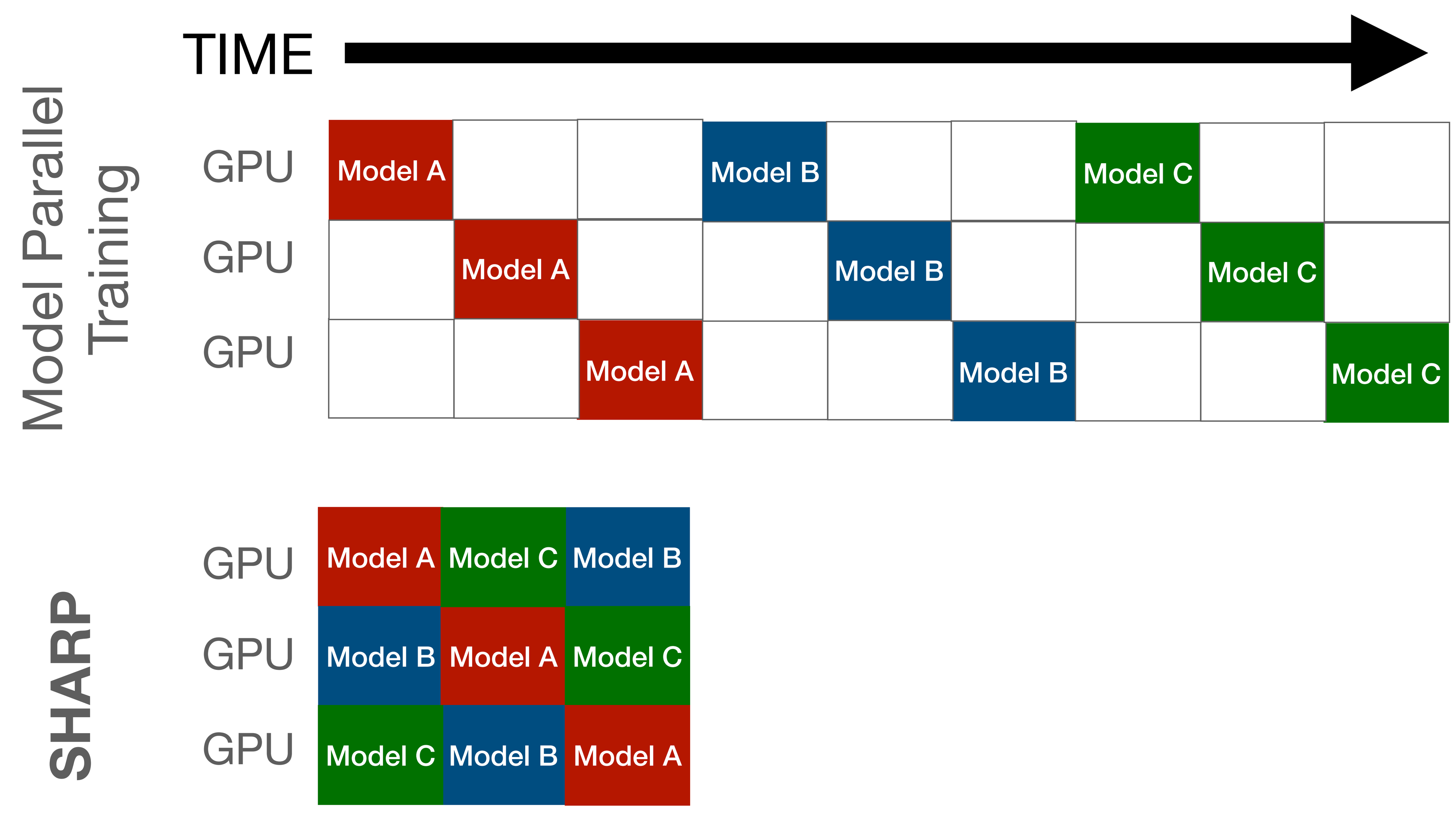
Task/Data Parallelism

Bottleneck: Traditional Model Parallelism uses multiple devices to handle the memory demands of a single model. But this reduces our ability to parallelize compute!

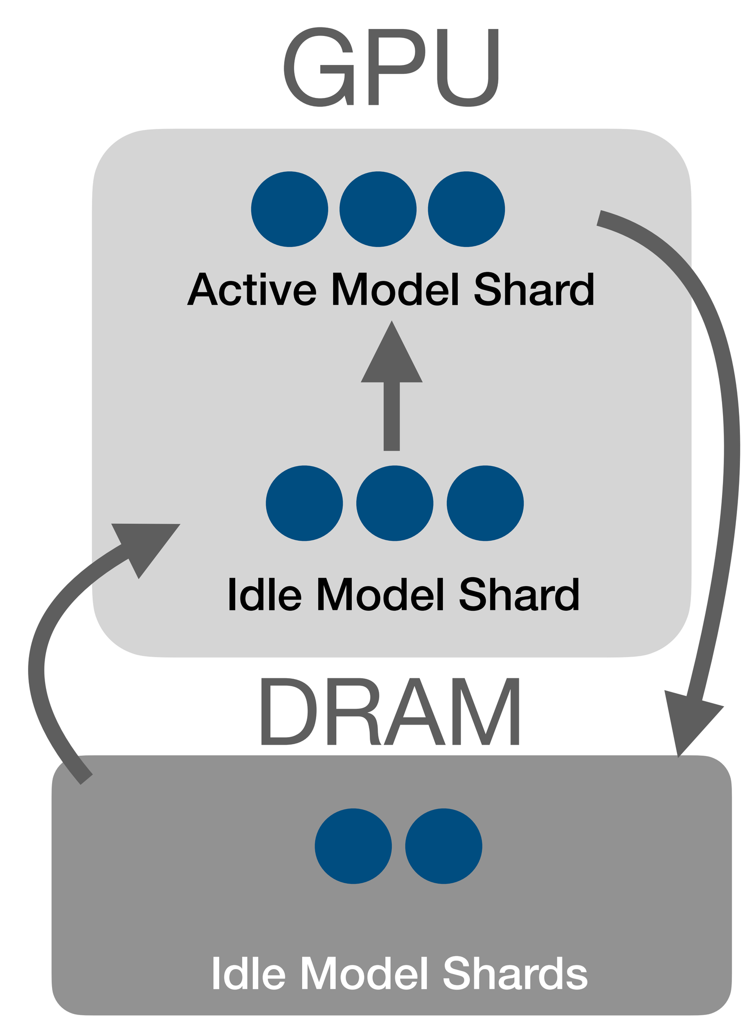
Hydra: Model Spilling, Shard Alternator Parallelism, and Double Buffering



Model Spilling
Detach training orchestration from GPU arrangement



Shard Alternator Parallelism (SHARP)
Blend model and task parallelism for high throughput training

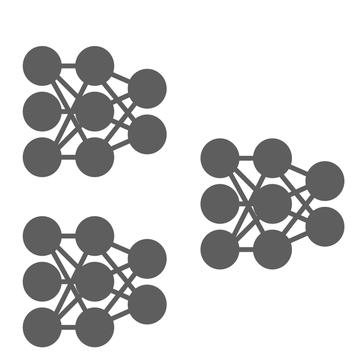


Double Buffering
Overlap communication with compute for low latency training

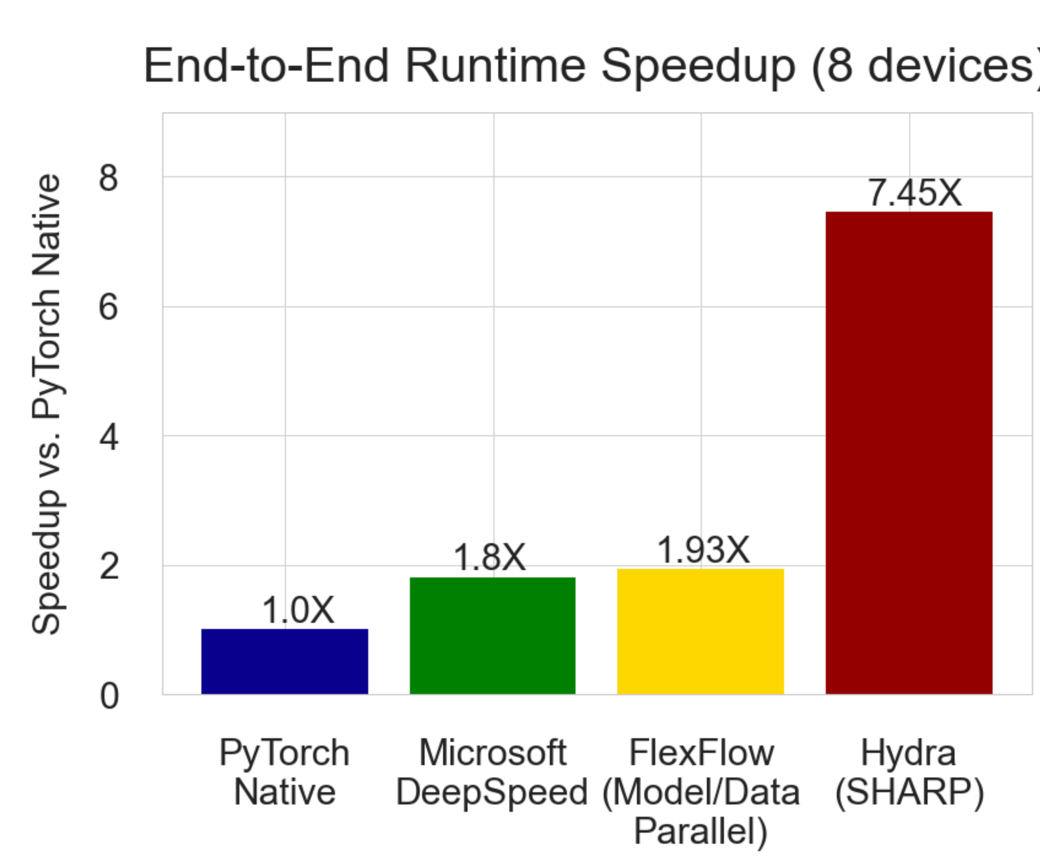
Evaluation

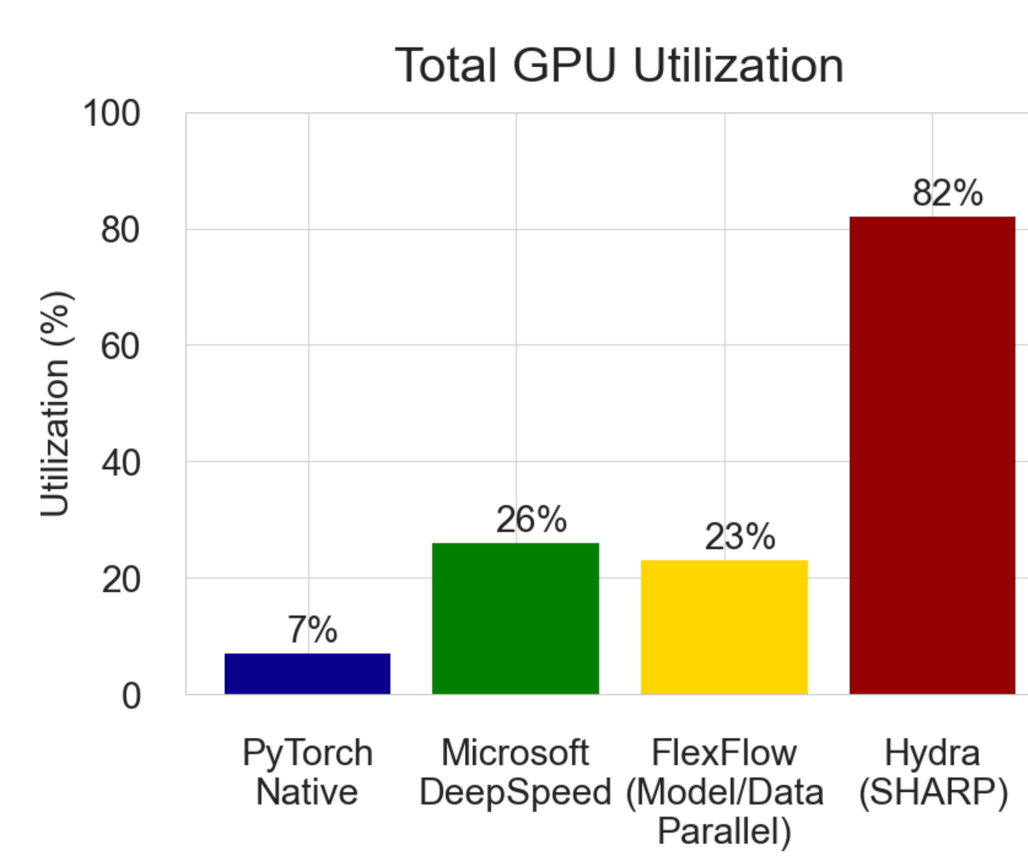
Benchmark Dataset:
WikiText-2

Workload:
Model Selection
- 12 1B+ parameter models
- Transformer pretraining task
- 8-32 batch size
- 128 sequence length



Hardware:
Single-node, 8 12GB GPUs

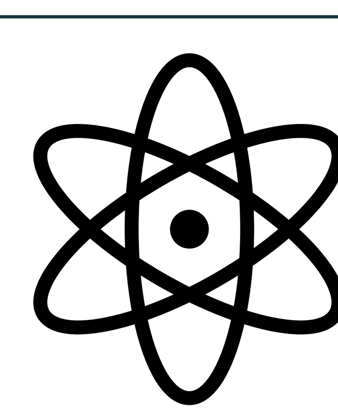




Hydra produces near-optimal speedups!
82% Average GPU Utilization
>7.4X Speedups with 8 Devices

Ongoing Work & Potential Impact

Data Parallelism



User Study - Deep Learning for Physical Simulations

Concurrent Training at Scale