Pre-training: Parallelism

CSC2541H1 Topics in Machine Learning, Winter 2025, UToronto

Chris J. Maddison

Announcements

- If you are assigned to **present on Feb 14, come talk** about presentations after class.
- I will try to get you feedback on the presentations within 2 weeks of your presentation.

Questions?

Recap & agenda

- Last week: understanding **the rate at which we can turn compute into better test loss** through scaling laws.
- This week: improvements in **the rate at which we can turn time into compute** through parallelism.
- **These two rates summarize the end-to-end performance of ML systems.**
- Most progress can be understood in these terms: Transformers (more efficient compute → test loss) and GPUs (more efficient time → compute)

Take-homes

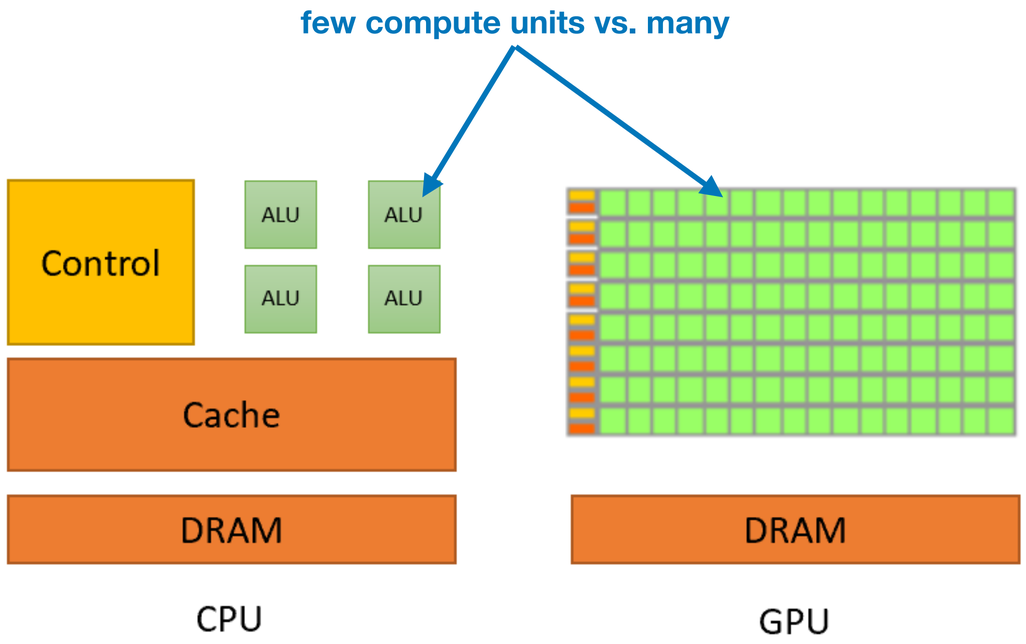
ML systems take-homes from Llama 3

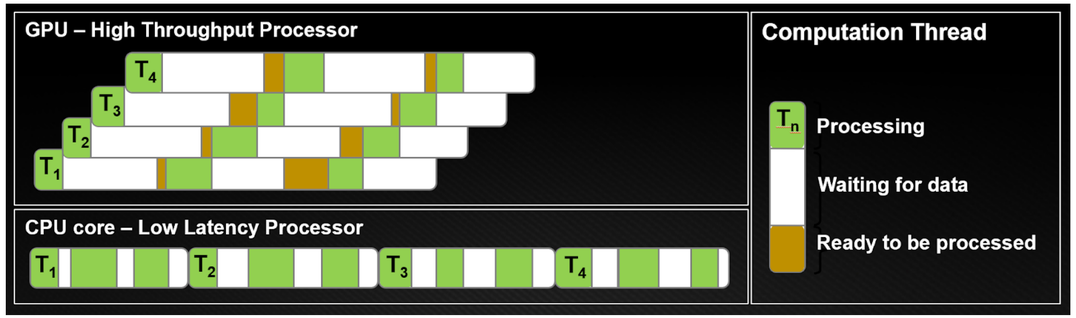
- A number of interesting systems take-homes from reading.
- But I will be focusing on the bigger picture: GPUs and multi-GPU setups.
  - **Caveat: I am not a systems expert!**

CPUs vs GPUs

High level

- Thread: a seq. of instructions.
  - Executed by a processor
  - Include read / write memory, floating point operations.
- CPUs are low latency
  - few threads, rarely waiting for data
- **GPUs are high throughput**
  - many threads, often waiting for data



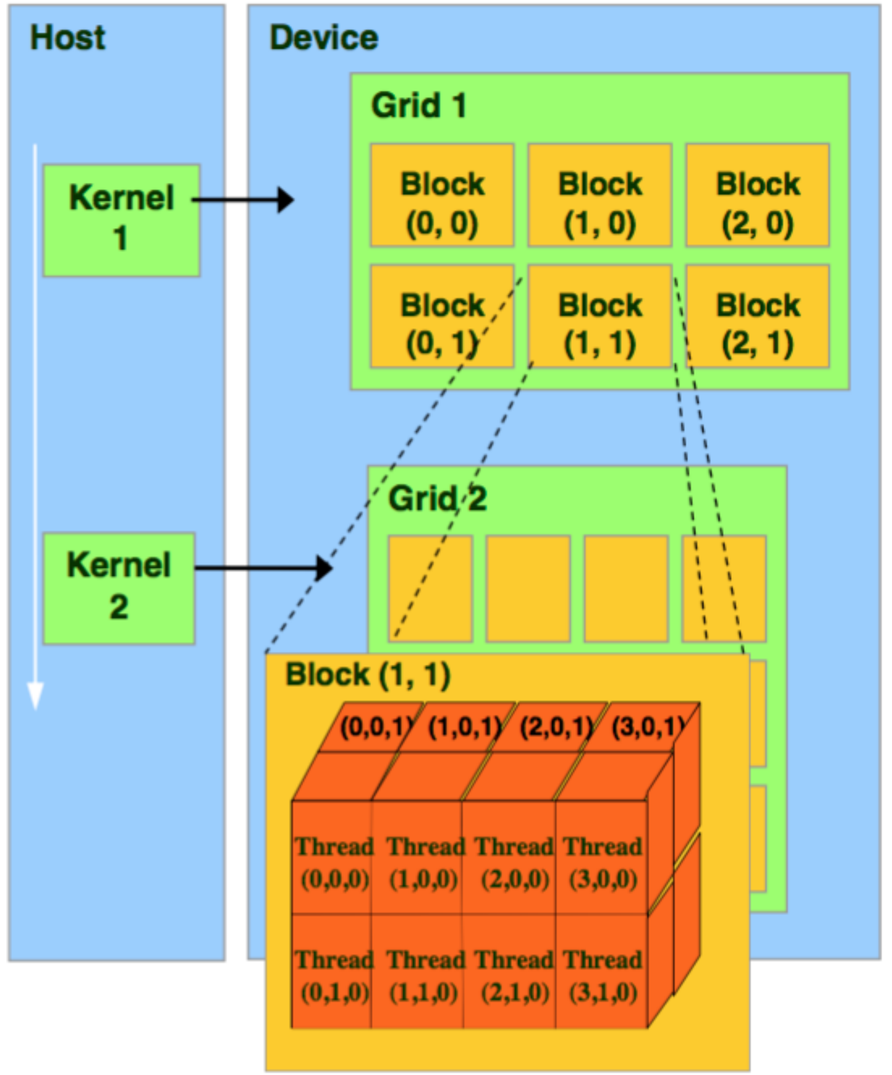


https://developer.nvidia.com/blog/cuda-refresher-reviewing-the-origins-of-gpu-computing/

GPU

Execution model

- Groups threads (warps) execute a single instruction, but applied to different data elements.
- **Vectorized instructions like matmul efficiently parallelized.**
- Bulk of Transformer compute is vectorized!
  - GPUs and AI/ML are a pair.

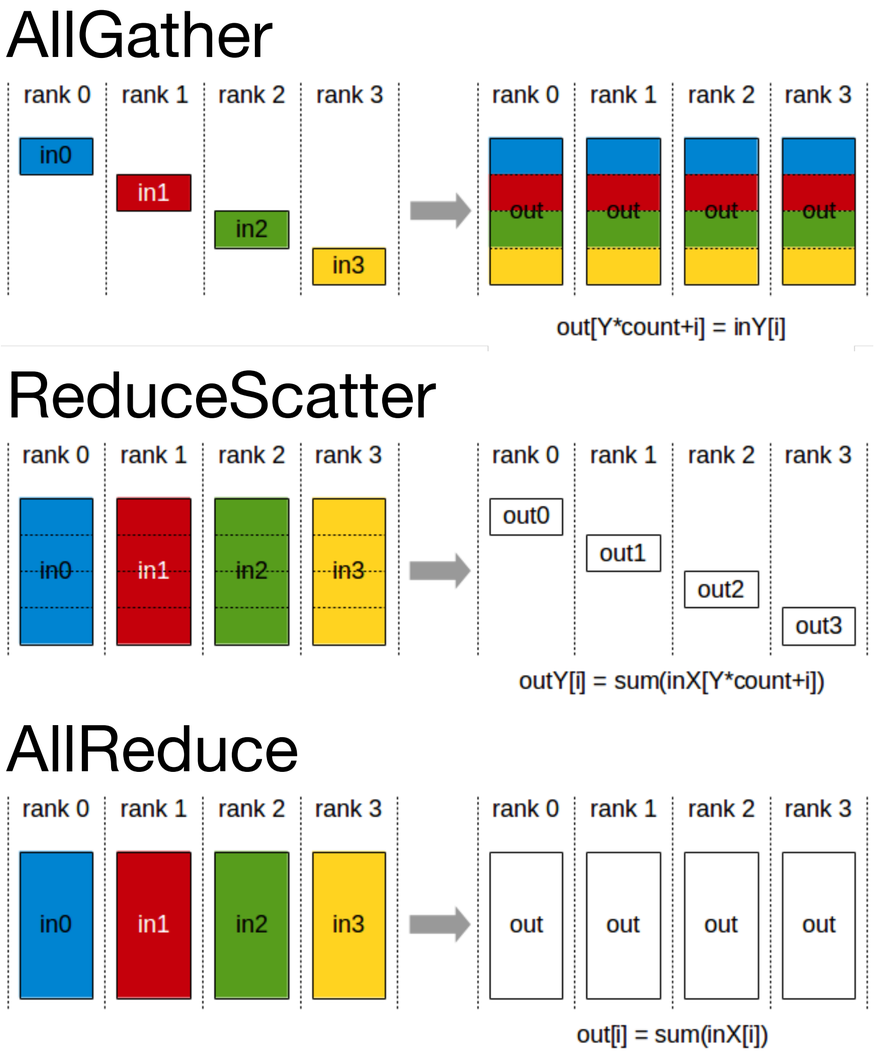


https://nyu-cds.github.io/python-gpu/02-cuda/

Multi-GPU

Parallelizing across GPUs

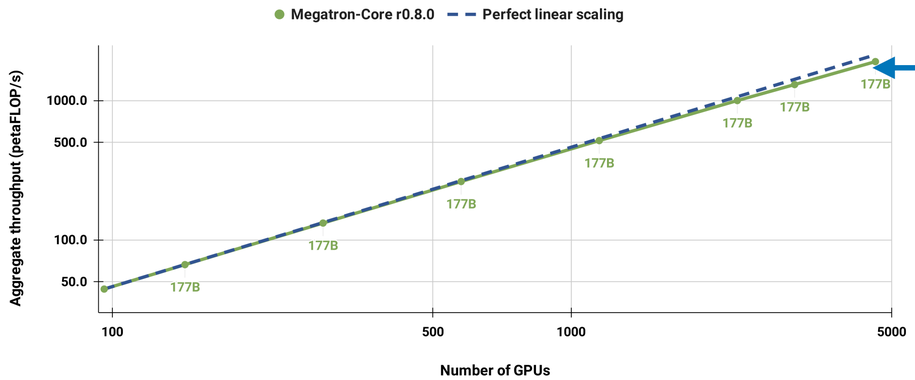
- As models scale, they no longer fit on a single GPU.
  - Distribute models (weights and activations) across GPU network
- **Communicate to synchronize state**
- **Clever overlapping of communication and computation needed**
  - As network scales, communication costs can hurt.



https://docs.nvidia.com/deeplearning/nccl/user-guide/docs/usage/collectives.html

Parallelism

- To summarize the efficiency of a multi-GPU setup, **we can measure the model FLOP utilization (MFU).**
  - Each GPU has a peak throughput measured in tera-FLOPs / s (TFLOPS).
  - MFU is [observed TFLOPS] / [theoretical TFLOPS aggregated across GPUs]



want MFU to be constant as you add GPUs

https://github.com/stas00/ml-engineering/blob/master/training/performance/README.md