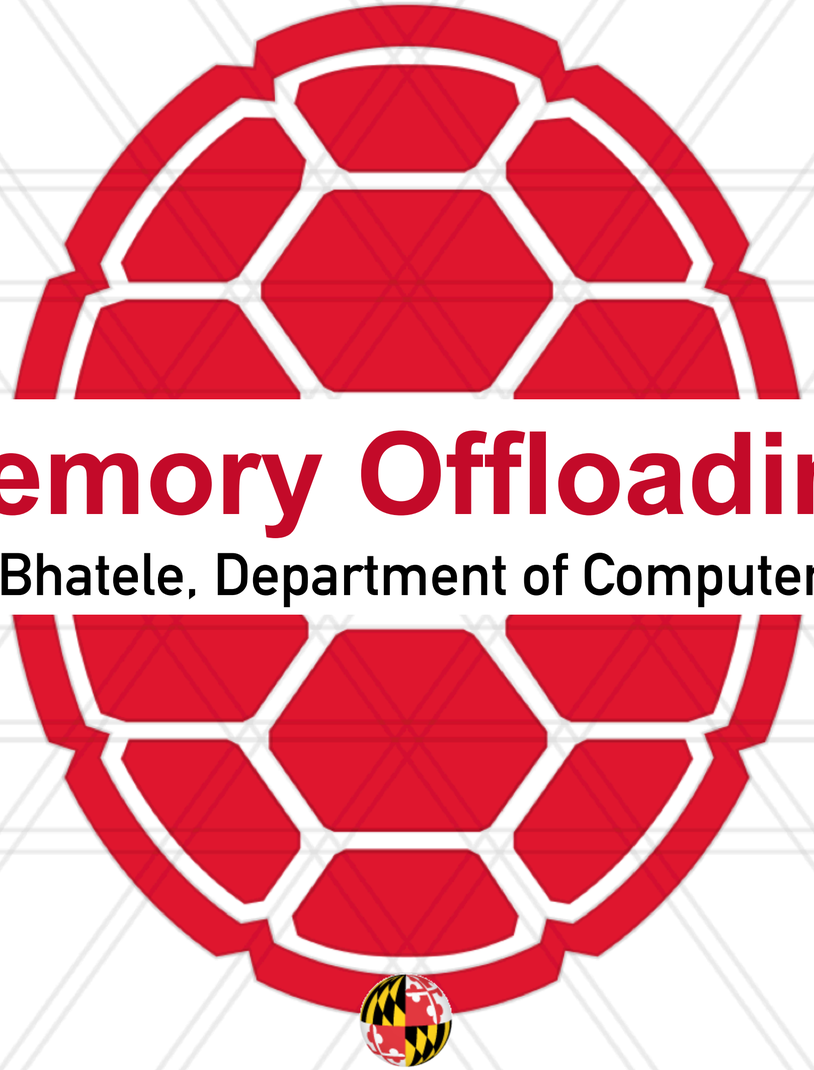
Systems for Machine Learning (CMSC828G)



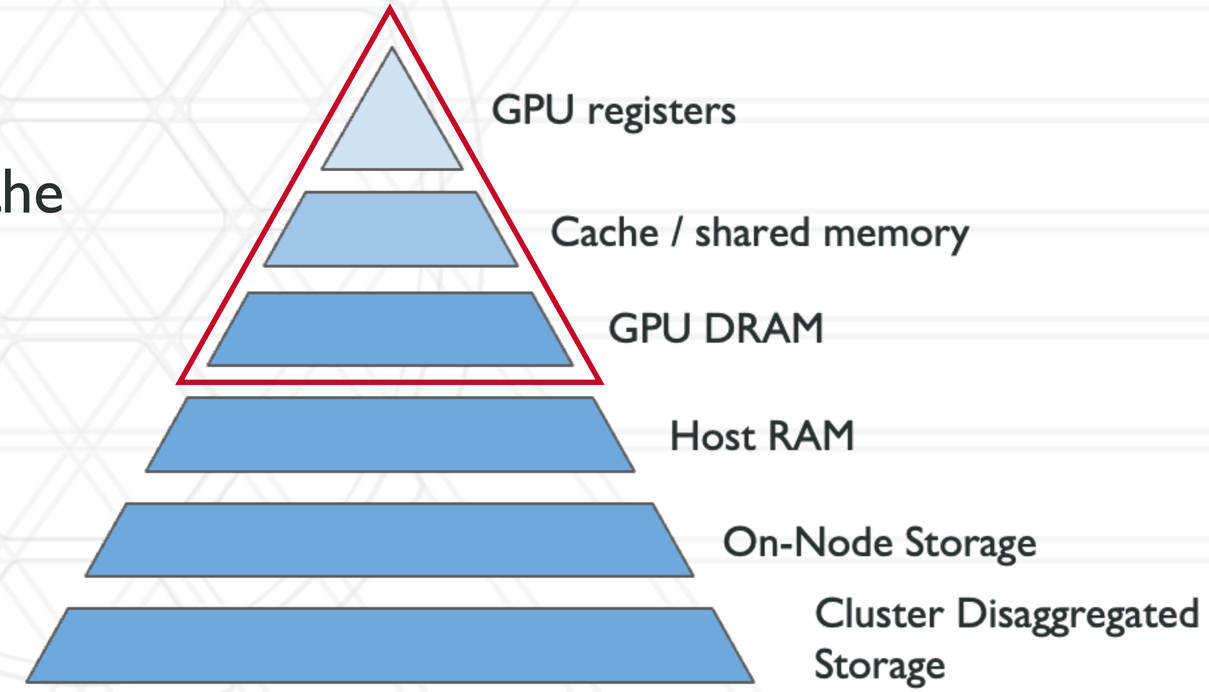
Memory Offloading

Abhinav Bhatele, Department of Computer Science

UNIVERSITY OF MARYLAND

Memory hierarchy

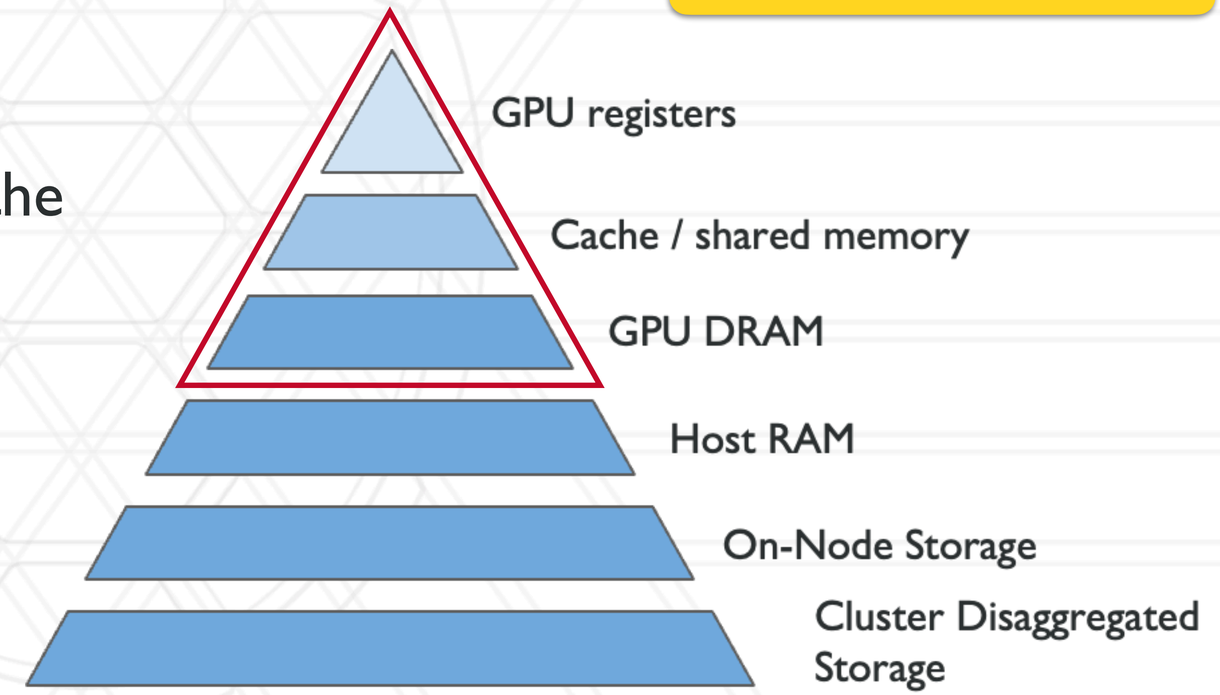
- If data doesn't fit in memory:
  - Use more GPUs - parallel training or inference
- OR offload to slower components in the hierarchy



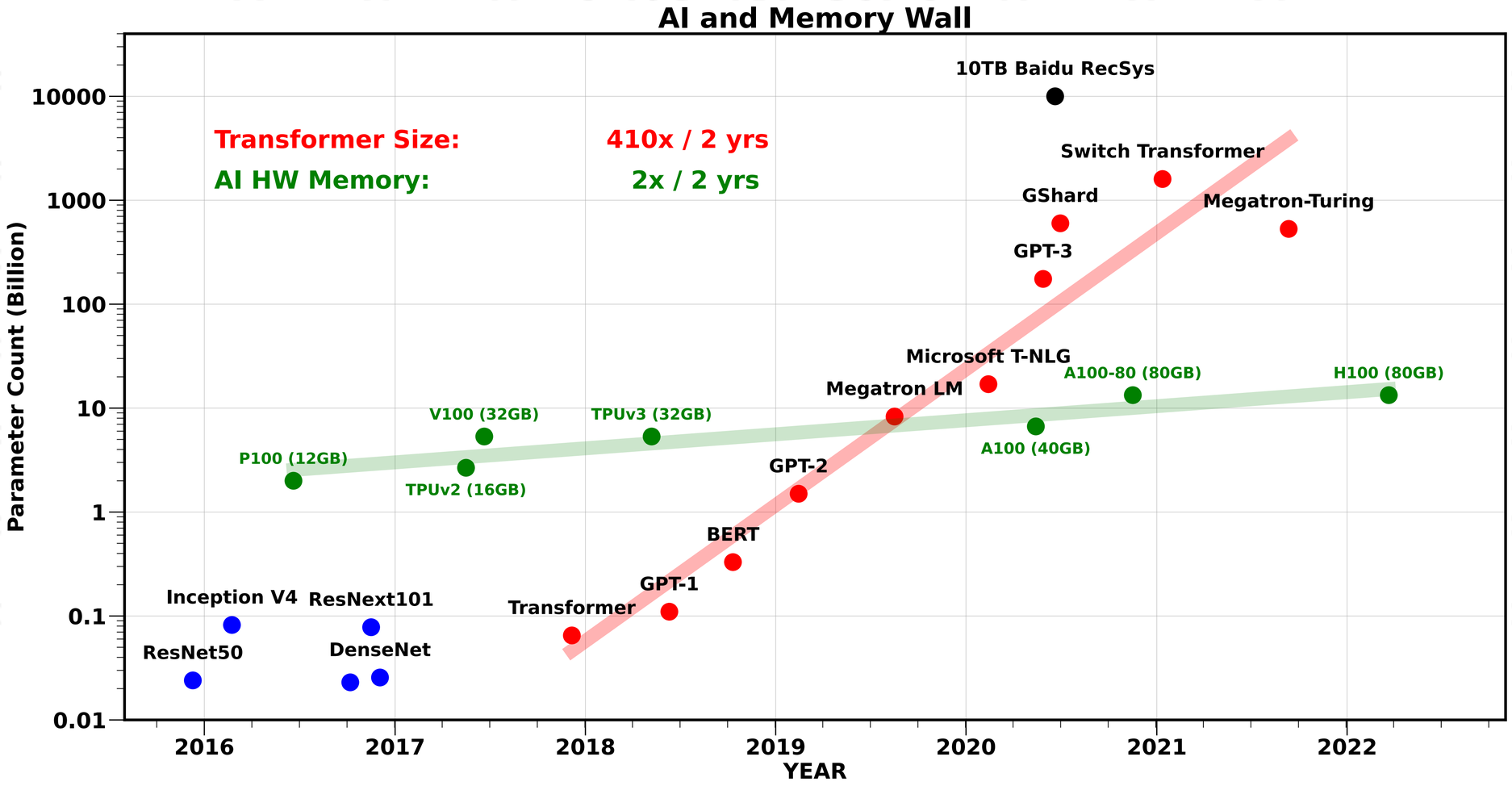
Memory hierarchy

- If data doesn't fit in memory:
  - Use more GPUs - parallel training or inference
- OR offload to slower components in the hierarchy

Typically data is kept here

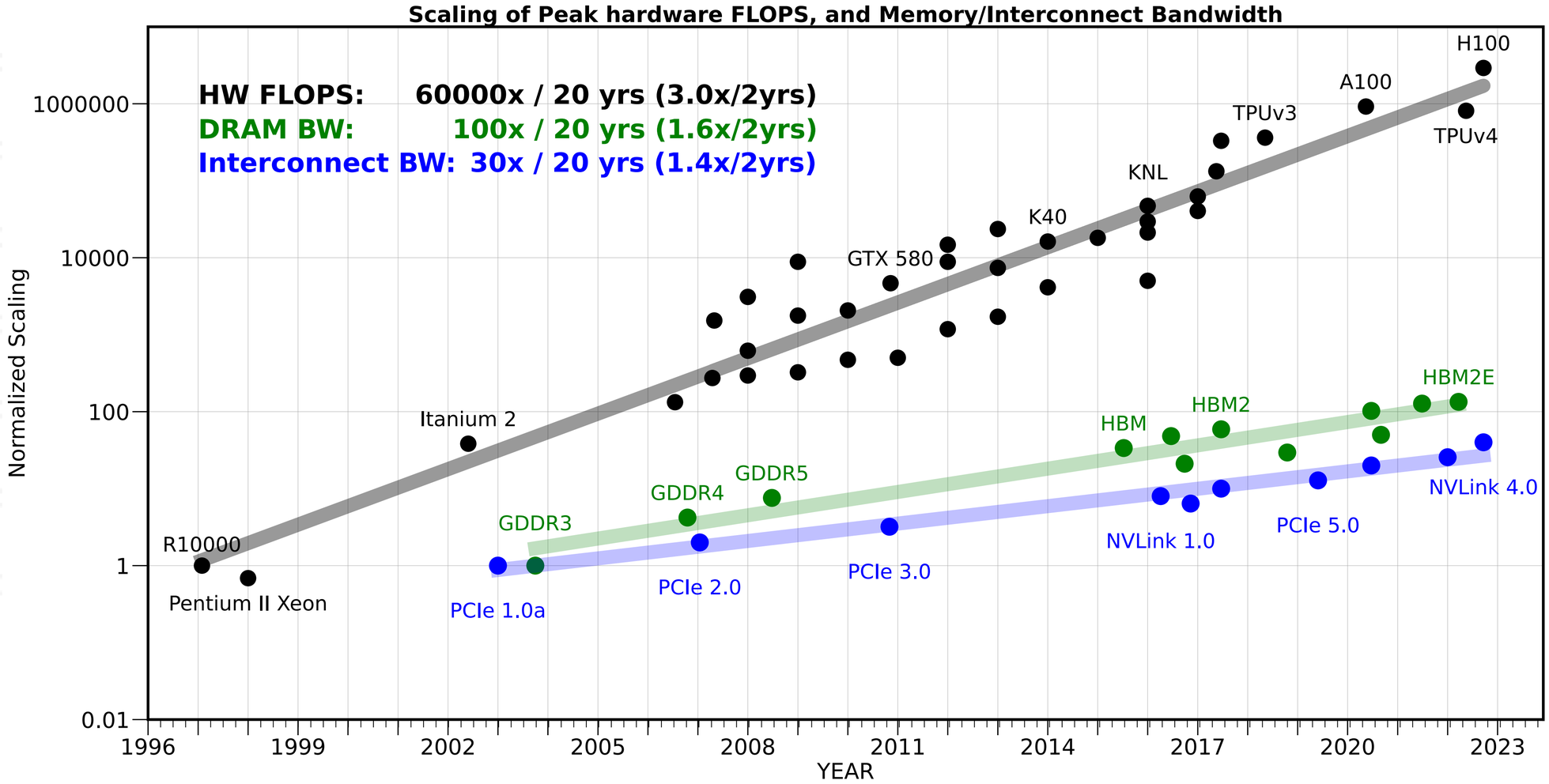


Memory is not scaling with model size



Gholam et al. https://arxiv.org/abs/2403.14123

Compute vs. memory vs. network



Memory offloading

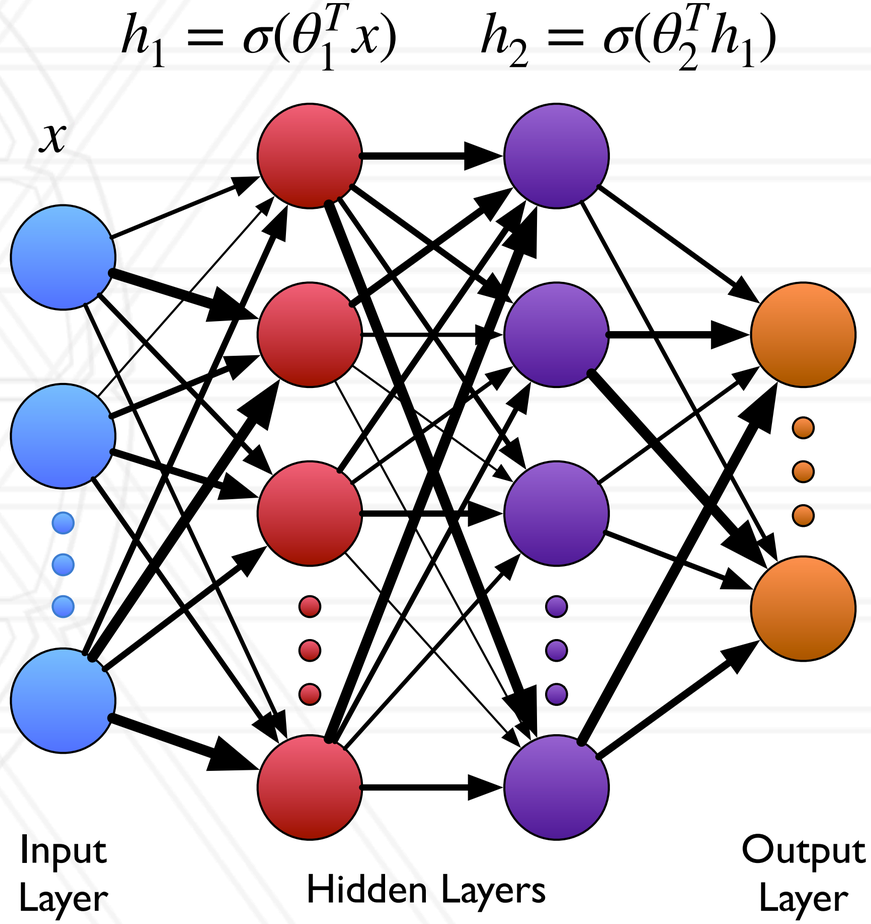
- Can we move data to CPU memory or disk to reduce memory usage on the GPU?
- It will slow things down but could make things feasible which weren't possible before
  - Running large models
  - Working with old hardware

What uses GPU memory?

- Parameters: θ_1, θ_2, …
- Activations: h_1, h_2, …
- Gradients: $\nabla_\theta L$
  - In mixed precision, two copies of gradients are stored: FP16 and FP32
- Optimizer states

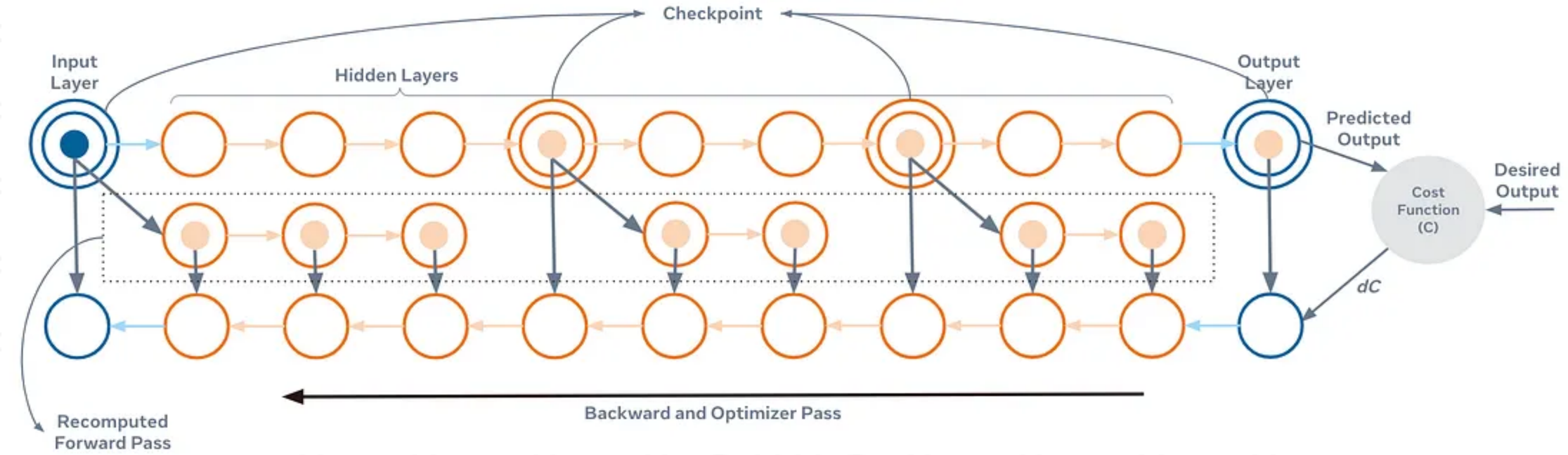
What uses GPU memory?

- Parameters: $\theta_1, \theta_2, \ldots$
- Activations: $h_1, h_2, \ldots$
- Gradients: $\nabla_\theta L$
  - In mixed precision, two copies of gradients are stored: FP16 and FP32
- Optimizer states



Activation checkpointing

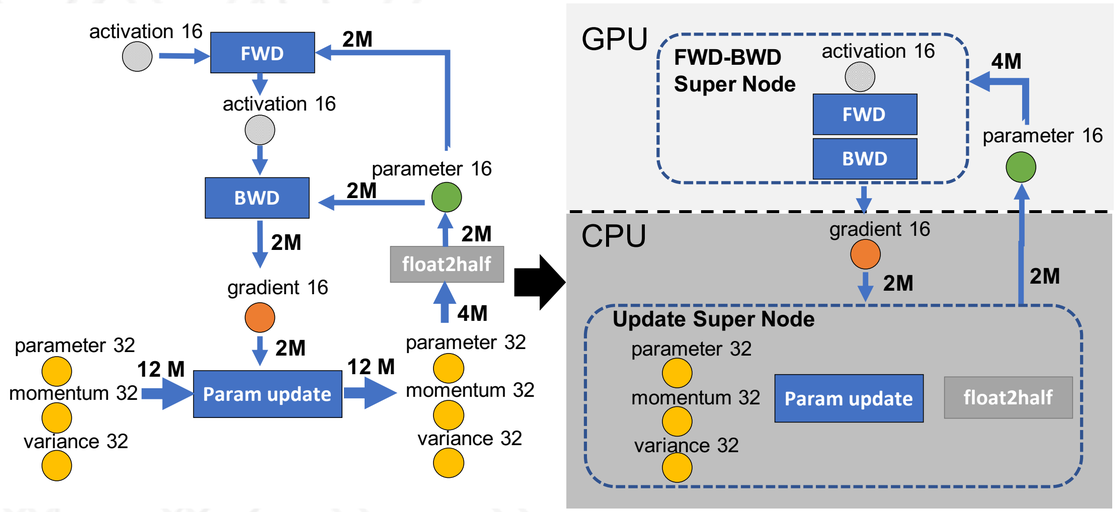
- Use less memory by recomputing some activations



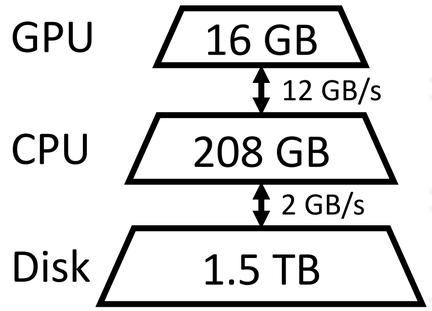
https://shivambharuka.medium.com/deep-learning-a-primer-on-distributed-training-part-1-d0ae0054bb1c

ZeRO-Offload

- Offload data and compute to the CPU
- Models training as a data-flow graph
- Partition graph between CPU and GPU
  - Give less work to CPU
  - Minimize data movement between CPU and GPU
  - Maximize memory savings



FlexGen: Offloading to CPU and Disk

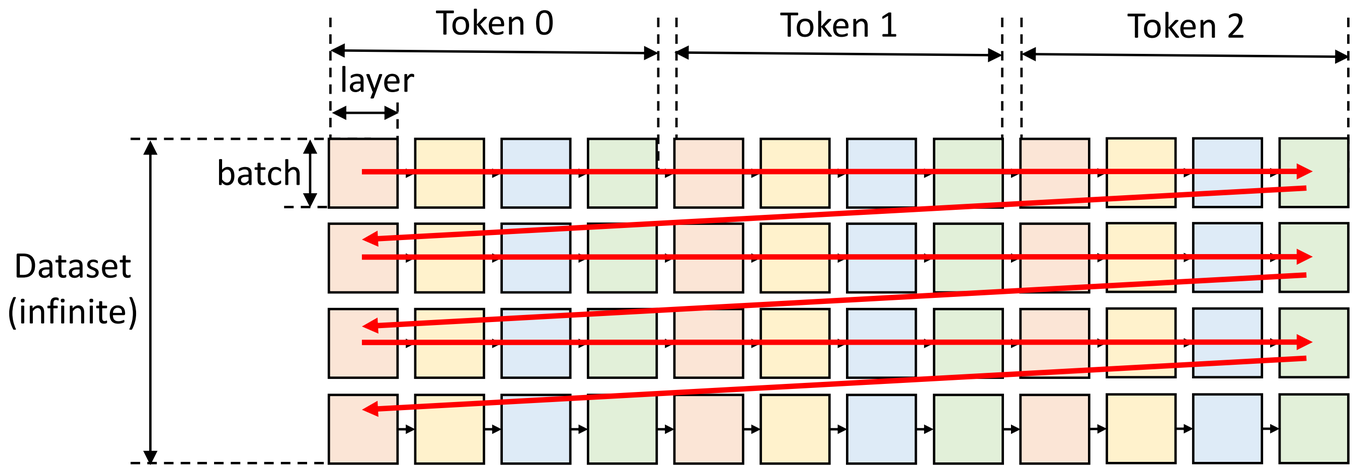


- Focus on single commodity GPU
- Three tensors: weights, activations, KV cache
- Define a search space of possible offloading strategies
- Linear programming based search algorithm to identify good candidates

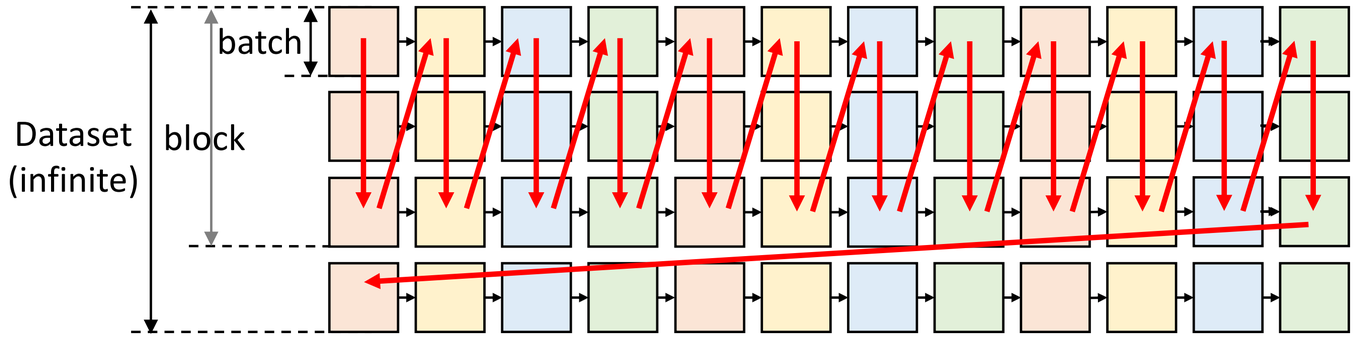
Example schedules

- Derive cost models for prefill and decode time
- Run an LP based search

$$
\begin{aligned}
\min_{p} \quad & T/bls \\
\text{s.t.} \quad & gpu\ peak\ memory < gpu\ mem\ capacity \\
& cpu\ peak\ memory < cpu\ mem\ capacity \\
& disk\ peak\ memory < disk\ mem\ capacity \\
& wg + wc + wd = 1 \\
& cg + cc + cd = 1 \\
& hg + hc + hd = 1
\end{aligned}
$$



(a) Row-by-row schedule



(b) Zig-zag block schedule

Comparison with other frameworks

- Generation throughput (tokens/s) on NVIDIA T4 instances (16 GB)
  - Intel Xeon CPU with 208 GB, SSD with 1.5 TB

Seq. length	512			1024		
Model size	6.7B	30B	175B	6.7B	30B	175B
Accelerate	25.12	0.62	0.01	13.01	0.31	0.01
DeepSpeed	9.28	0.60	0.01	4.59	0.29	OOM
Petals	8.25	2.84	0.08	6.56	1.51	0.06
FlexGen	25.26	7.32	0.69	13.72	3.50	0.35
FlexGen (c)	29.12	8.70	1.12	13.18	3.98	0.42

UNIVERSITY OF
MARYLAND