ODTÜ METU

TÜBİTAK

EUROCC 4SEE

Optimizing Deep Learning Systems for Hardware

Assoc. Prof. Erdem AKAGÜNDÜZ, METU

ncc@ulakbim.gov.tr

Part IV : System-level Optimization

- PIV.a : Parallelism
- PIV.b : Mixed-Precision Training
- PIV.c : Other Techniques

“Other” types of System-level Optimization

- This section is meant to capture additional system-level strategies that improve memory efficiency, training throughput, or hardware utilization
  - but don't fall strictly under "parallelism" or "mixed-precision."
- We aim:
  - Memory savings,
  - compute efficiency,
  - scalability

  beyond parallelism and precision

“Other” types of System-level Optimization

- This section is meant to capture additional system-level strategies that improve memory efficiency, training throughput, or hardware utilization
  - but don't fall strictly under "parallelism" or "mixed-precision."
- We aim:
  - Memory savings,
  - compute efficiency,
  - scalability

  beyond parallelism and precision

Activation Checkpointing

- Definition: Trade computation for memory by re-computing activations during backpropagation instead of storing them all during forward pass.
- Enables training of deeper/larger models without exceeding memory limits.
- System-level: Controlled at runtime or framework level (e.g., PyTorch’s torch.utils.checkpoint).i

Operator Fusion

- Definition: Combines multiple small ops into a single kernel to reduce memory access and kernel launch overhead.
- Reduces latency and improves GPU utilization.
- System-level: Done by compilers/runtimes (e.g., TensorRT, XLA, TorchScript).

Memory Offloading / Paging

- Definition: Move activations or optimizer states to CPU/NVMe when not in use.
- Enables training of very large models even on limited GPU memory.
- System-level: Used in frameworks like DeepSpeed

Zero Redundancy Optimizer

- Definition: Breaks optimizer states, gradients, and parameters across devices to reduce memory redundancy in data-parallel training.
- Scales large models with limited memory per device.
- System-level: Implemented in DeepSpeed; transparent to model code.

Next: Part V

- Part I : Fundamentals
- Part II : Hardware Types & Memory Hierarchy
- Part III : Model-Level Optimizations
- Part IV : System-Level Optimizations
- Part V : Introduction to Scaling Deep Learning in HPC

Thanks!

Co-funded by
the European Union

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101191697. The JU receives support from the Digital Europe Programme and Germany, Türkiye, Republic of North Macedonia, Montenegro, Serbia, Bosnia and Herzegovina.