CASE SCHOOL OF ENGINEERING
CASE WESTERN RESERVE UNIVERSITY

CSDS 451: Designing High Performant Systems for AI

Lecture 25

12/4/2025

Sanmukh Kuppannagari

sanmukh.kuppannagari@case.edu

https://sanmukh.research.st/

Case Western Reserve University

Outline

- Note about where to go from here

Outline

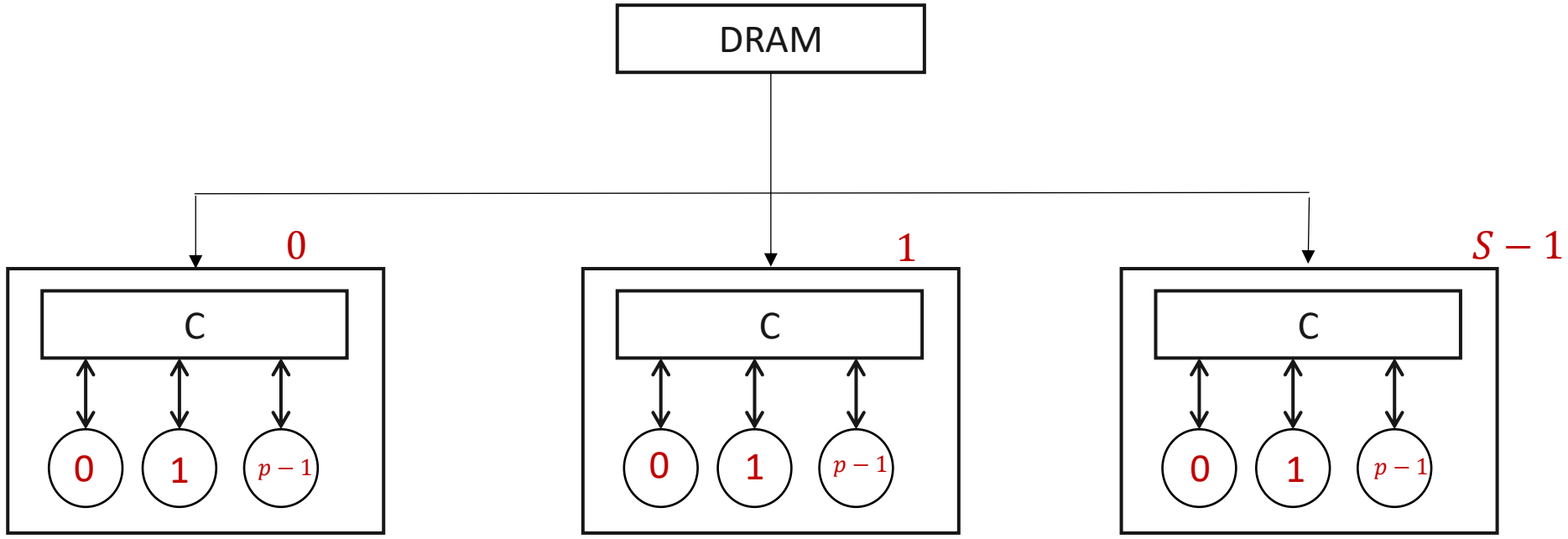
- Note about where to go from here

Trends in AI

- The AI Index Report
- An Yearly report on the trends in AI by Stanford University
- A good resource to keep track of
- https://aiindex.stanford.edu/report/

What did you learn in the course?

- GPU Modeling



- S Blocks
- p Threads per block

GPU Modeling

- Have a mental model of how programs execute on GPUs
- Data parallel programming – all cores execute same instructions on different data
- Within SMP, data sharing and synchronization is relatively easy
- Across SMP, sharing and synchronization is slower

GPU Modeling

- Programming Languages
- We studied HIP
- Learn CUDA too
  - Very similar to HIP

GPU Modeling

- Programming Languages
- Most of Type #1 projects are targeted towards this. You also get to modify various layers of DNN models which again is an important skill
- Explore as much as you can beyond what you proposed in the proposal. Keep on updating your github projects

Parallel Algorithm Design

- Data parallelism is dominating the key computational kernel implementations

- At the entire model level a combination of data and task parallelism is dominating
  - 3D parallelism: data, model/pipeline, tensor parallelism

Parallel Algorithm Design

- Understand key computational kernels and how to accelerate them
- Convolution -> Matrix Multiplication: Blocked Matrix Multiplication
- Sparse Matrix Multiplication -> several sophisticated techniques exist.
- Prefix Scan/Sorting -> Usually recursive doubling based

Parallel Algorithm Design

- At a model level, understand 3D parallelism
- Play around with Pytorch-lightning, if possible, DeepSpeed
- Type #2 projects have focused on this. Explore, put on your resume
- (You should have HPC access for next several weeks, you can try it out and put it on your resumes)

AI/ML Models

- We looked at:
  - Convolutional Neural Networks
  - Transformer Based Models

- We could not look at:
  - Graph Neural Networks
  - Vision Transformers

AI/ML Models

- CNN models, or at least the key kernel – Convolution will remain relevant moving forward
  - 3D Convolution – 3D version of convolution
- Increasingly being incorporated in Vision Transformers
  - Vision Transformers: convolution plus transformer layers
- Critical to biomedical applications
  - Most popular model: U-net

AI/ML Models

- Convolution Use Cases
- One of my previous students needs to run heavy image segmentation models on CPU only client systems in real-time
- Researchers in Biomedical engineering need to find out ways to run image processing extremely high resolution images

AI/ML Models

- Transformer based Models
  - Go way beyond Natural Language Processing
- Genomics, Protein sequences
- Challenge here: How to model long context lengths (millions or billions)
  - Existing LLMs can handle around ~100,000 context length

AI/ML Models

- Modeling Long Contexts
- Specific sparsity patterns (e.g., ring attention) to achieve sequence parallelism
- In general, the sparse attention mask have not been very successful
  - Reason: Existing implementations rely on Dense matrix multiplications
  - We discussed a few techniques in the course. Research on this topic is increasingly becoming relevant

AI/ML Models

- Modeling Long Context
- Fourier Domain Models
- Key Idea: polynomial multiplication in fourier domain takes $O(L\ LogL)$ time (FFT - $O(L\ LogL)$, elementwise produce - $O(L)$)
- Better complexity that attention mechanism - $O(L^2)$
- Illustrative Work: HyenaDNA
  - Can model up to 1 million context lengths on DNA sequences

AI/ML Models

- Graph Neural Networks
- Learning on Graph data
- Important model in network analysis, molecular dynamic simulations, etc.

CASE SCHOOL OF ENGINEERING
CASE WESTERN RESERVE UNIVERSITY

AI/ML Models

- Some models to look out for
- Neural Operators (Caltech) – Gaining popularity in tasks such as weather forecasting
- Geometric Graph Neural Networks – Extensions of GNNs that are gaining popularity in scientific simulations
- Liquid Time Constant Networks – Gaining popularity in robotics

AI/ML Models

- Read their papers, go to their repos.
- See if you can do some interesting self directed projects
- See if you can accelerate them using some techniques that we learned in class
  - Untapped opportunity. No one has looked into accelerating these models yet.

Programming Skills

- Pytorch is good for building models
- But to really run them fast, you need C/C++
  - Its may be frustrating, not many people like it
  - That's what makes it a coveted skill
- Hardware Acceleration – CUDA/Sycl/OpenCL/HIP/ROCm
- DNN Model Programming – Torch C++ Api
  - Look at the philosophy of when to use this: https://pytorch.org/cppdocs/frontend.html

Summary of Current Hot Skills in AI Systems

- Thanks to Ketan Singh, Meta (Last year's Career Talk Speaker)

- Compilers, Distributed systems, ML courses
- Learn graph manipulation frameworks like FX transforms in torch.compile, JAX
- C++ services, Python based model authoring and boundaries in between.
  - Calling c++ libraries via python wrappers for performance and dev efficiency.
- Distributed systems
  - Learn fundamental protocols of communication in a distributed setting.
  - Small companies: May need to build infra (using cloud services)
  - Large companies: Infra is built already, you need to use it properly
  - For model training/serving
    - Think about data transfers between hosts
    - Single host efficiency: CPU/GPU communication, pipelining instructions on GPU

Next Class

- Life as an AI Engineer?

CASE SCHOOL OF ENGINEERING
CASE WESTERN RESERVE UNIVERSITY

Thank You

- Questions?

- Email: sanmukh.kuppannagari@case.edu