ASTRA-sim: Enabling SW/HW Co-Design Exploration for Distributed Deep Learning Training Platforms

Tushar Krishna
Associate Professor
School of ECE
Georgia Institute of Technology

tushar@ece.gatech.edu

http://tusharkrishna.ece.gatech.edu
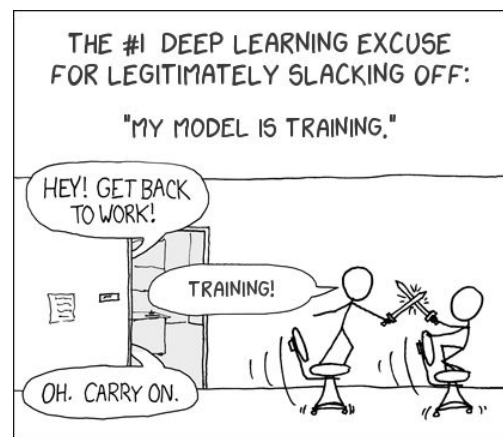
Modsim Workshop

October 7, 2021

Joint Work with Saeed Rashidi, William Won (Georgia Tech), Srinivas Sridharan (Facebook), Sudarshan
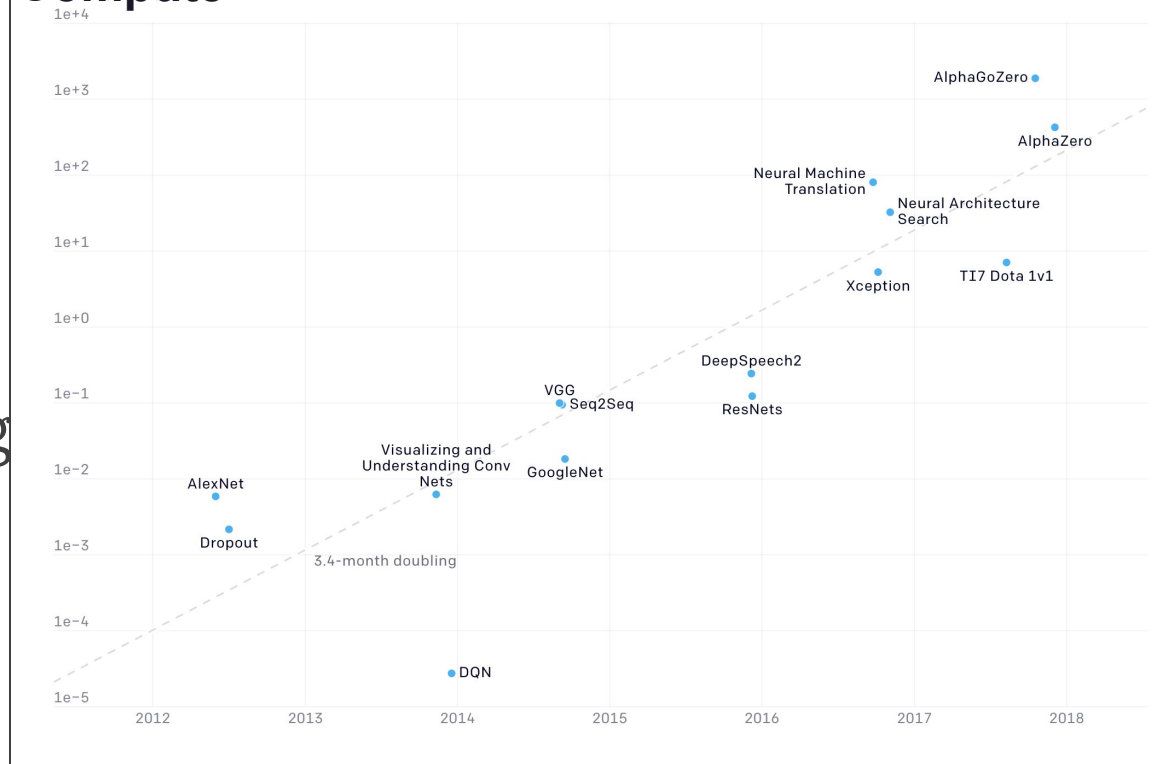
Deep Learning Training Challenge

- **Training time is increasing**
  - DNNs are becoming larger
    - Megatron LM: 8.3B Parameter
    - Turning-NLG: 17.2 B Parameters
    - GPT-3: 175B Parameters
    - Switch Transformer: 1T Parameters
  - Training samples are becoming larg
  - Moore's Law has ended

Training a deep neural network (DNN) involves feeding it a training dataset to update its weights to model the underlying function representing the dataset
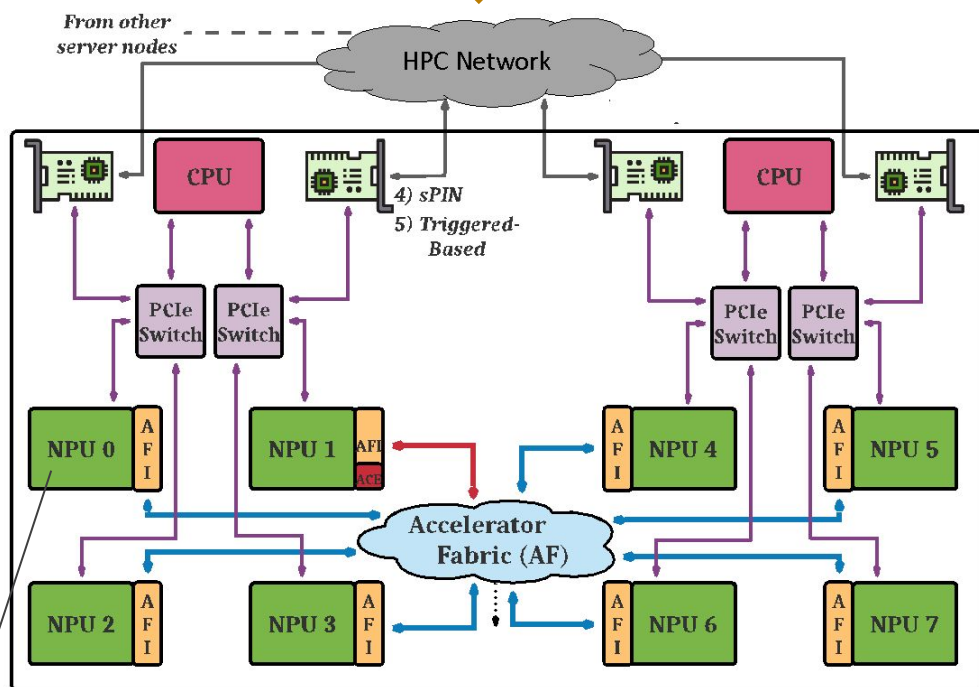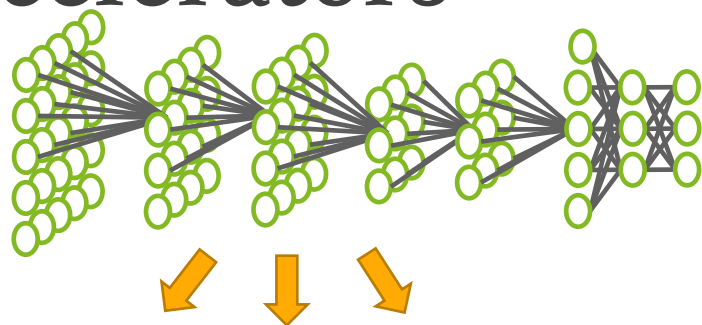
THE #1 DEEP LEARNING EXCUSE FOR LEGITIMATELY SLACKING OFF:

"MY MODEL IS TRAINING."

HEY! GET BACK TO WORK!

TRAINING!

OH. CARRY ON.

AlexNet to AlphaGo Zero: A 300,000x Increase in Compute

Source: https://openai.com/blog/ai-and-compute/

Solution: Distributed Training over Accelerators

GPU/TPU

HPC Network Link (e.g., Infiniband, Ethernet) PCIe link AFI Network Link (e.g., NVlink) NIC

Challenges

- **Communication!**
  - Inevitable in any distributed algorithm

- **What does communication depend on?**
  - **synchronization scheme:** synchronous vs. asynchronous
  - **parallelism approach:** *data-parallel, model-parallel, hybrid-parallel, pipeline-parallel*

- **What is the communication pattern?**
  - Collective
    - All-Reduce (Reduce-Scatter + All-Gather)
    - All-to-All

- **Is it a problem?**
  - Depends … can we hide it behind compute?

Understanding the Design-space for DL Training

We can view the design-space as three layers:

- **Workload layer (*aka* training loop):**
  - Parallelism approach
  - Communication size & type
  - Dependency order

- **System layer:**
  - Collective communication algorithm
  - Message sizes
  - Schedule of compute
  - Schedule of collectives

- **Network layer:**
  - Physical topology
  - Congestion control, transport protocol
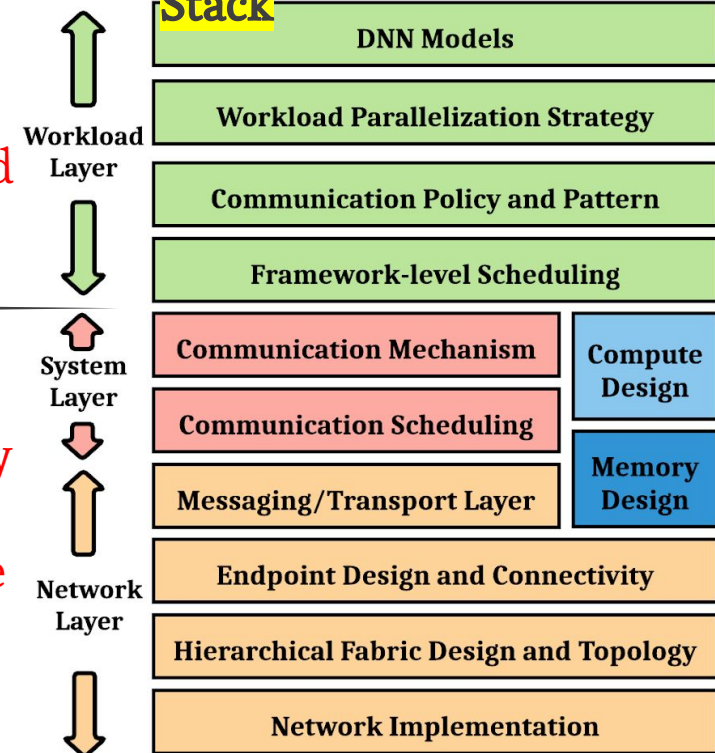 - Link BW, latency, buffers, routing algorithm

Implemented by libraries like NVIDIA NCCL.

Not too many tools to explore these aspects

Many existing tools (e.g., Garnet, Booksim, NS3)

DL Training Co-Design Stack

Workload Layer
- DNN Models
- Workload Parallelization Strategy
- Communication Policy and Pattern
- Framework-level Scheduling

System Layer
- Communication Mechanism
- Communication Scheduling
- Messaging/Transport Layer

Compute Design

Memory Design

Network Layer
- Endpoint Design and Connectivity
- Hierarchical Fabric Design and Topology
- Network Implementation

S. Rashidi et al.,"**ASTRA-SIM: Enabling SW/HW Co-Design Exploration for Distributed DL Training Platforms**", ISPASS 2020

*Figure Courtesy:
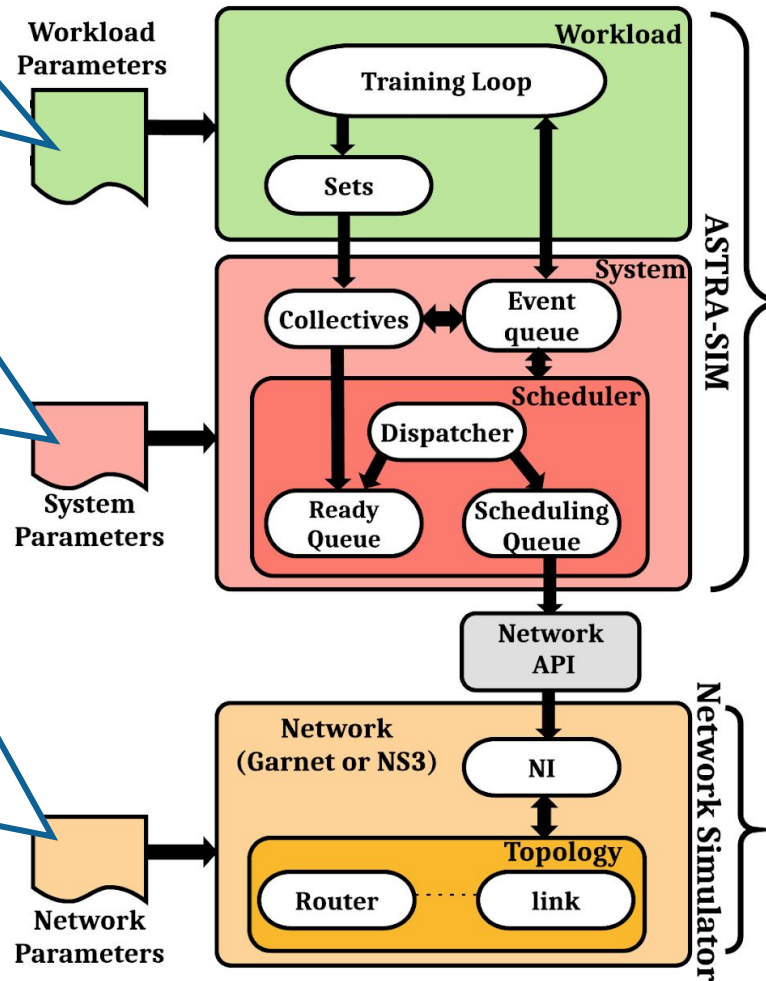Srinivas Sridharan (Facebook)*

Modeling the Design-space using ASTRA-sim

- ✔ Supports Data-Parallel, Model-Parallel, Hybrid-Parallel training loops
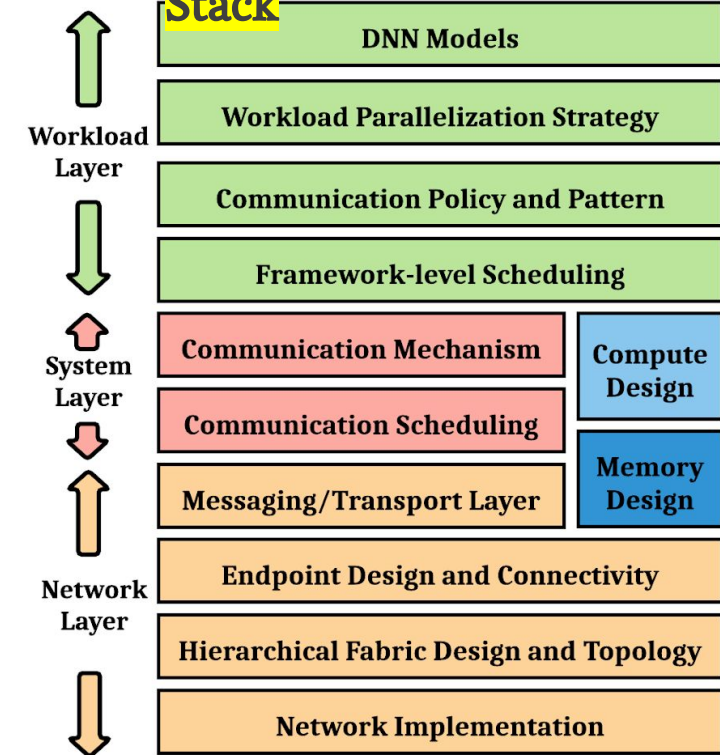- ✔ Extensible to more training loops

- ✔ Ring based, Tree-based, AlltoAll based, and multi-phase collectives
- ✔ Variety of scheduling policies
- ✔ Compute times fed via offline system measurements or compute simulator

- ✔ Various topologies, flow-control, link bandwidth, congestion control
- ✔ Plug-and-play options
  - ✔ Analytical (roofline) - released
  - ✔ Garnet (credit-based) - released
 - ✔ NS3 (TCP, RDMA) - to be released

http://github.com/astra-sim/astra-sim

Workload Parameters → **Workload** (ASTRA-SIM): Training Loop → Sets

System Parameters → **System**: Collectives ↔ Event queue; Scheduler: Dispatcher → Ready Queue, Scheduling Queue

Network API

Network Parameters → **Network (Garnet or NS3)** (Network Simulator): NI → Topology: Router ···· link

DL Training Co-Design Stack

Workload Layer:
- DNN Models
- Workload Parallelization Strategy
- Communication Policy and Pattern
- Framework-level Scheduling

System Layer:
- Communication Mechanism
- Communication Scheduling
- Messaging/Transport Layer
- Compute Design
- Memory Design

Network Layer:
- Endpoint Design and Connectivity
- Hierarchical Fabric Design and Topology
- Network Implementation

*S. Rashidi et al.,"**ASTRA-SIM: Enabling SW/HW Co-Design Exploration for Distributed DL Training Platforms**", ISPASS 2020*

*S. Rashidi, et al., "**Scalable Distributed Training of Recommendation Models: An ASTRA-SIM + NS3 case-study with TCP/IP transport**", Hot Interconnects 2020*
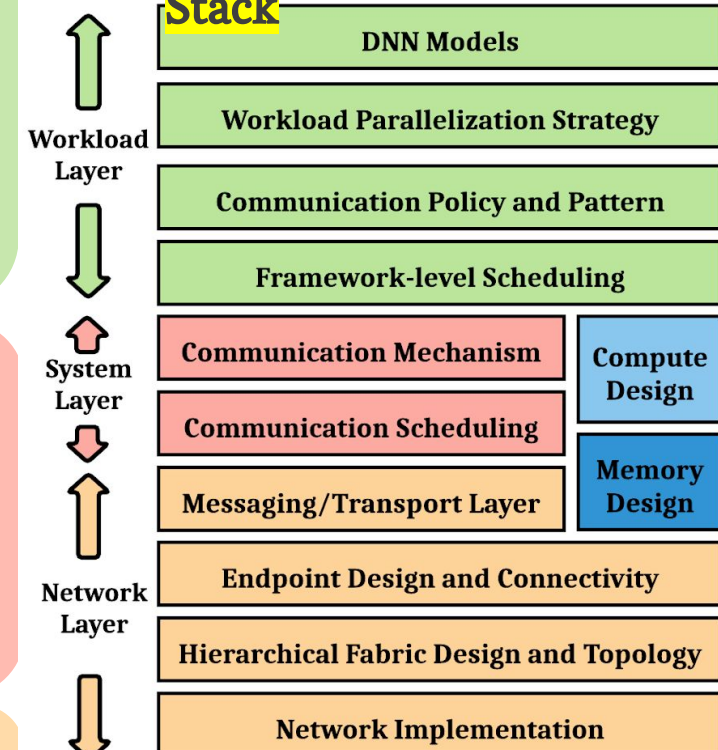
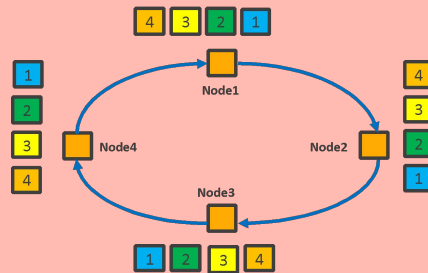# Navigating the Inter-dependent Design-space
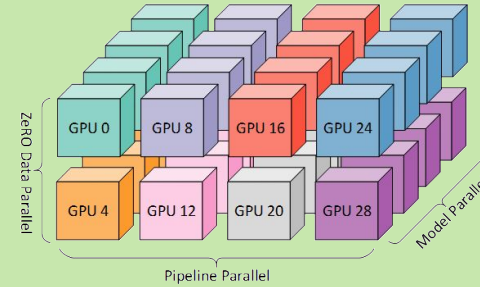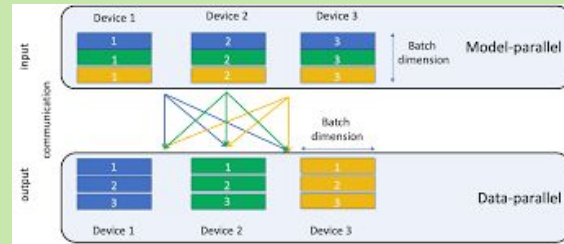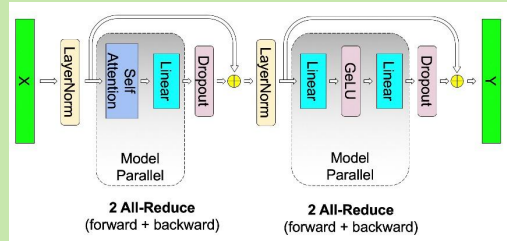
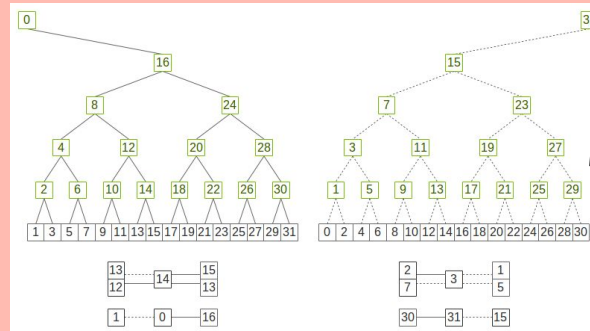Megatron LM (NVIDIA)

DLRM (Facebook)

ZeRO (Microsoft)

Ring All Reduce

Double-binary Tree All Reduce

Torus (Google Cloud TPU)

Pt-to-Pt (NVIDIA DGX-1)

Switch (NVIDIA DGX-2)

DL Training Co-Design Stack

DNN Models	
Workload Parallelization Strategy	Workload Layer
Communication Policy and Pattern	
Framework-level Scheduling	
Communication Mechanism	**Compute Design**
Communication Scheduling	System Layer
Messaging/Transport Layer	**Memory Design**
Endpoint Design and Connectivity	
Hierarchical Fabric Design and Topology	Network Layer
Network Implementation	

*S. Rashidi et al.,"**ASTRA-SIM: Enabling SW/HW Co-Design Exploration for Distributed DL Training Platforms**", ISPASS 2020*

http://github.com/astra-sim/astra-sim