https://synergy.ece.gatech.edu

ISPASS 2023
April 25, 2023

ASTRA-sim2.0:
Modeling Hierarchical Networks and Disaggregated Systems for Large-model Training at Scale

William Won[1*] Taekyung Heo[1*] Saeed Rashidi[1*] Srinivas Sridharan[2] Sudarshan Srinivasan[3] Tushar Krishna[1]






[1]Georgia Institute of Technology [2]Meta [3]Intel [*]Equal contribution

Joint work with Georgia Tech, Meta, and Intel

https://astra-sim.github.io

paper

Outline

- Distributed Training
- Background
- ASTRA-sim
- Limitations of ASTRA-sim
- ASTRA-sim2.0
- Case Studies and Results
- Conclusion

Large Language Models (LLMs)

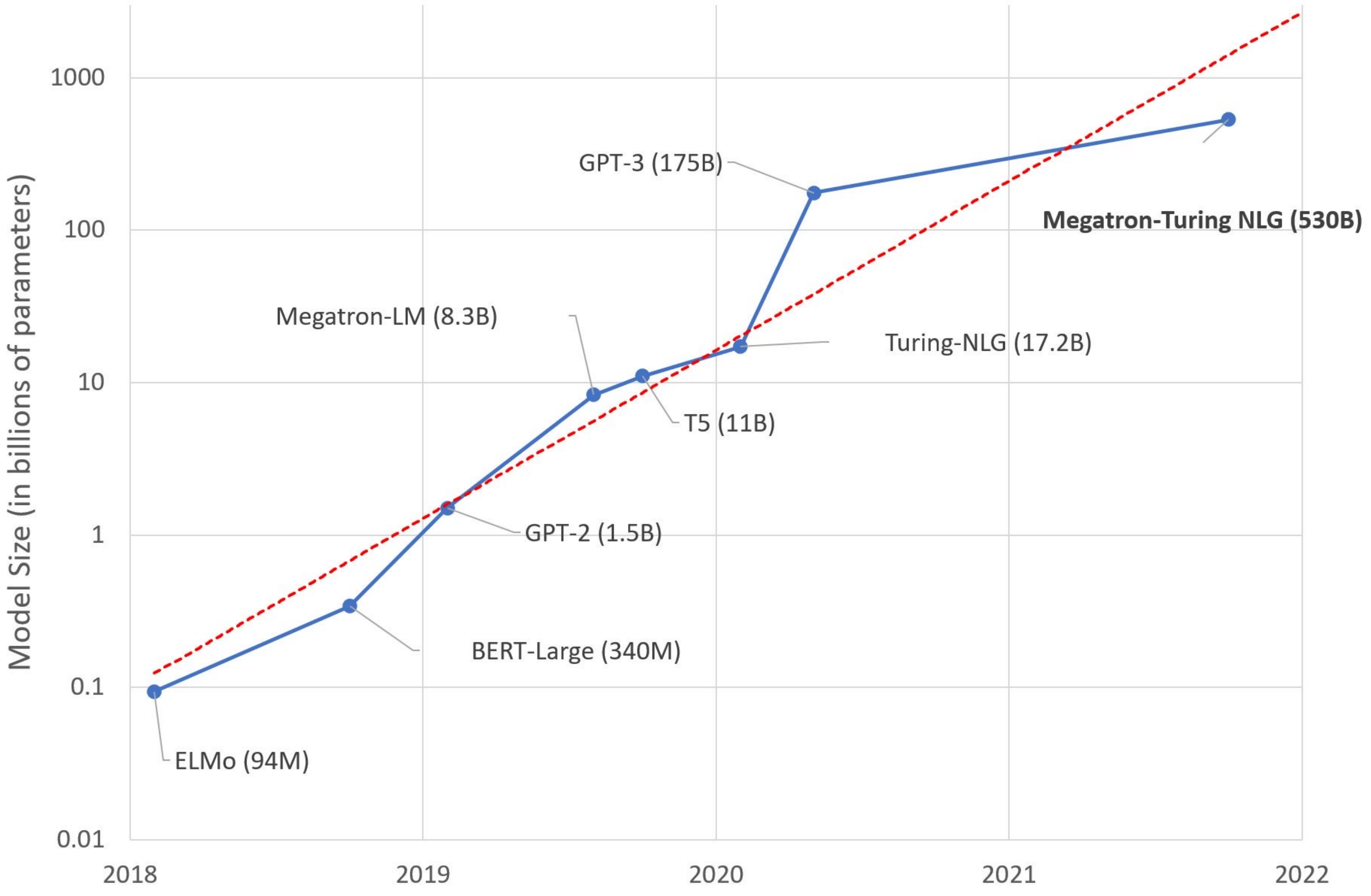
ChatGPT

Bard AI

cerebras

Models are Becoming Larger

- Deep learning (DL) **models are scaling** at an unprecedented rate



https://huggingface.co/blog/large-language-models

Dataset is also Becoming Larger

BA How large was your training dataset?

As a language model, I am a pre-trained model that has been trained on a large corpus of text data to learn patterns and relationships between words, phrases, and sentences. The specific dataset used to train me depends on the version of the GPT architecture that I am built on top of.

For example, the largest version of the GPT architecture, GPT-3, was trained on a diverse and massive dataset of web pages, books, and other sources of text data, totaling over 570 GB of uncompressed text data. This dataset included a wide range of sources, including books, websites, and even programming code, to help provide a broad and comprehensive understanding of the English language.

"570 GB of uncompressed text data"

"1.56 trillion words"

How large was your training dataset?

View other drafts

My training dataset was 1.56 trillion words of text and code. This dataset was collected from a variety of sources, including books, articles, code, and human conversations. The dataset was carefully curated to ensure that it was high-quality and representative of the real world.

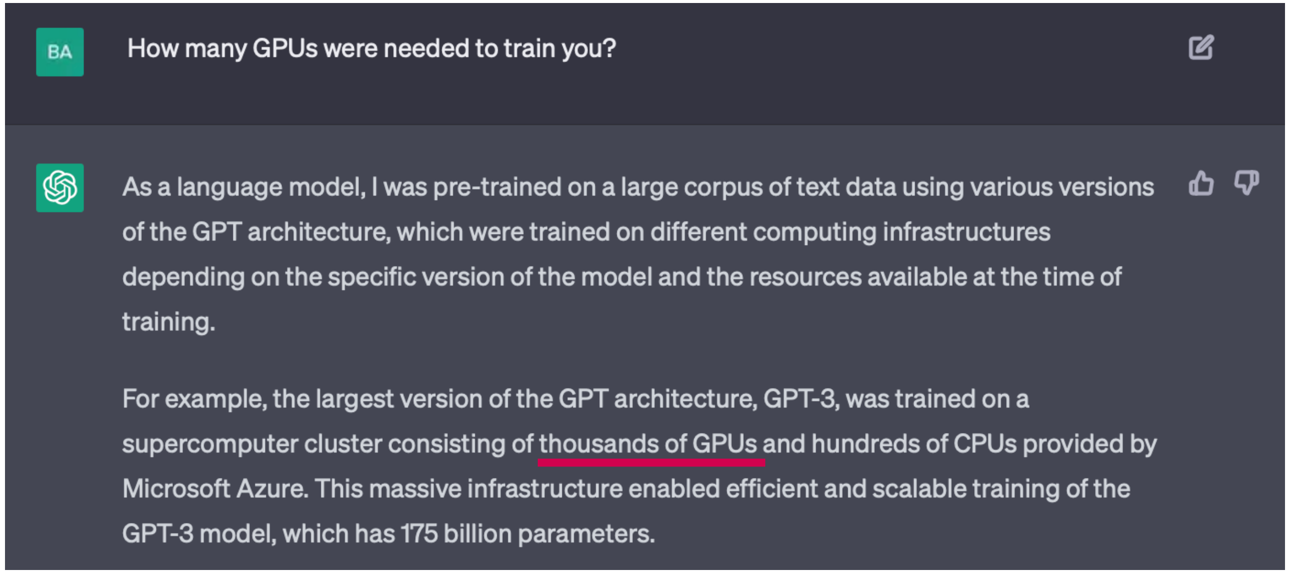
Training is a Key Challenge

- Trillion-parameter Models
  - Zeta-scale floating-point operations
  - 10s of TB of memory

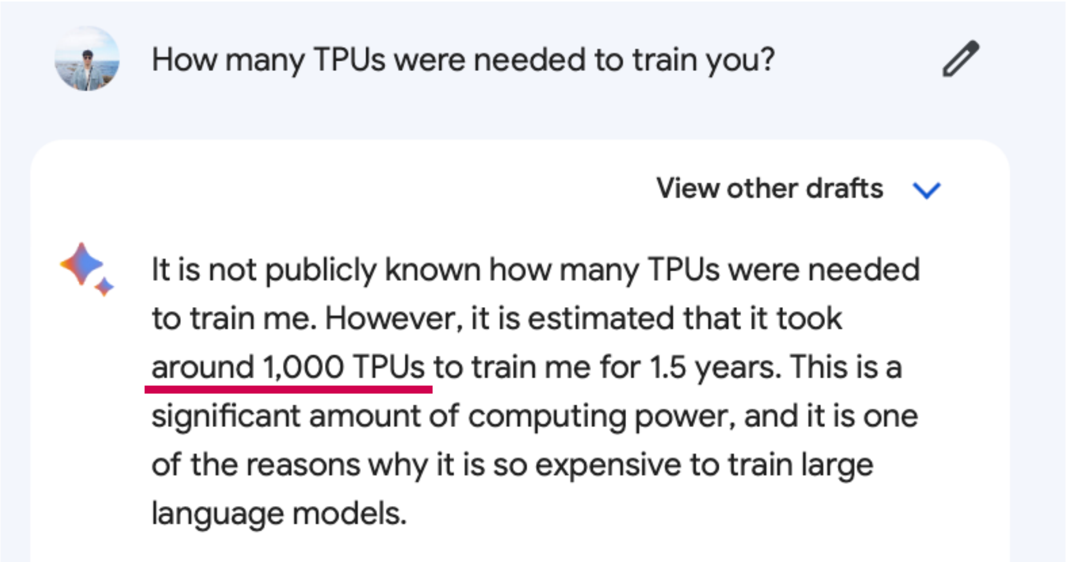
- Impractical to be trained on a single GPU
  - **355 GPU-years to train** (using NVIDIA V100)

Distributed Training is Inevitable

- **Shard model/data across NPUs** (Neural Processing Unit)



"thousands of GPUs"



"around 1,000 TPUs"

Distributed Training is Inevitable

- **Shard model/data across NPUs** (Neural Processing Unit)

Distributed training is necessitated

supercomputer cluster consisting of thousands of GPUs and hundreds of CPUs provided by Microsoft Azure. This massive infrastructure enabled efficient and scalable training of the GPT-3 model, which has 175 billion parameters.

"thousands of GPUs"

significant amount of computing power, and it is one of the reasons why it is so expensive to train large language models.

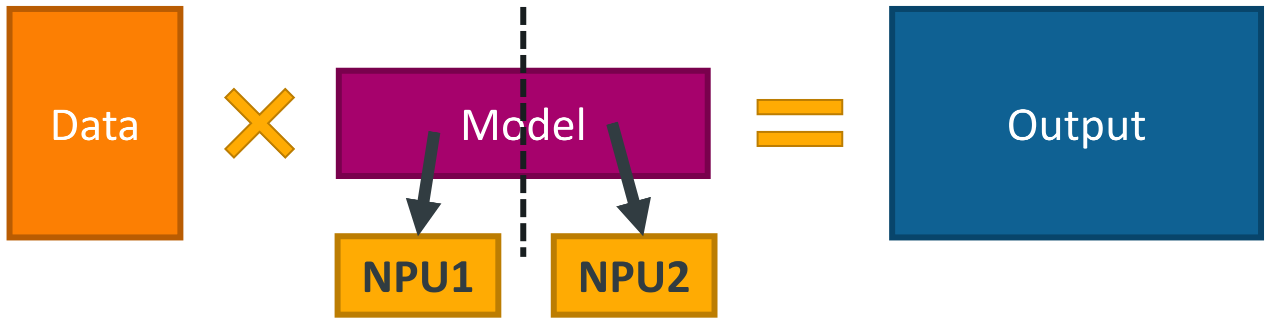
"around 1,000 TPUs"

Outline

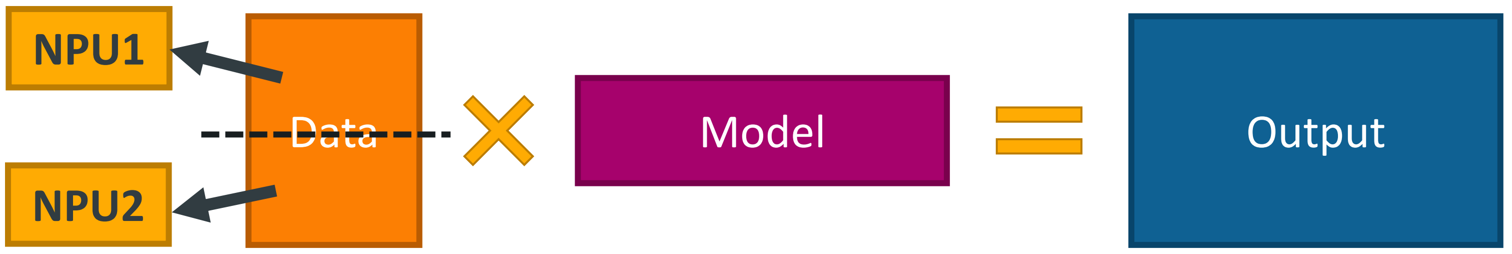
- Distributed Training
- Background
- ASTRA-sim
- Limitations of ASTRA-sim
- ASTRA-sim2.0
- Case Studies and Results
- Conclusion

Parallelization Strategy

- Model Parallelism (MP)



- Data Parallelism (DP)



Design-space of Distributed Training

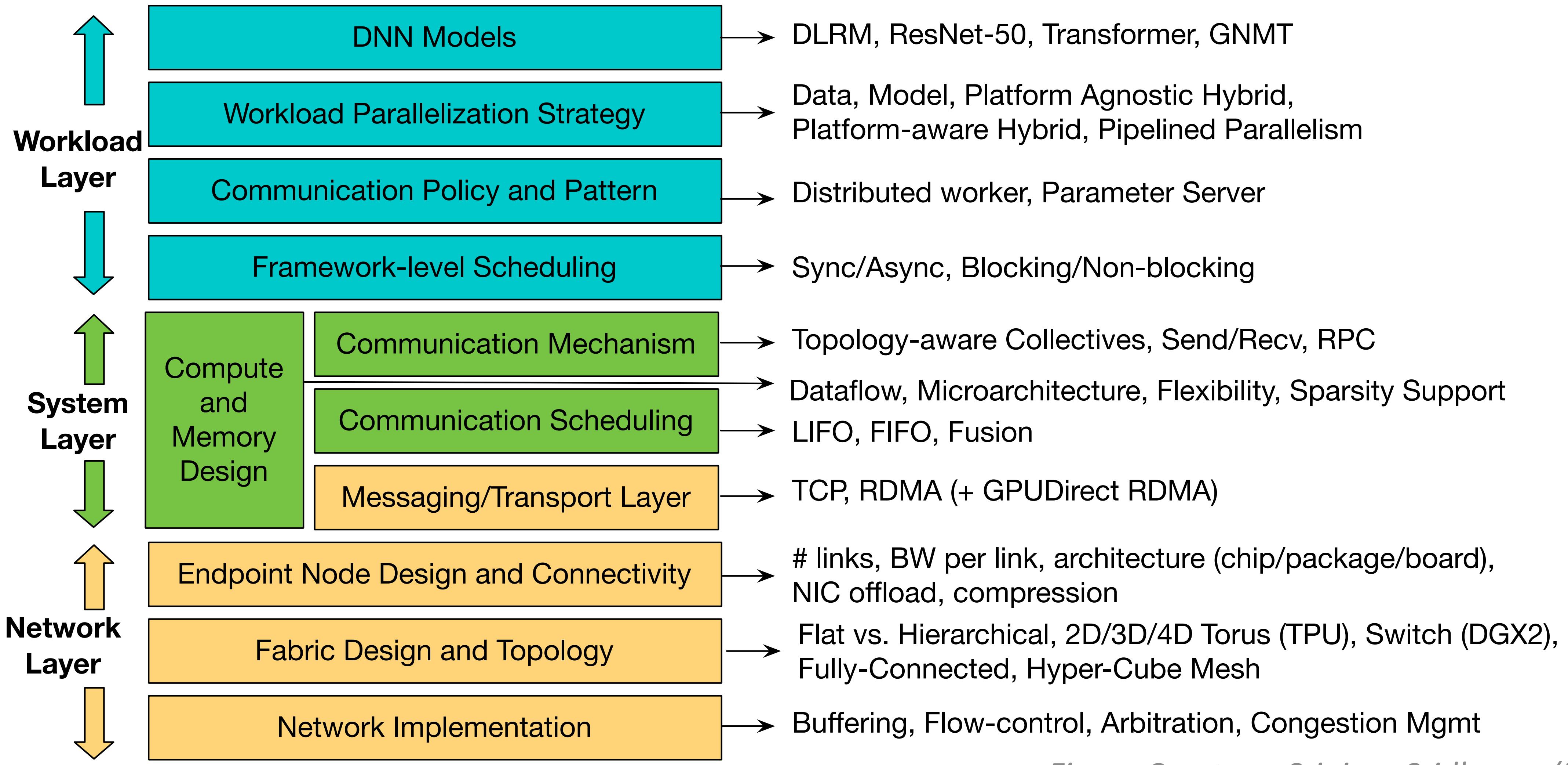
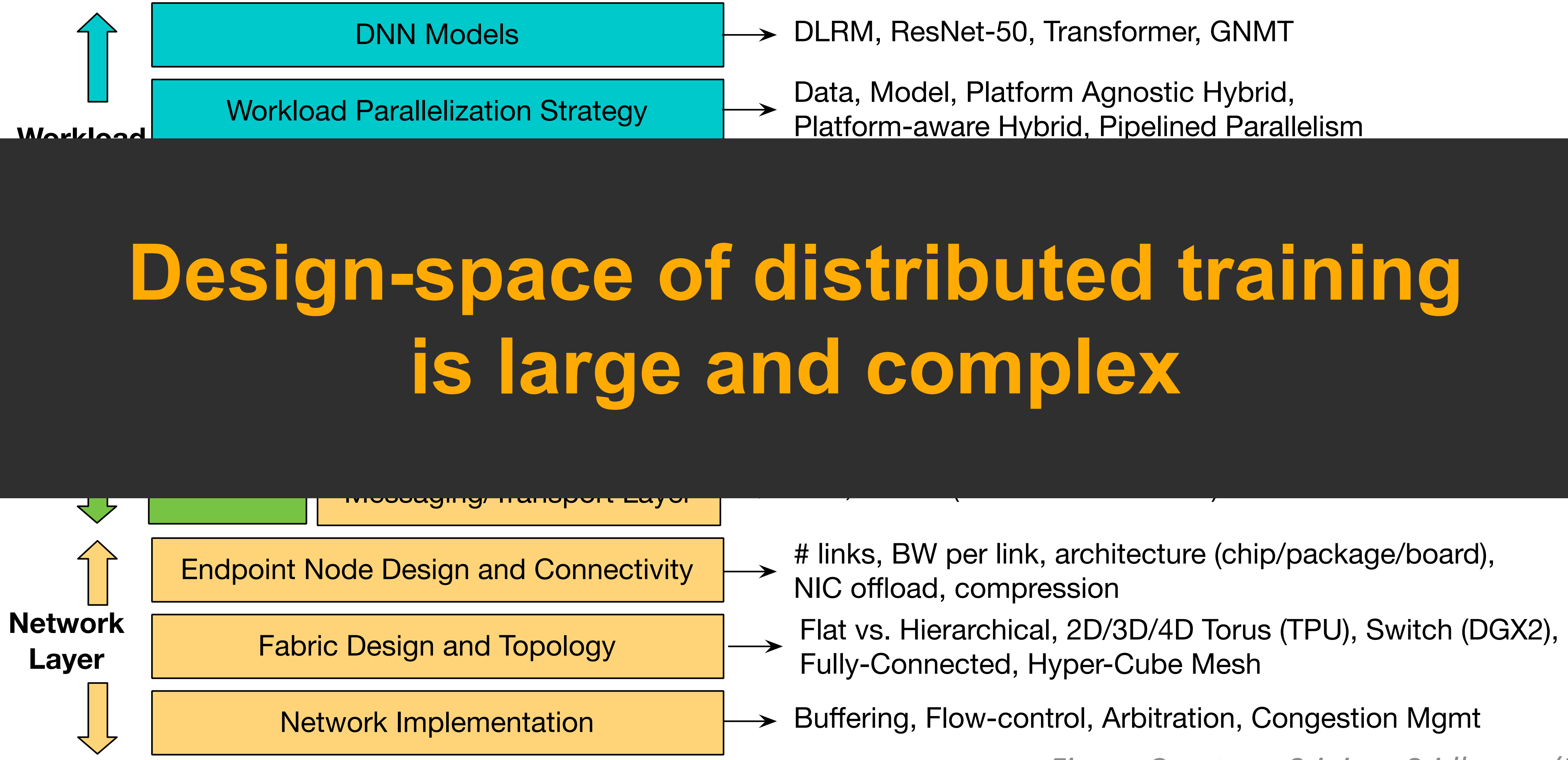


Figure Courtesy: Srinivas Sridharan (Meta)

Design-space of Distributed Training



Design-space of distributed training is large and complex

Figure Courtesy: Srinivas Sridharan (Meta)

Outline

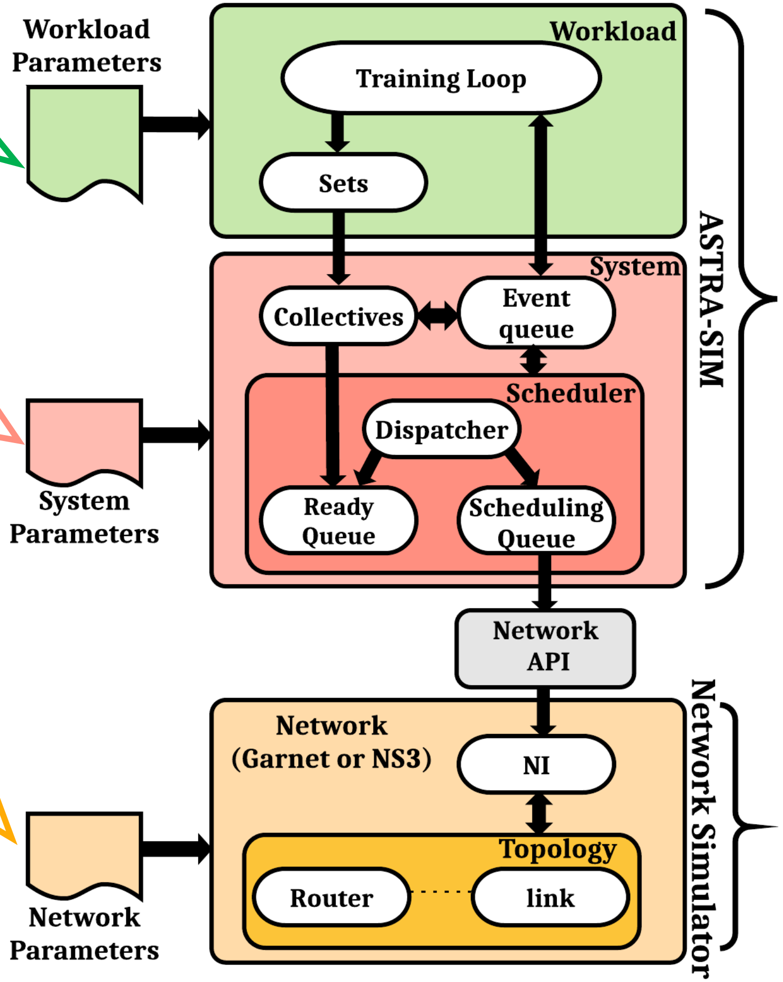
- Distributed Training
- Background
- ASTRA-sim
- Limitations of ASTRA-sim
- ASTRA-sim2.0
- Case Studies and Results
- Conclusion

ASTRA-sim

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

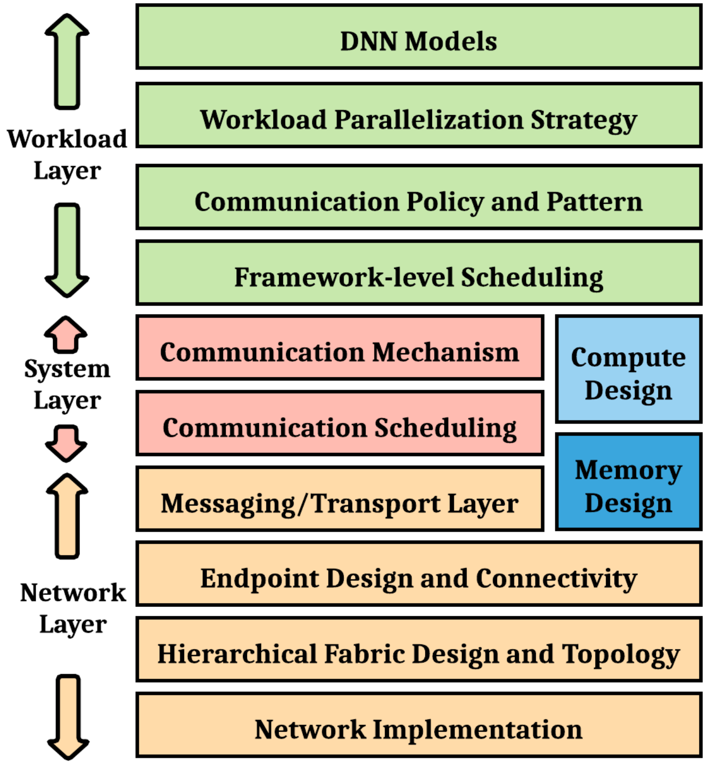
- ✓ Ring based, Tree-based, All-to-All based, and multi-phase collectives
- ✓ Variety of scheduling policies
- ✓ Compute times fed via offline system measurements or compute simulator

- ✓ Various topologies, flow-control, link bandwidth, congestion control
- ✓ Plug-and-play options
  - ✓ Garnet (credit-based)



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

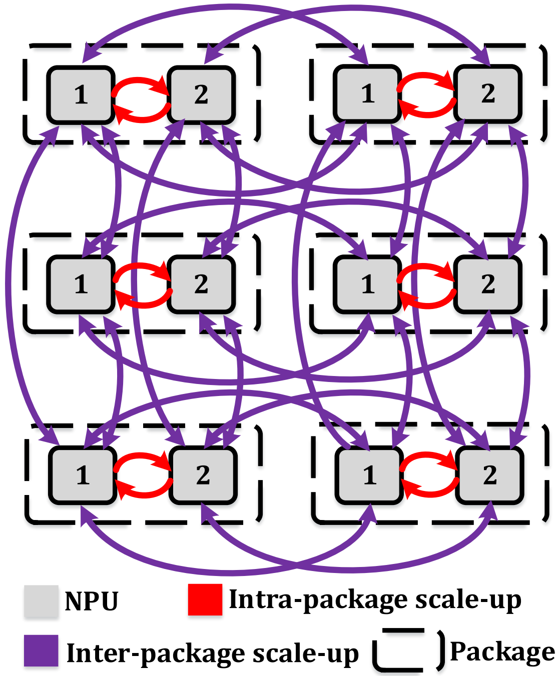
DL Training Co-Design Stack



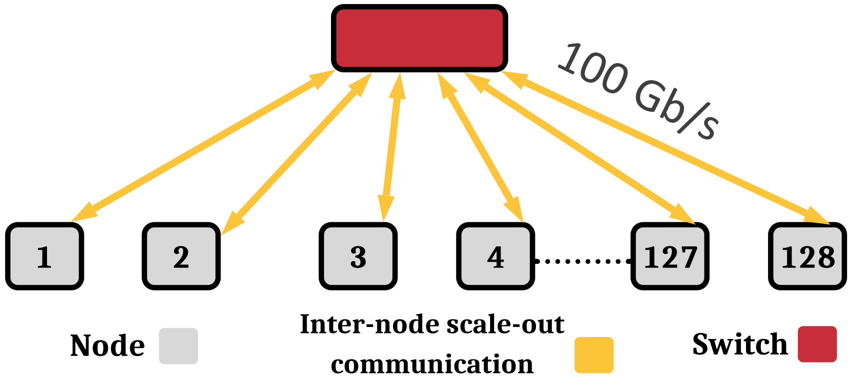
S. Rashidi et al.,"***ASTRA-SIM: Enabling SW/HW Co-Design Exploration for Distributed DL Training Platforms***", ISPASS 2020

S. Rashidi, et al., "***Scalable Distributed Training of Recommendation Models: An ASTRA-SIM + NS3 case-study with TCP/IP transport***", Hot Interconnects 2020

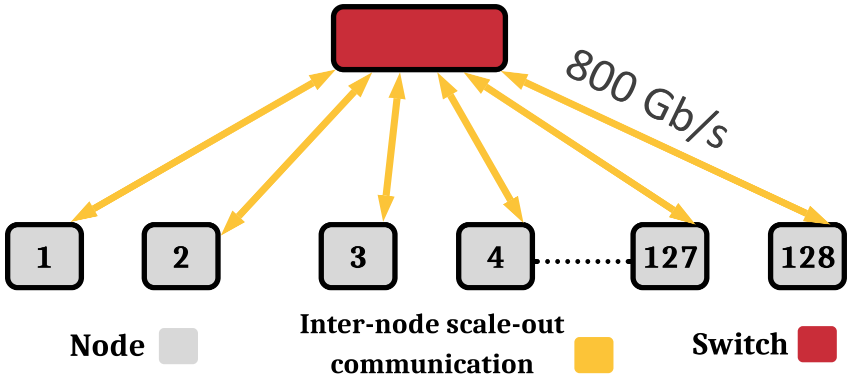
ASTRA-sim Capabilities



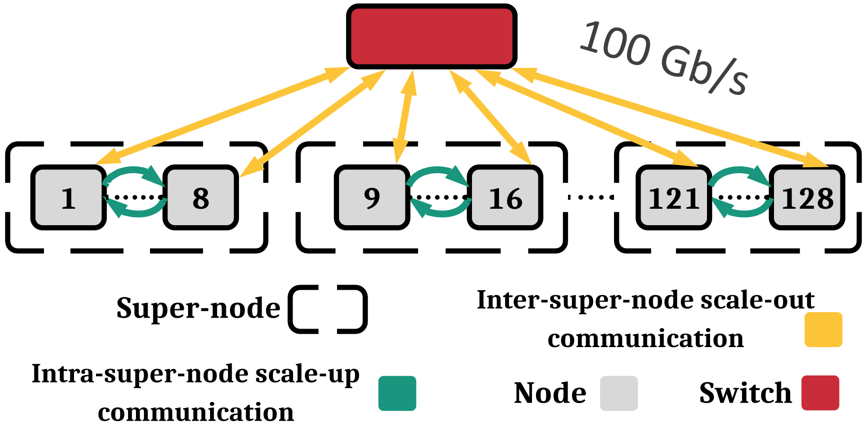
Torus3D



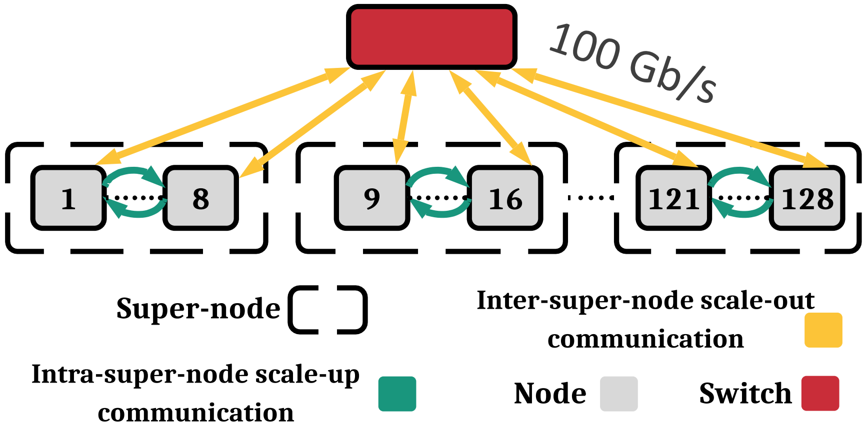
Flat100G



Flat800G

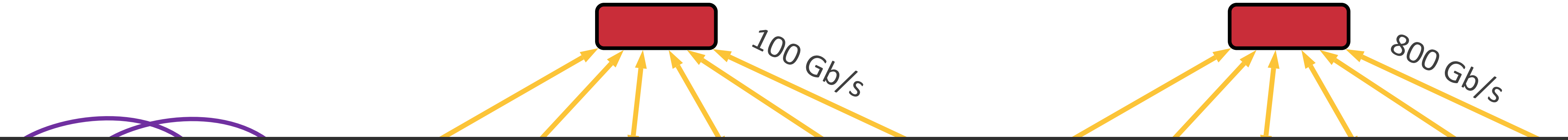


Hier

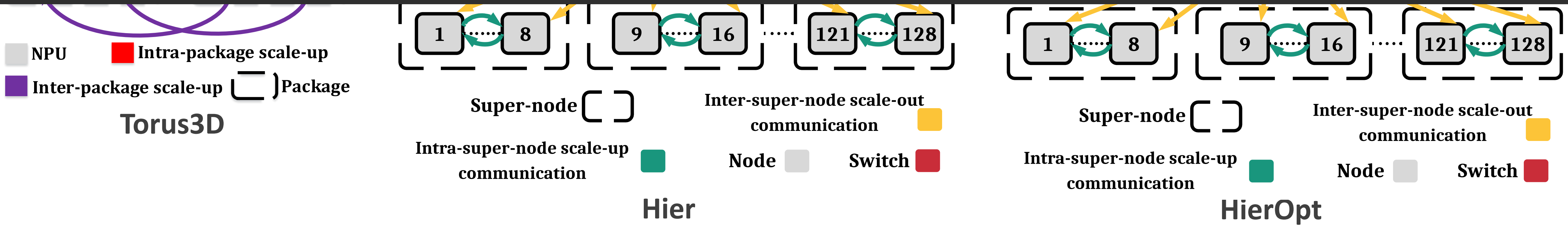


HierOpt

ASTRA-sim Capabilities



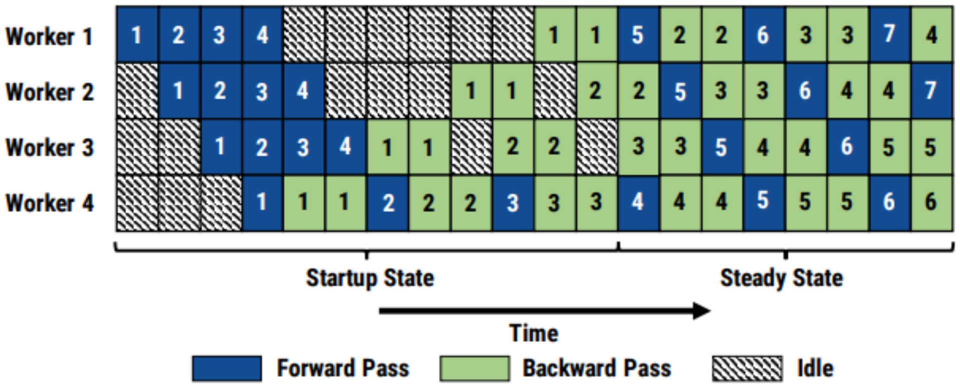
ASTRA-sim captures/simulates complex design-space of distributed training



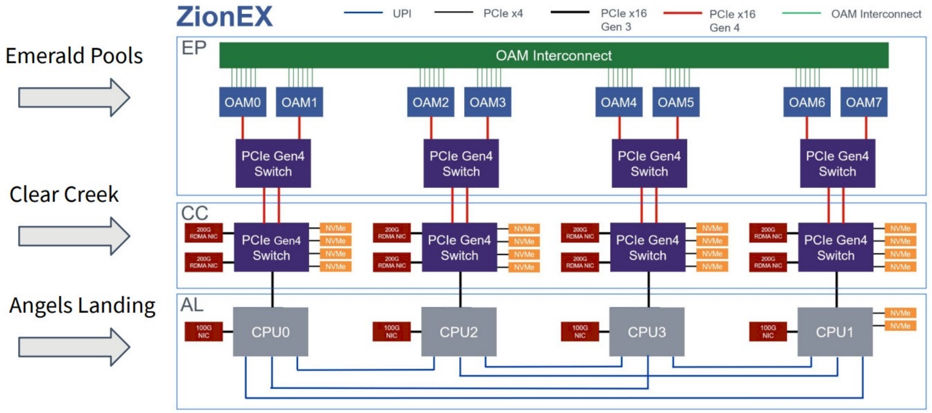
Outline

- Distributed Training
- Background
- ASTRA-sim
- Limitations of ASTRA-sim
- ASTRA-sim2.0
- Case Studies and Results
- Conclusion

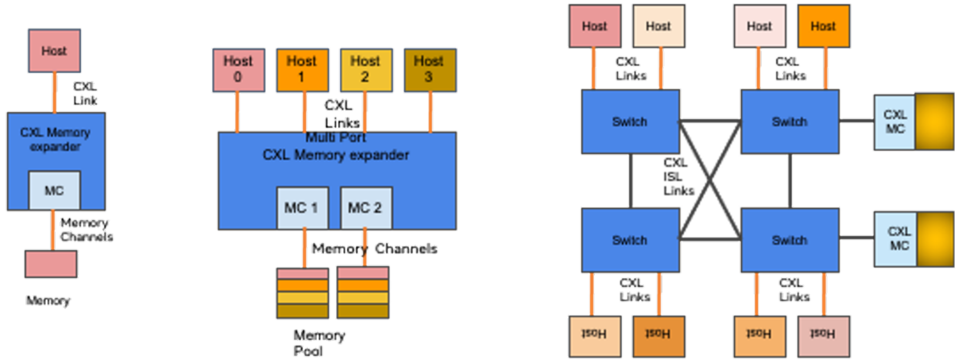
Emerging Platforms



Pipeline Parallelism



Multi-dimensional Networks



Local Memory

Pooled Memory

Memory Switches/Systems

Novel Memory Systems through CXL

Limitations of ASTRA-sim

- Rigid **parallelization strategy**
- Pre-defined **network topology** with limited scale
- Lack of **memory system modeling**

Limitations of ASTRA-sim

- Rigid **parallelization strategy**
- Pre-defined **network topology** with limited scale

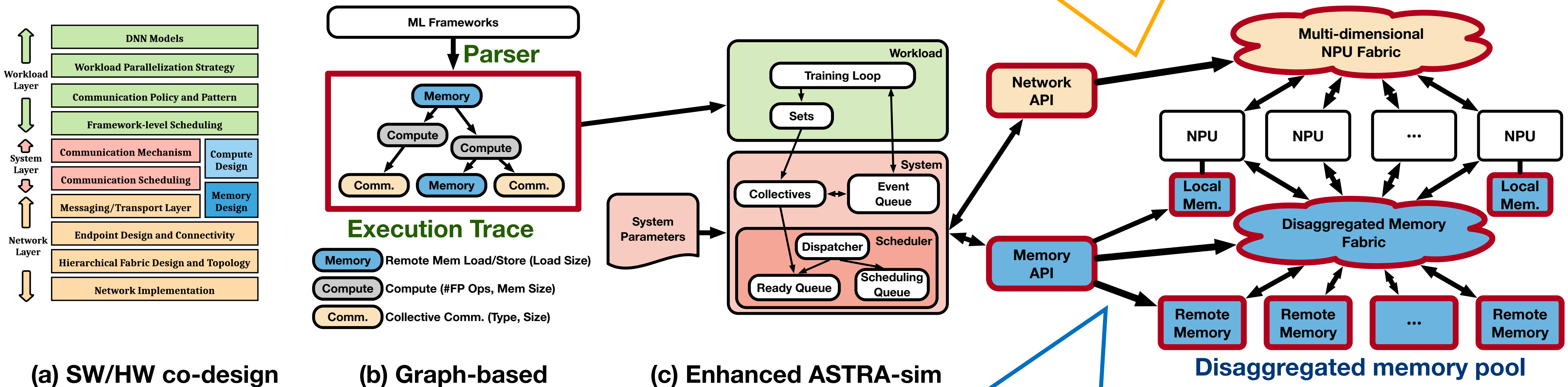
ASTRA-sim cannot model emerging training platforms

Outline

- Distributed Training
- Background
- ASTRA-sim
- Limitations of ASTRA-sim
- ASTRA-sim2.0
- Case Studies and Results
- Conclusion

Overview: ASTRA-sim2.0

- ✓ Simulates multi-dimensional networks at scale
- ✓ Multi-dimensional topology representation
- ✓ Analytical network backend



(a) SW/HW co-design stack of distributed training

(b) Graph-based Execution Engine

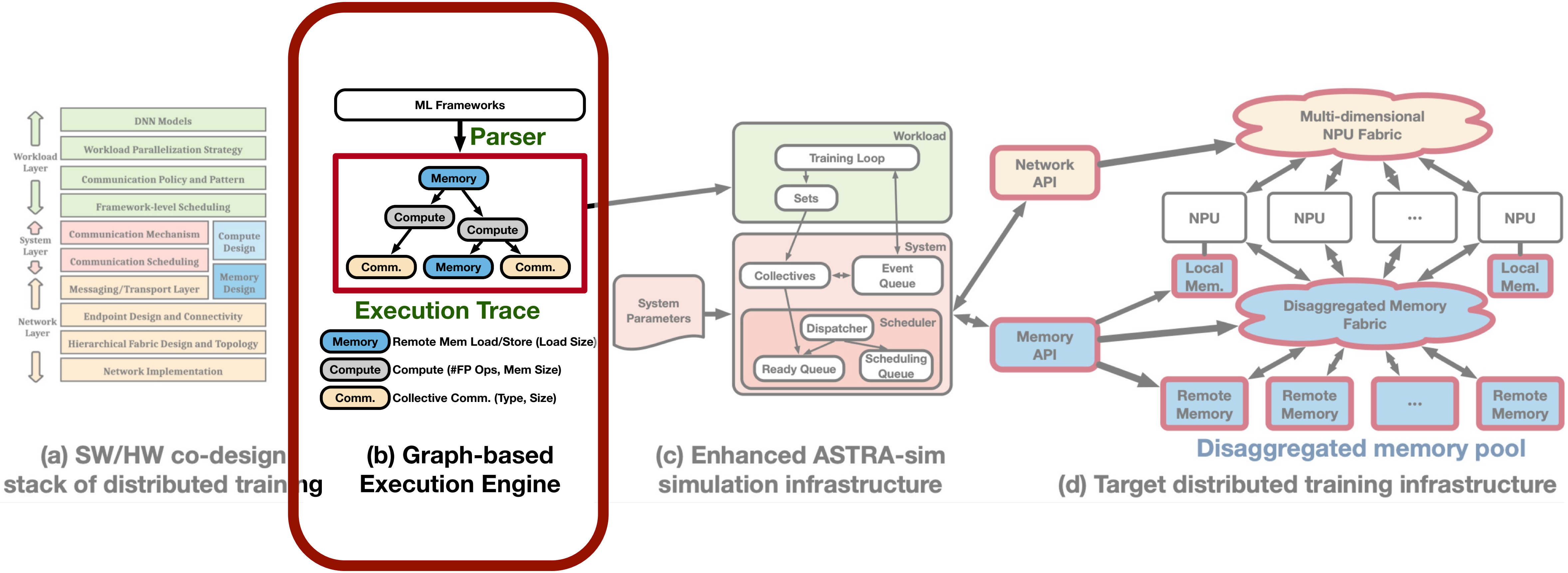
(c) Enhanced ASTRA-sim simulation infrastructure

Target distributed training infrastructure

- ✓ Supports arbitrary parallelization strategies
- ✓ Graph-based Execution Engine
- ✓ Execution Traces (ETs)

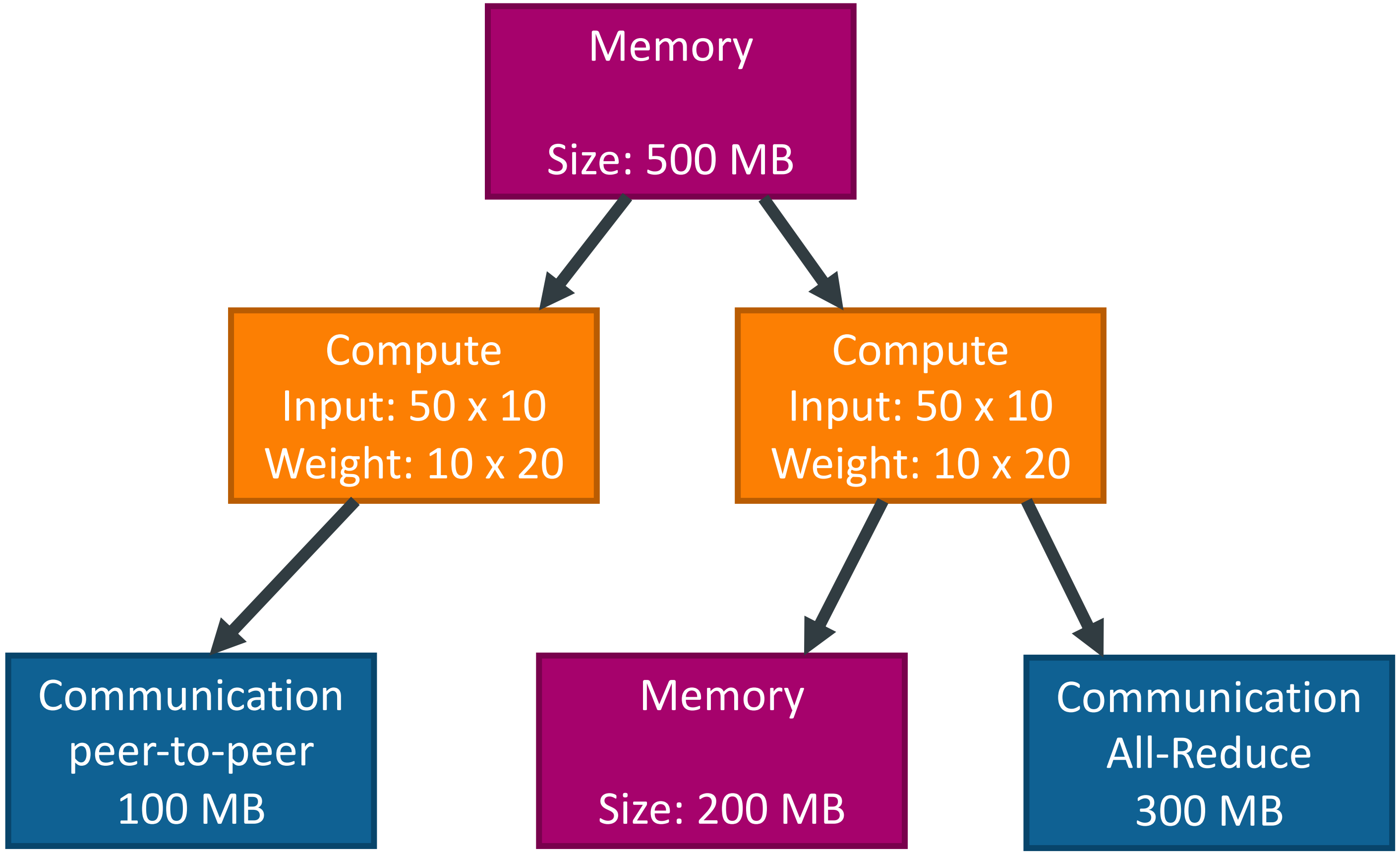
- ✓ Captures memory systems through MemoryAPI
- ✓ Local memory model
- ✓ Remote memory model

Graph-based Execution Engine



Graph-based Execution Engine

- Parallelization is represented in Execution Trace (ET)



Collecting Execution Trace

- ETs could be easily **collected from PyTorch models**

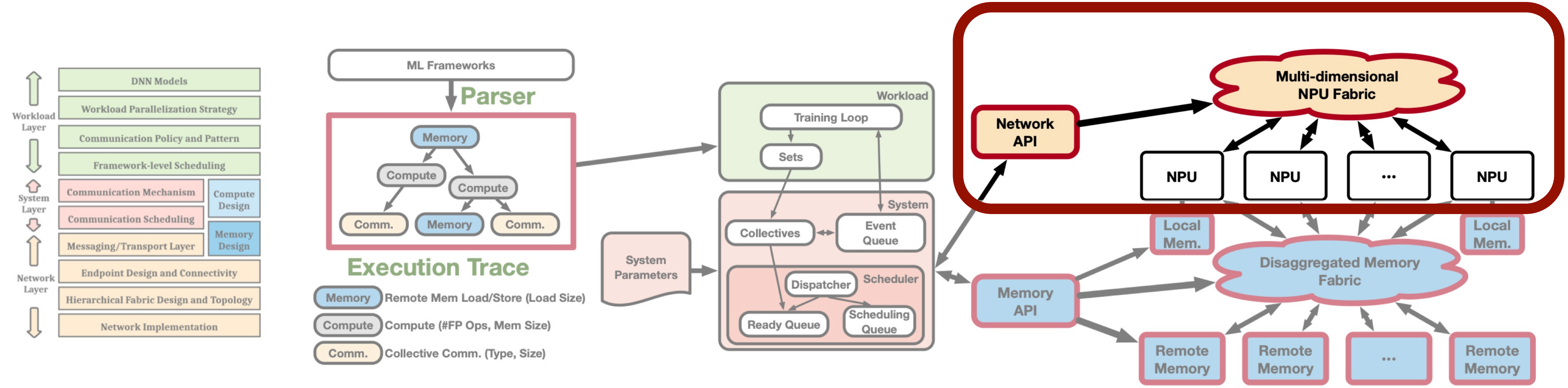
```
et = ExecutionGraphObserver()
et.register_callback("et_file.json")
et.start()

# run PyTorch model

et.stop()
et.unregister_callback()
```

Start
ET collection

Run model

Stop collection

Multi-dimensional Network Modeling



(a) SW/HW co-design stack of distributed training

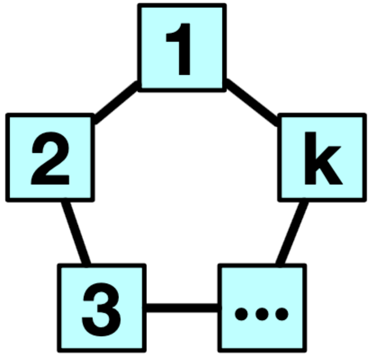
(b) Graph-based Execution Engine

(c) Enhanced ASTRA-sim simulation infrastructure

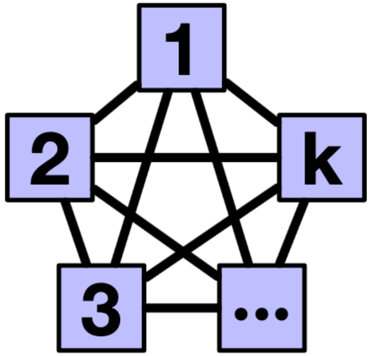
(d) Target distributed training infrastructure

Network Building Blocks

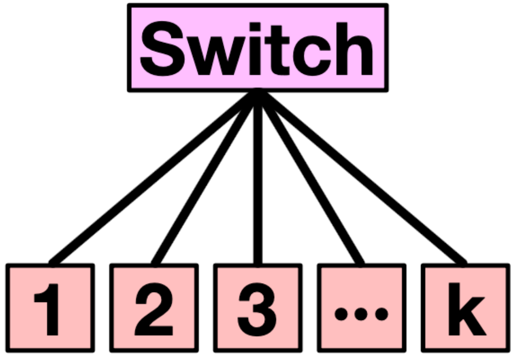
- Basic building blocks of multi-dimensional networks



Ring



FullyConnected



Switch

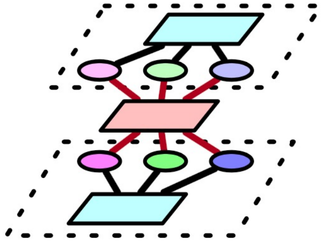
- **No network congestion** while running collective communication

Topology Building Block	Topology-aware Collective Algorithm
Ring	Ring
FullyConnected	Direct
Switch	HalvingDoubling

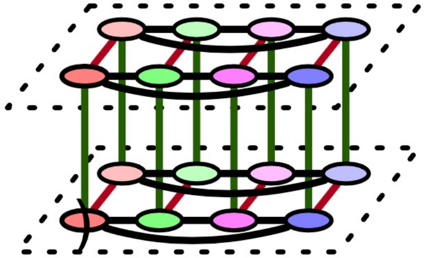
Representing Real Systems

- Captures state-of-the-art training platforms

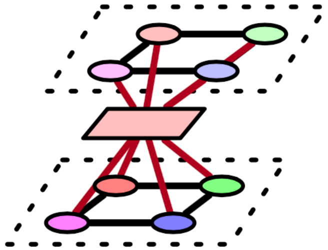
Dimension	Component (Networking)
Dim 1	Chiplet (on-chip)
Dim 2	Package (NVLink)
Dim 3	Node (NVLink)
Dim 4	Pod (NIC)



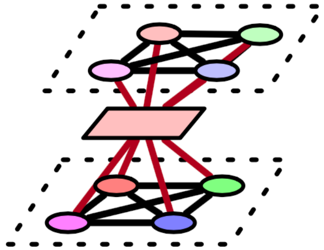
SW(3)_SW(2)
NVIDIA DGX-2
NVIDIA DGX-A100



R(4)_R(2)_R(2)
Google TPUv4



R(4)_SW(2)
Meta Zion
NVIDIA DGX-1



FC(4)_SW(2)
Intel
Habana

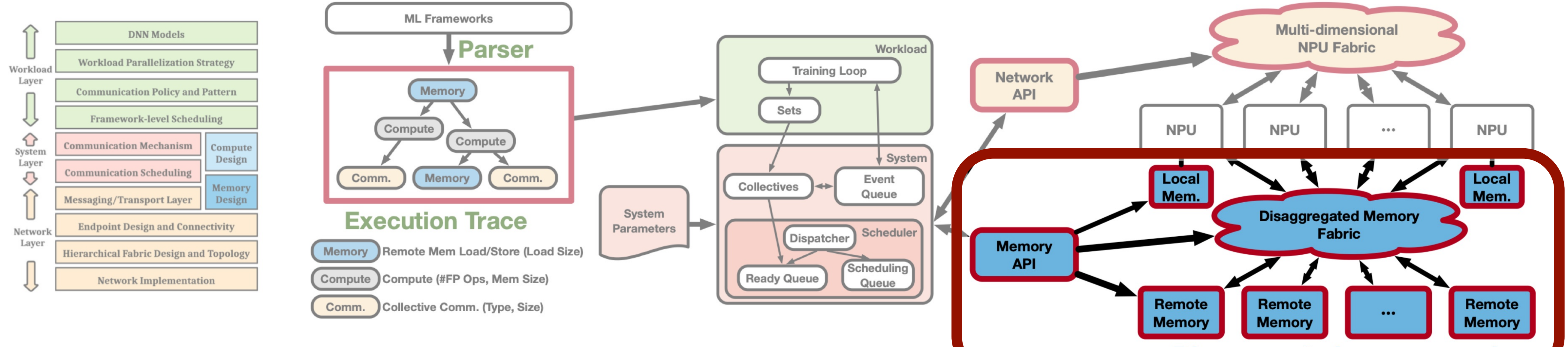
Analytical Backend

- Boost up simulation by **analytically modeling communications**

$$send(src \rightarrow dest,\ msg\_size) = \underbrace{\#hops(src \rightarrow dest) \times link\_latency}_{\text{link delay}} + \underbrace{\frac{msg\_size}{link\ bandwidth}}_{\text{serialization delay}}$$

- Suitable when there's no network contention
  - Topology-aware collective communication

Modeling Emerging Memory Systems



(a) SW/HW co-design stack of distributed training

(b) Graph-based Execution Engine

(c) Enhanced ASTRA-sim simulation infrastructure

(d) Target distributed training infrastructure

Modeling Emerging Memory Systems

- ASTRA-sim2.0 adds a **MemoryAPI**
  - Could be used for both local/remote memory models
- **Local Memory Model**

$$access(tensor\_size) = memory\ access\ latency + \frac{tensor\_size}{memory\ bandwidth}$$

- **Remote Memory Model**
  - Mix and match per design choices (e.g., pipelining multiple stages)
- **In-switch Collective Communication**
  - Reduction happens on-the-fly inside network switches

Modeling Emerging Memory Systems

- ASTRA-sim2.0 adds a **MemoryAPI**
  - Could be used for both local/remote memory models

ASTRA-sim2.0 models futuristic training characteristics

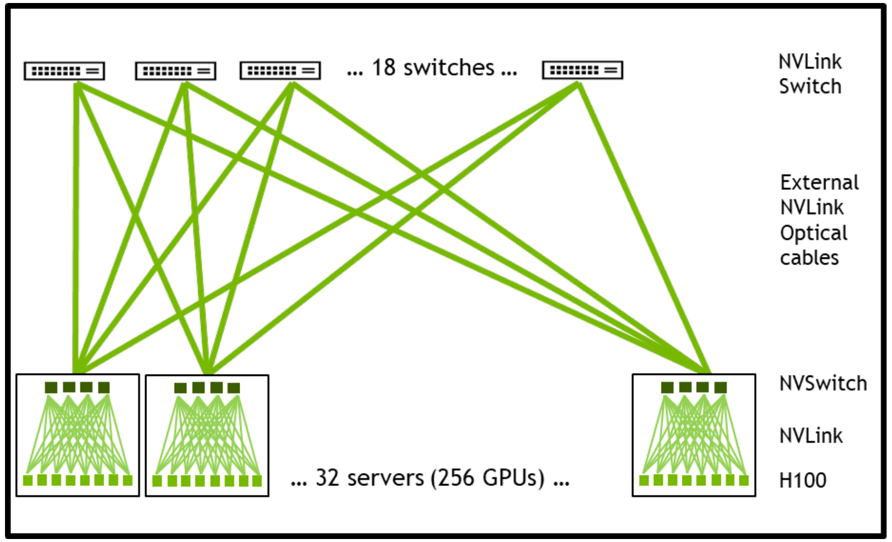
- **In-switch Collective Communication**
  - Reduction happens on-the-fly inside network switches

Outline

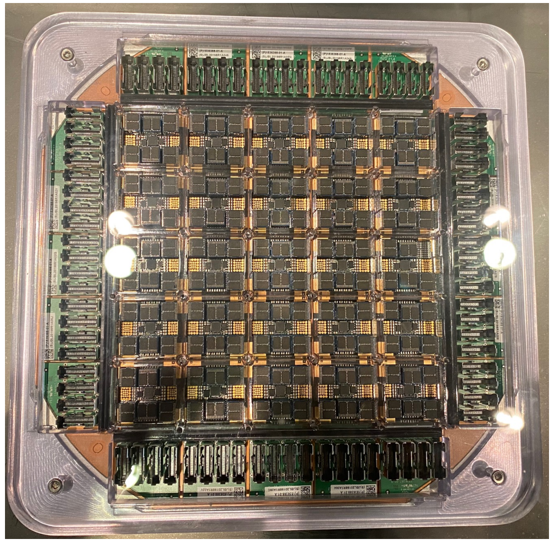
- Distributed Training
- Background
- ASTRA-sim
- Limitations of ASTRA-sim
- ASTRA-sim2.0
- Case Studies and Results
- Conclusion

Case Study 1: Conventional vs. Wafer-scale

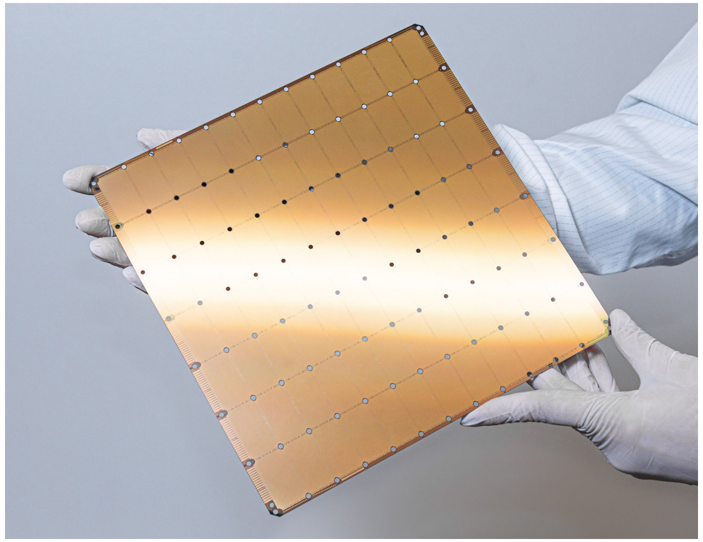
- Conventional Systems: multi-dimensional with diminishing BW
- Wafer-scale Systems: 1-2D topology with very-high-BW



NVIDIA HGX-H100



Tesla D1



Cerebras WSE-2

- https://developer.nvidia.com/blog/introducing-nvidia-hgx-h100-an-accelerated-server-platform-for-ai-and-high-performance-computing
- https://www.lrz.de/presse/ereignisse/2022-05-25-NextGenAISystem/

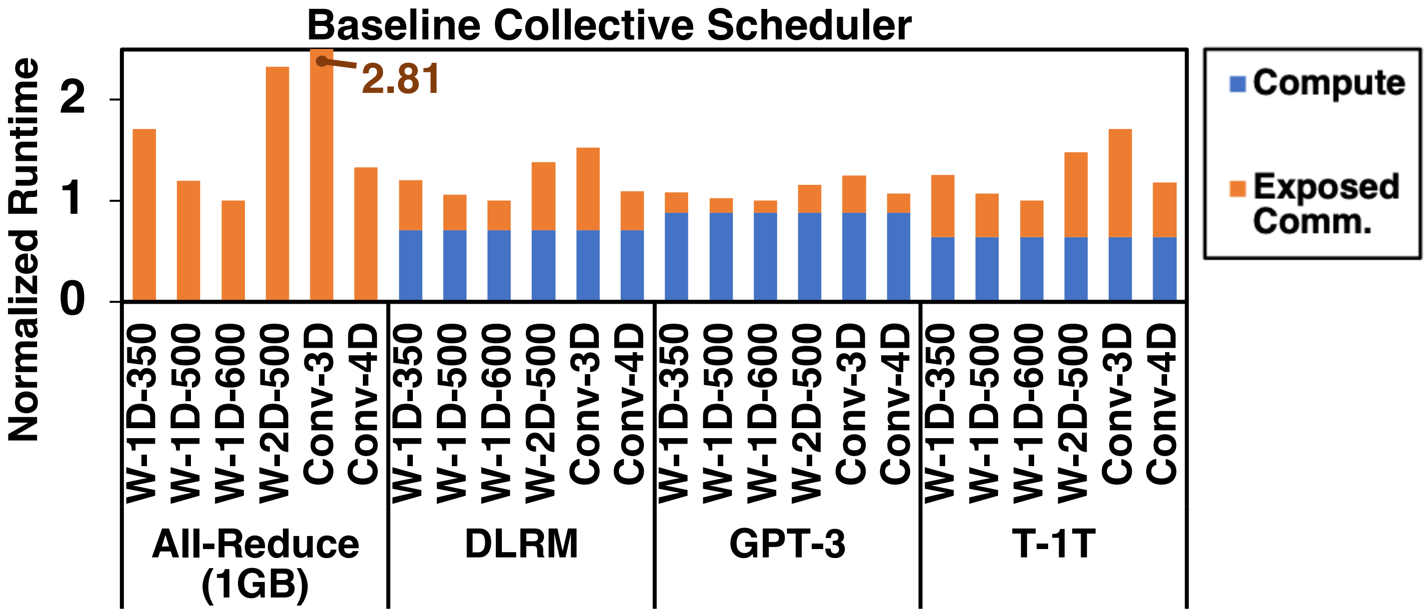
Case Study 1: Conventional vs. Wafer-scale

- Wafer-scale: 1-2D
  - With very high BW per each Dim
- Conventional Systems: 3-4D
  - With diminishing network BW with higher network dimension

Topology	**Shape**	**NPU Size**	**BW (GB/s)**
W-1D	Switch	512	350, 500, 600
W-2D	Switch_Switch	32×16	250_250
Conv-3D	Ring_FC_Switch	16×8×4	200_100_50
Conv-4D	Ring_FC_Ring_Switch	2×8×8×4	250_200_100_50

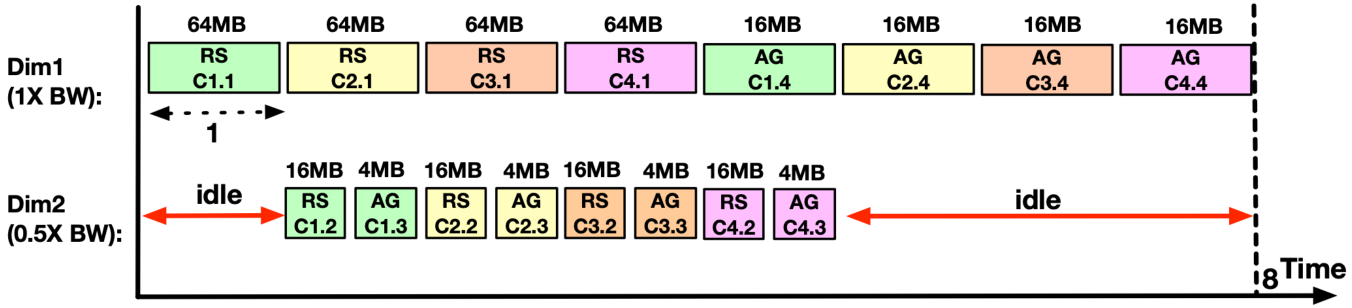
Case Study 1: Result

- **Overhead running multi-dimensional collective** communication
  - W-1D (with higher BW) yields overall best performance
- Conv-4D is still powerful
  - Driving **higher BW per NPU**

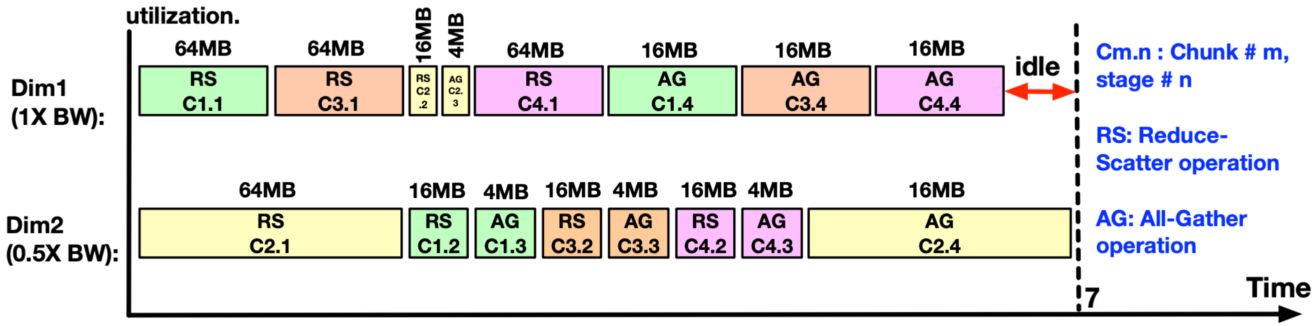


Case Study 2: Chunk Scheduling Policy

- **Themis: Greedy-based chunk scheduling policy**
  - To maximize BW utilization of multi-dimensional collective communication



Baseline Scheduling

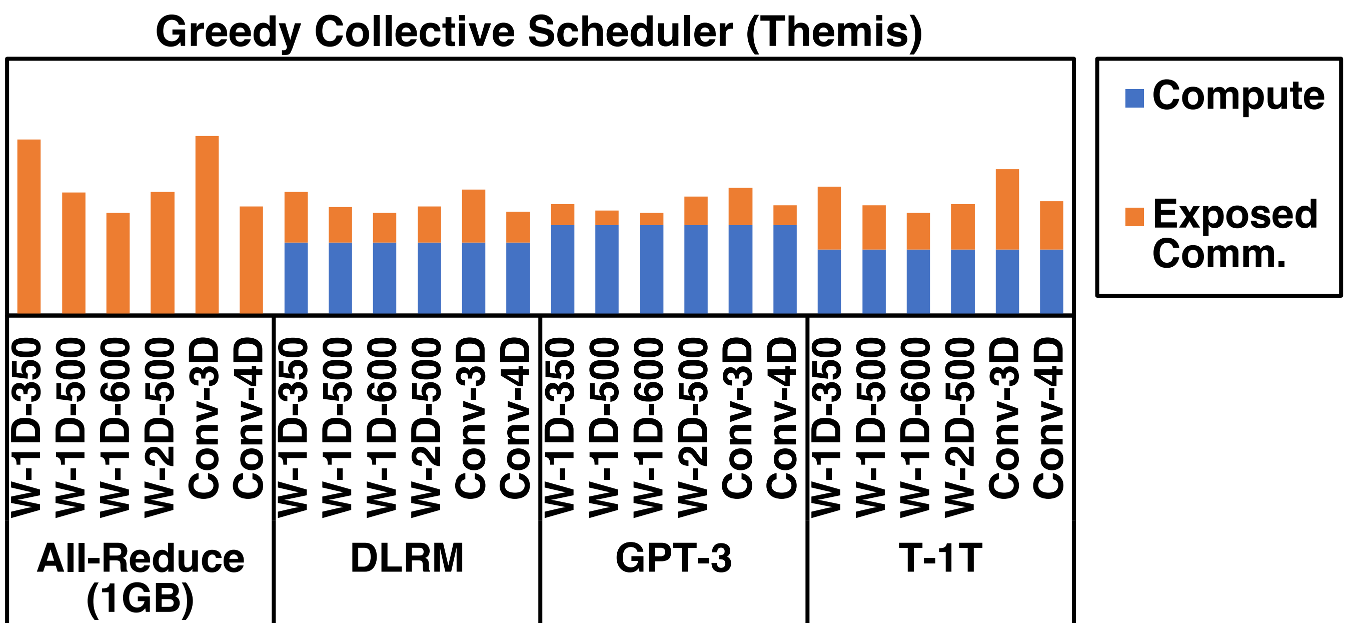


Themis Scheduling

Saeed Rashidi *et al.*, "Themis: A Network Bandwidth-Aware Collective Scheduling Policy for Distributed Training of DL Models," ISCA 2022

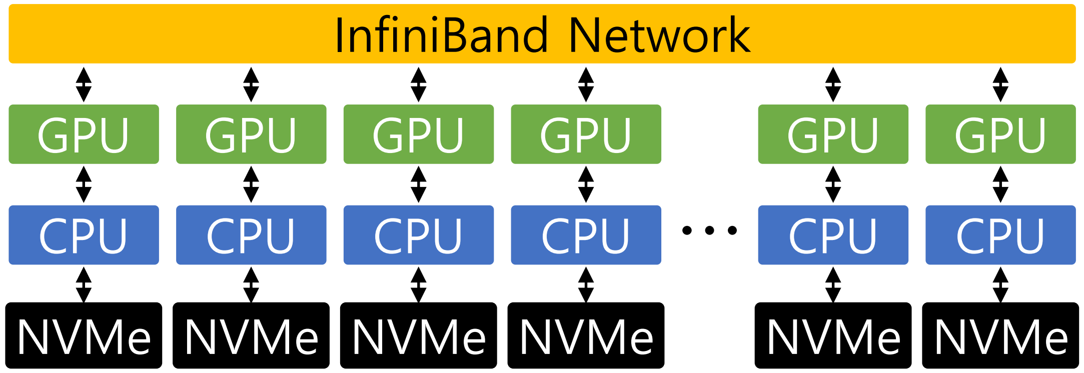
Case Study 2: Result

- No difference in W-1D, but **huge gain in W-2D, Conv-3/4D**
- If equal BW/NPU is provisioned, **yields near identical performance**
  - Regardless of network dimensionality

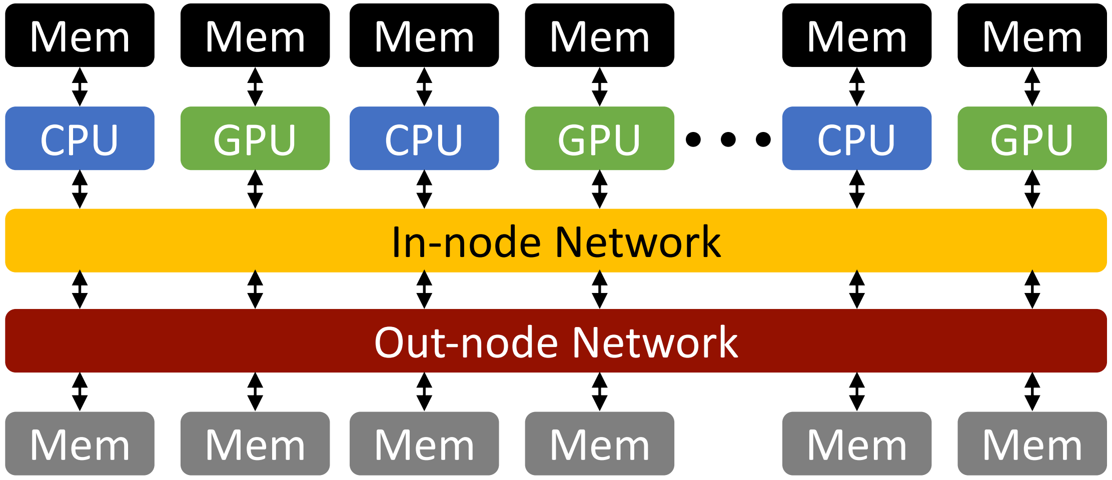


Case Study 3: Comparing Memory Systems

- ZeRO-Infinity: leveraging local memory (NVMe)
- HierMem: disaggregated memory systems with in-switch collective



ZeRO-Infinity



HierMem

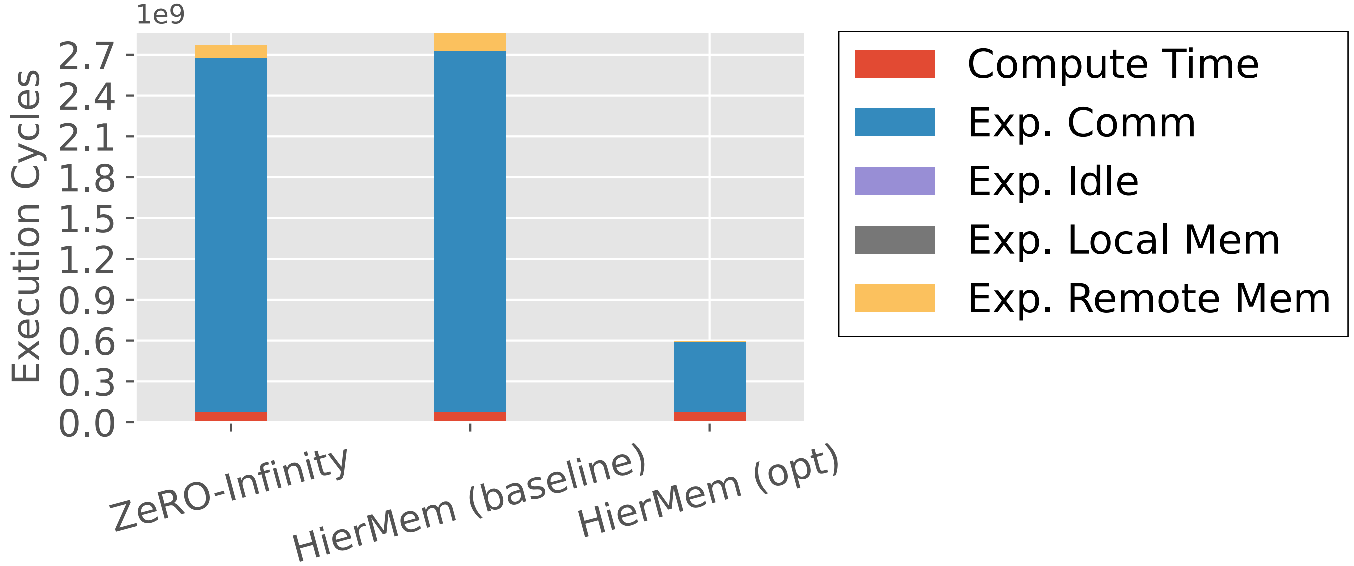
Case Study 3: Comparing Memory Systems

- ZeRO-Infinity: Baseline
- HierMem:
  - Baseline: equivalent configuration as ZeRO-Infinity
  - Opt: fine-tuned configuration for Mixture-of-Experts (MoE) Model

	ZeRO-Infinity	**HierMem (Baseline)**	**HierMem (Opt)**
GPU Peak Perf (TFLOPS)	2048	2048	2048
GPU Local HBM BW (GB/sec)	4096	4096	4096
In-node Pooled Fabric BW (GB/sec)	-	256	**512**
Num of Out-node Switches	-	16	16
Num of Remote Memory Groups	256	256	256
Remote Mem Group BW (GB/sec)	100	100	**500**

Case Study 3: Result

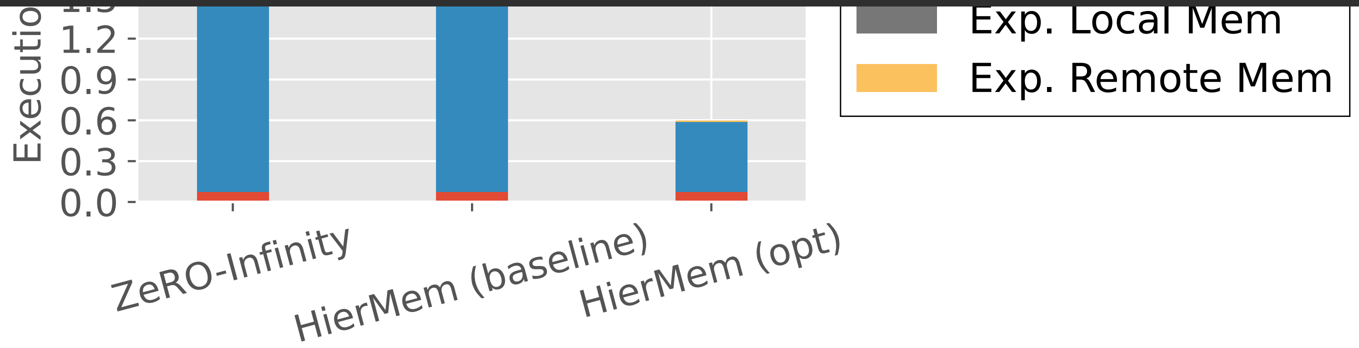
- ZeRO-Infinity and HierMem (baseline) is **near-identical**
- Fine-tuned HierMem shows **4.6x better runtime**
  - **In-switch collective communication** reduces exposed communication



Case Study 3: Result

- ZeRO-Infinity and HierMem (baseline) is **near-identical**
- Fine-tuned HierMem shows **4.6x better runtime**

ASTRA-sim2.0 enables design-space exploration of emerging training platforms



Outline

- Distributed Training
- Background
- ASTRA-sim
- Limitations of ASTRA-sim
- ASTRA-sim2.0
- Case Studies and Results
- Conclusion

Conclusion

- Needs to **navigate the design-space of distributed training**
  - Large models and huge training dataset makes distributed training inevitable
  - Design space is complex: parallelism, memories, networks, etc.
- **ASTRA-sim2.0**: modeling emerging systems
  - Arbitrary parallelization strategies
  - Multi-dimensional networks at scale
  - Disaggregated memory system modeling

Thank You!

Reach out to me at:
william.won@gatech.edu

https://astra-sim.github.io

ASTRA-sim2.0 paper