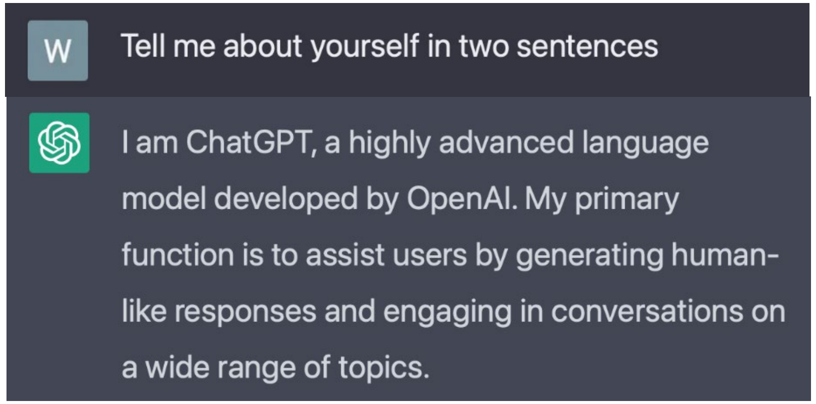
nsdi '23

TopoOpt: Co-optimizing Network Topology and Parallelization Strategy for Distributed Training Jobs

Weiyang (Frank) Wang, MIT CSAIL

Moein Khazraee, Zhizhen Zhong, Manya Ghobadi, Zhihao Jia, Dheevatsa Mudigere, Ying Zhang, Anthony Kewitsch

The era of large deep neural networks (DNNs)



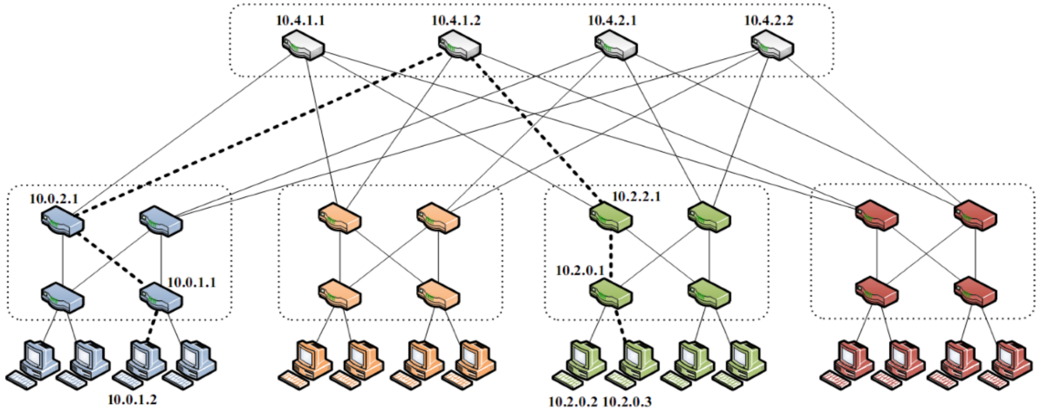
GPT-4
Large Language Model

Deep Learning Recommendation Model
Recommendation Model

DALL.E 2
Image Generation Model

- The growth of large DNN models creates demands efficient distributed DNN training systems

State-of-the-art training clusters: Fat-Tree network topology



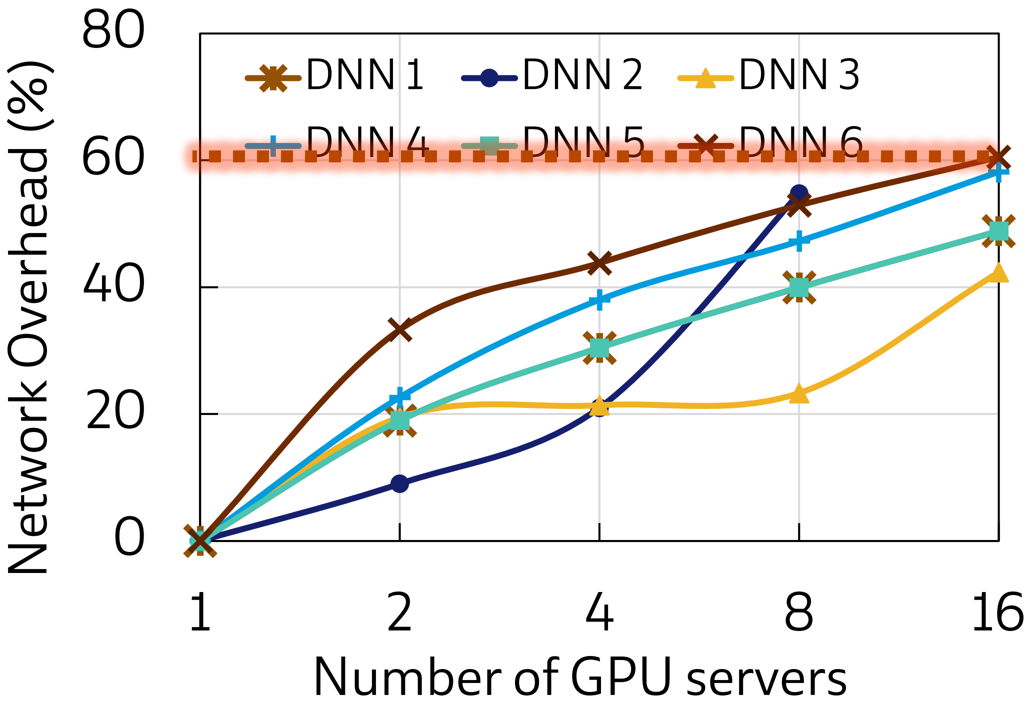
A Scalable, Commodity Data Center Network Architecture
Mohammad Al-Fares et al., SIGCOMM '08

- Fat-Trees provide uniform bandwidth and latency between server pairs
- Ideal when the workload is unpredictable and consists mostly of short transfers

- Fat-Tree networks are not the best network topology for DNN training!

Network is becoming a bottleneck of DNN training

- Fat-Tree based DNN training infrastructures are facing a network bottleneck
  - Network Bottleneck: the amount of time spent on communication only



Previous work on distributed DNN training optimization does not consider physical topology

Compression and encoding
Qsgd [NeurIPS '17]

Schedulers
Themis [NSDI '20]

Asynchronous transmit
DC-ASGD [PMLR '17]

Computation
+
Communication

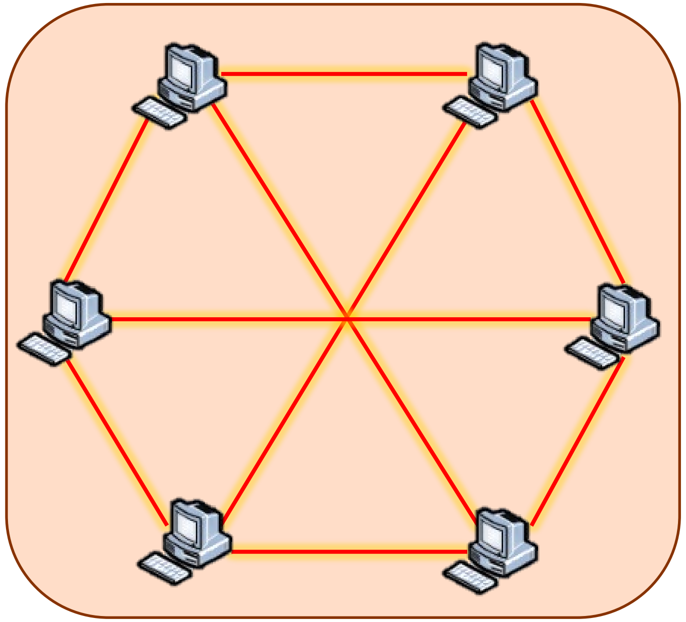
Parallelization strategy
FlexFlow [MLSys '19]

Collective communication
BytePS [OSDI '20]

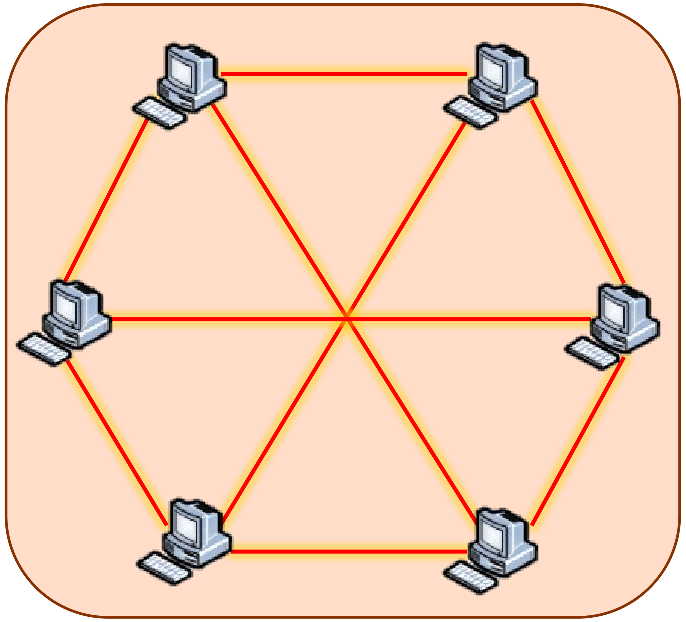
Hyper parameters
ASHA [MLSys '20]

Network topology
?

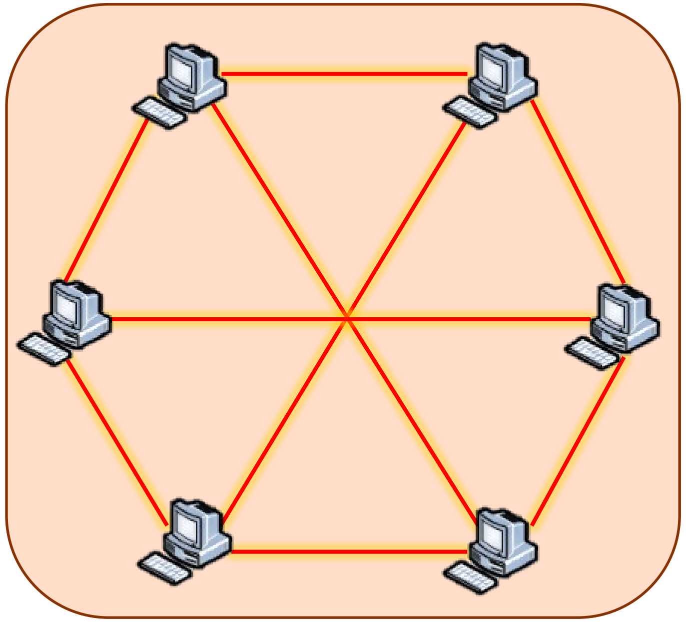
Reconfiguring physical network topology



Topology A

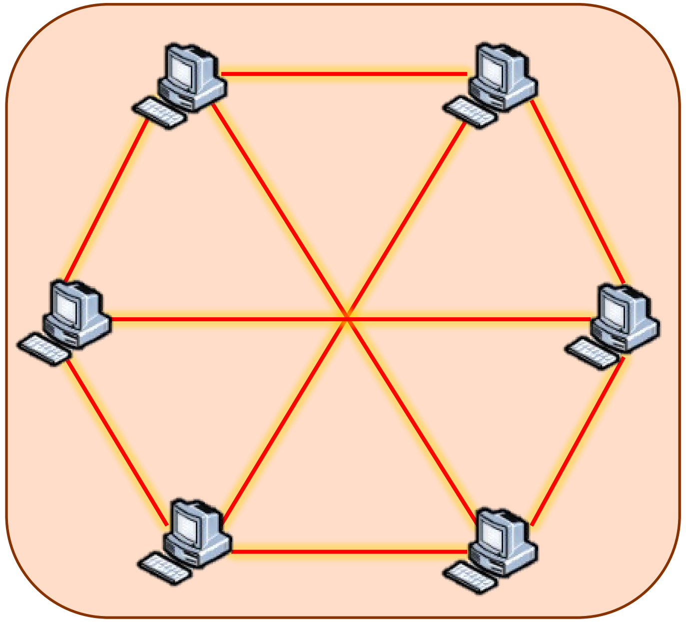


Topology A

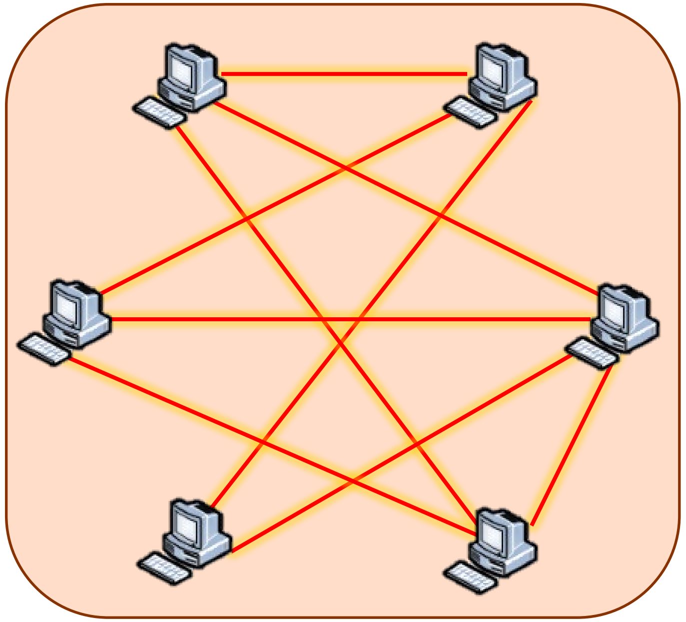


Topology A

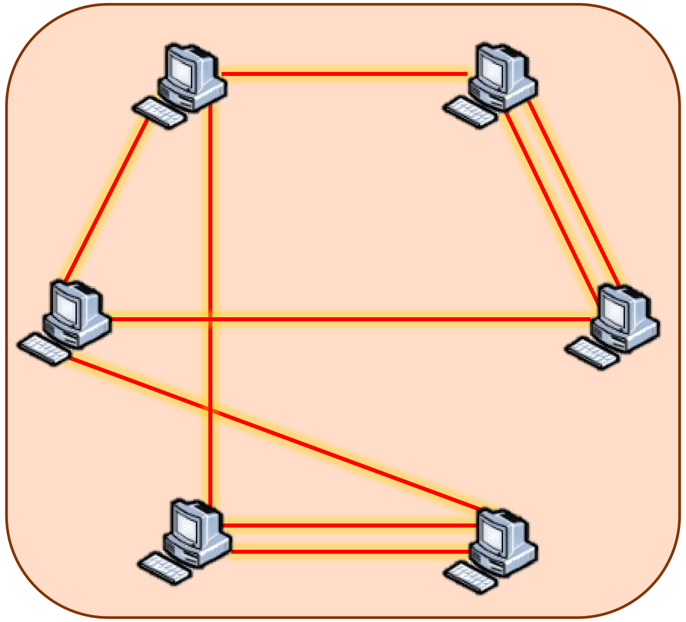
Reconfiguring physical network topology



Topology A

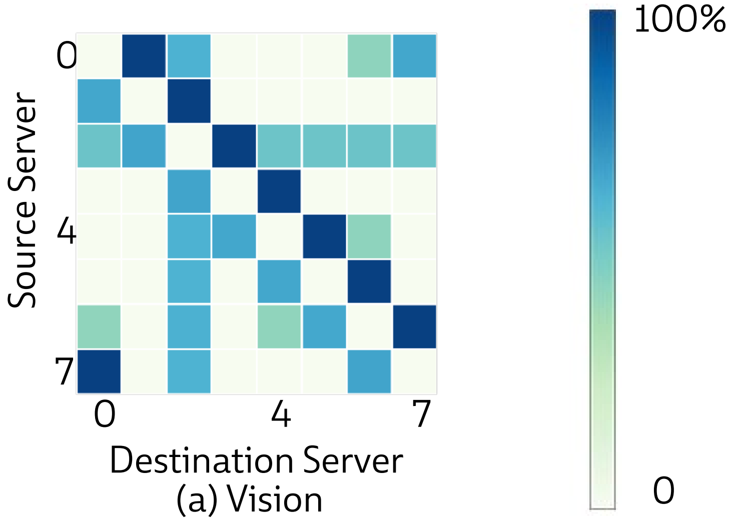


Topology B



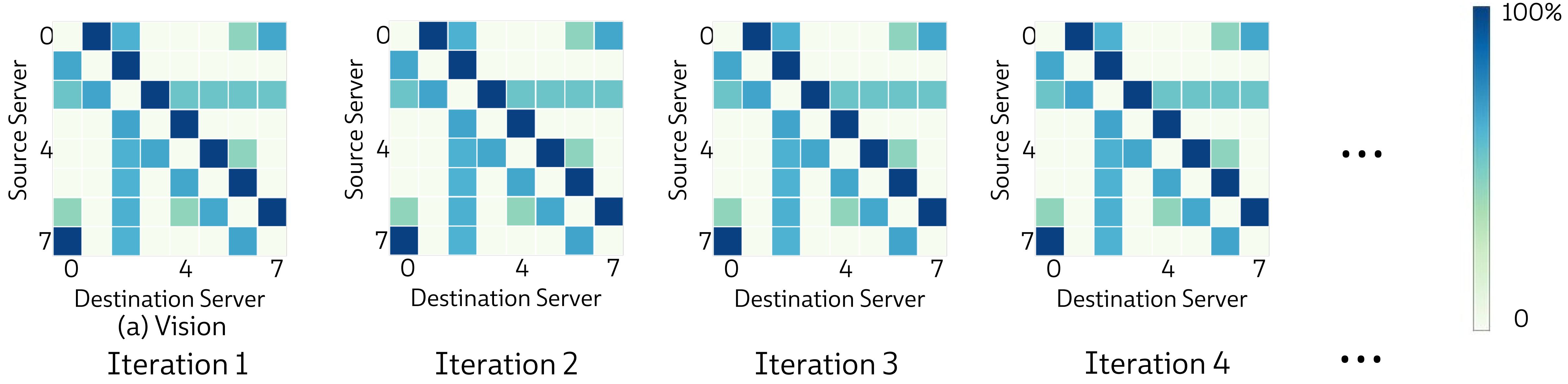
Topology C

DNNs training traffic has different properties



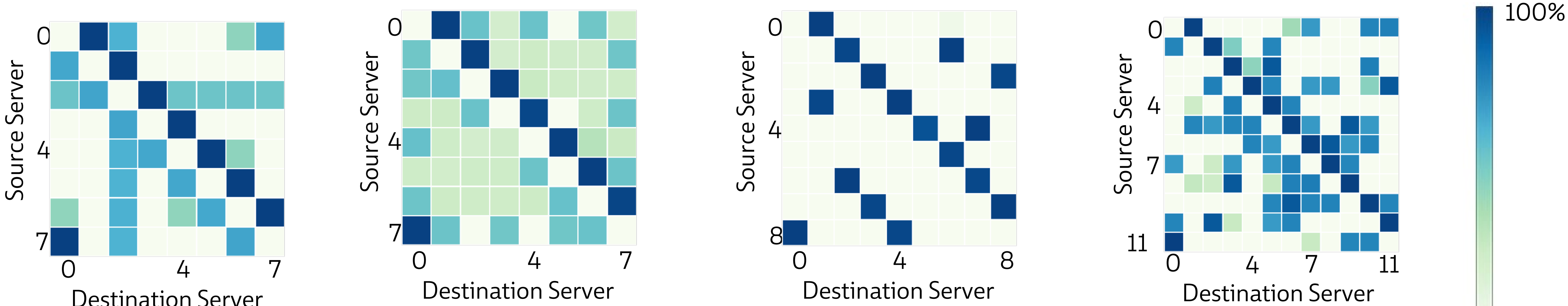
(a) Vision

DNNs training traffic has different properties



- Key observations:
  1. Traffic patterns are predictable, and do not change across training iterations

DNNs training traffic has different properties



(a) Vision (b) Image processing (c) Object Tracking (d) Speech Recognition

- Key observations:
    1. Traffic patterns are predictable, and do not change across training iterations
    2. Traffic patterns are model-dependent

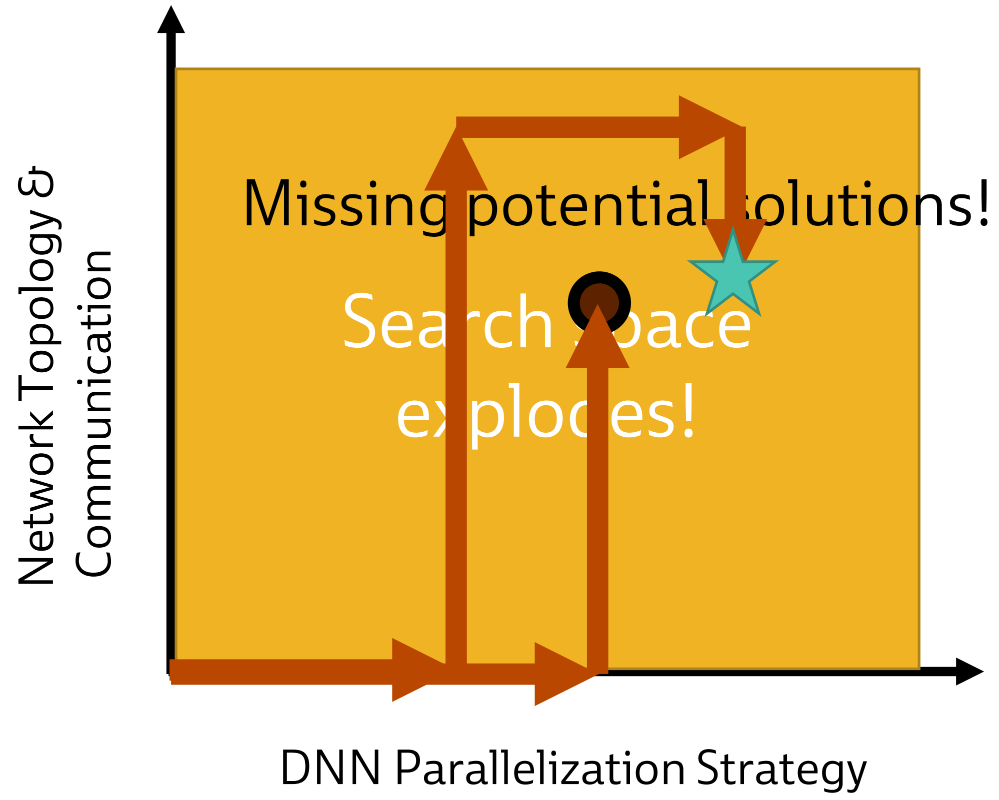
TopoOpt

The first system to leverage reconfigurable network, to co-optimize network topology and parallelization strategy for distributed training

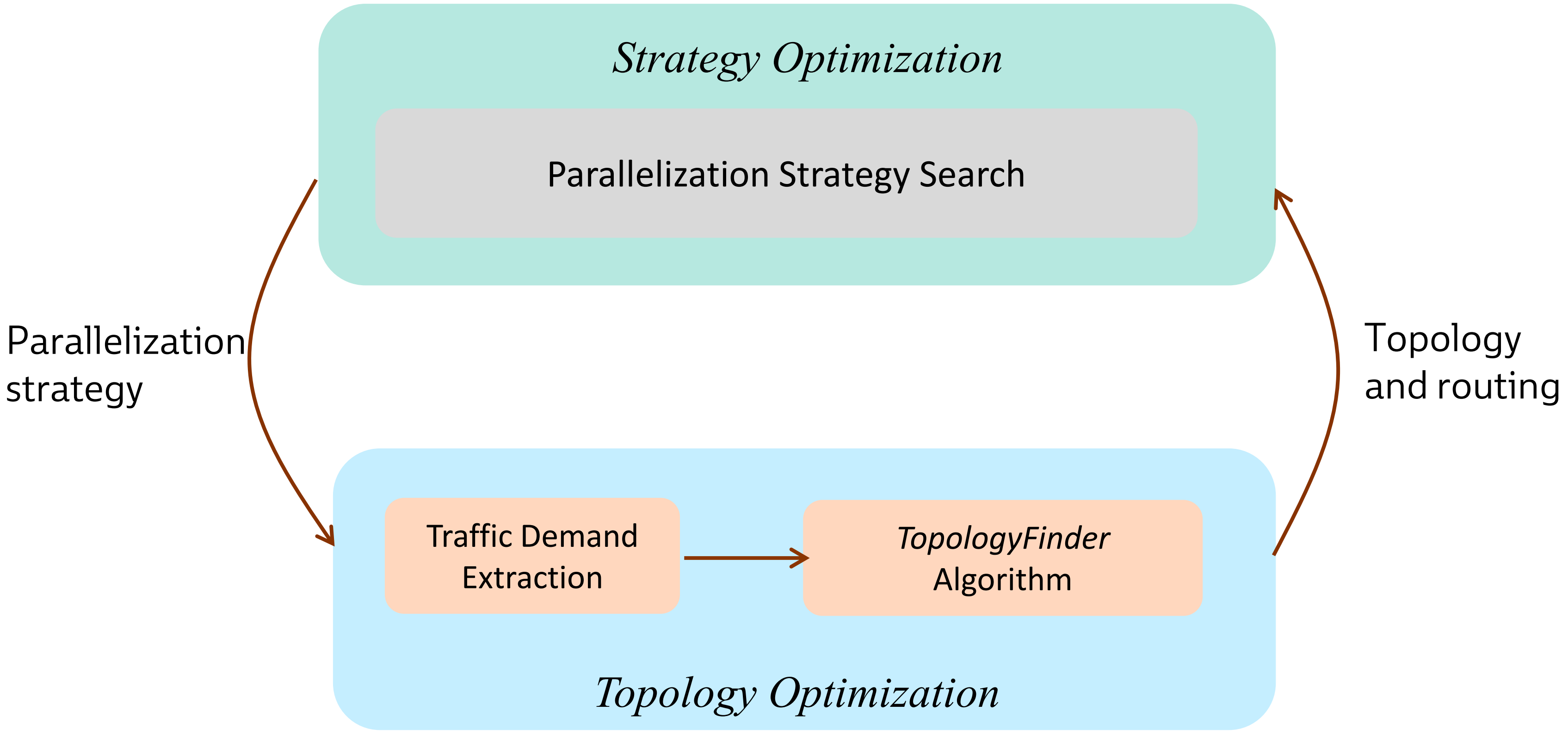
TopoOpt achieves 3.4x faster training time for DNN training

Co-optimization challenge: Huge search space for optimal DNN training

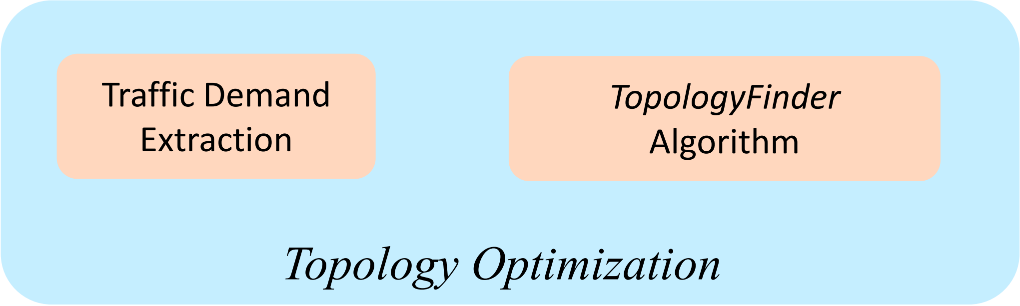
- The configuration space is huge!



Alternating optimization framework to co-optimize DNN parallelization strategy and network topology

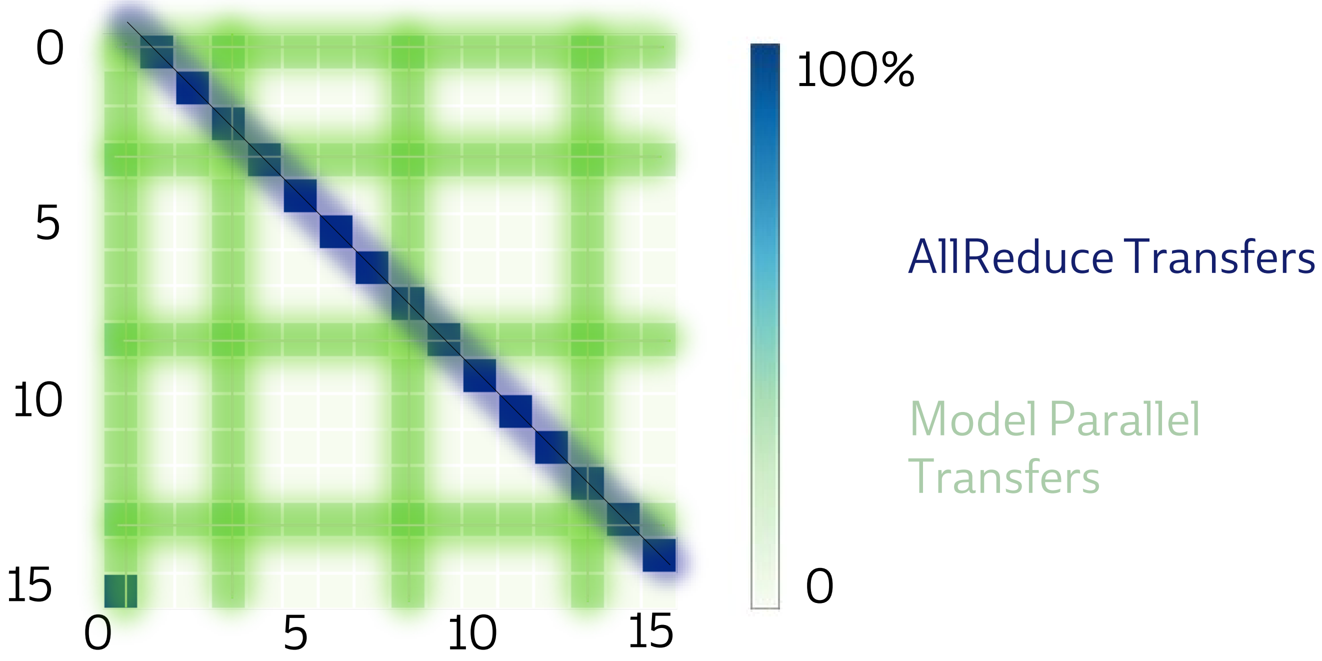


Alternating optimization framework to co-optimize DNN parallelization strategy and network topology



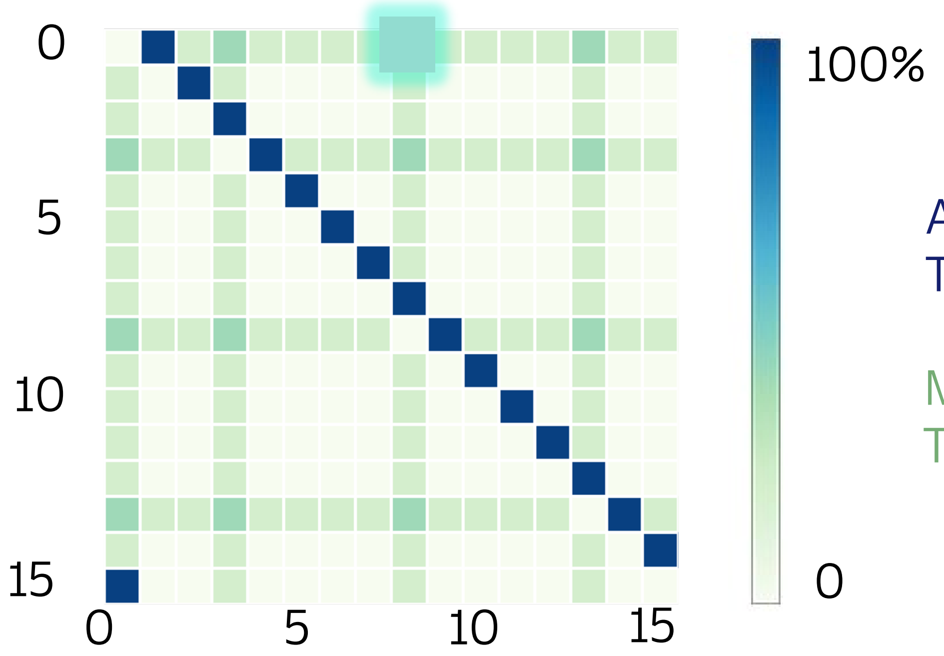
What algorithm should we use to find the topology in this framework?

Characteristics of DNN training traffic



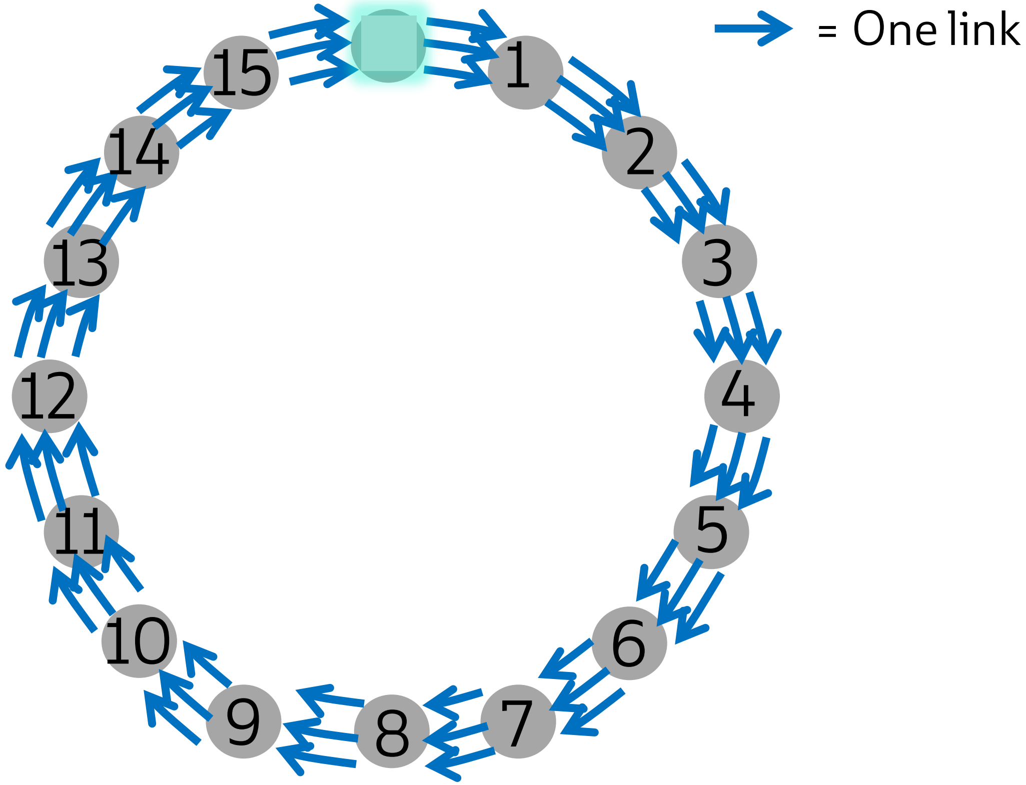
Challenge: finding a good network topology for both AllReduce and Model-Parallel transfers

- Degree (d) = 3, unidirectional



AllReduce Transfers

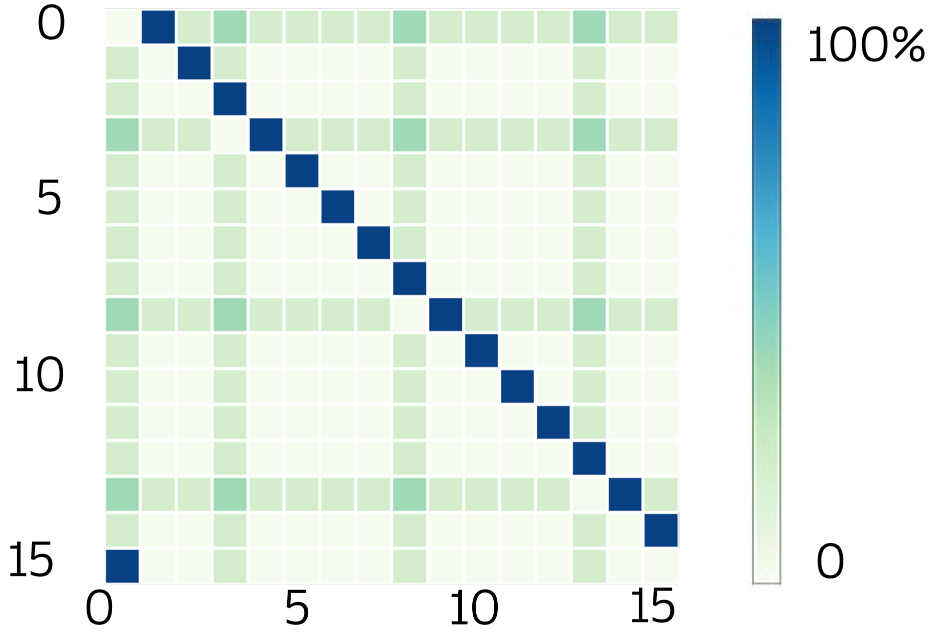
Model Parallel Transfers



8 hops!

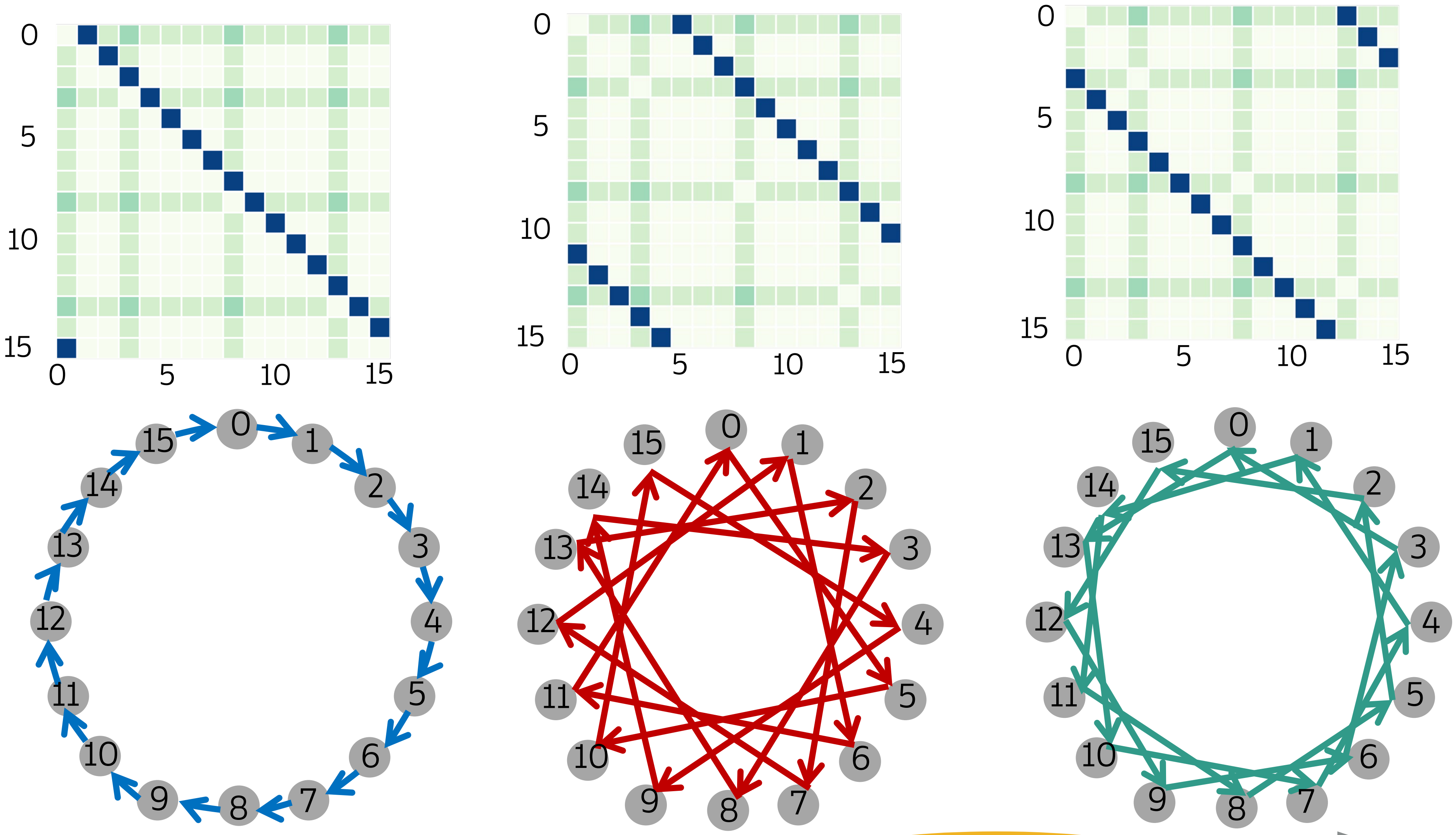
Meeting the requirements of both AllReduce and Model-Parallel transfers

- Degree (d) = 3, unidirectional



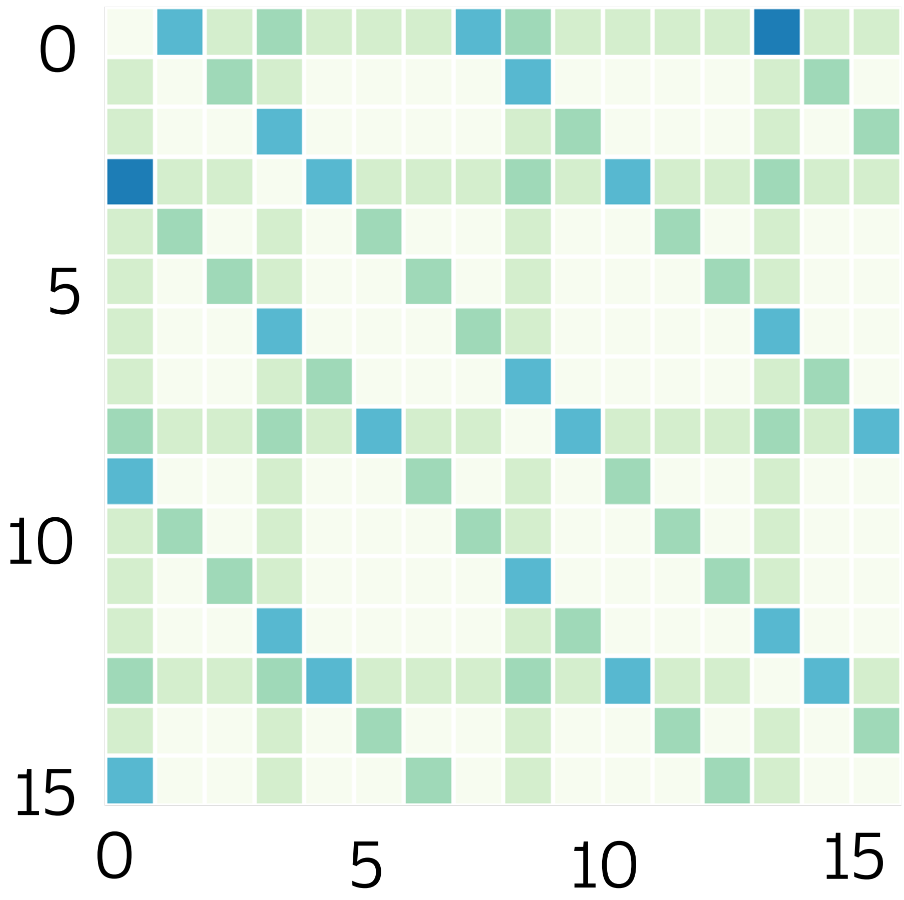
Transfer Type	Characteristics	Network Requirement
AllReduce Transfers	Large, Sparse	Ample Bandwidth
Model Parallel Transfers	Small, Dense	Low hop-count

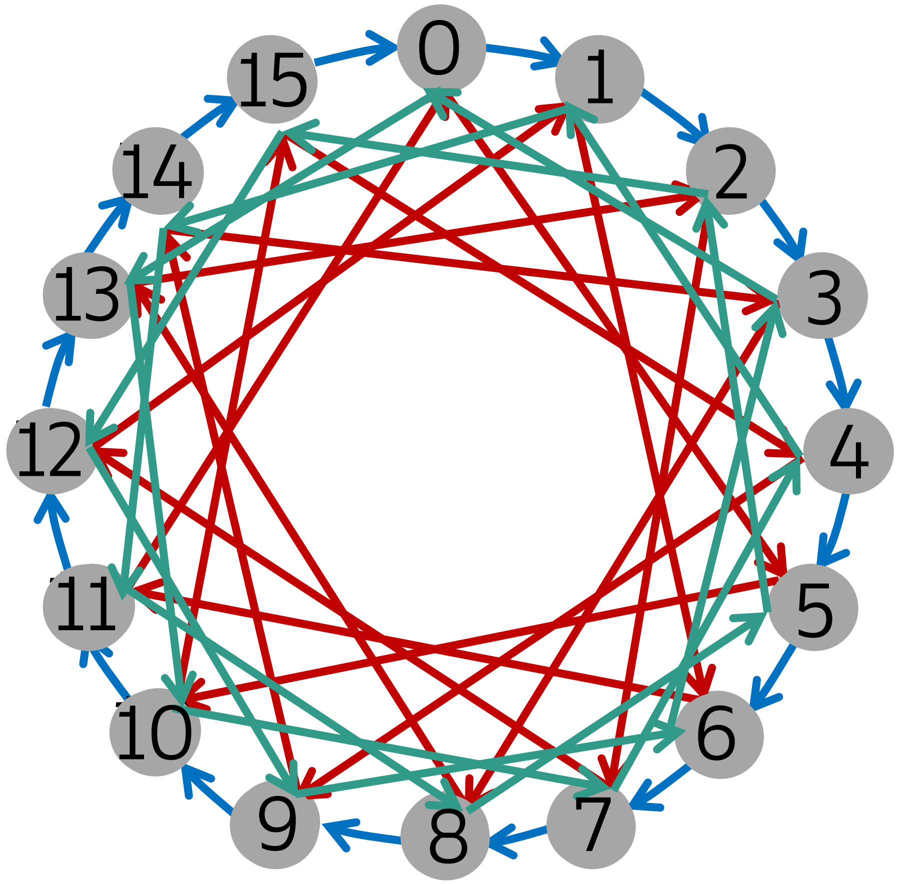
Key idea: **mutate the traffic matrix**



AllReduce transfers are **mutable.** Model-Parallel transfers are not mutable.

Splitting AllReduce traffic

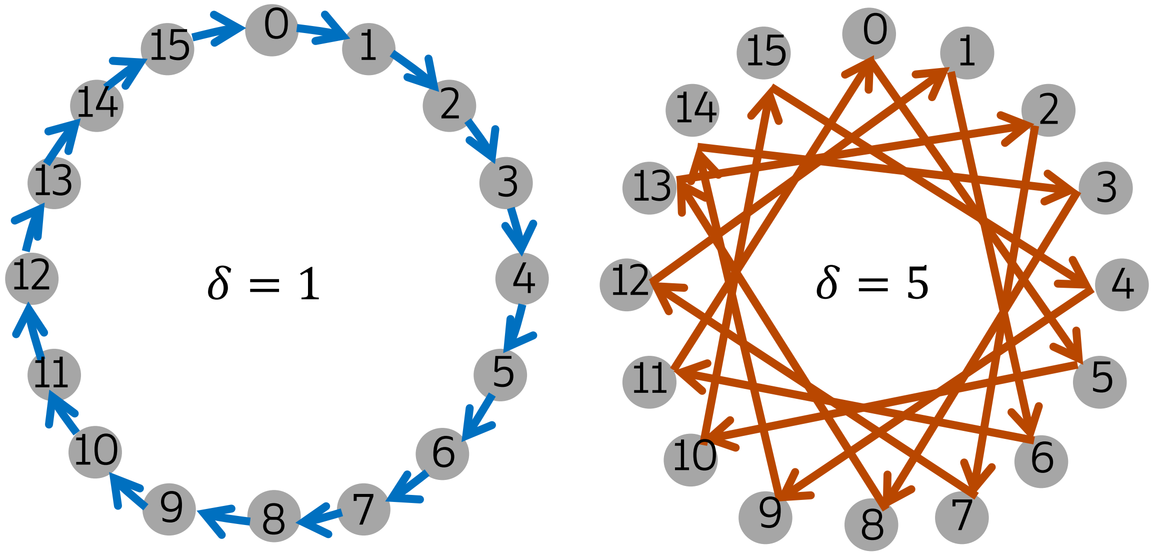


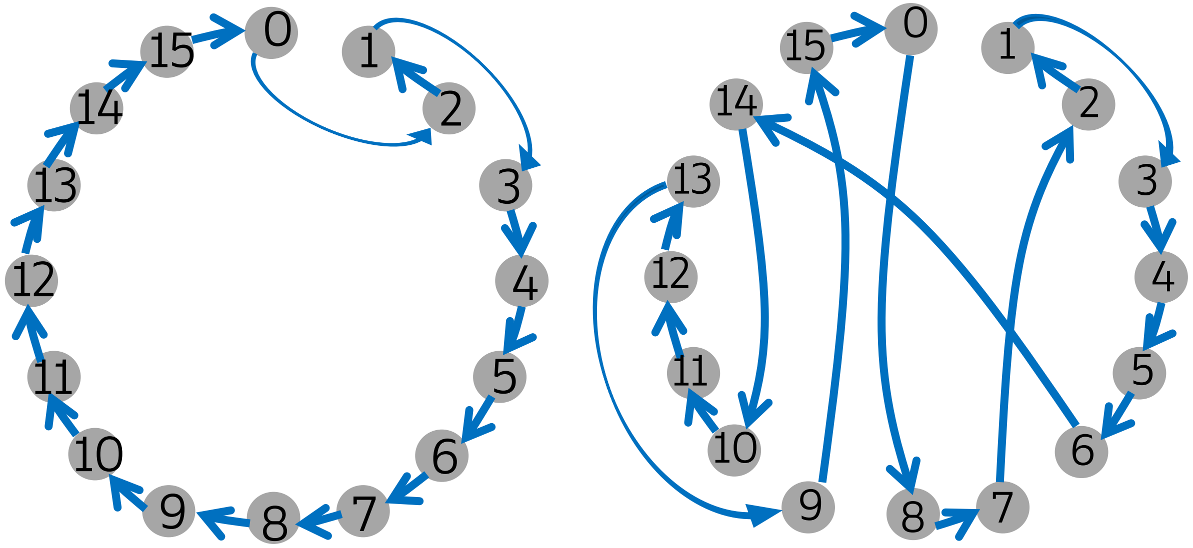


Leverage the mutability of AllReduce transfers to achieve high bandwidth for AllReduce & low hop-count for Model-Parallel!

Key technique: Regular permutations

- n total accelerator, each with degree d





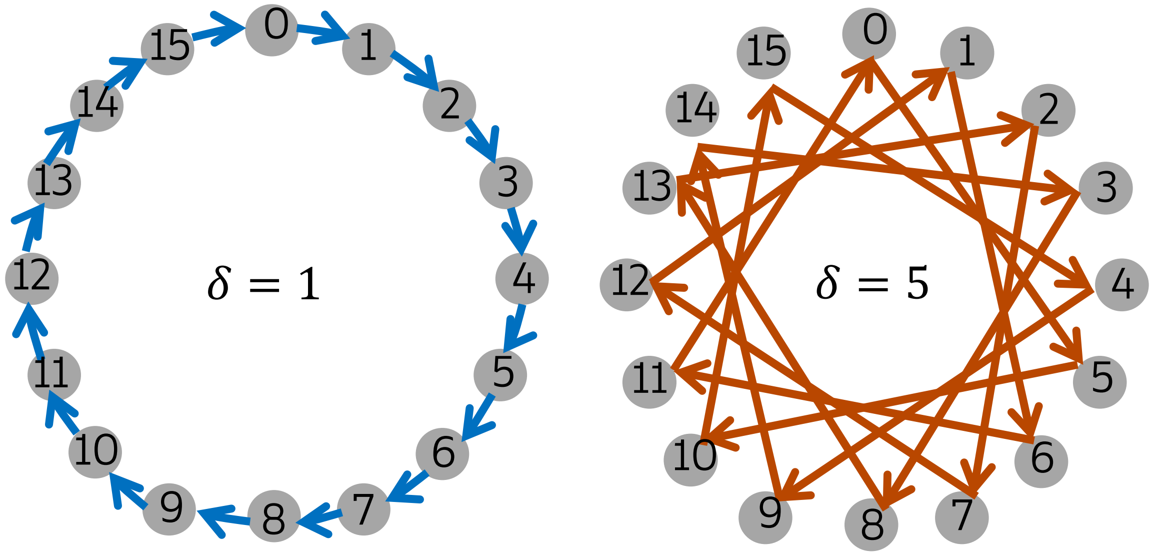
Irregular permutations

Regular permutations – every server connects to another one with a fixed distance δ

$\boldsymbol{O(n!)}$ different permutations

Key technique: Regular permutations

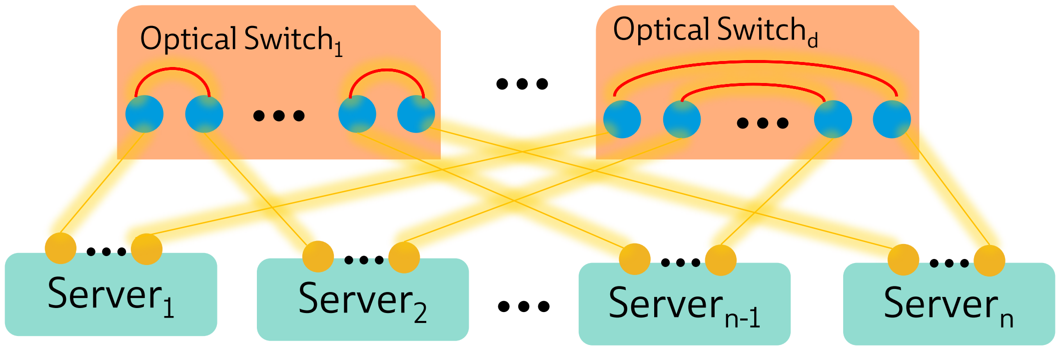
- n total accelerator, each with degree d



- The possible set of δ are the positive integers less than n, such that $\gcd(\delta, n) = 1$
  -> $\boldsymbol{O(n)}$ search space!
- Among all possible δ distances, choose a set of them within the degree to minimize the cluster diameter
- The technique of permuting labels work for other AllReduce algorithms as well

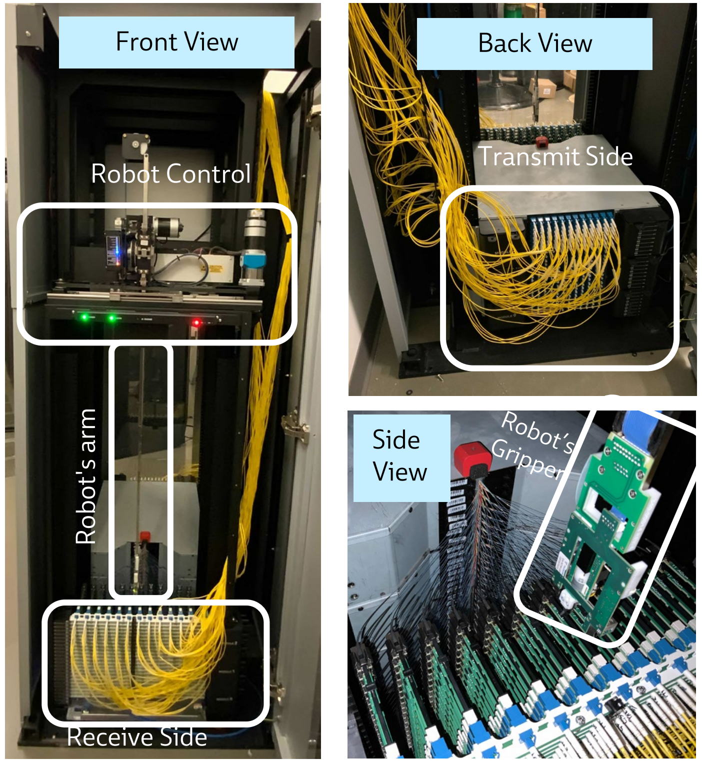
TopoOpt bounds the cluster diameter to $O(d \cdot \sqrt[d]{n})$

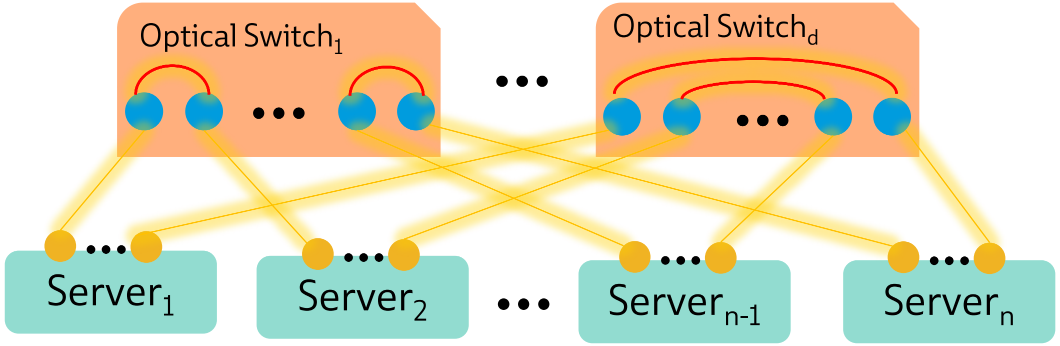
TopoOpt uses optical switches



TopoOpt uses optical switches

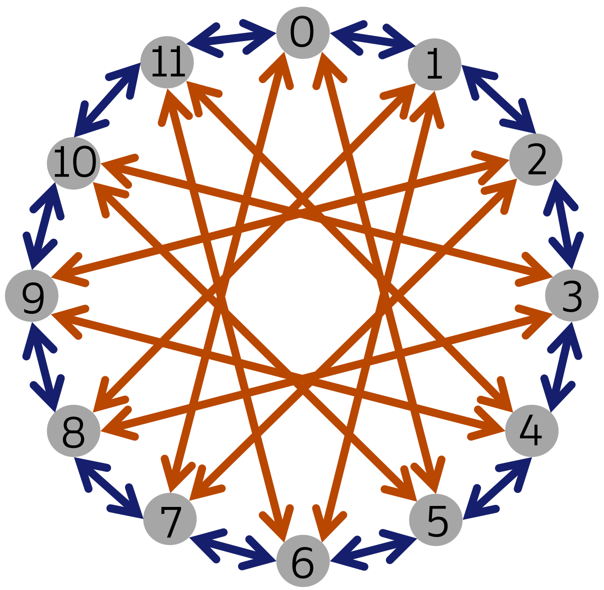
- Fully functional 12-node, degree 4 testbed integrated with NCCL





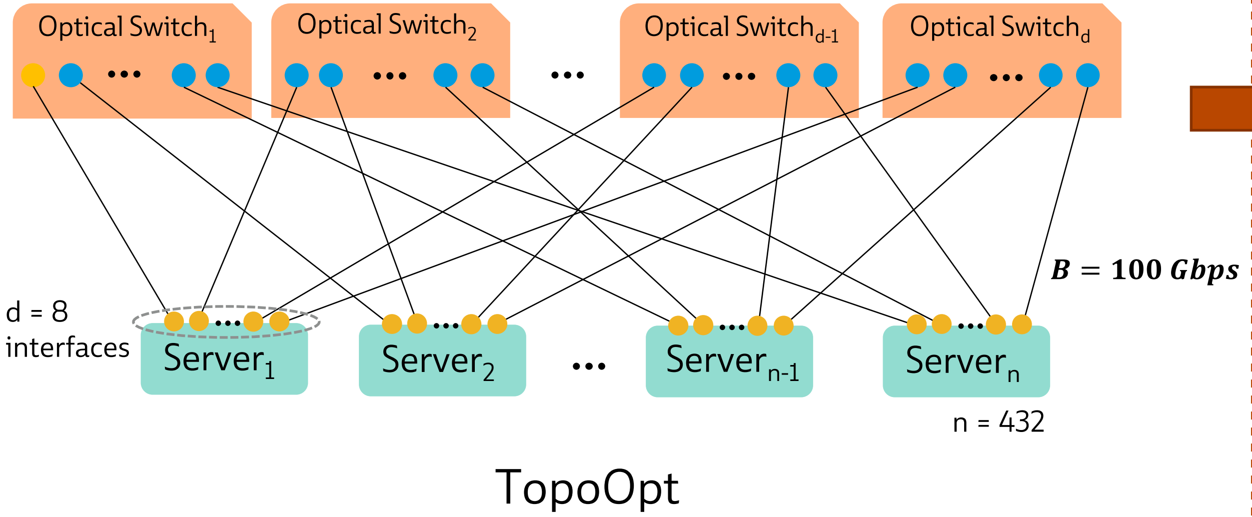
TopoOpt uses optical switches

- Fully functional 12-node, degree 4 testbed integrated with NCCL

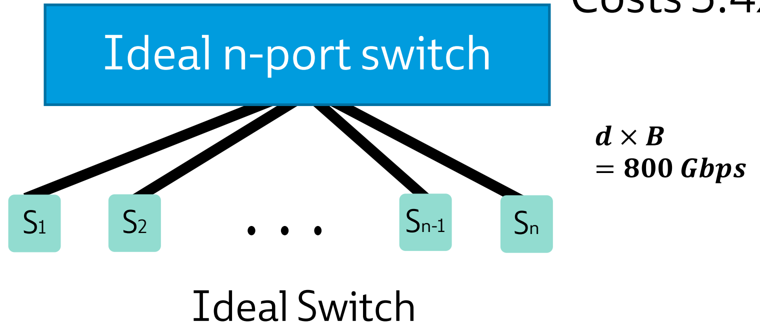


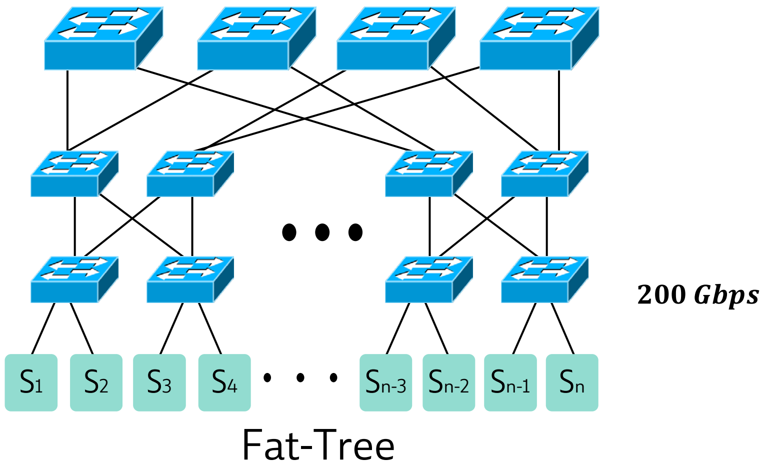
Evaluation

- We evaluate TopoOpt with large scale simulation and a small-scale prototype
- Artifact code can be found at http://TopoOpt.csail.mit.edu



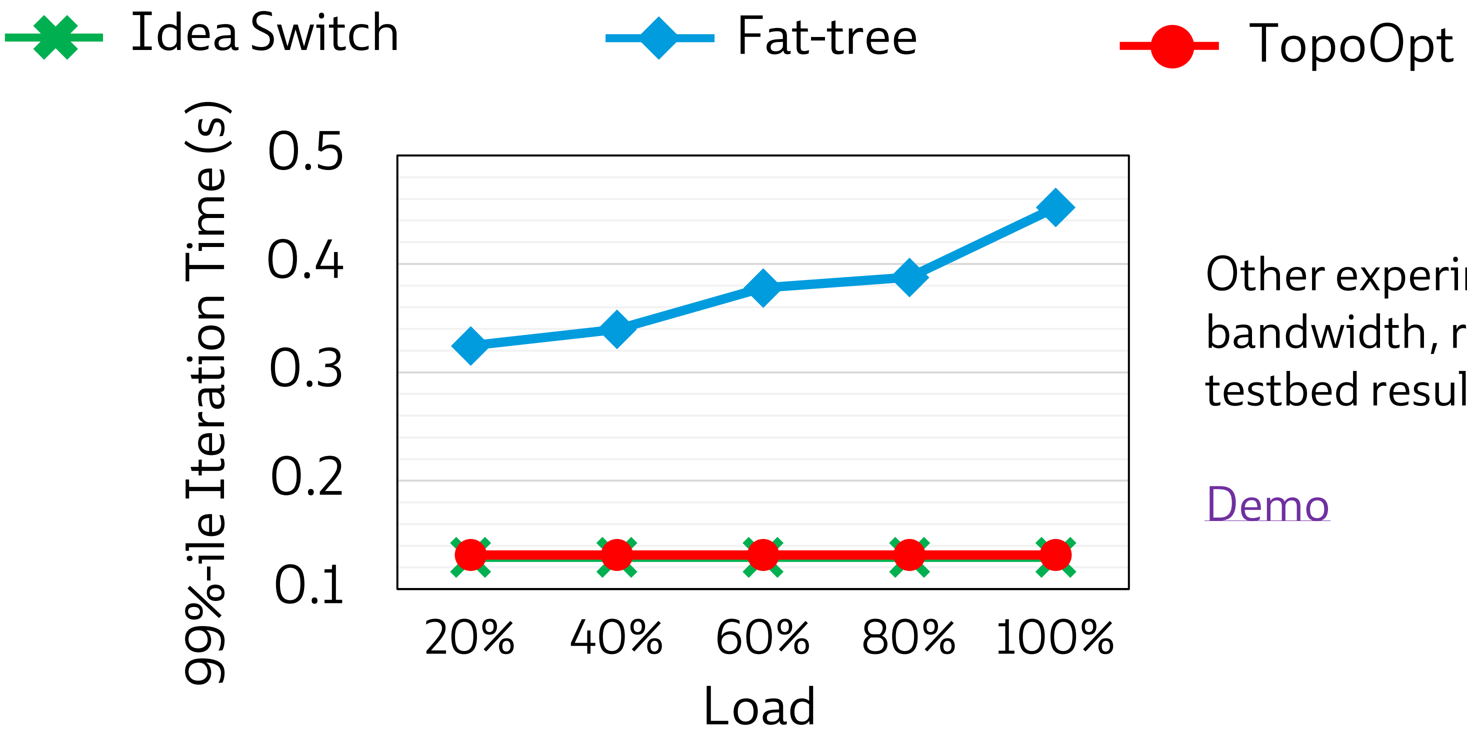
Costs 3.4x!





Simulation – tail completion time

- Running several jobs together on a 432 node, d = 8, 100Gbps TopoOpt system, compared to several other options

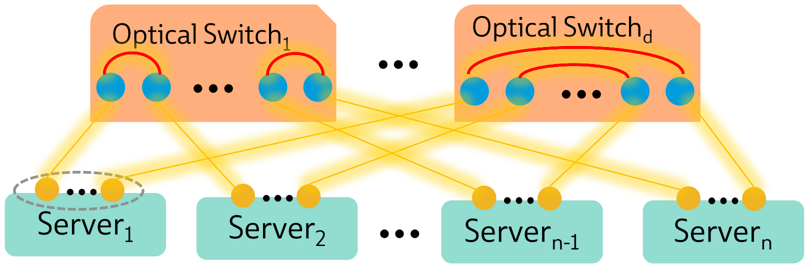


Other experiments with varying bandwidth, reconfiguration delay and testbed results are in the paper!

Demo

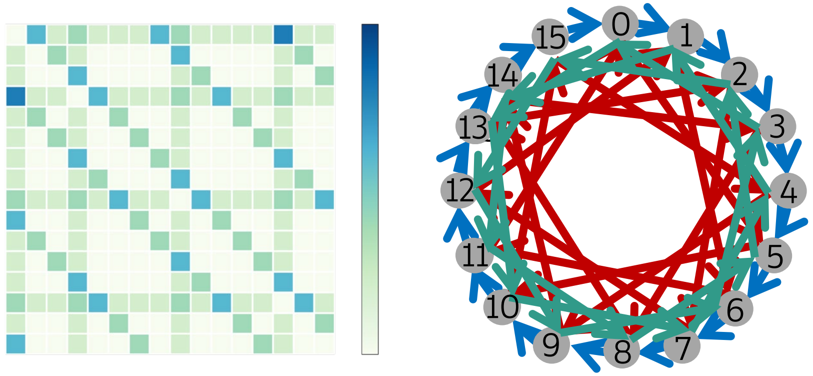
TopoOpt achieves up to **3.4x** faster 99%-tile latency compared to cost-equivalent Fat-trees

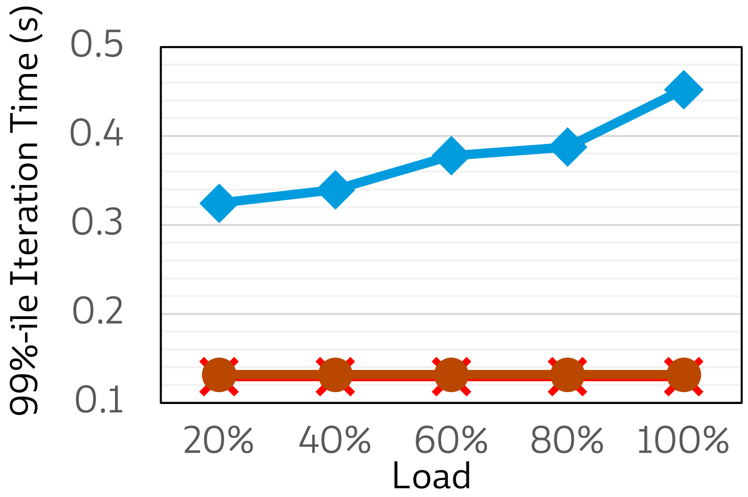
Summary



TopoOpt: the first system to co-optimize DNN training with demand-aware network topology

Leverages the mutability of DNN training traffic to search and construct the best topology





Achieves up to 3.4x faster 99%-ile training iteration time compared to cost equivalent Fat-trees