Lecture 4: Networking and Deep Learning

CS 256: Systems and Machine Learning

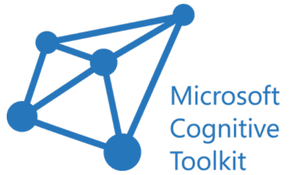
Sangeetha Abdu Jyothi

UCIrvine

Parts of this lecture were adapted from talks on Parameter Server, Horovod, and TicTac

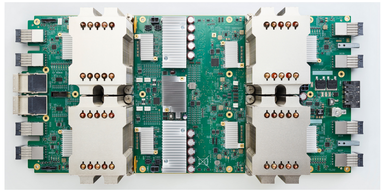
Previous Lectures

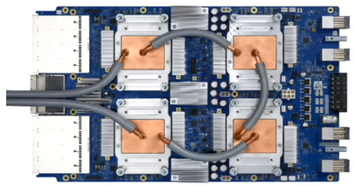
Deep Learning Frameworks

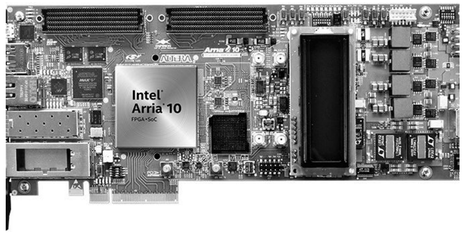


Deep Learning Compilers

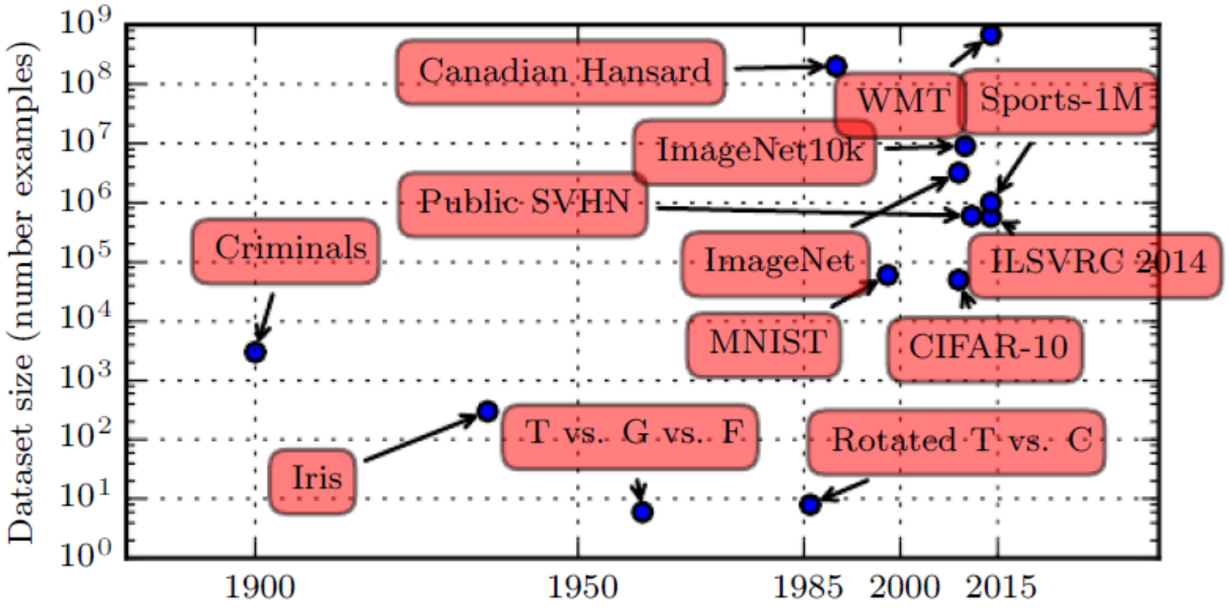
Hardware

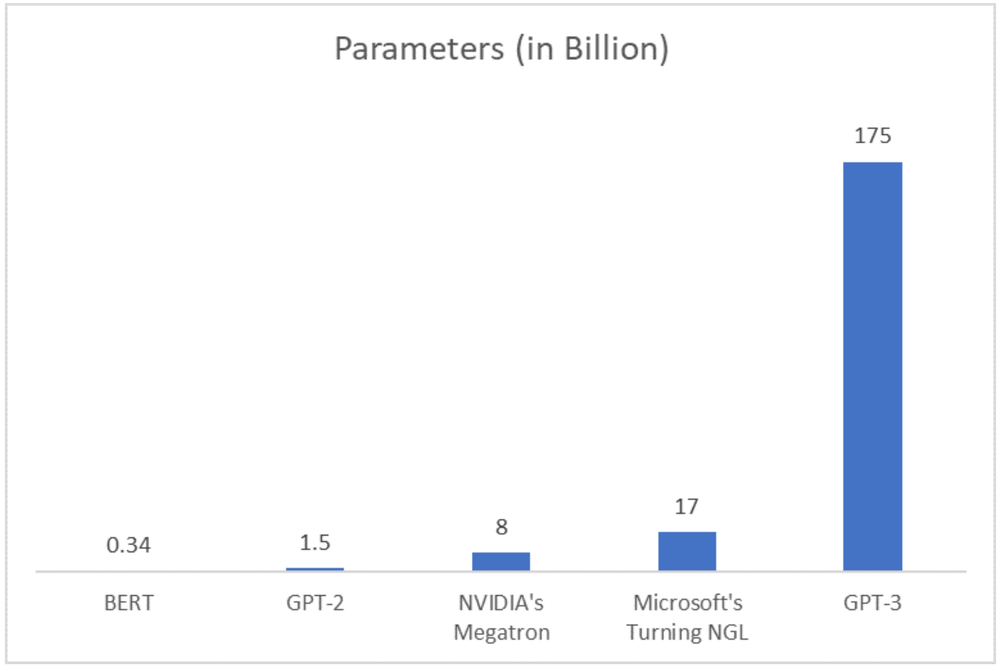






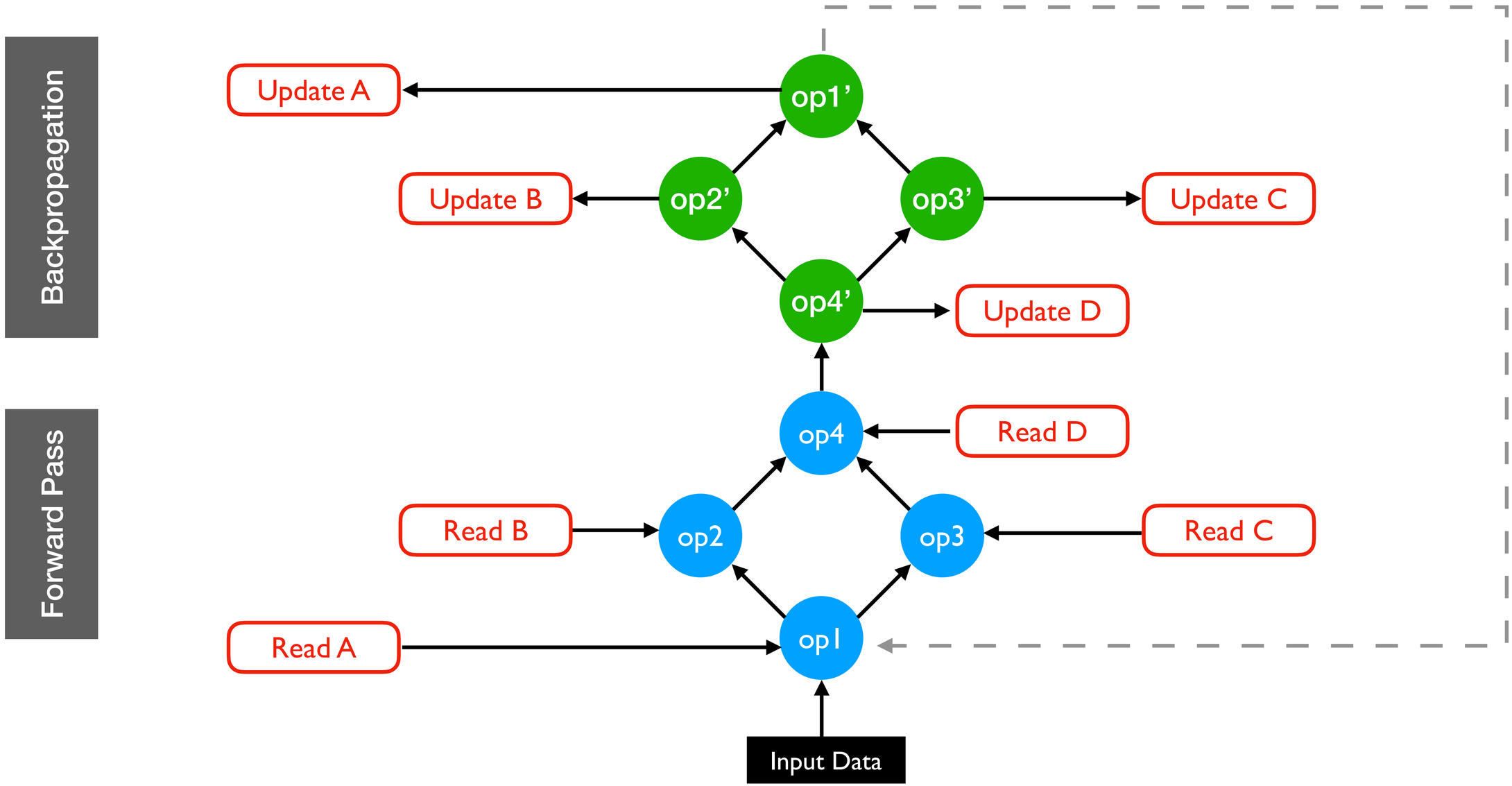
Rapid Growth



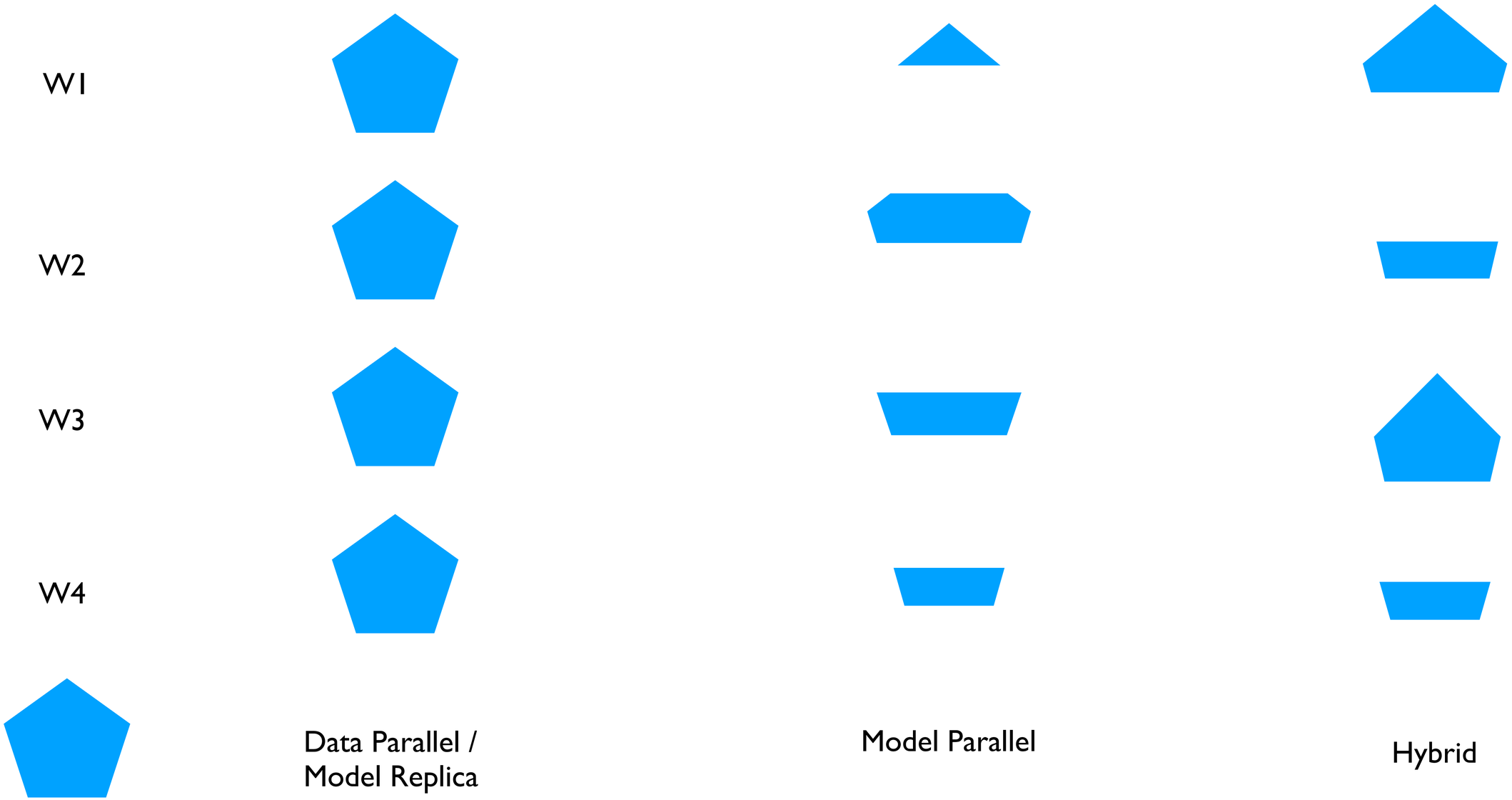


Datasets and Models are rapidly growing in size ➡️ Distributed training is necessary

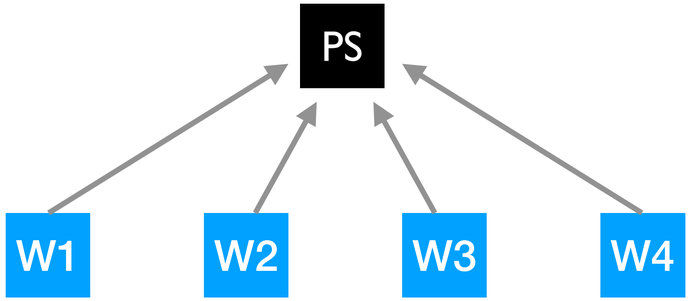
Training Recap



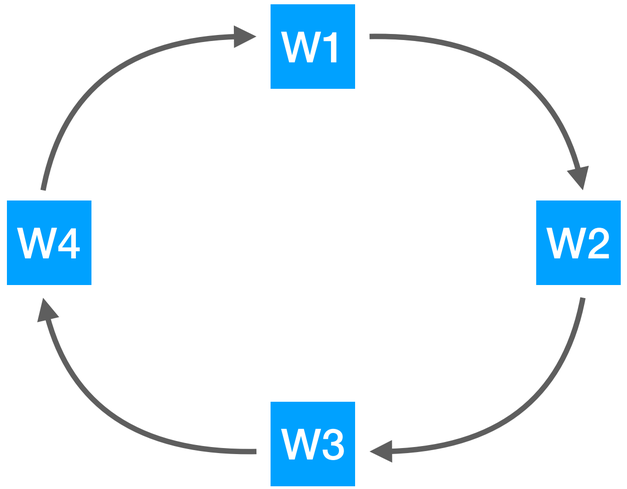
Distribution Patterns



Popular Modes of Network Aggregation



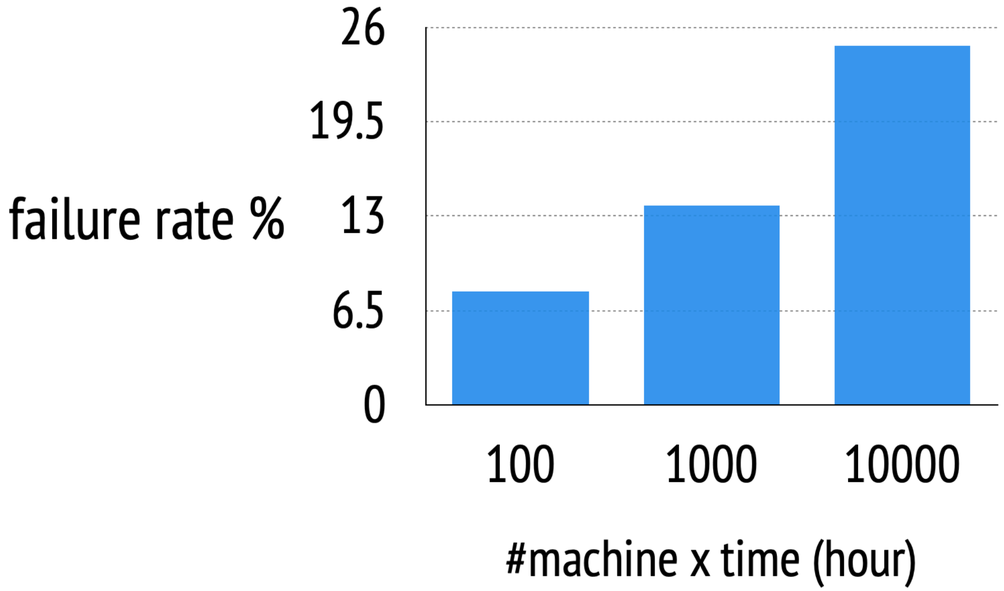
Parameter Server



Decentralized Aggregation

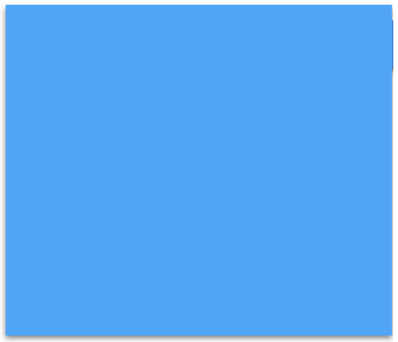
Parameter Server [OSDI'14]

- Goals
  - Scale to industry-scale problems
    - billions of samples and features
    - hundreds of machines
  - Enable efficient communication
  - Fault tolerance
  - Easy to use

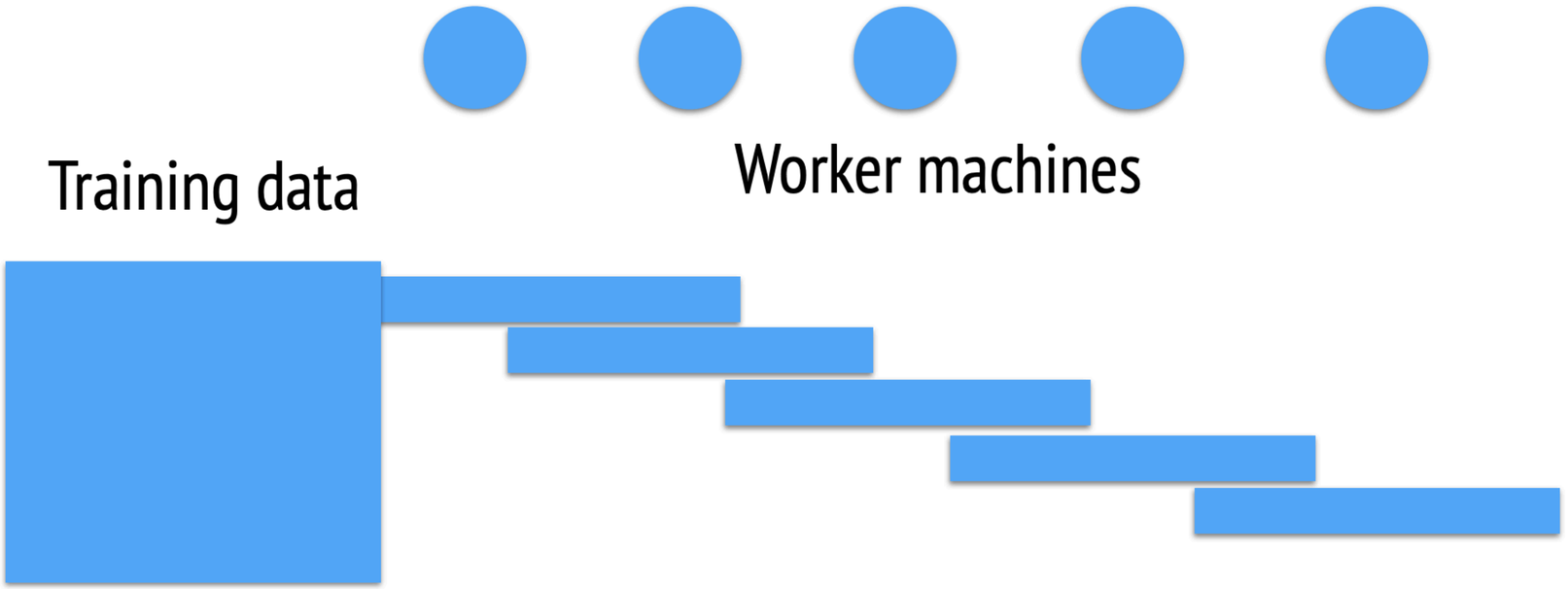


Data and Model Partitioning

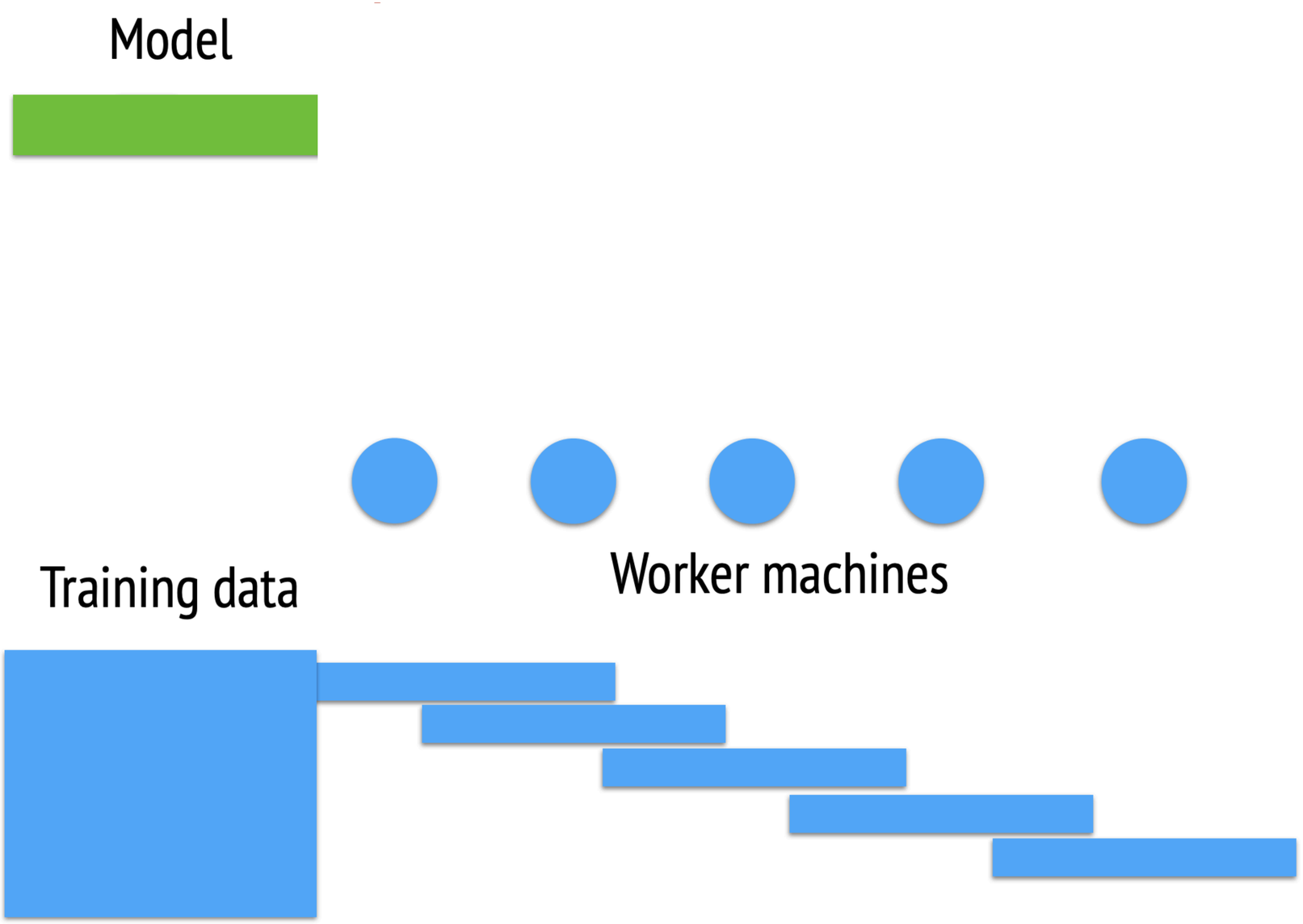
Training data



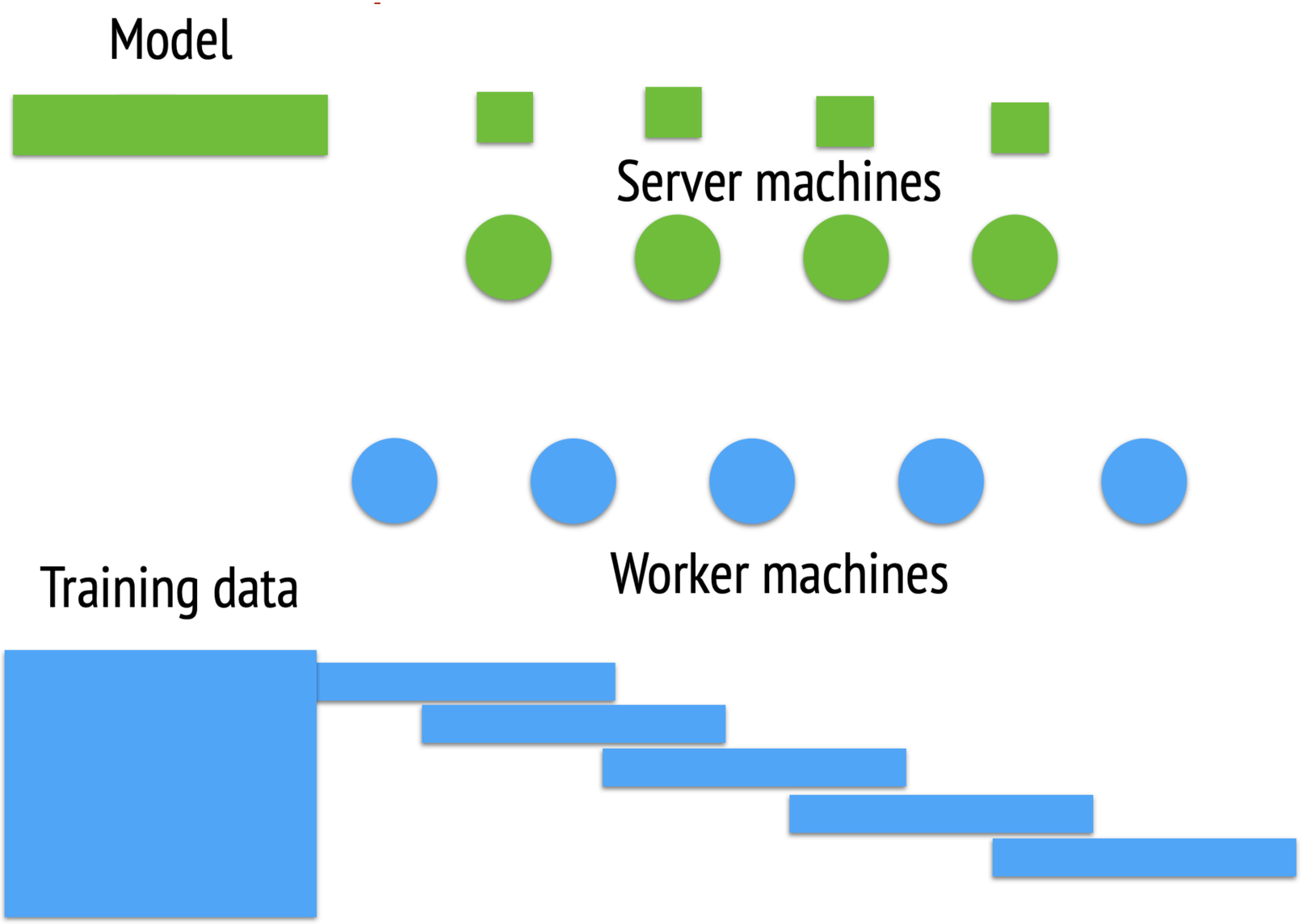
Data and Model Partitioning



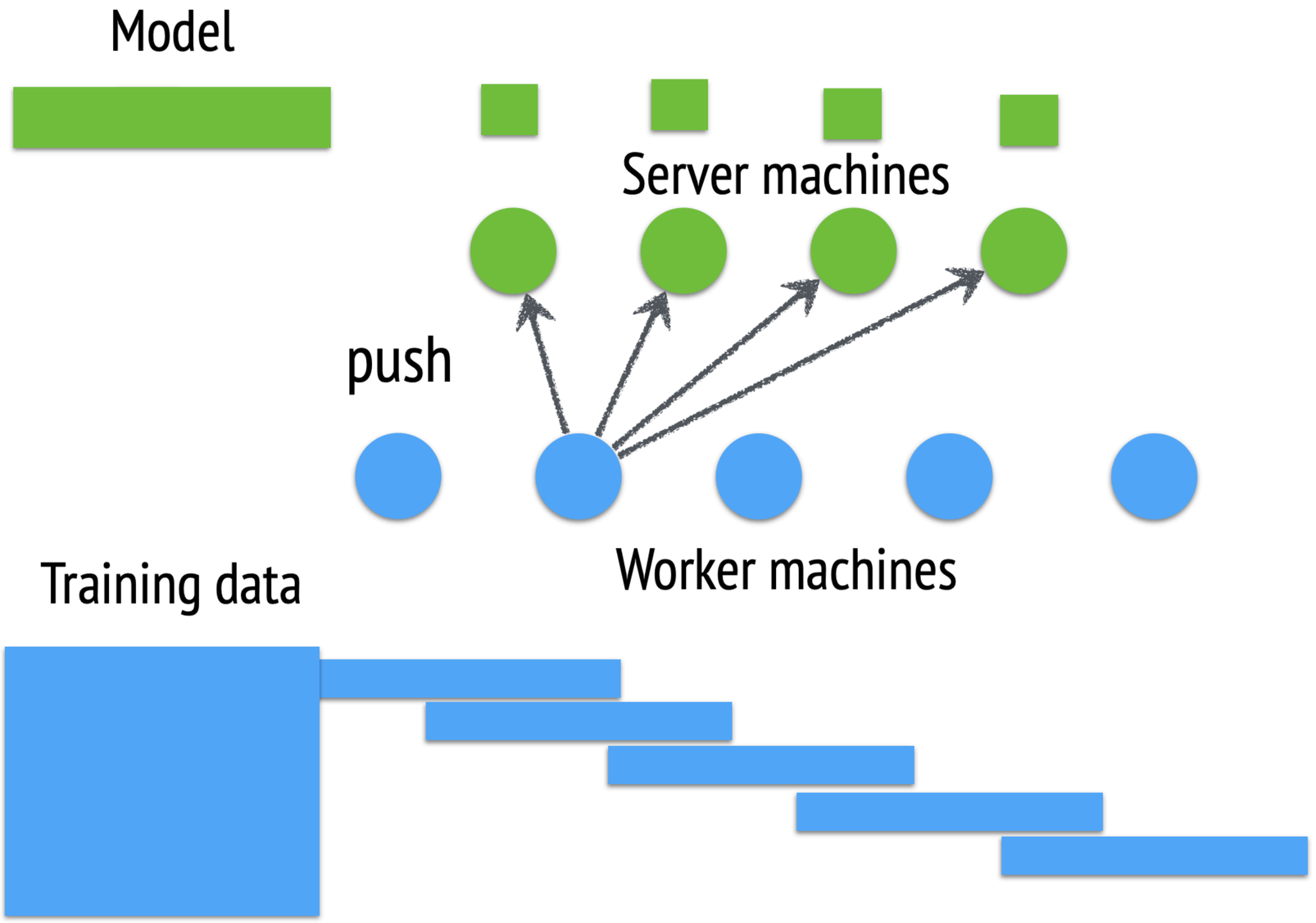
Data and Model Partitioning



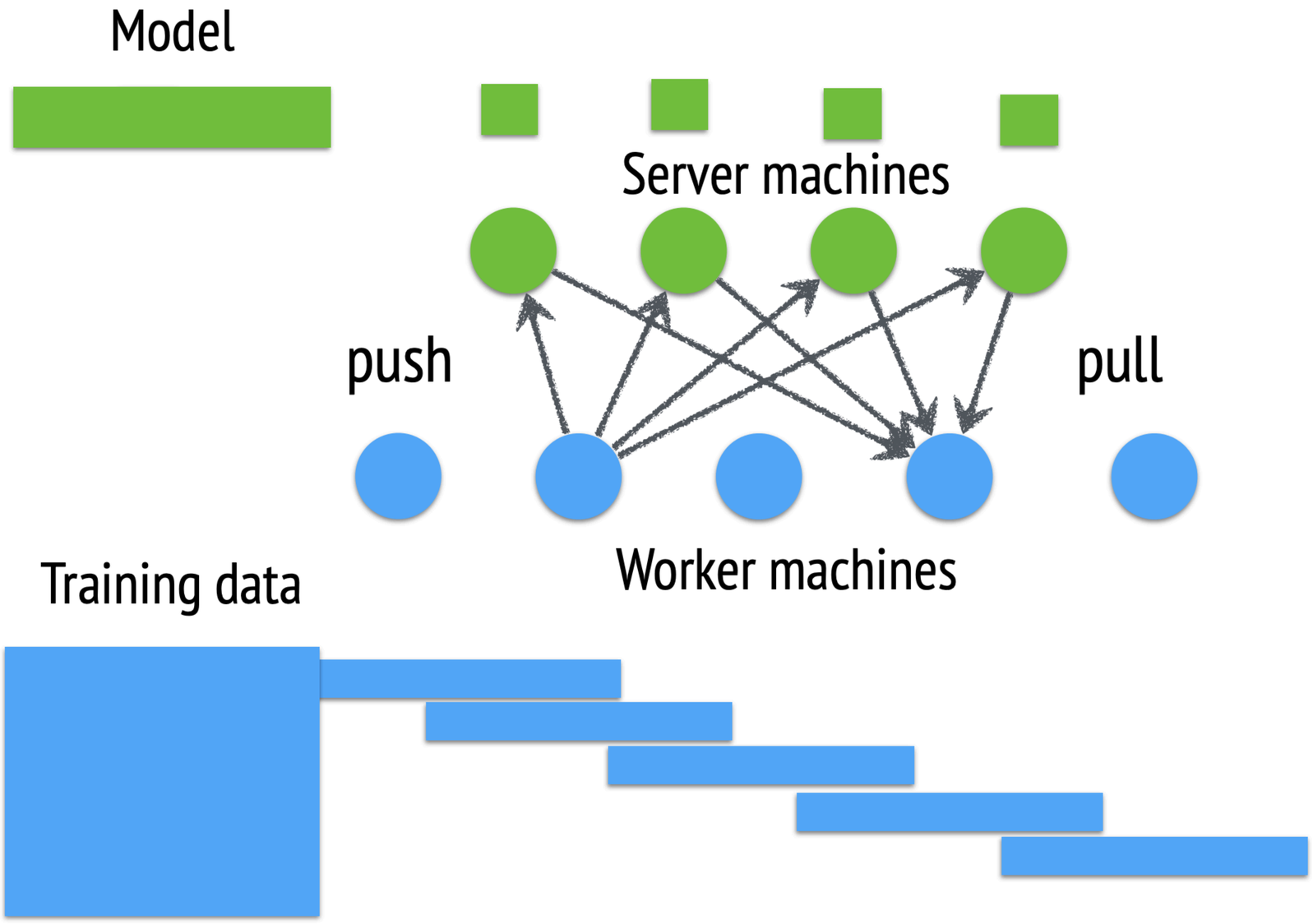
Data and Model Partitioning



Communication Operations

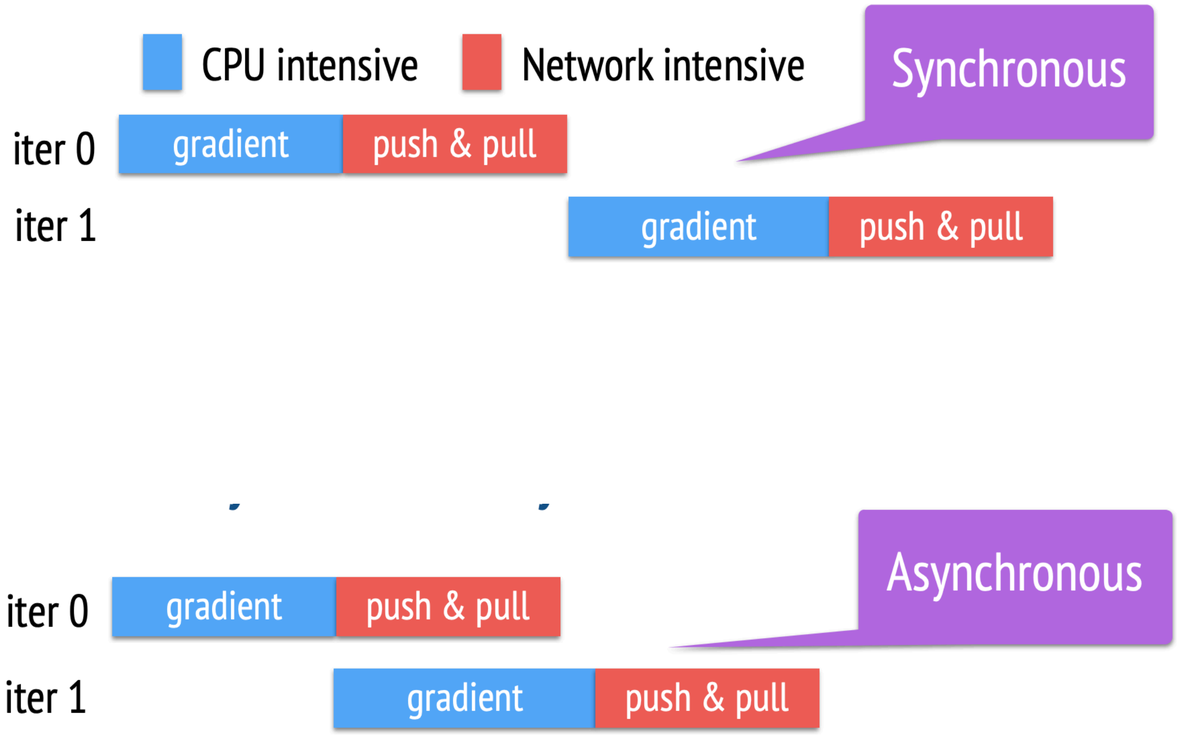


Communication Operations



Challenges in the vanilla model

- Massive communication traffic
  - Frequent access to the shared model
- Expensive global barriers between iterations



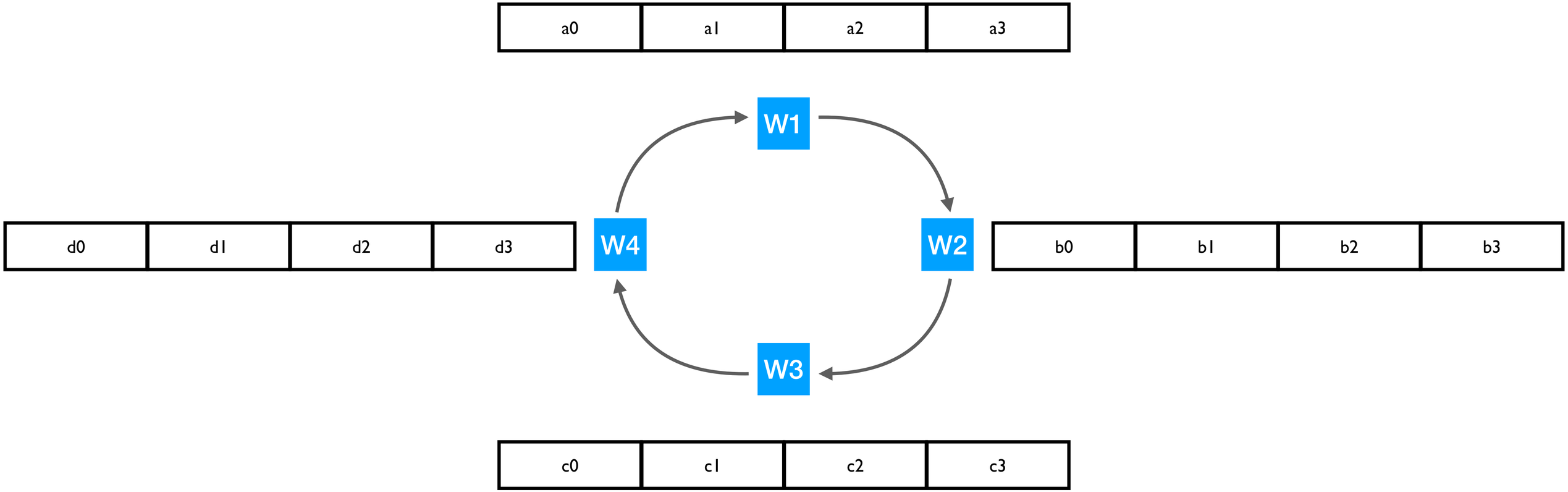
Issue with Parameter Server

Even with distributed PS architecture,

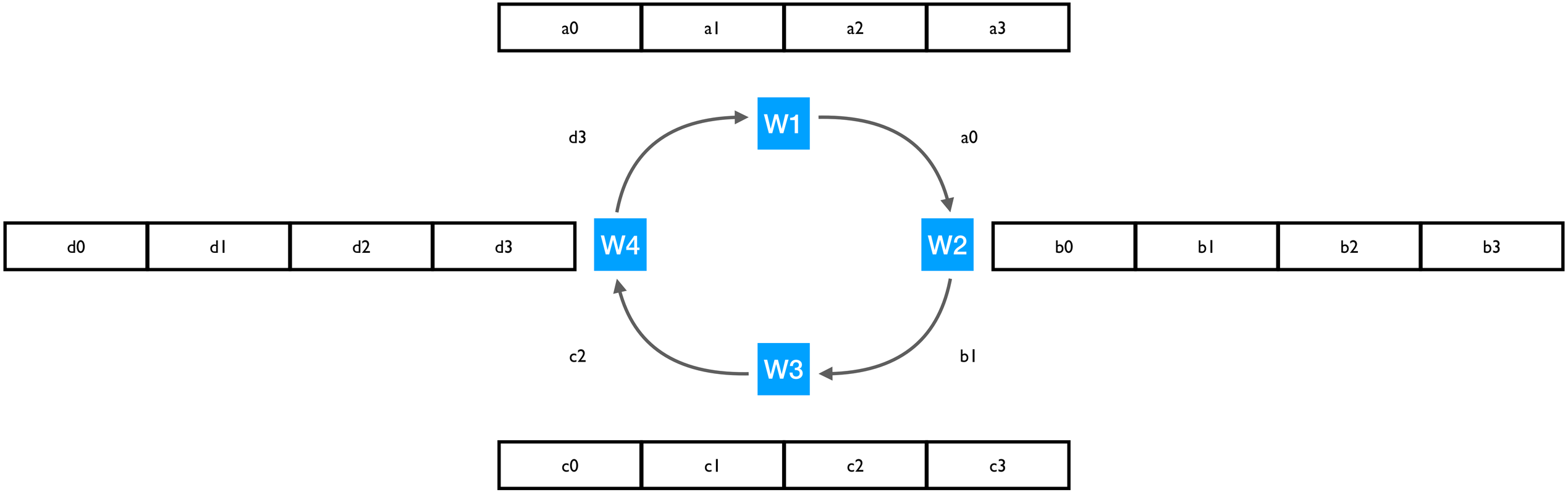
there can be network congestion at the parameter servers

Solution: Decentralized Aggregation

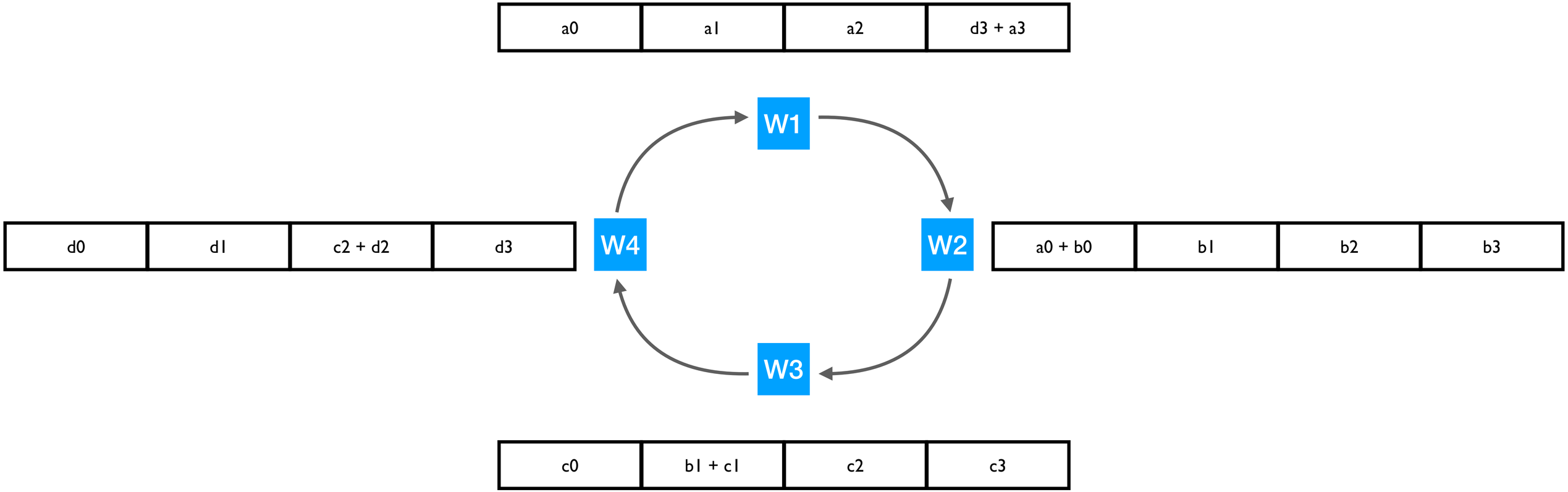
Ring AllReduce - Decentralized Aggregation



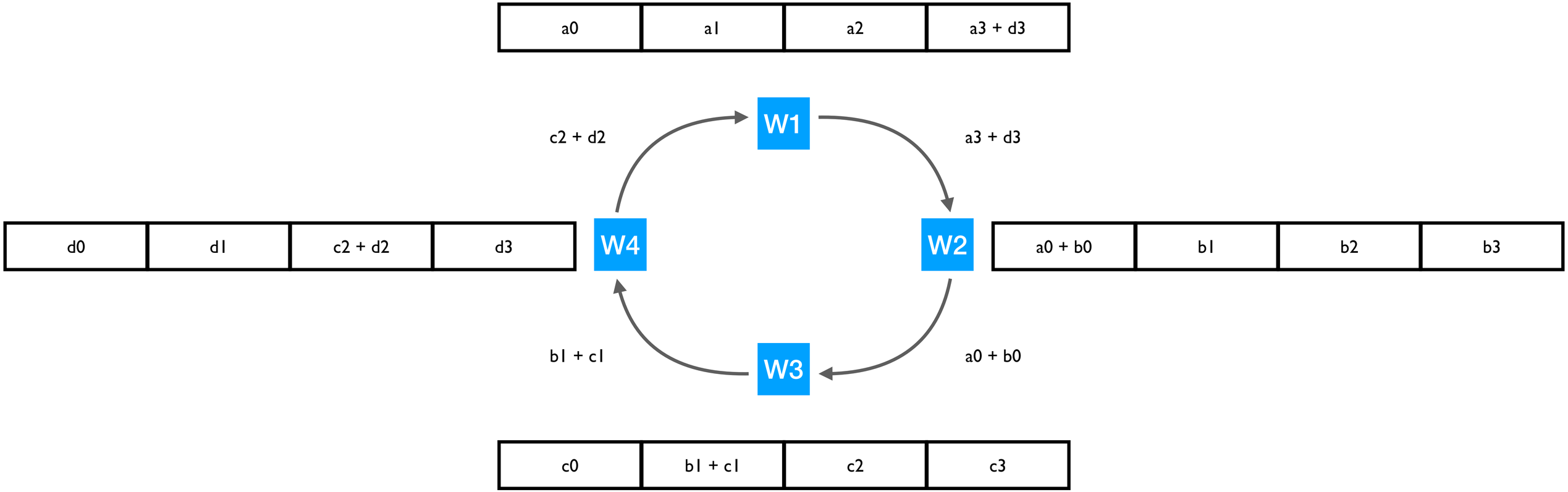
Ring AllReduce



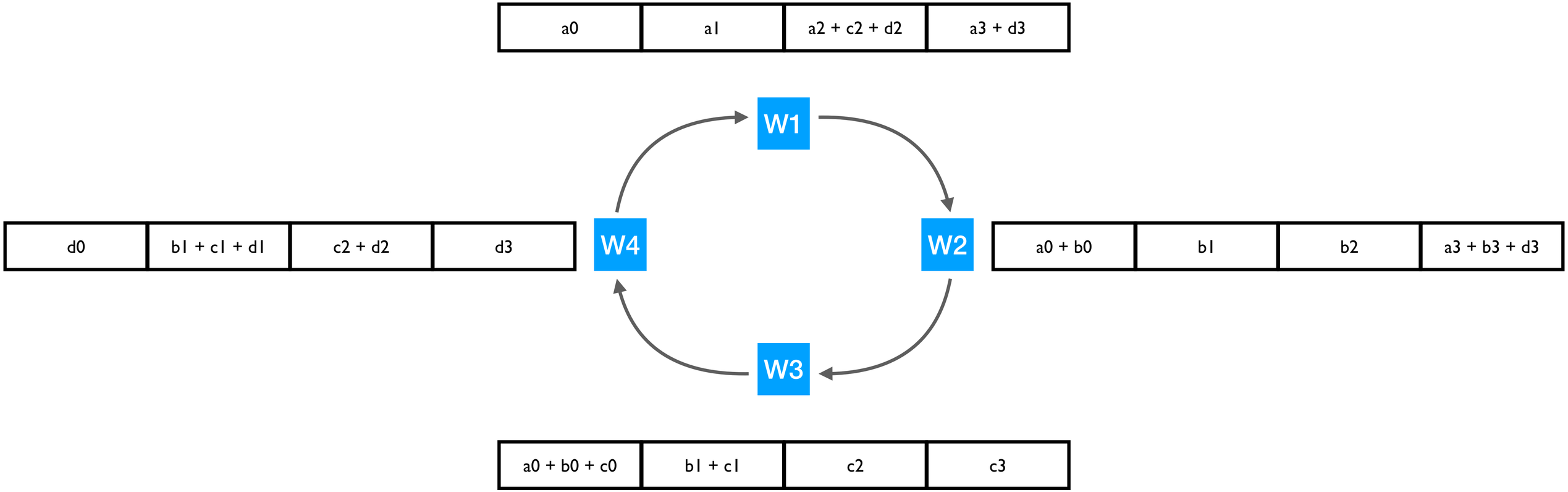
Ring AllReduce



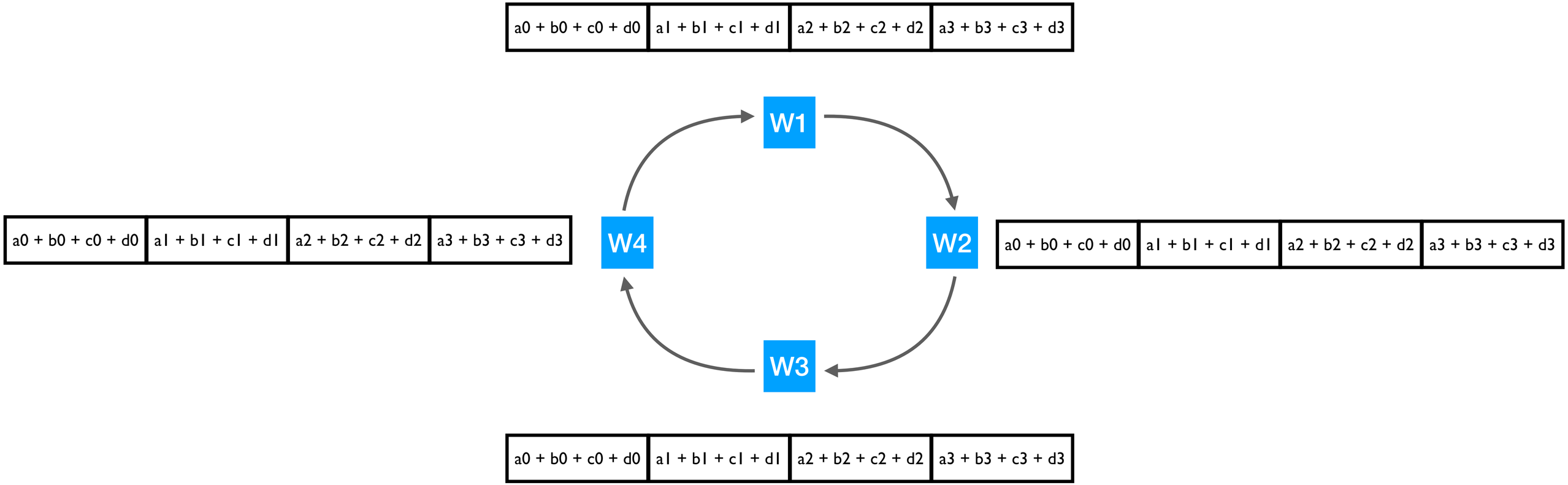
Ring AllReduce



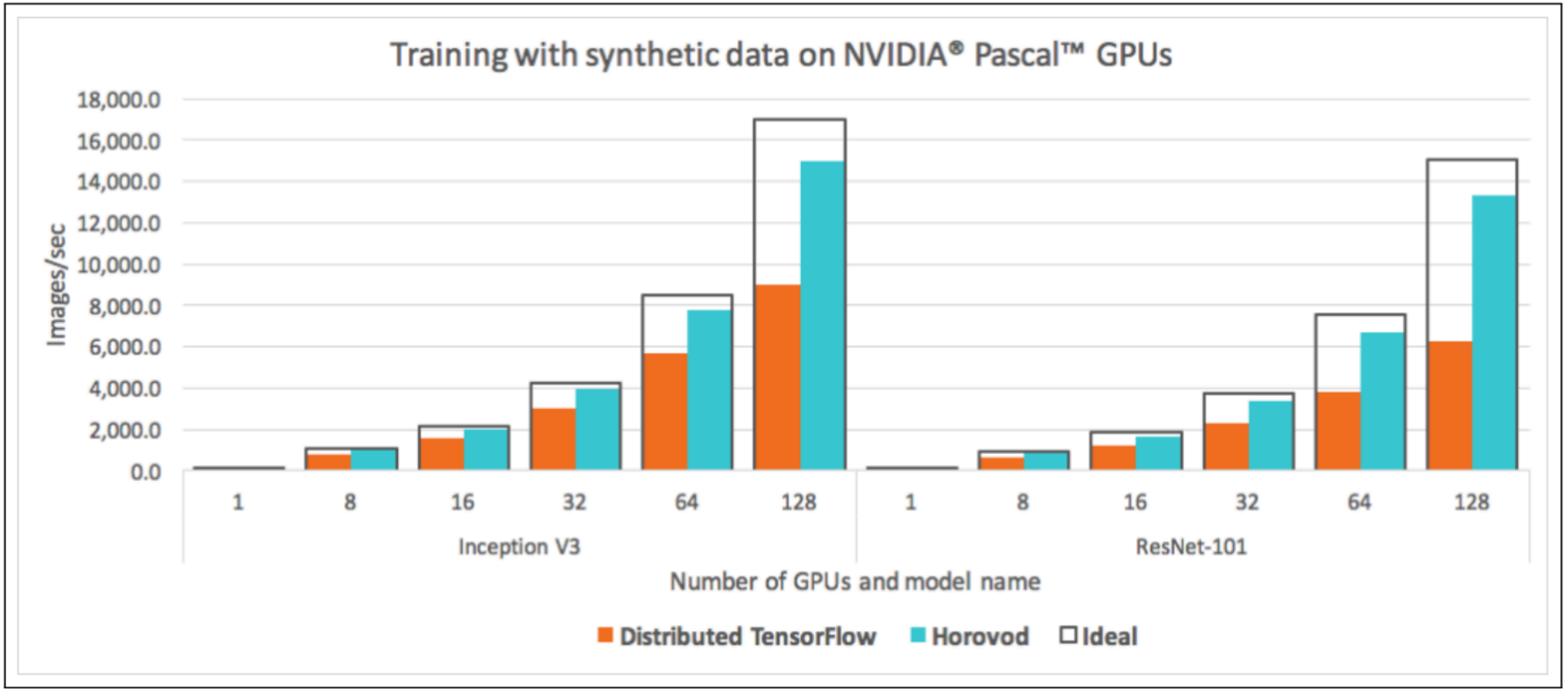
Ring AllReduce



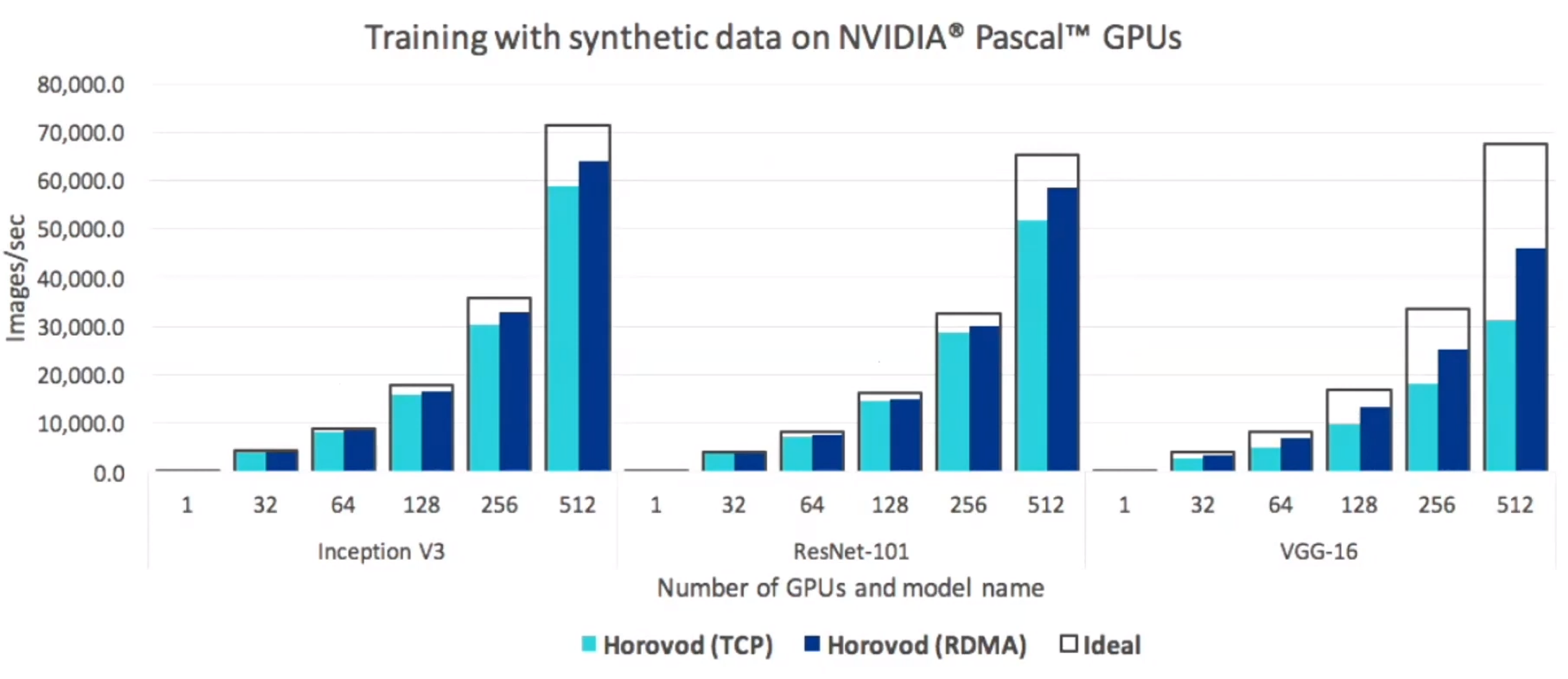
Ring AllReduce



Performance of Horovod (Ring AllReduce Implementation)



Performance



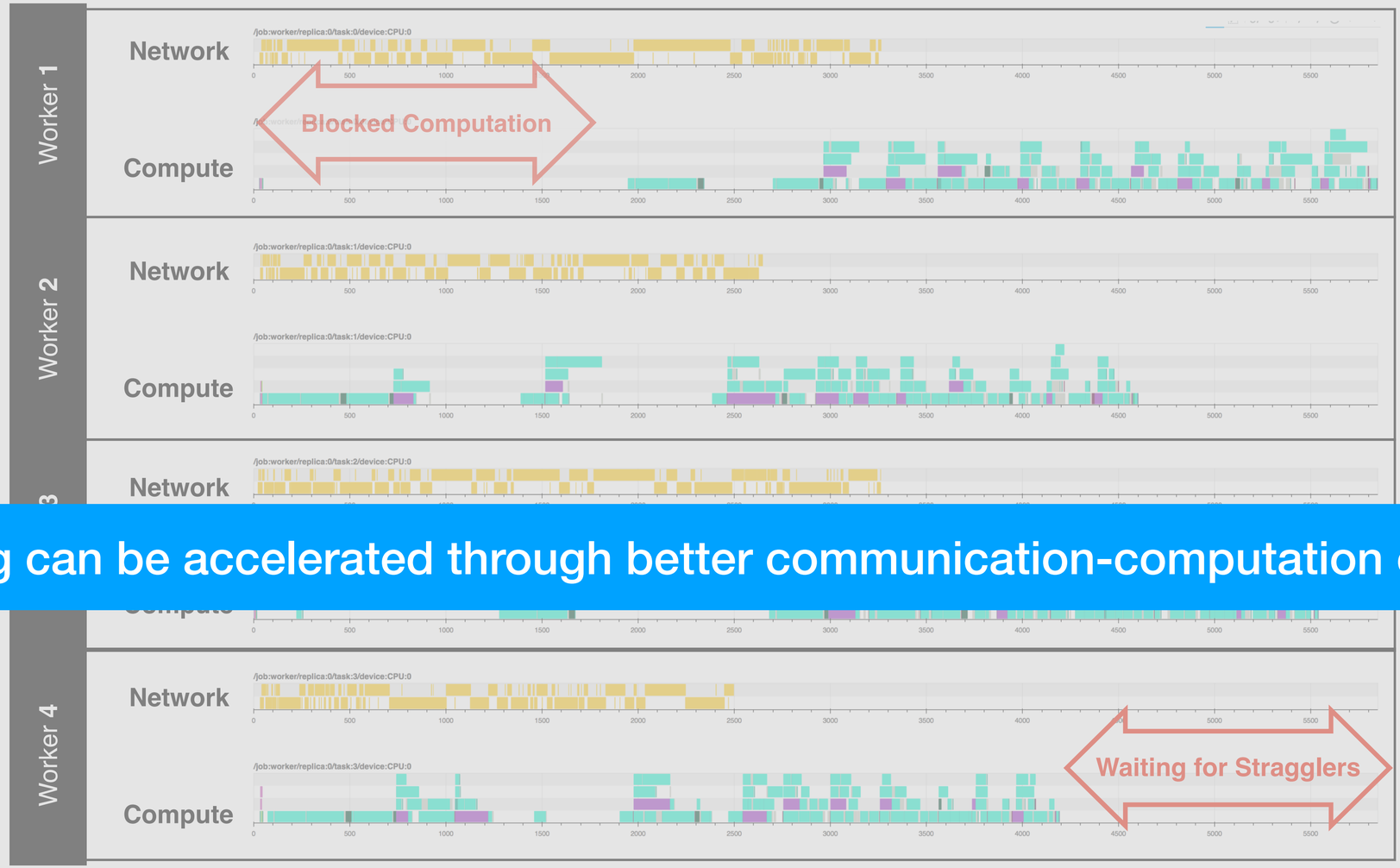
AllReduce advantages

- Better performance
- More scalable
- Fits well with Torus topology

An issue with both PS and AllReduce

Compute under-utilization

Understanding Compute Underutilization

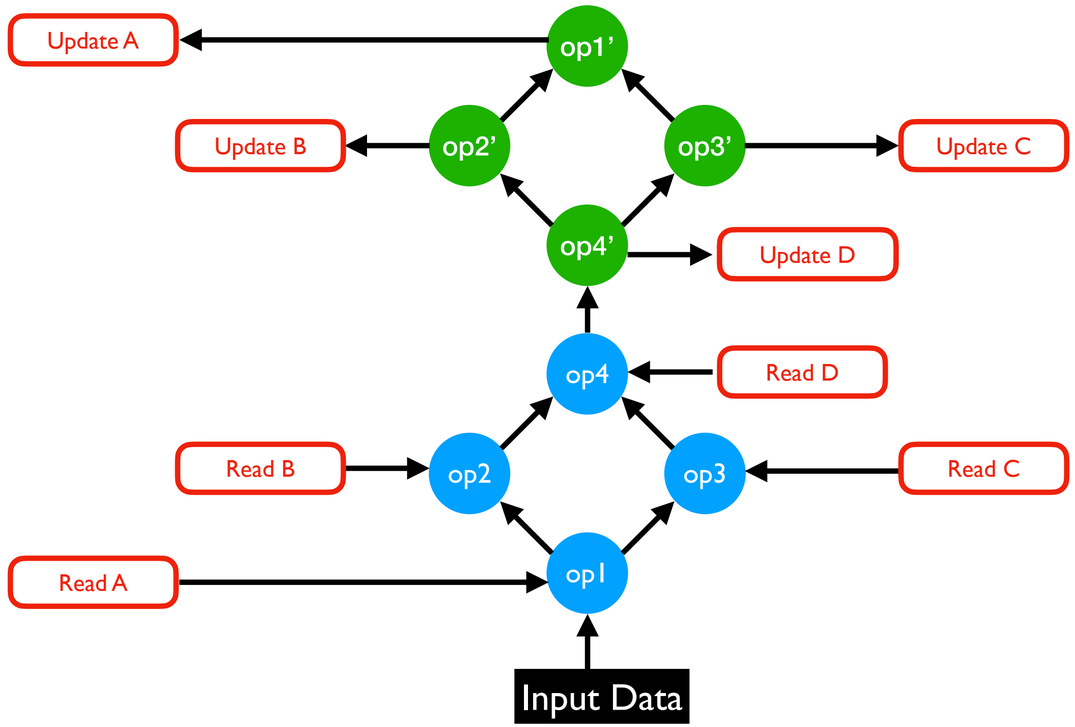


Training can be accelerated through better communication-computation overlap

Inception v3
Data-Parallel with Parameter Server
TensorFlow
Mustang: CPU

Cause: Random Order of Parameter Transfers

- In this example, the computation cannot start until parameter A is received
- B, C, or D may be transferred before A, thereby blocking the computation
- To make things worse, parameters that are updated last are consumed first



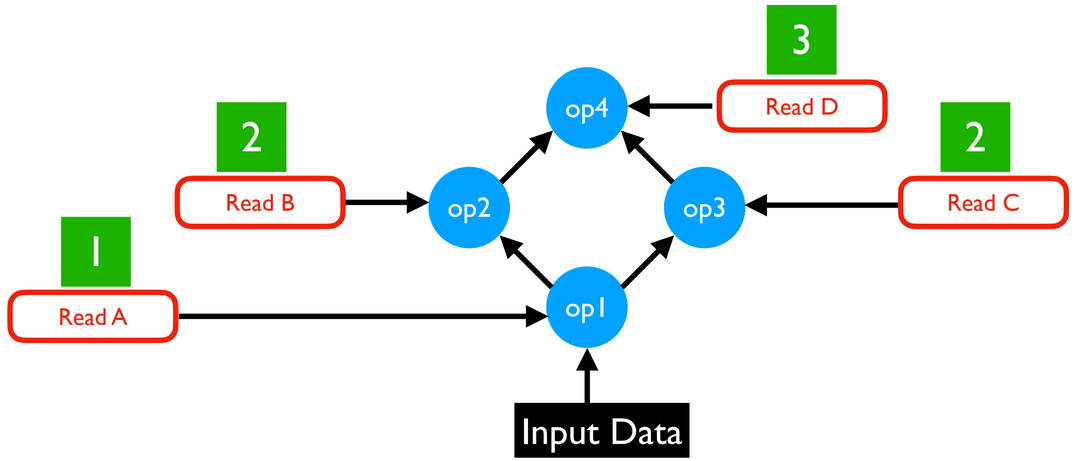
TicTac and P3 [MLSys'19] High-level idea

- Improve iteration time through better communication-computation overlap in Parameter Server based aggregation

- Achieved through parameter transfer scheduling

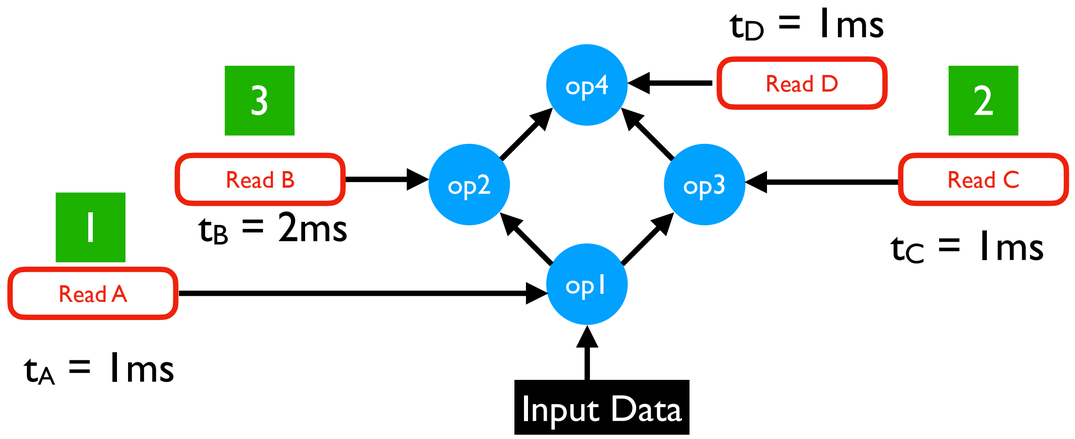
Timing Independent Computation Scheduling

- Uses DAG structure only
- Assign priorities based on the number of communication operations *dependent* on a given transfer
- In the e.g, A has no other transfers dependent on it. Hence, it gets the highest priority
- B and C each have one dependency. Hence, the next priority
- D assigned lowest priority



Timing Aware Computation Scheduling

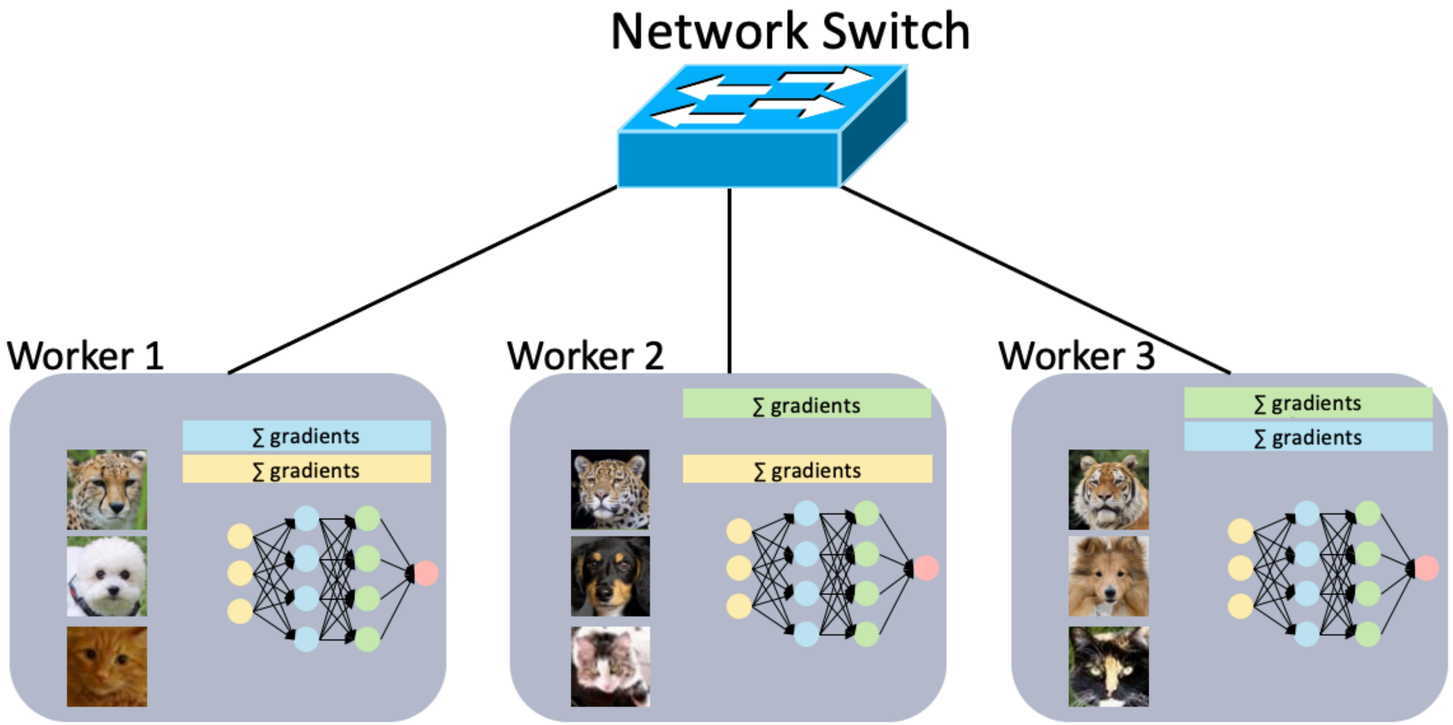
- Uses DAG structure and time taken by each operation
- Reduce blocking on the critical path
- A assigned highest priority
- C is the next smallest blocking transfer
- Followed by B, then D



DNN Training Acceleration

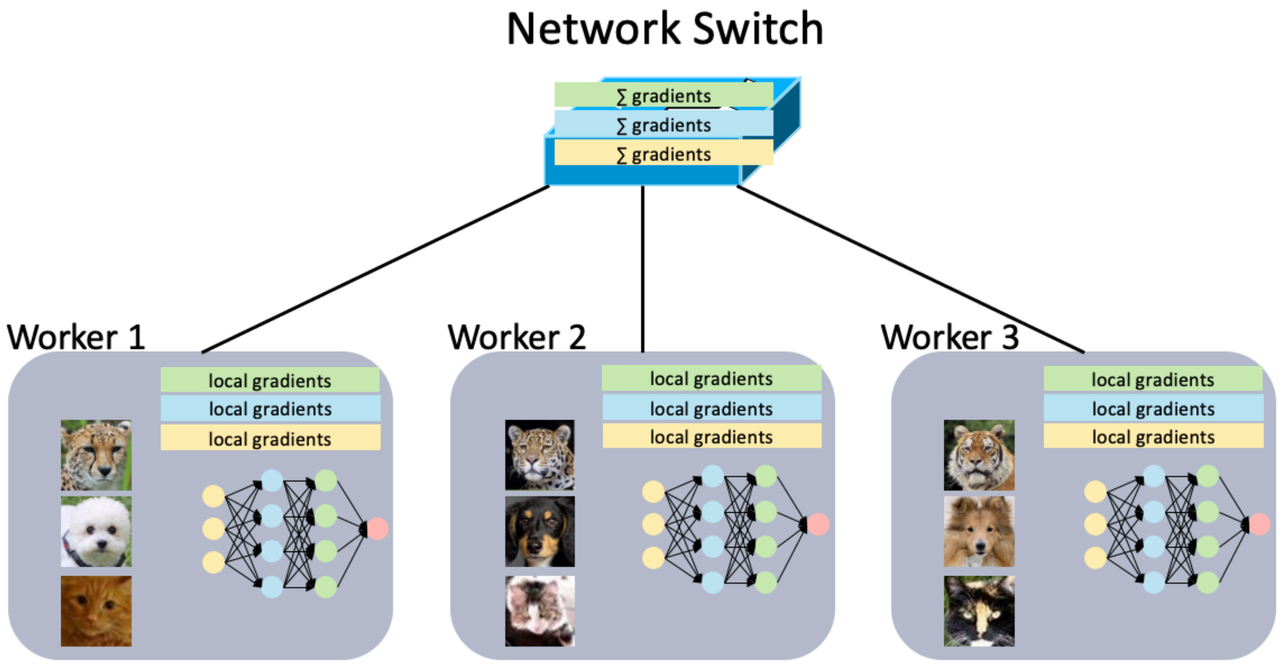
- Improving communication-computation overlap
    - Communication Scheduling: TicTac [MLSys'19], P3 [Jayarajan, MLSys'19], ByteScheduler [SOSP'19]
    - Computation scheduling: BytePS [OSDI'20], Caramel [arXiv'20]
    - Hybrid mode: PipeDream [SOSP'19]
- Increasing computation time
    - Increase batch size [Iandola et al., 2016]
    - Model-dependent solution [Goyal et al., 2017; Cho et al., 2017; You et al., 2017; Akiba et al., 2017]
- Decreasing Communication Time
    - Reduce Number of Messages [Alistarh et al., 2017; Wen et al., 2017; Zhang et al., 2017]
    - Decrease Message Size [Vanhoucke et al., 2011; Courbariaux et al., 2015; Gupta et al., 2015]

In-network Aggregation for Shared Machine Learning Clusters [MLSys'21]



Data-Parallel DNN Training Using Ring-AllReduce

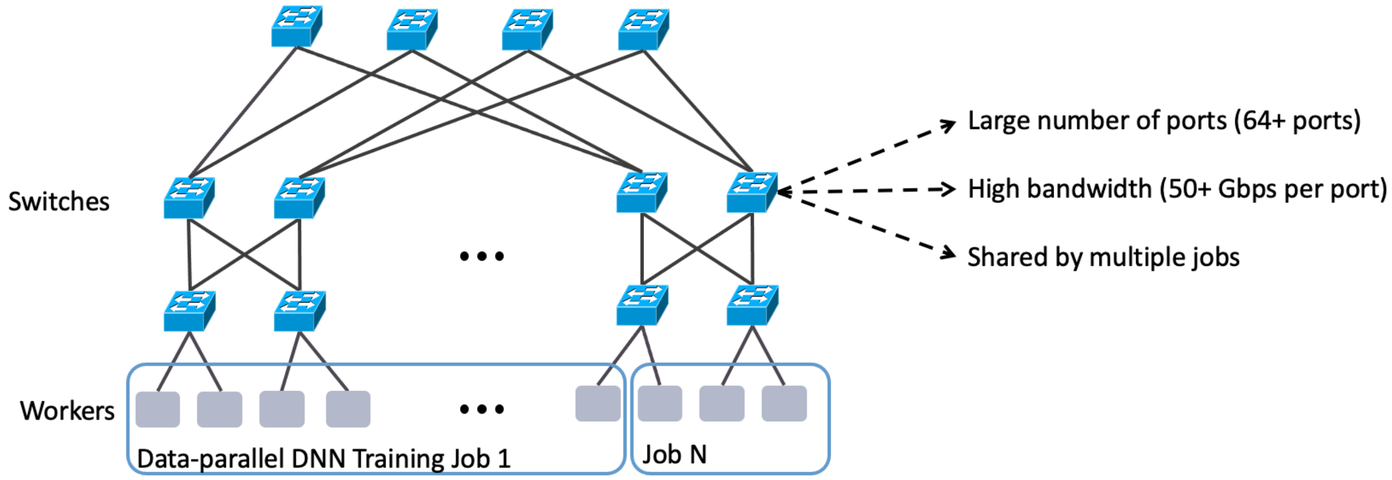
In-network aggregation



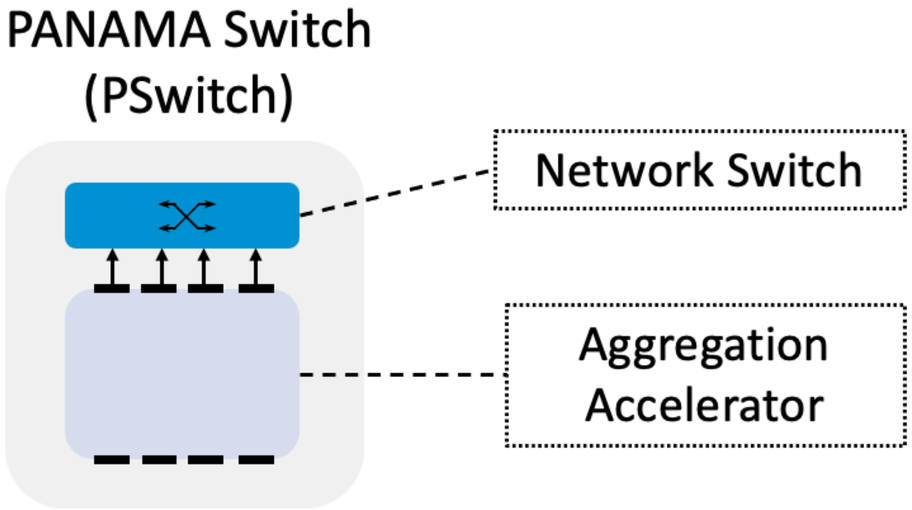
Challenges for Wide Adoption of In-network Aggregation

- Efficient hardware support for in-network aggregation at large-scale

- Fair and balanced use of network resources by aggregation and non-aggregation flows

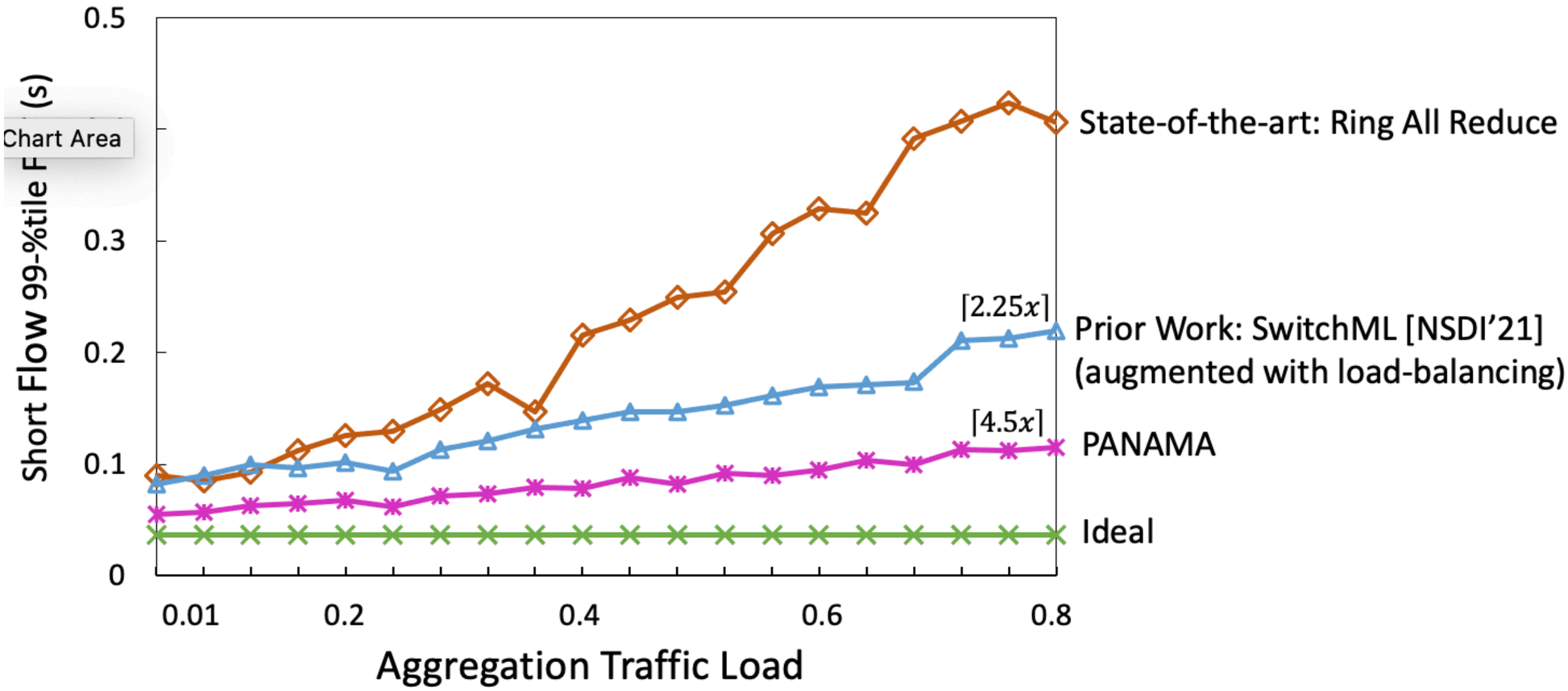


PANAMA Switch



- Routing and switching
- In-network aggregation specialization
  - Support for floating point computation

Short Flow Latency



Other Related Work

- Considering stragglers in compute during scheduling
- SmartNICs for AllReduce
- Handling heterogeneity

Thanks!