# Multi-Node Training

# Different Kinds of "Multi-Node Jobs"

- model fits into single GPU
- model needs multiple GPUs but fits on single node
- model needs multiple nodes
- model is replicated across multiple GPUs on a single node
- model is replicated across multiple nodes

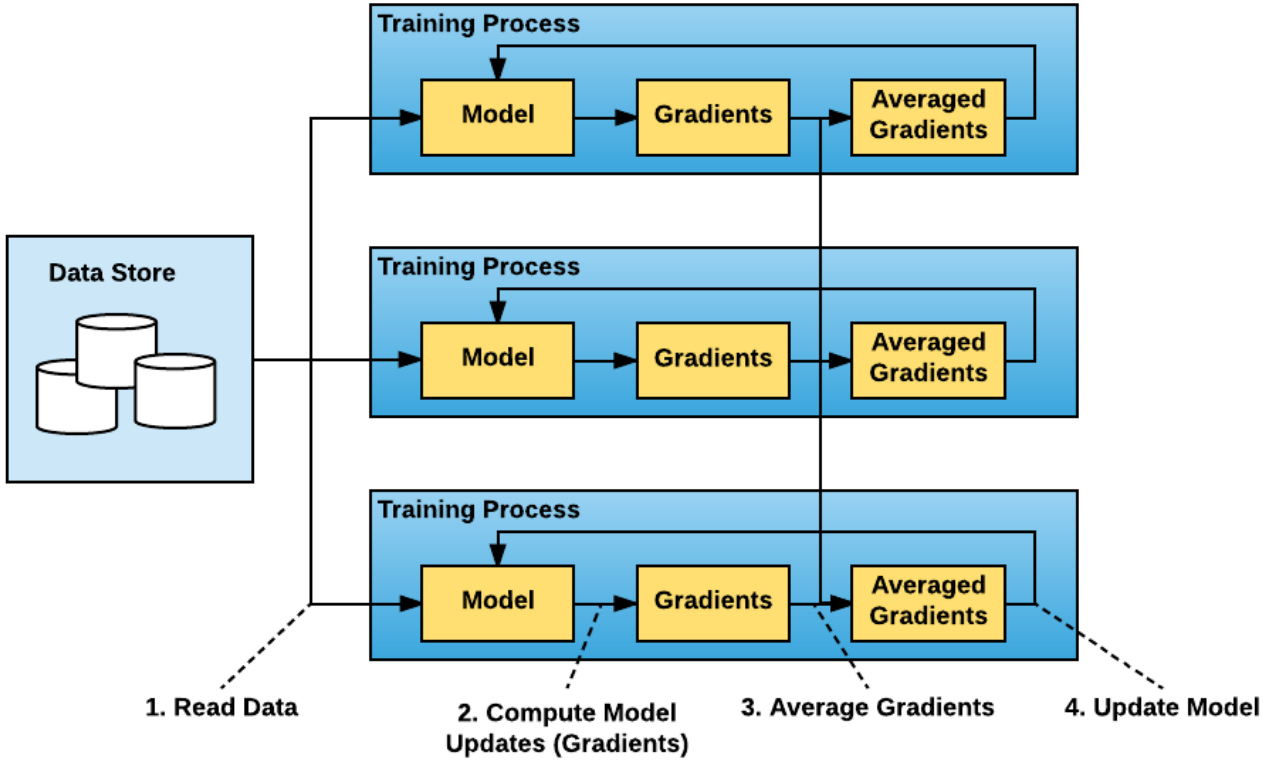
# Mixed Communications

- PCI bus for CPU, CPU↔GPU (computations involving single model)
- RDMA to CPU memory (data)
- GPUdirect for I/O to GPU, RDMA (data loading, inter-node parameters)
- NVLINK between GPUs (multi-GPU SGD within a node)
- standard Ethernet to CPU memory (control messages, maybe data)

# Bottlenecks of Single Node Jobs

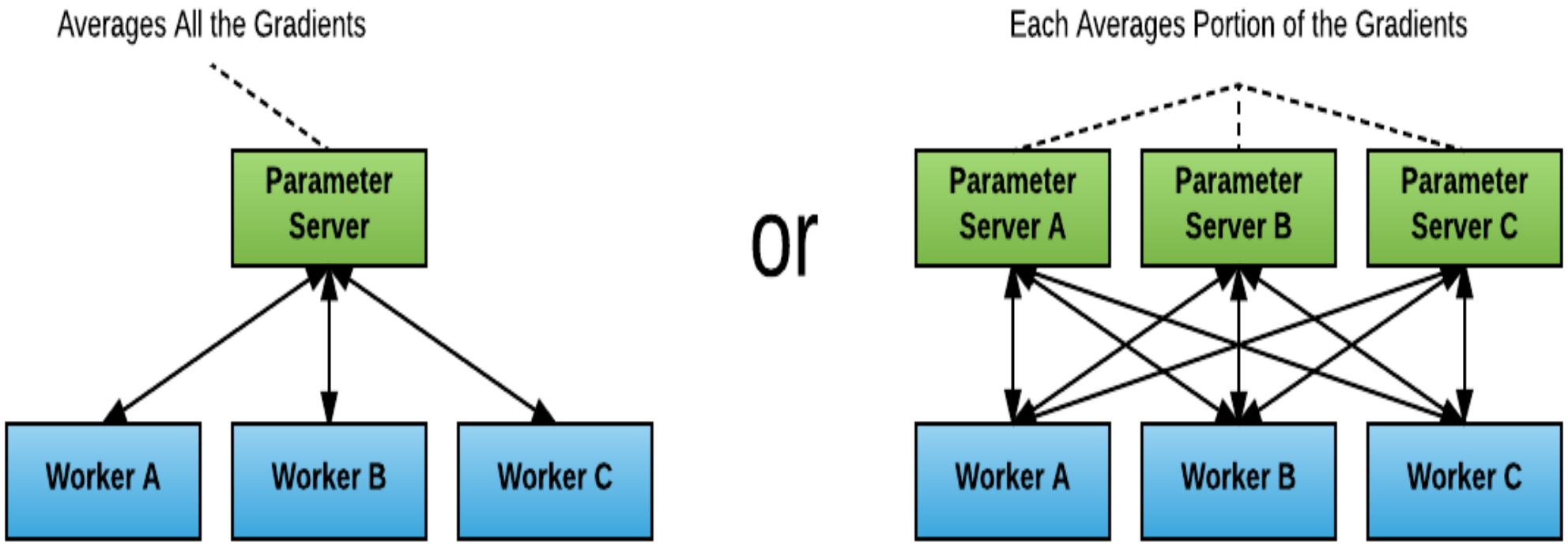
- limited number of GPUs per node
- limited PCI bus bandwidth
- limited CPU power
- limited local storage

# Multi-GPU / Multi-Node Training



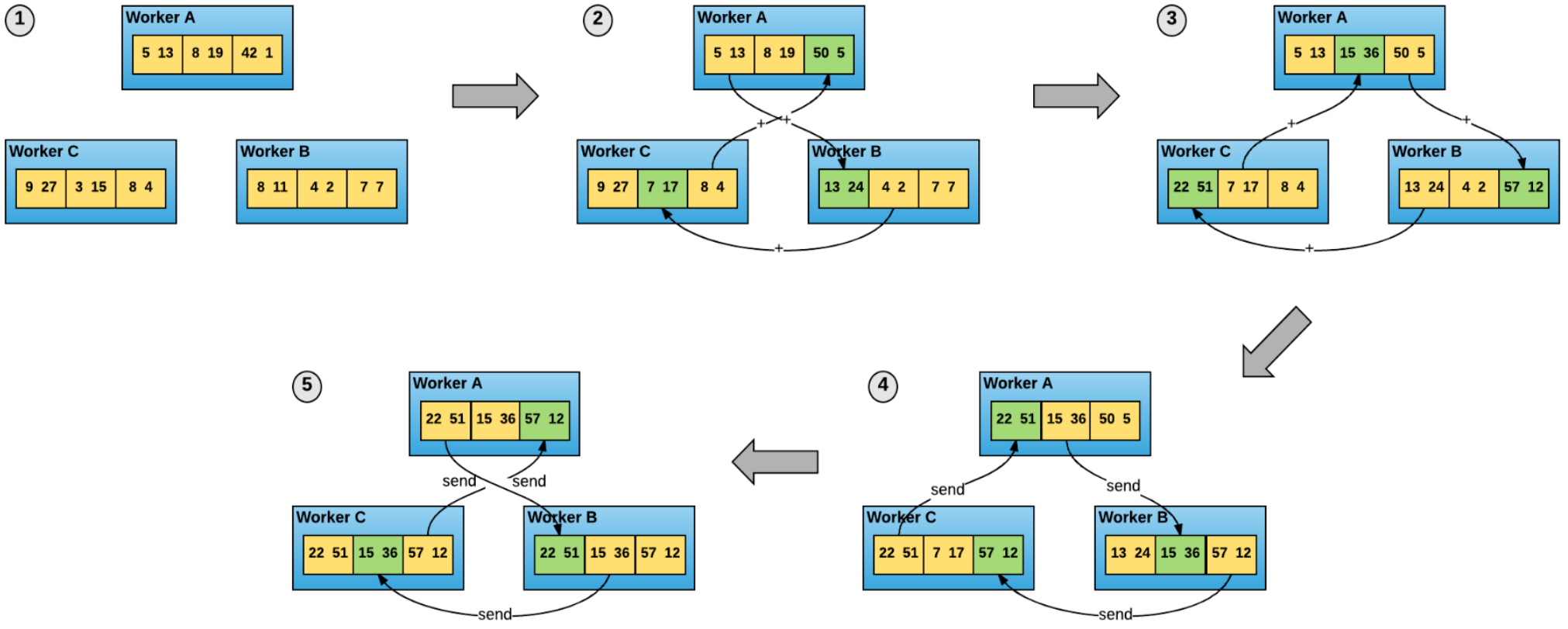

(source: Horovod)

# Parameter Servers




(source: Horovod)

# Direct Parameter Exchanges




(source: Horovod)

# How do you distribute?

- starting up jobs across clusters / machines
  - Ansible
  - Kubernetes
  - Slurm
- communications libraries
  - sockets, ZMQ
  - NCCL, Gloo
  - torch.distributed, Horovod
- all-in-one
  - MPI

# My Toolkit

- Ansible
- Kubernetes
- ZMQ, NCCL
- toch.distributed