Framework Design

2019/09/07

Horovod/Bluefog Consideration

Comm.
MPI
NCCL
Gloo

Framework
TensorFlow
PyTorch
MXNet

Device
CPU
GPU

Type
Int
Double
Float

Alg. Style
Async
Sync

Operation
Send/Recv
Broadcast
Reduce

Comm Style
All
Neighbor
Point to Point

Fault Tolerance
Timeout
Backup
Self-healing?

Bluefog Consideration (First Priority)

Comm.
MPI
NCCL
~~Gloo~~

Framework
~~TensorFlow~~
PyTorch
~~MXNet~~

Device
CPU
GPU

Type
~~Int~~
~~Double~~
Float

Alg. Style
Async
Sync

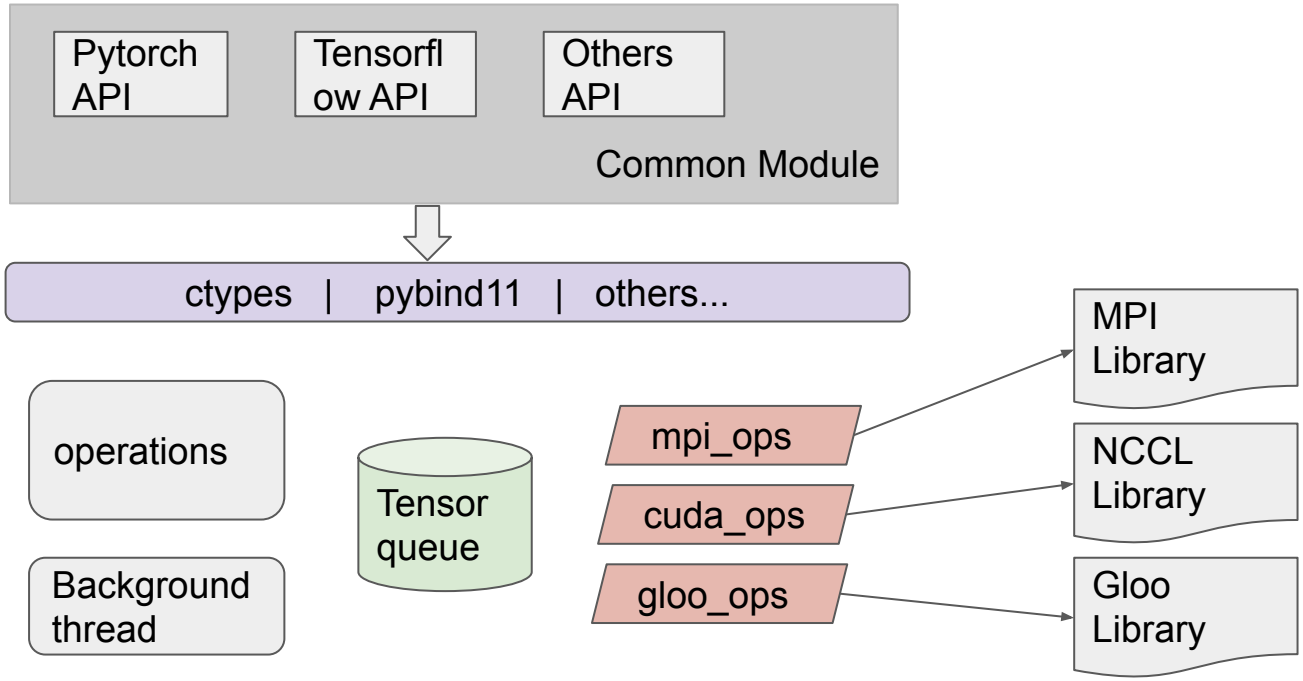
Operation
Send/Recv
Broadcast
Reduce

Comm Style
~~All~~
Neighbor
~~Point to Point~~

Fault Tolerance
~~Timeout~~
~~Backup~~
~~Self-healing?~~

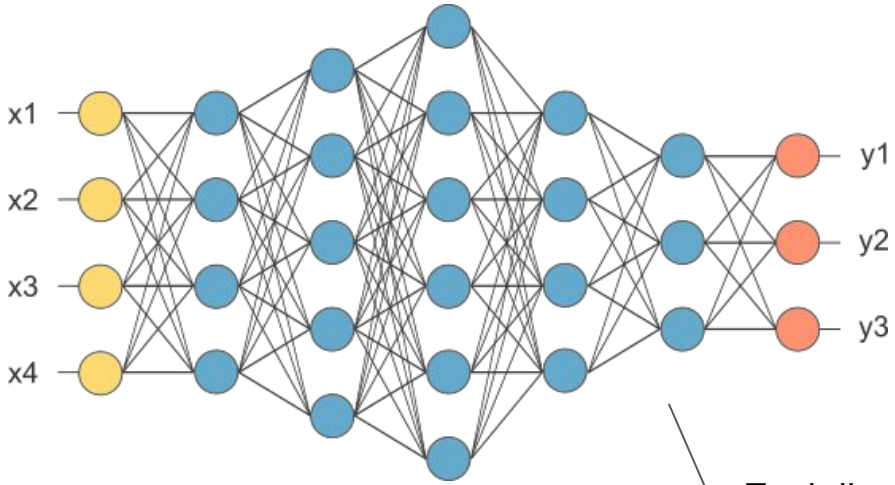
Code Design

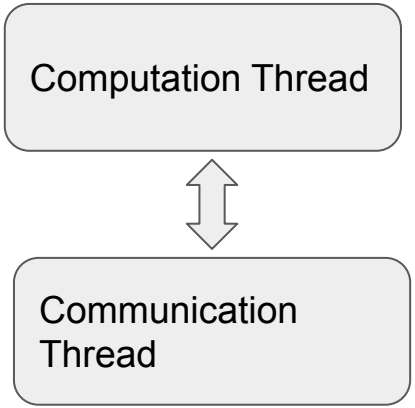
In order to manage and be able to easily extensibility for code, we have to decouple the logic well. A big picture of horovod code structure:



Horovod main idea

Paralleling communication with computation

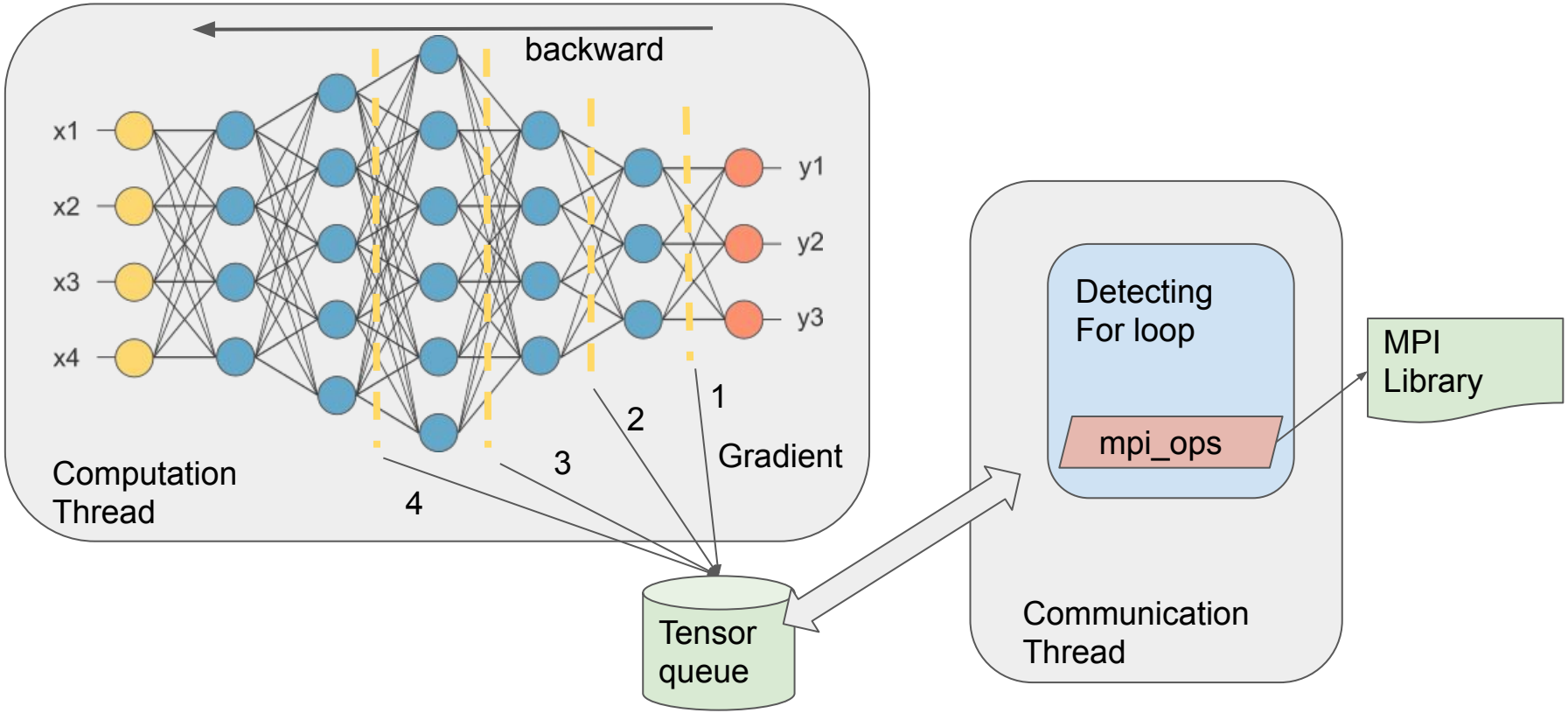




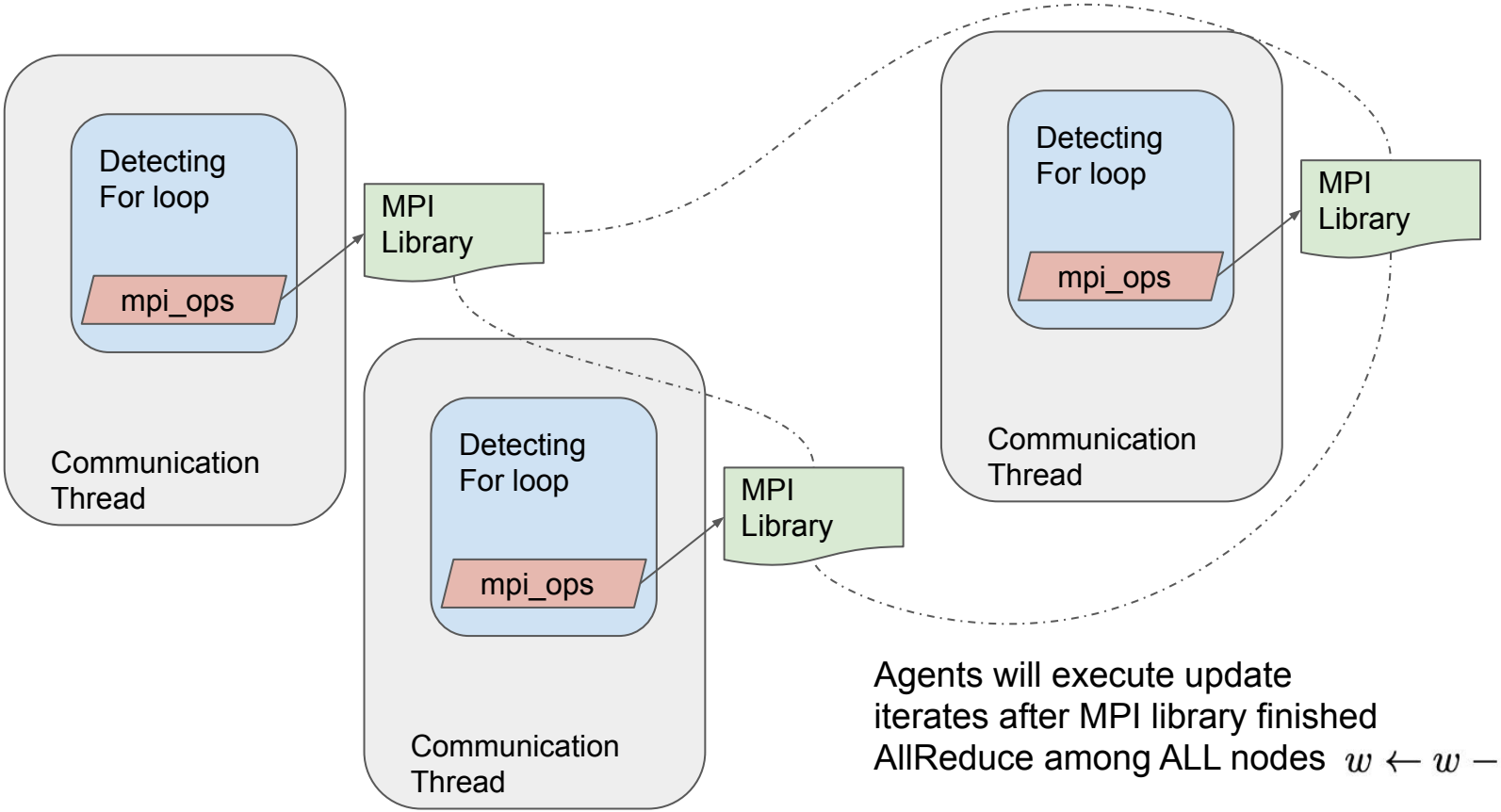
1. Forward ---- $y = f(w)$
2. Backward--- $\nabla f(w)$
3. Step ---- $w \leftarrow w - \mu \nabla f(w)$

Each line represents the parameters w_{i,m}^k in the forward pass and its corresponding gradient In the backward pass

Paralleling communication with computation



Ring All-Reduce and Synchronized Update



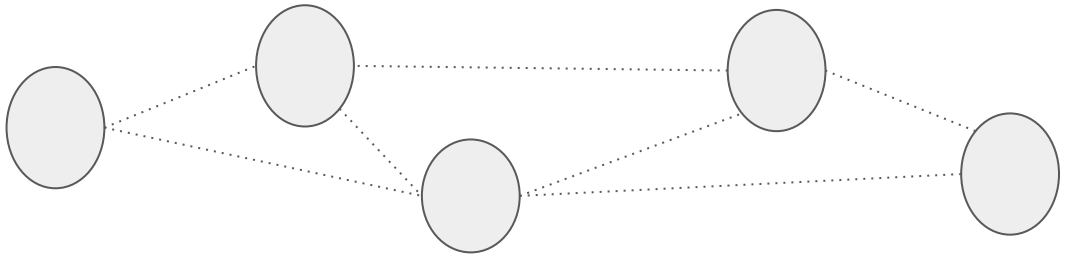
Agents will execute update
iterates after MPI library finished
AllReduce among ALL nodes $w \leftarrow w - \mu \nabla f(w)$

Bluefog Three Main Differences

1. Computing average on iterate w instead of gradient

$$w_{i,k} \leftarrow \sum_{\ell \in \mathcal{N}_k} a_{k\ell} w_{i-1,\ell}$$

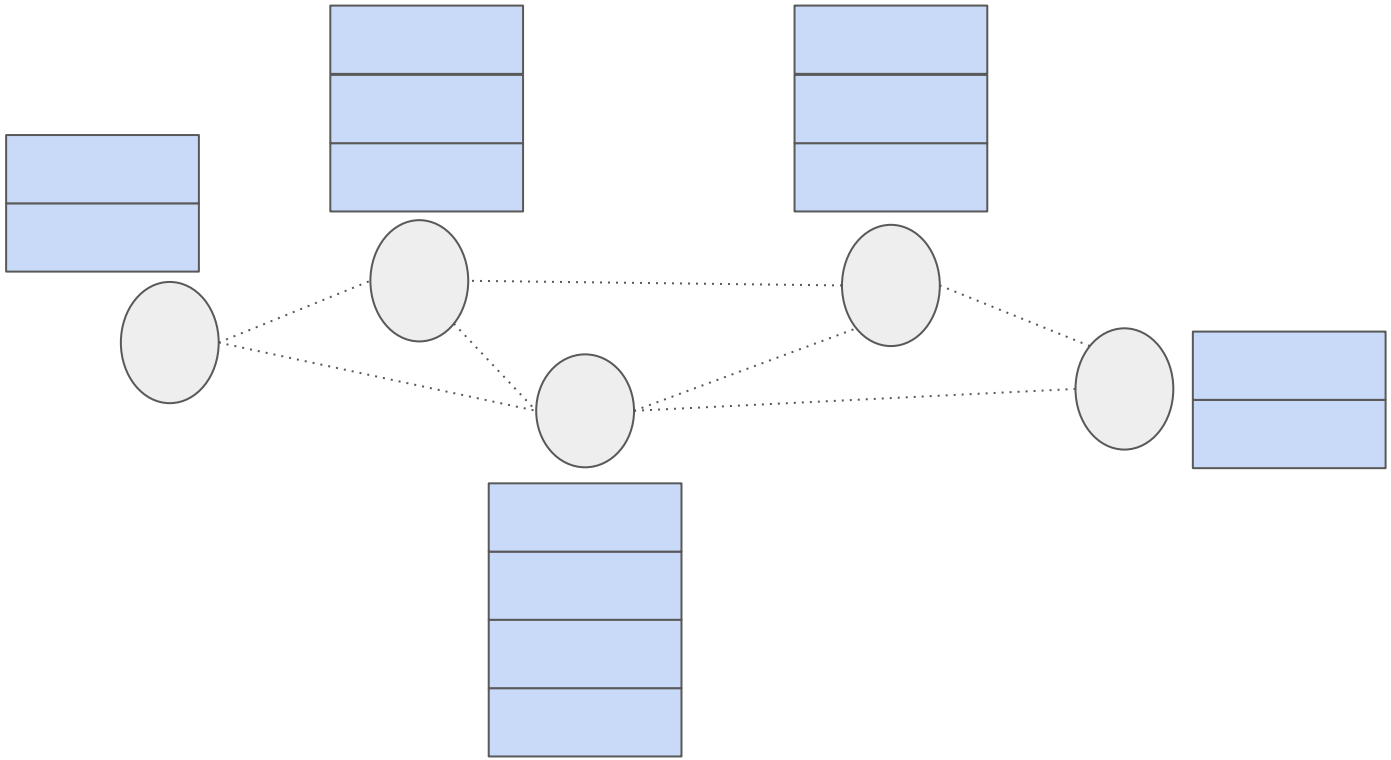
2. Local/neighbor communication (need topology)



file:///Users/ybc/Documents/dev/picpic-blue-fog/python/gossip_simulations/simulations_result/push_sum.html

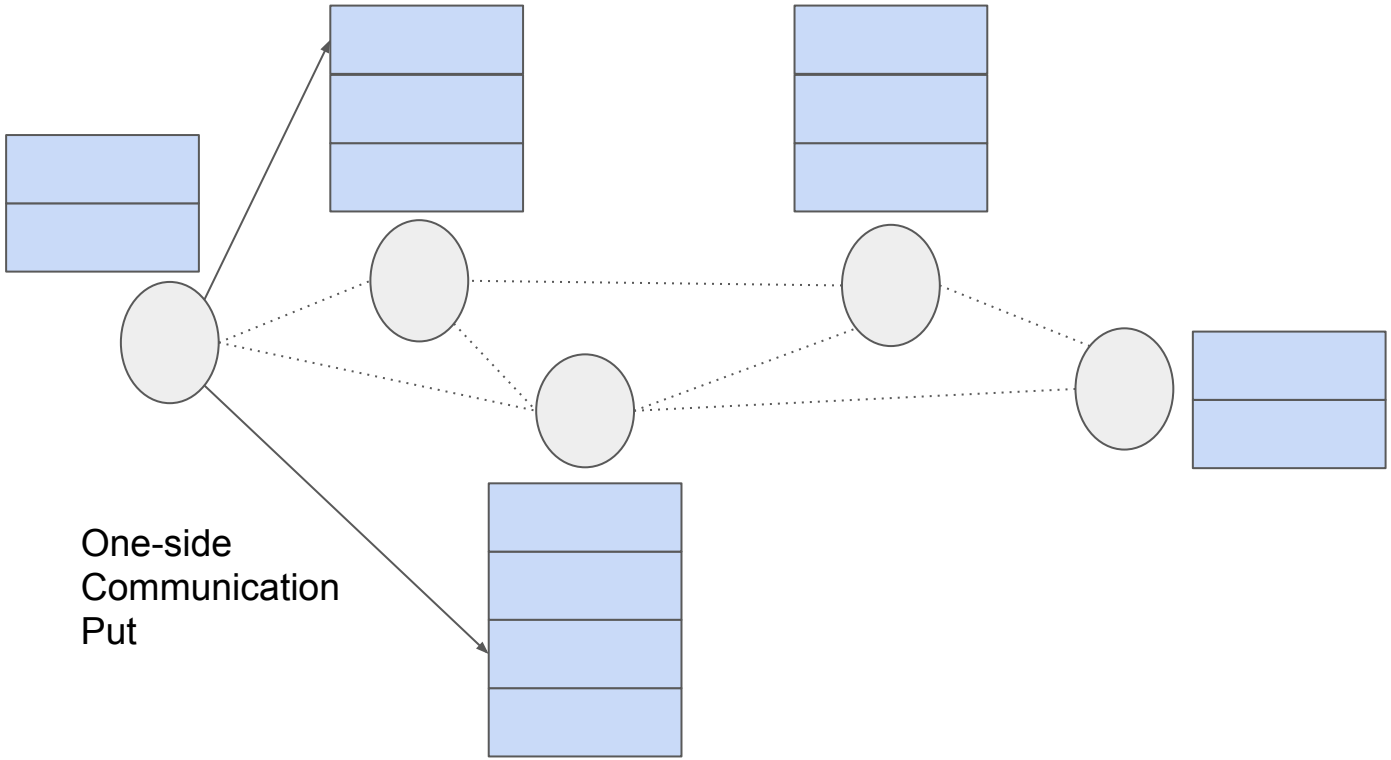
3. Asynchronous algorithm

Our asynchronization



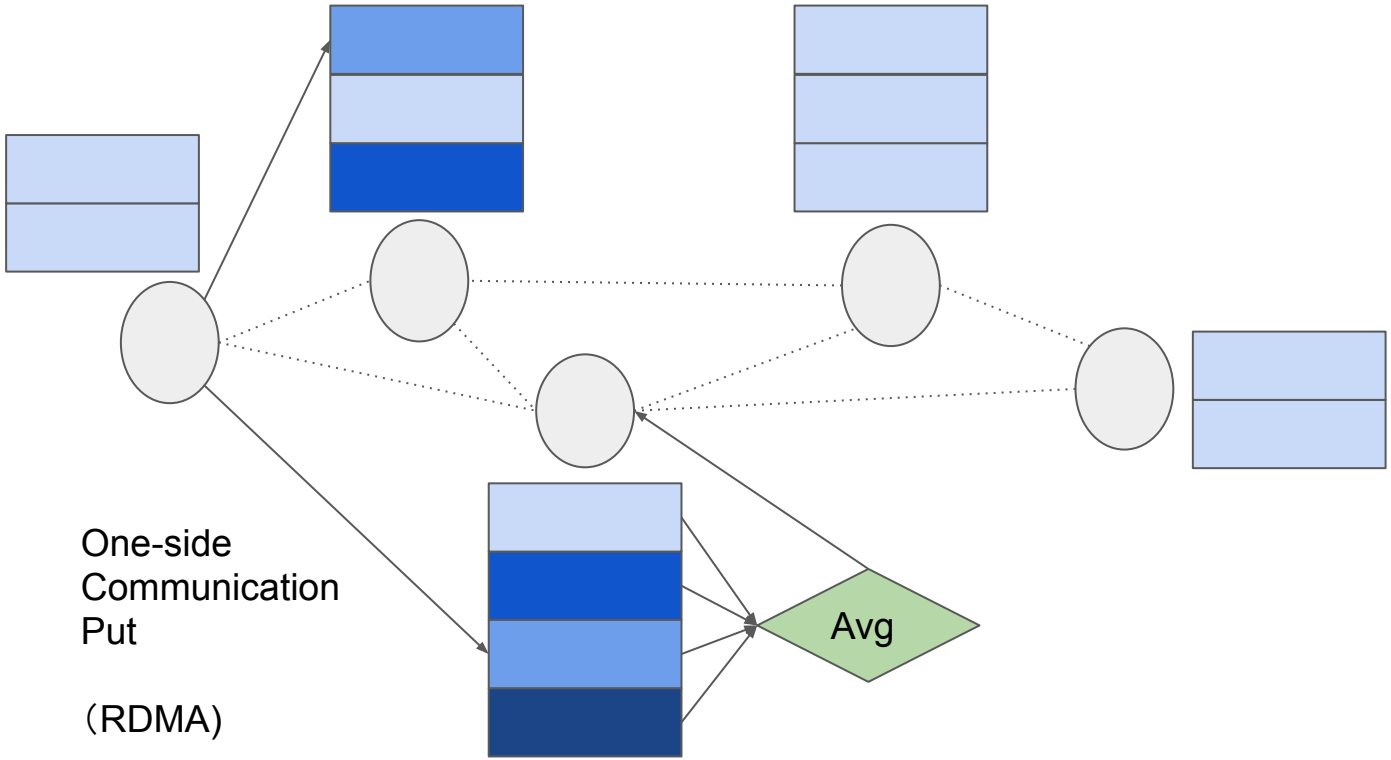
Extra Memory
Buffer for
Each neighbor

Our asynchronization



Extra Memory
Buffer for
Each neighbor

Our asynchronization



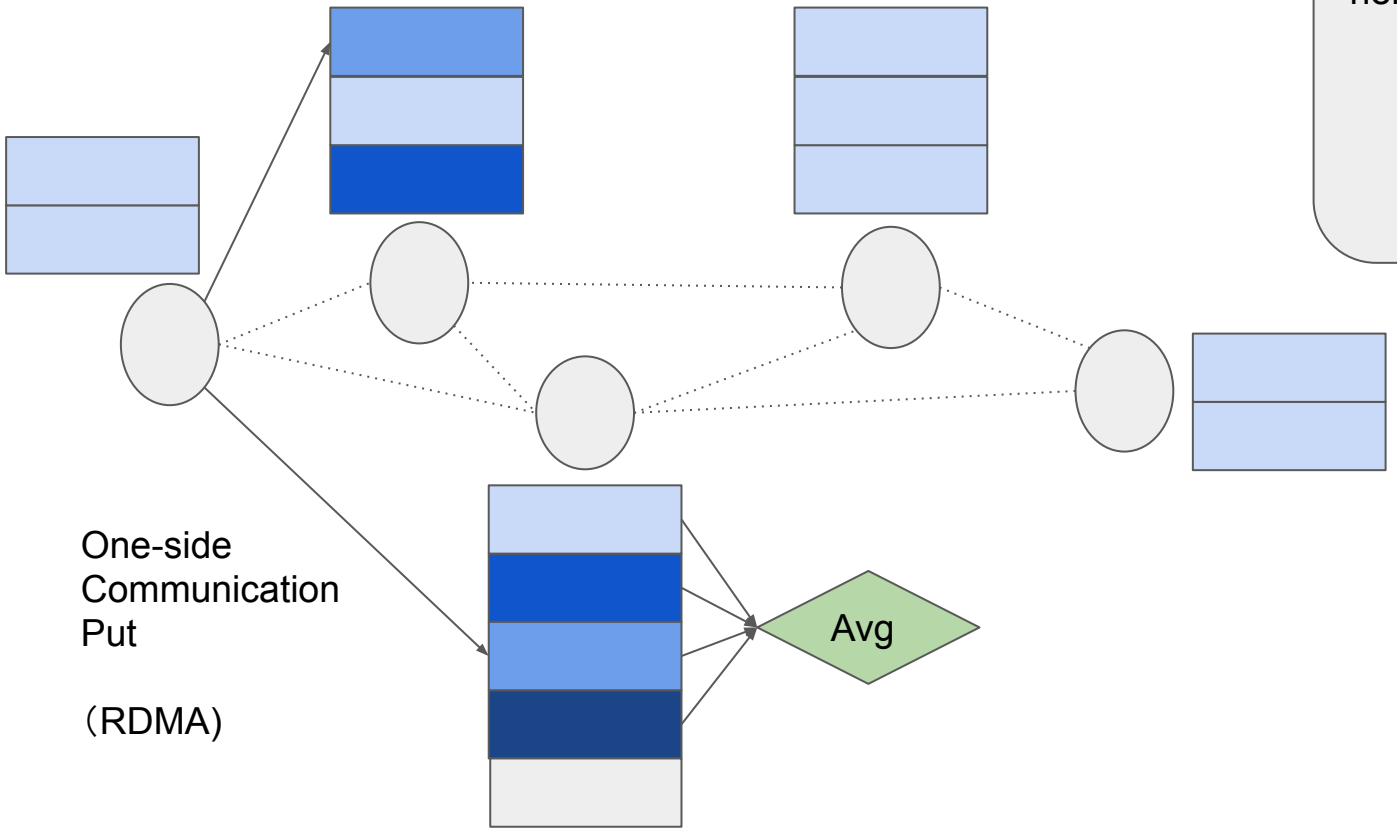
One-side
Communication
Put

（RDMA)

Extra Memory
Buffer for
Each neighbor

The neighbor information stored in the buffer may be out-of-date in different dealy

Our asynchronization



One thread for each neighbor

Extra Memory Buffer for Each neighbor

The neighbor information stored in the buffer may be out-of-date in different dealy