Lab 6-1
Distributed Training

Parallel Programming
2022/12/29

Outline

- Distributed ML Architecture
  - Parameter Server
  - Ring Allreduce
- Make Model Distributed
  - Tensorflow Distributed Strategies
  - Horovod
- Lab

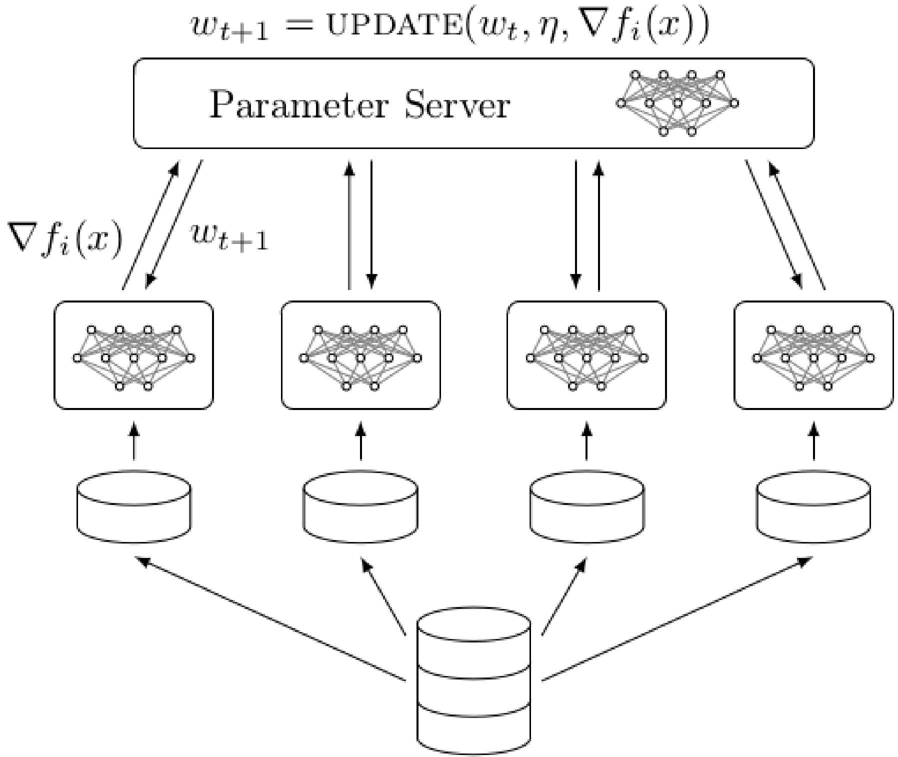
Why distributed training?

- Deeper / Larger model
- Larger dataset

Parameter Server

- Parameter server(s) hold the model parameters
- Worker pull parameters from Parameter server and perform local training
- Worker push training result to parameter server
- Data parallelism

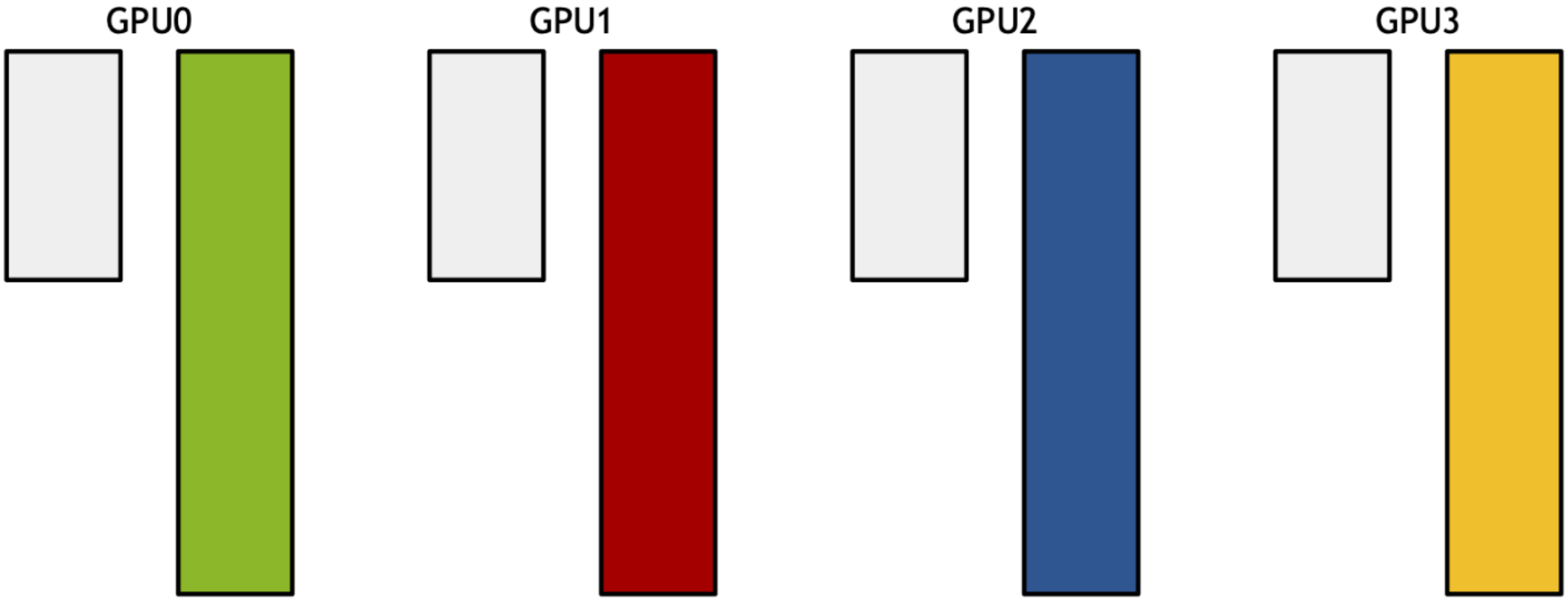
Parameter Server



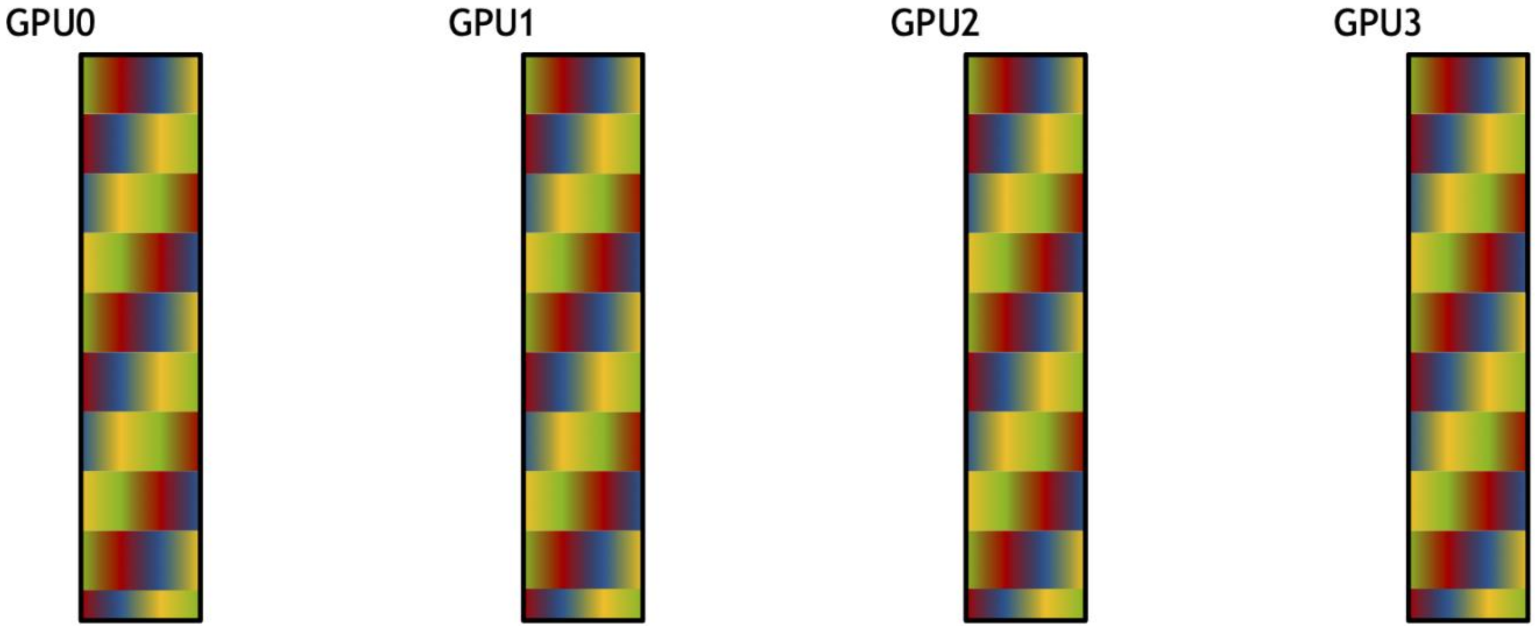
Ring-Allreduce

- 2-step allreduce
  - scatter-reduce
  - allgather
- Communication cost is independent to the number of GPUs, but bounded by the slowest connection
- Very efficient

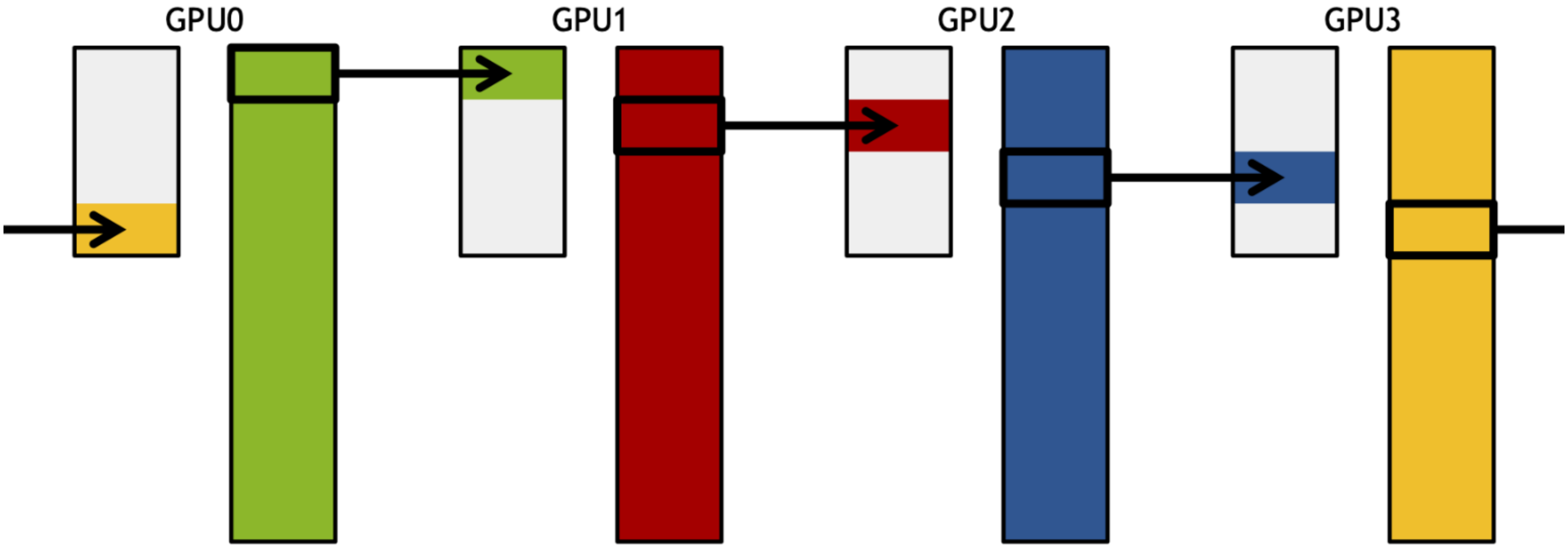
Ring-Allreduce



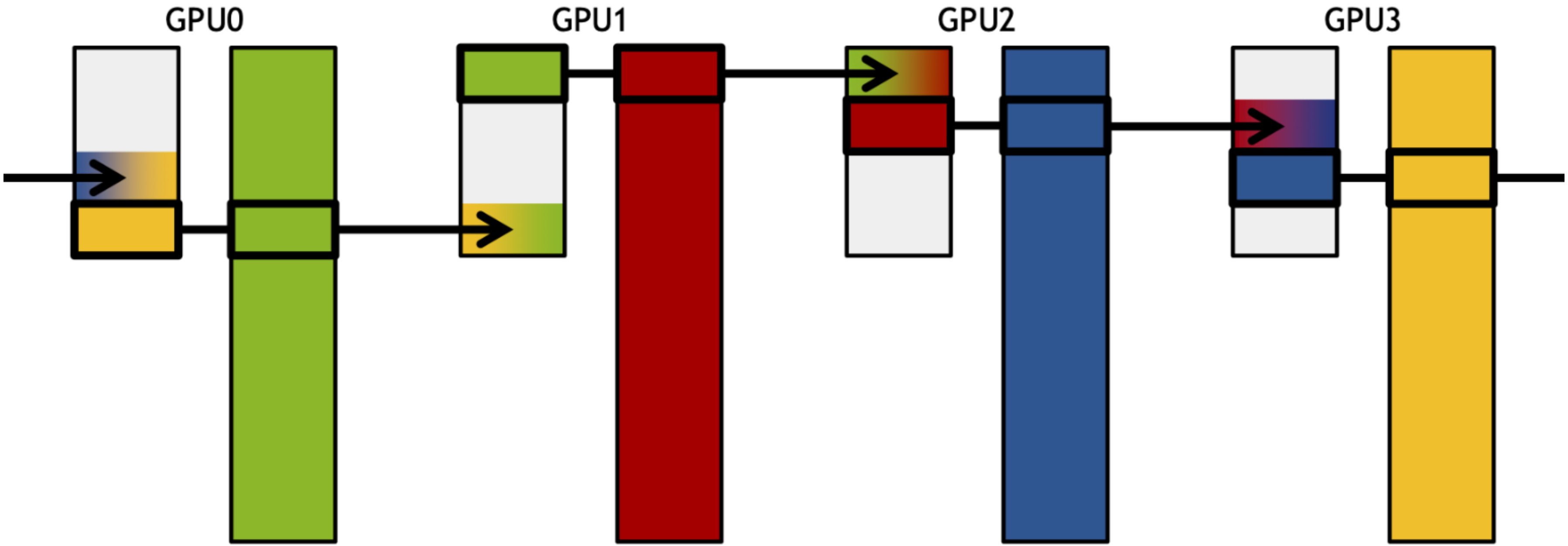
Ring-Allreduce



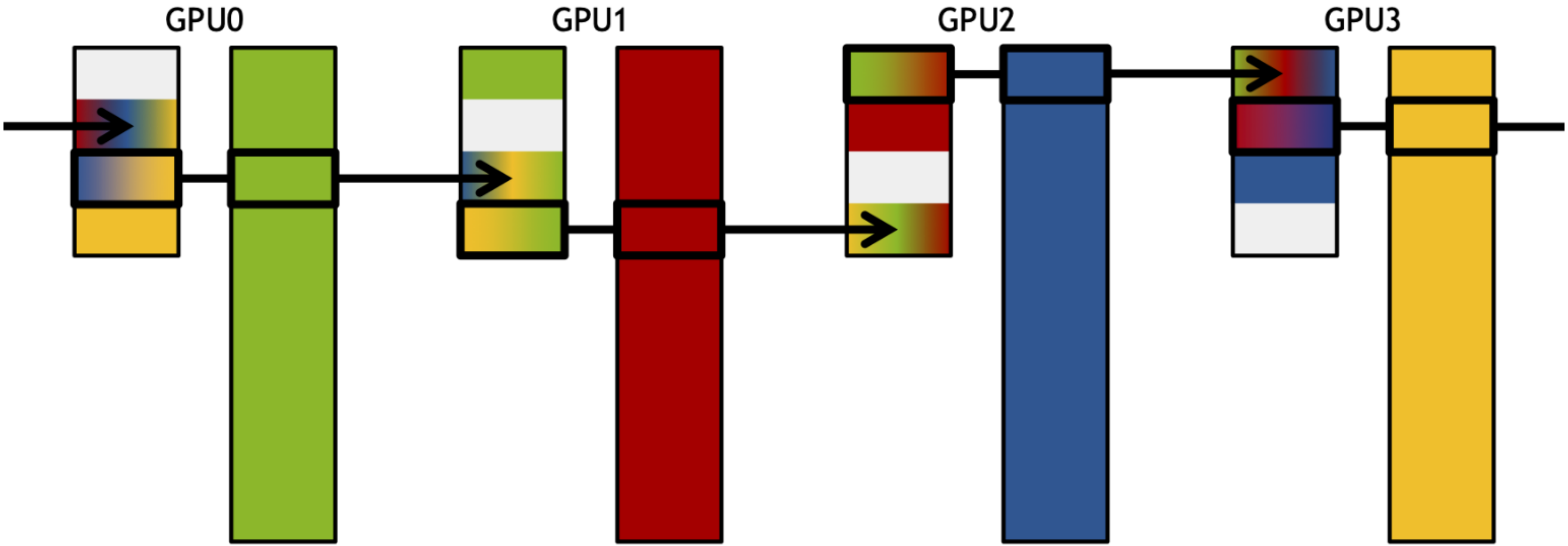
Ring-Allreduce



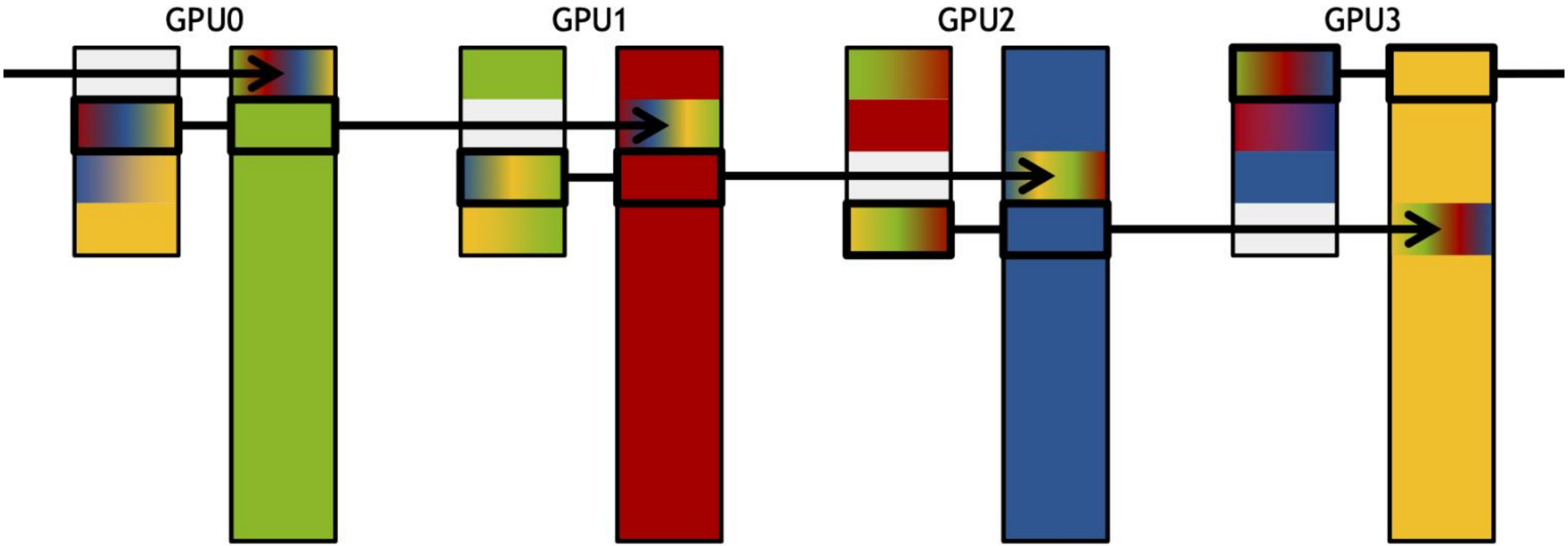
Ring-Allreduce



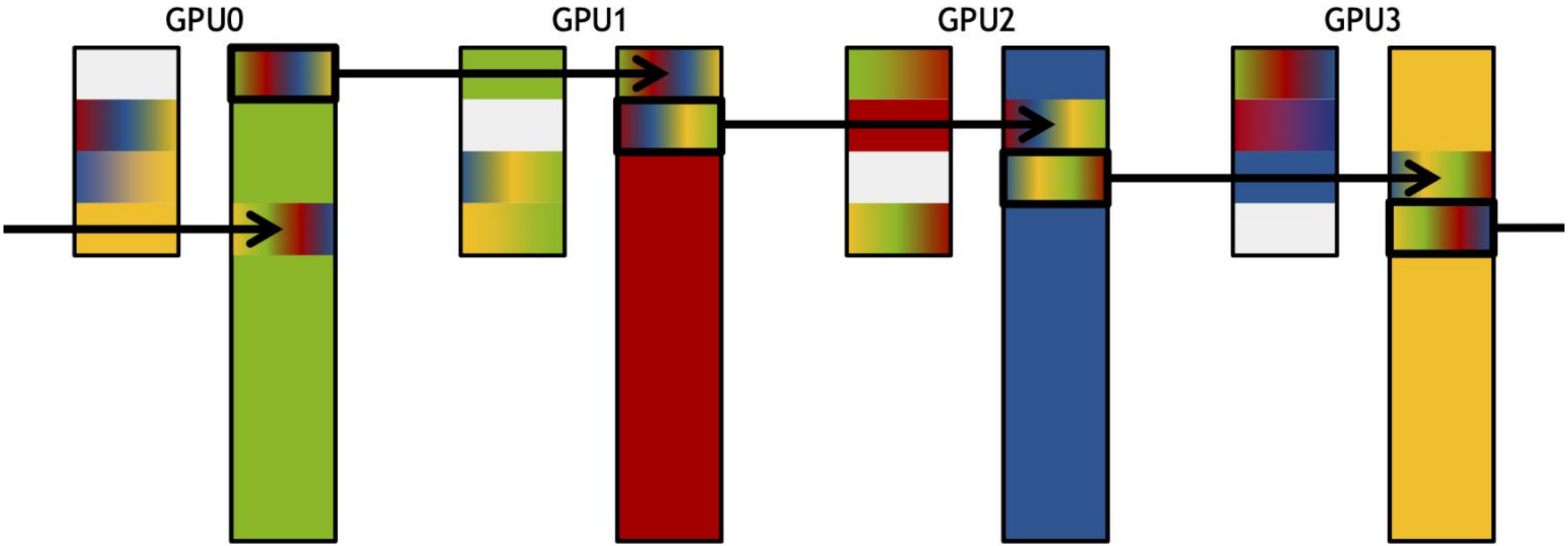
Ring-Allreduce



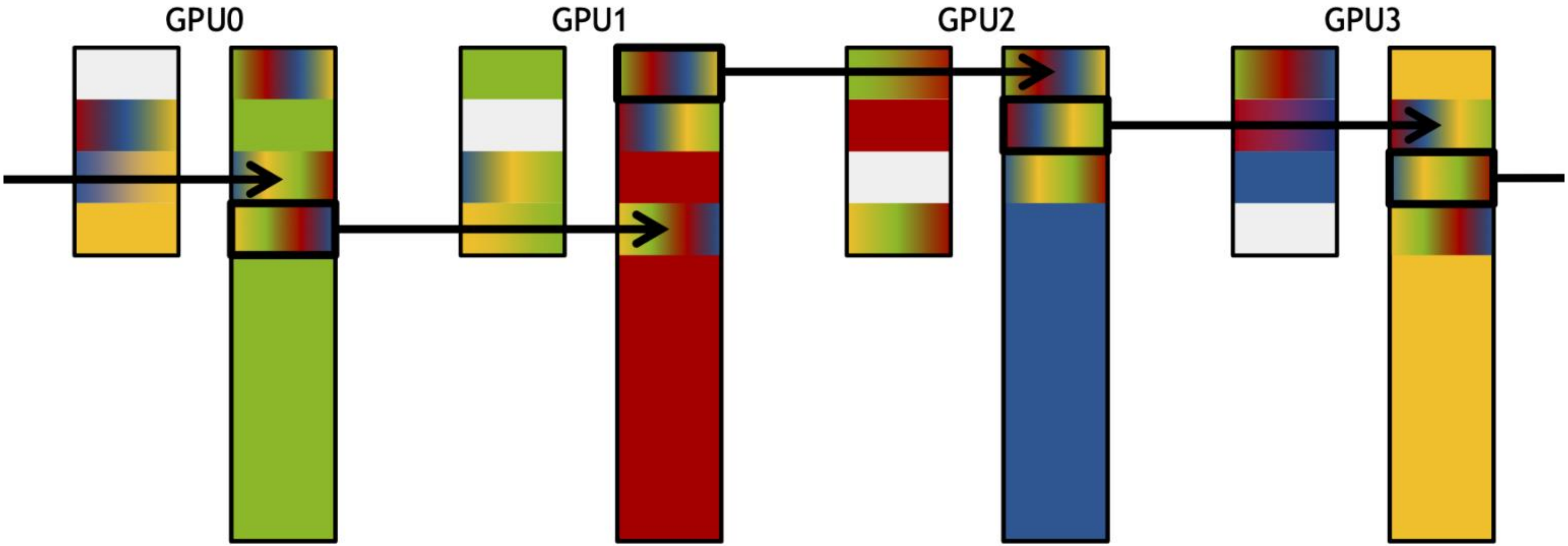
Ring-Allreduce



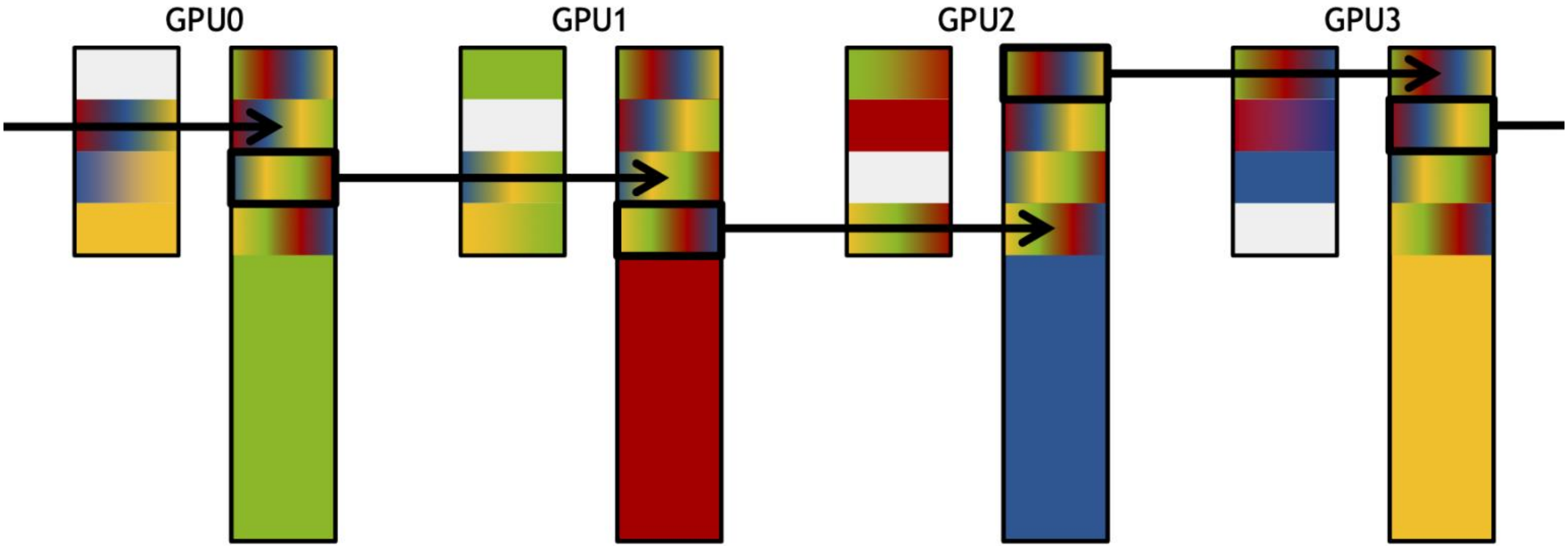
Ring-Allreduce



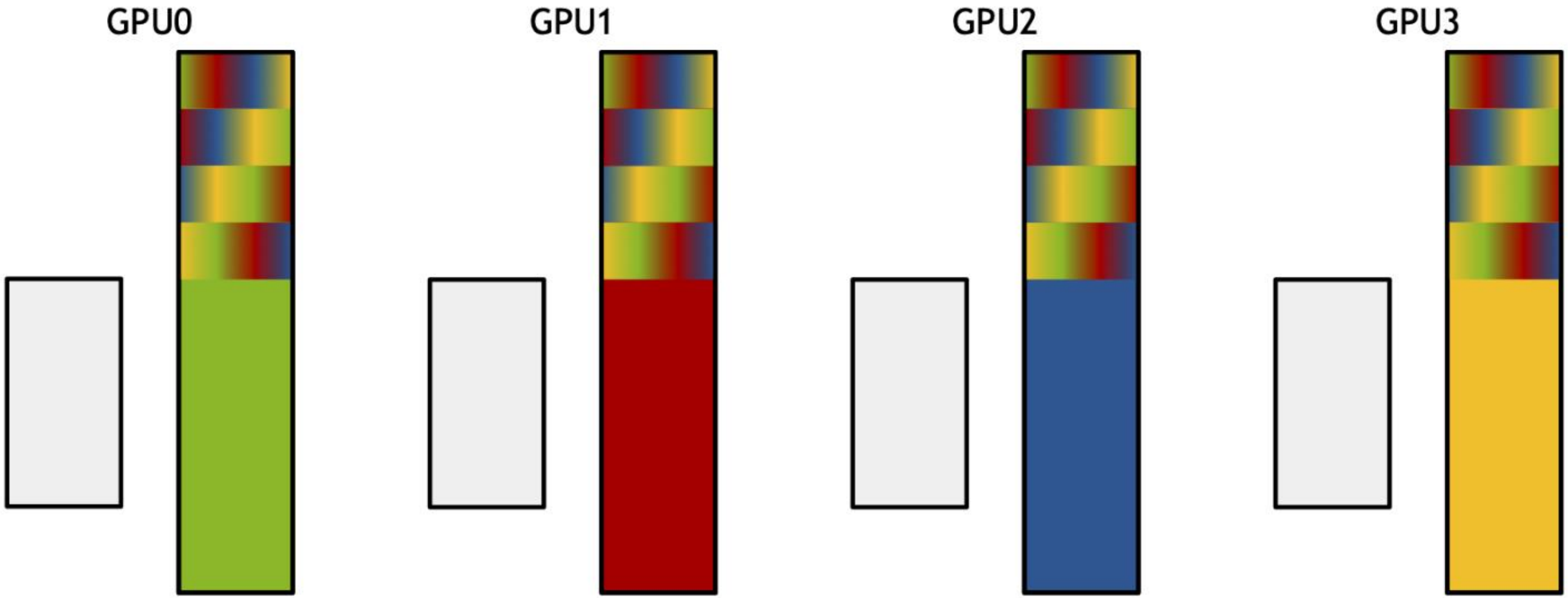
Ring-Allreduce



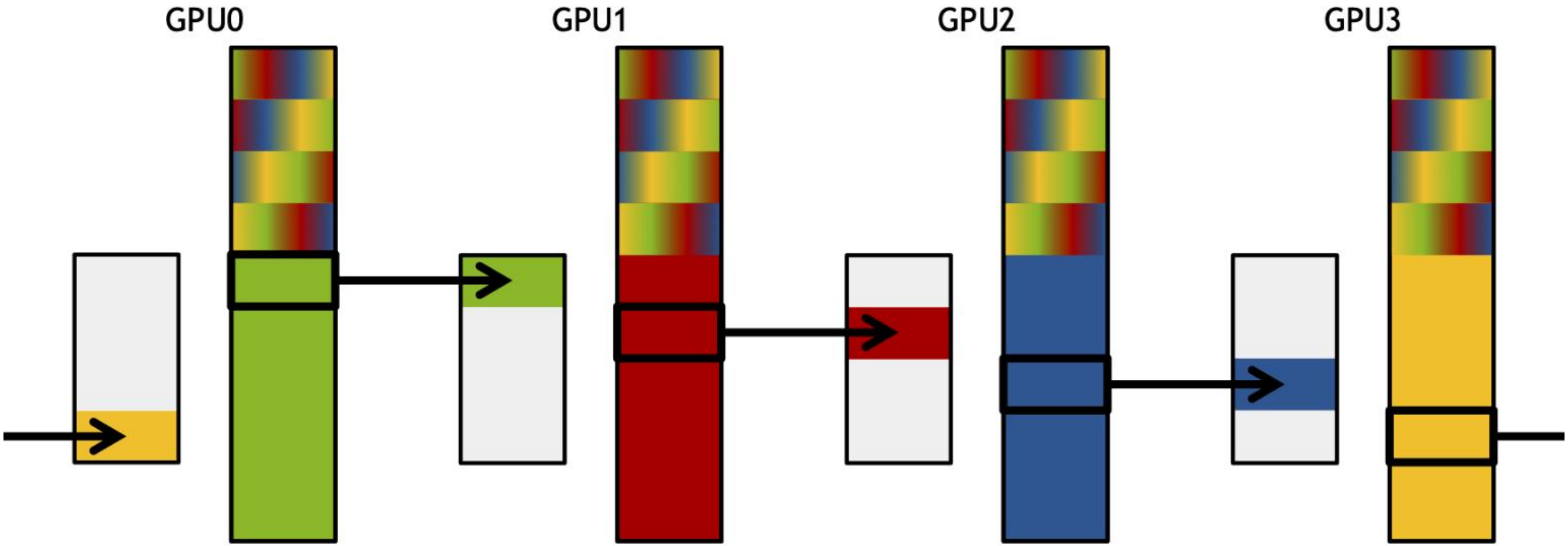
Ring-Allreduce



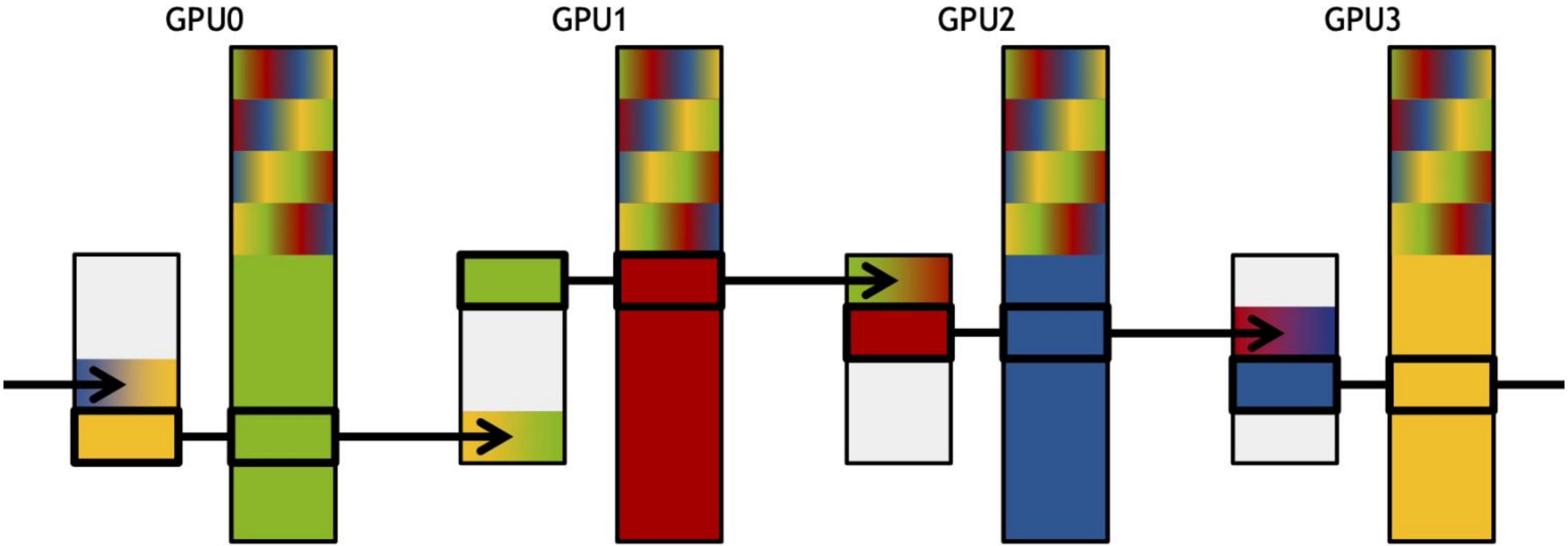
Ring-Allreduce



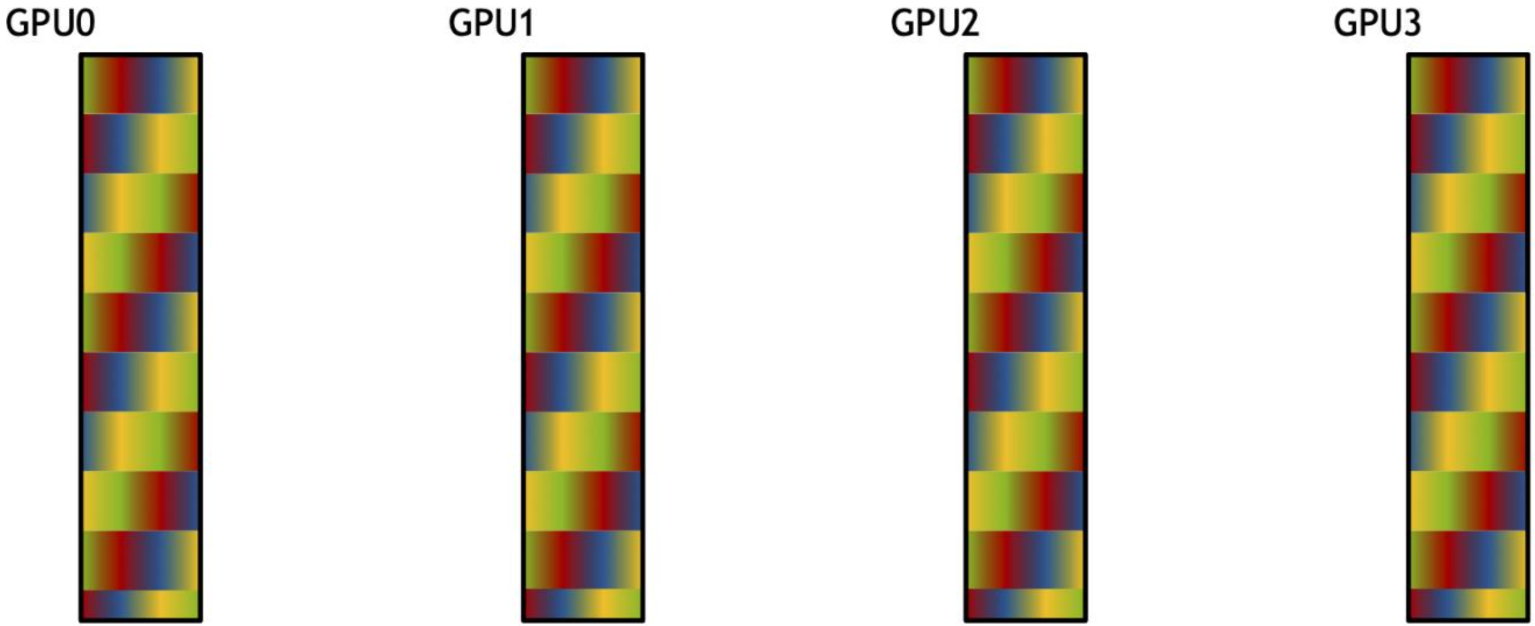
Ring-Allreduce



Ring-Allreduce



Ring-Allreduce



Tensorflow Distributed Strategies

- MirroredStrategy
- CentralStorageStrategy
- MultiWorkerMirroredStrategy
- ParameterServerStrategy

MirroredStrategy

- Single node
- With multiple GPUs or CPUs
- Create variable within strategy's scope

```
strategy = tf.distribute.MirroredStrategy(["GPU:0", "GPU:1"])
with strategy.scope():
   x = tf.Variable(1.)
```

- This will keep those variables synchronized across all devices.
- Using NCCL all-reduce by default.

CentralStorageStrategy

- Single node
- With single or multiple GPU.
- By default, CPU handles parameters

MultiWorkerMirroredStrategy

- Multi node
- With multiple GPUs or CPUs
- Need to config worker’s IP

```
TF_CONFIG = '{"cluster": {"worker": ["localhost:12345", "localhost:23456"]},
"task": {"type": "worker", "index": 0} }'
```

- Keep variables synchronized like MirroredStrategy.
- Two communication backend
  - CollectiveCommunication.RING: gRPC
  - CollectiveCommunication.NCCL: NCCL

ParameterServerStrategy

- Multi node with multiple GPUs or CPUs
- Create one or multiple parameter servers.
- Server distribute work to workers.

Horovod

- Implemented with OpenMPI & ring-allreduce
- Developed by Uber
- Support TensorFlow, Keras, PyTorch, MXNet
- Few code changes
- Easy to submit job on modern supercomputer

Horovod with Tensorflow2 - 1

- Initialize horovod `hvd.init()`
- Ping GPU to a single process

```
gpus = tf.config.experimental.list_physical_devices('GPU')
for gpu in gpus:
    tf.config.experimental.set_memory_growth(gpu, True)
if gpus:
    tf.config.experimental.set_visible_devices(gpus[hvd.local_rank()], 'GPU')
```

Horovod with Tensorflow2 - 2

- Scale training steps and learning rate
- We see more data in 1 mini-batch, so to speedup training, we have to use larger learning rate
- Horovod recommend N*lr, where N is the number of workers.

Horovod with Tensorflow2 - 3

- Using tf.GradientTape and wrap the tape in hvd.DistributedGradientTape

```
with tf.GradientTape() as tape:
    probs = mnist_model(images, training=True)
    loss_value = loss(labels, probs)
tape = hvd.DistributedGradientTape(tape)
```

- Broadcast the initial variable states from rank 0 to all other processes.
    - This is necessary to ensure consistent initialization of all workers when training is started with random weights or restored from a checkpoint. (broadcast the weights after first step)
    - For TensorFlow2, use hvd.broadcast_variables after models and optimizers have been initialized.

    ```
    hvd.broadcast_variables(variables, root_rank=0)
    ```

Lab - horovod

- Use apollo server
- `cp -r /home/pp22/share/lab6/horovod ~/`
- Search TODO to complete the code.
- `sbatch run.sh` to run the training task for horovod.
- **Submit screenshot of the output message to eeclass**.
    - Including number of processes, loss.
    - You can ignore the warning message.

```
#!/bin/bash
#SBATCH -p pp22
#SBATCH -N 1
#SBATCH -n 8
#SBATCH -o horovod.out.%j
```