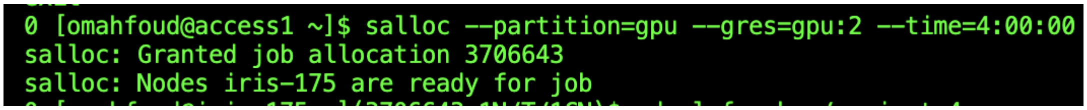
December 4, 2024

# Programming ML Algorithms for HPC - Homework 4

Othmane Mahfoud & Anton Zaitsev

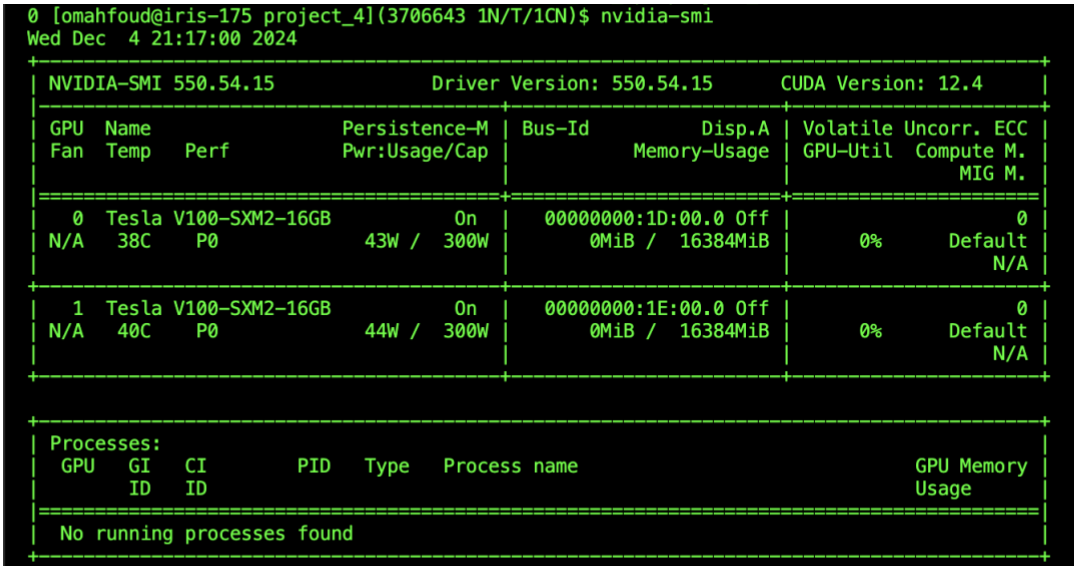
## 1. Server Setup

We start by connecting to the Iris HPC and requesting GPU resources



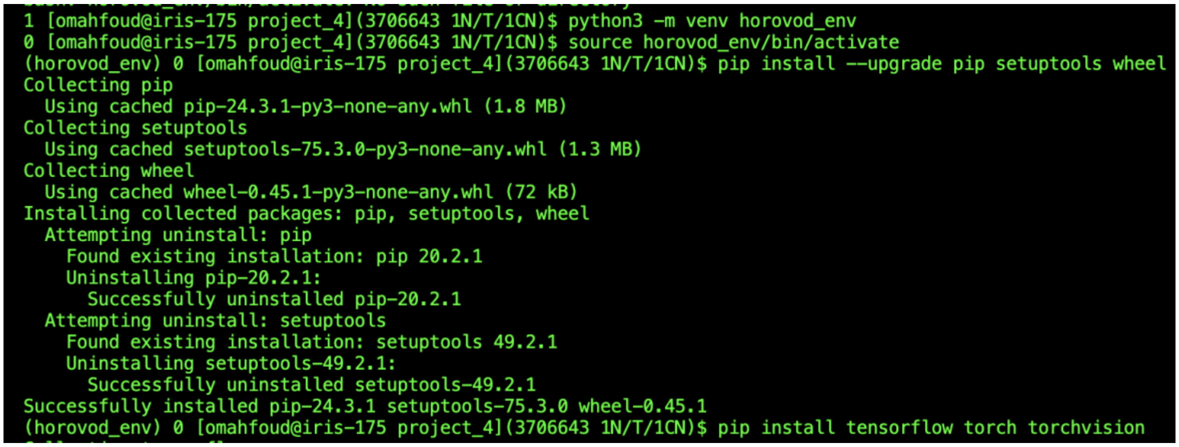
```
0 [omahfoud@access1 ~]$ salloc --partition=gpu --gres=gpu:2 --time=4:00:00
salloc: Granted job allocation 3706643
salloc: Nodes iris-175 are ready for job
```

We then check GPU access



```
0 [omahfoud@iris-175 project_4](3706643 1N/T/1CN)$ nvidia-smi
Wed Dec  4 21:17:00 2024
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.54.15              Driver Version: 550.54.15      CUDA Version: 12.4     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  Tesla V100-SXM2-16GB           On  |   00000000:1D:00.0 Off |                    0 |
| N/A   38C    P0             43W /  300W |       0MiB /  16384MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   1  Tesla V100-SXM2-16GB           On  |   00000000:1E:00.0 Off |                    0 |
| N/A   40C    P0             44W /  300W |       0MiB /  16384MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                              GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
```

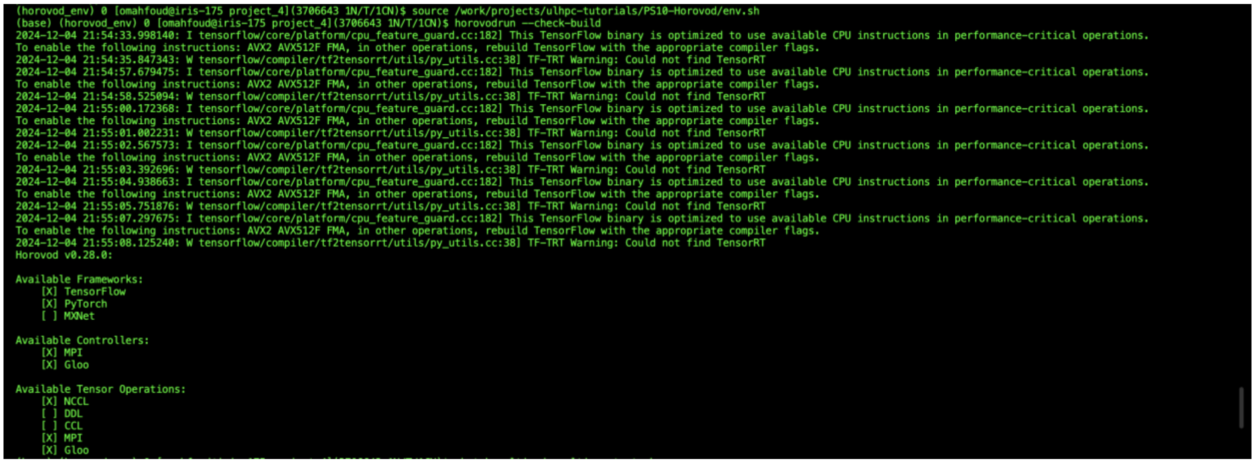
Then we create and activate a python virtual environment to isolate the installations for this project. We also need to update pip and install the required dependencies for horvord (tensorflow and pytorch)



```
1 [omahfoud@iris-175 project_4](3706643 1N/T/1CN)$ python3 -m venv horovod_env
0 [omahfoud@iris-175 project_4](3706643 1N/T/1CN)$ source horovod_env/bin/activate
(horovod_env) 0 [omahfoud@iris-175 project_4](3706643 1N/T/1CN)$ pip install --upgrade pip setuptools wheel
Collecting pip
  Using cached pip-24.3.1-py3-none-any.whl (1.8 MB)
Collecting setuptools
  Using cached setuptools-75.3.0-py3-none-any.whl (1.3 MB)
Collecting wheel
  Using cached wheel-0.45.1-py3-none-any.whl (72 kB)
Installing collected packages: pip, setuptools, wheel
  Attempting uninstall: pip
    Found existing installation: pip 20.2.1
    Uninstalling pip-20.2.1:
      Successfully uninstalled pip-20.2.1
  Attempting uninstall: setuptools
    Found existing installation: setuptools 49.2.1
    Uninstalling setuptools-49.2.1:
      Successfully uninstalled setuptools-49.2.1
Successfully installed pip-24.3.1 setuptools-75.3.0 wheel-0.45.1
(horovod_env) 0 [omahfoud@iris-175 project_4](3706643 1N/T/1CN)$ pip install tensorflow torch torchvision
```

## 2. Horvord Setup

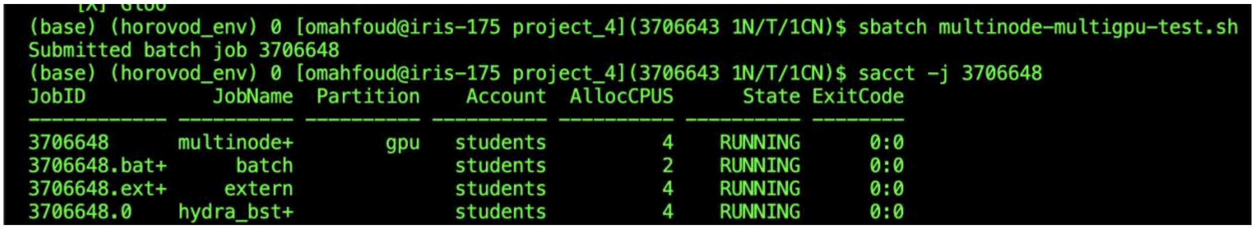
We first install Horvord as shown in the guide and check the installation, we obtain the expected output as shown below



```
(horovod_env) 0 [omahfoud@iris-175 project_4](3706643 1N/T/1CN)$ source /work/projects/ulhpc-tutorials/PS10-Horovod/env.sh
(base) (horovod_env) 0 [omahfoud@iris-175 project_4](3706643 1N/T/1CN)$ horovodrun --check-build
2024-12-04 21:54:33.998140: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 AVX512F FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2024-12-04 21:54:35.847343: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
2024-12-04 21:54:57.679475: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 AVX512F FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2024-12-04 21:54:58.525094: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
2024-12-04 21:55:00.172368: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 AVX512F FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2024-12-04 21:55:01.002231: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
2024-12-04 21:55:02.567573: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 AVX512F FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2024-12-04 21:55:03.392696: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
2024-12-04 21:55:04.938663: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 AVX512F FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2024-12-04 21:55:05.751876: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
2024-12-04 21:55:07.297675: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 AVX512F FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2024-12-04 21:55:08.125240: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
Horovod v0.28.0:

Available Frameworks:
    [X] TensorFlow
    [X] PyTorch
    [ ] MXNet

Available Controllers:
    [X] MPI
    [X] Gloo

Available Tensor Operations:
    [X] NCCL
    [ ] DDL
    [ ] CCL
    [X] MPI
    [X] Gloo
```

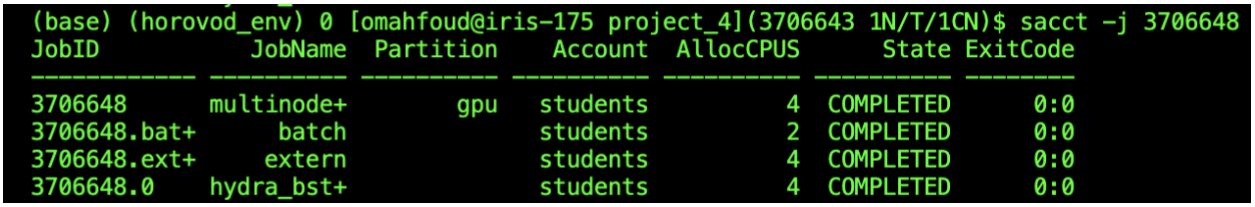
## 3. Testing multi-node multi-GPU Horovod

For testing large-scale training we launch test_horovod.py on 2 nodes, we then check if the job is successfully running using sacct



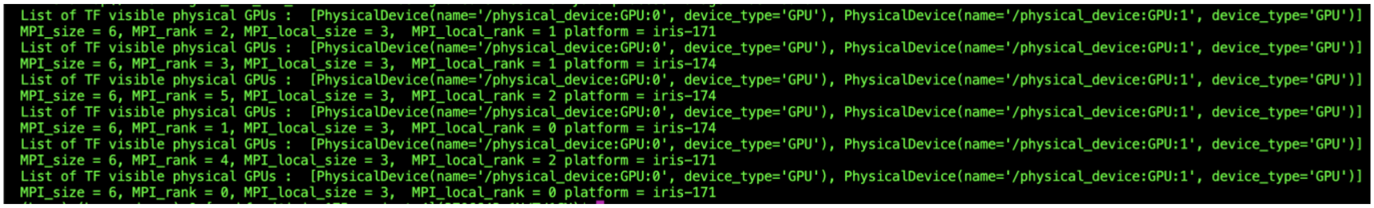
```
    [X] Gloo
(base) (horovod_env) 0 [omahfoud@iris-175 project_4](3706643 1N/T/1CN)$ sbatch multinode-multigpu-test.sh
Submitted batch job 3706648
(base) (horovod_env) 0 [omahfoud@iris-175 project_4](3706643 1N/T/1CN)$ sacct -j 3706648
JobID           JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
3706648      multinode+        gpu   students          4    RUNNING      0:0
3706648.bat+      batch              students          2    RUNNING      0:0
3706648.ext+     extern              students          4    RUNNING      0:0
3706648.0    hydra_bst+              students          4    RUNNING      0:0
```

And finally, when it finishes running



```
(base) (horovod_env) 0 [omahfoud@iris-175 project_4](3706643 1N/T/1CN)$ sacct -j 3706648
JobID           JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
3706648      multinode+        gpu   students          4  COMPLETED      0:0
3706648.bat+      batch              students          2  COMPLETED      0:0
3706648.ext+     extern              students          4  COMPLETED      0:0
3706648.0    hydra_bst+              students          4  COMPLETED      0:0
```

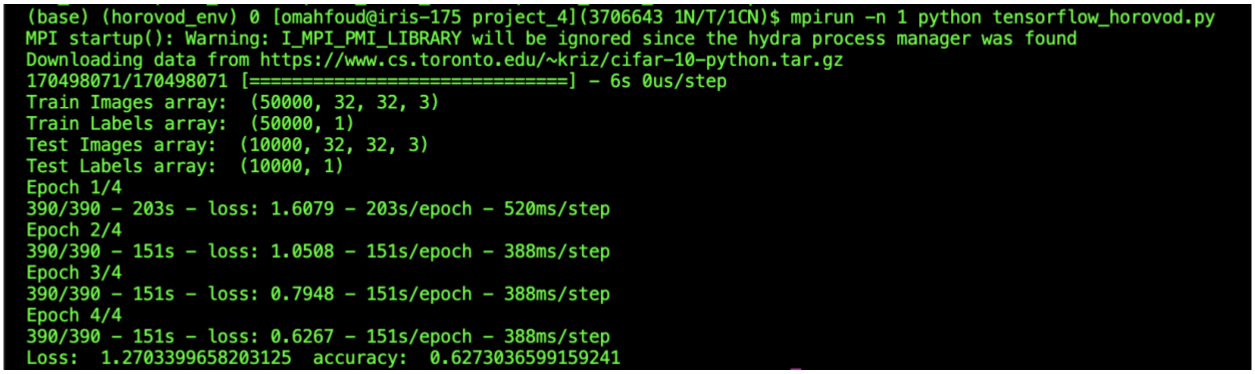
Now let’s check the output in the slurm output file



```
 List of TF visible physical GPUs :  [PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU'), PhysicalDevice(name='/physical_device:GPU:1', device_type='GPU')]
 MPI_size = 6, MPI_rank = 2, MPI_local_size = 3,  MPI_local_rank = 1 platform = iris-171
 List of TF visible physical GPUs :  [PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU'), PhysicalDevice(name='/physical_device:GPU:1', device_type='GPU')]
 MPI_size = 6, MPI_rank = 3, MPI_local_size = 3,  MPI_local_rank = 1 platform = iris-174
 List of TF visible physical GPUs :  [PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU'), PhysicalDevice(name='/physical_device:GPU:1', device_type='GPU')]
 MPI_size = 6, MPI_rank = 5, MPI_local_size = 3,  MPI_local_rank = 2 platform = iris-174
 List of TF visible physical GPUs :  [PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU'), PhysicalDevice(name='/physical_device:GPU:1', device_type='GPU')]
 MPI_size = 6, MPI_rank = 1, MPI_local_size = 3,  MPI_local_rank = 0 platform = iris-174
 List of TF visible physical GPUs :  [PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU'), PhysicalDevice(name='/physical_device:GPU:1', device_type='GPU')]
 MPI_size = 6, MPI_rank = 4, MPI_local_size = 3,  MPI_local_rank = 2 platform = iris-171
 List of TF visible physical GPUs :  [PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU'), PhysicalDevice(name='/physical_device:GPU:1', device_type='GPU')]
 MPI_size = 6, MPI_rank = 0, MPI_local_size = 3,  MPI_local_rank = 0 platform = iris-171
```

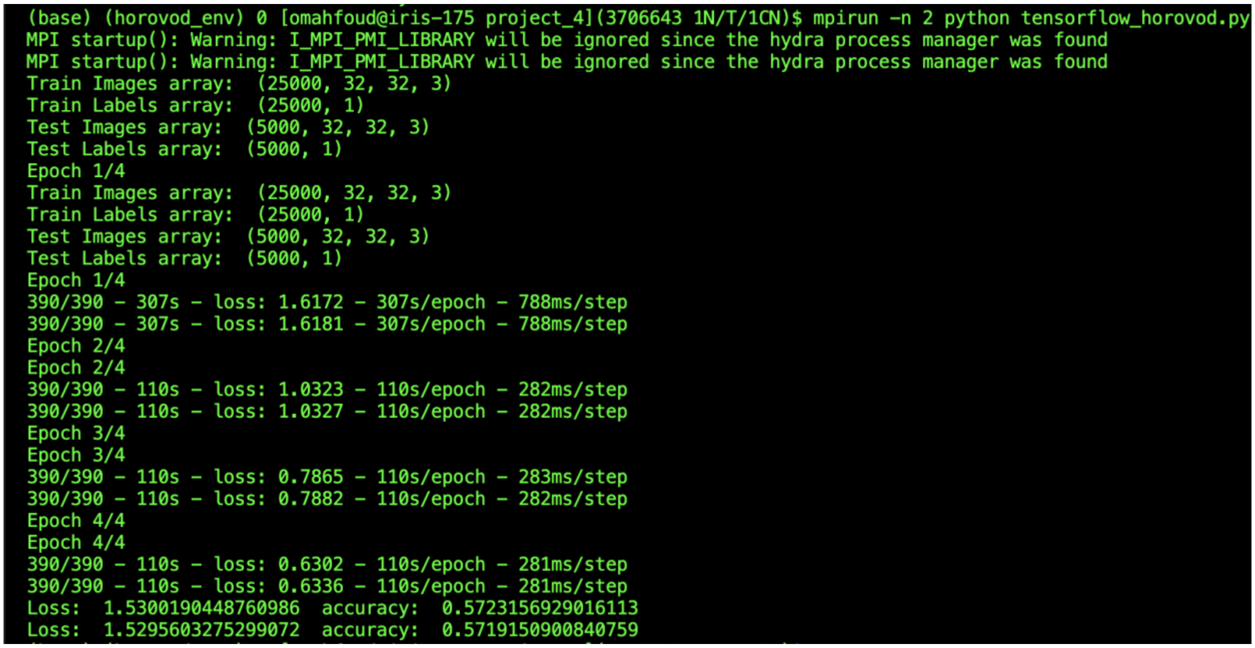
## 4. Tensorflow Examples

First, we run the Tensorflow code with 1 node



```
(base) (horovod_env) 0 [omahfoud@iris-175 project_4](3706643 1N/T/1CN)$ mpirun -n 1 python tensorflow_horovod.py
MPI startup(): Warning: I_MPI_PMI_LIBRARY will be ignored since the hydra process manager was found
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170498071/170498071 [==============================] - 6s 0us/step
Train Images array:  (50000, 32, 32, 3)
Train Labels array:  (50000, 1)
Test Images array:  (10000, 32, 32, 3)
Test Labels array:  (10000, 1)
Epoch 1/4
390/390 - 203s - loss: 1.6079 - 203s/epoch - 520ms/step
Epoch 2/4
390/390 - 151s - loss: 1.0508 - 151s/epoch - 388ms/step
Epoch 3/4
390/390 - 151s - loss: 0.7948 - 151s/epoch - 388ms/step
Epoch 4/4
390/390 - 151s - loss: 0.6267 - 151s/epoch - 388ms/step
Loss:  1.2703399658203125  accuracy:  0.6273036599159241
```

Then we run again with 2 nodes

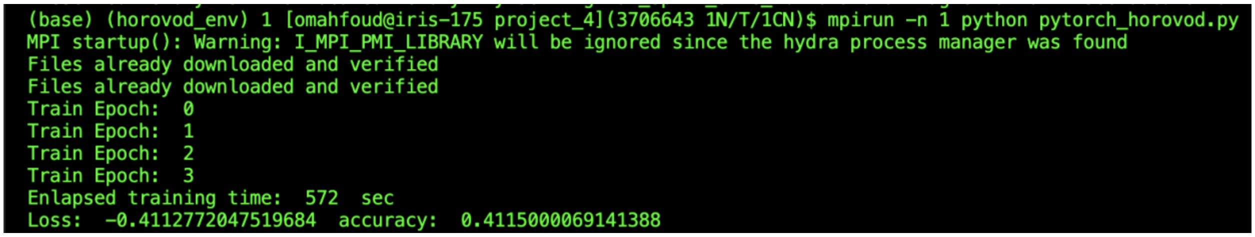


```
(base) (horovod_env) 0 [omahfoud@iris-175 project_4](3706643 1N/T/1CN)$ mpirun -n 2 python tensorflow_horovod.py
MPI startup(): Warning: I_MPI_PMI_LIBRARY will be ignored since the hydra process manager was found
MPI startup(): Warning: I_MPI_PMI_LIBRARY will be ignored since the hydra process manager was found
Train Images array:  (25000, 32, 32, 3)
Train Labels array:  (25000, 1)
Test Images array:  (5000, 32, 32, 3)
Test Labels array:  (5000, 1)
Epoch 1/4
Train Images array:  (25000, 32, 32, 3)
Train Labels array:  (25000, 1)
Test Images array:  (5000, 32, 32, 3)
Test Labels array:  (5000, 1)
Epoch 1/4
390/390 - 307s - loss: 1.6172 - 307s/epoch - 788ms/step
390/390 - 307s - loss: 1.6181 - 307s/epoch - 788ms/step
Epoch 2/4
Epoch 2/4
390/390 - 110s - loss: 1.0323 - 110s/epoch - 282ms/step
390/390 - 110s - loss: 1.0327 - 110s/epoch - 282ms/step
Epoch 3/4
Epoch 3/4
390/390 - 110s - loss: 0.7865 - 110s/epoch - 283ms/step
390/390 - 110s - loss: 0.7882 - 110s/epoch - 282ms/step
Epoch 4/4
Epoch 4/4
390/390 - 110s - loss: 0.6302 - 110s/epoch - 281ms/step
390/390 - 110s - loss: 0.6336 - 110s/epoch - 281ms/step
Loss:  1.5300190448760986  accuracy:  0.5723156929016113
Loss:  1.5295603275299072  accuracy:  0.5719150900840759
```

We notice indeed an improvement in training times using 2 nodes instead of 1.

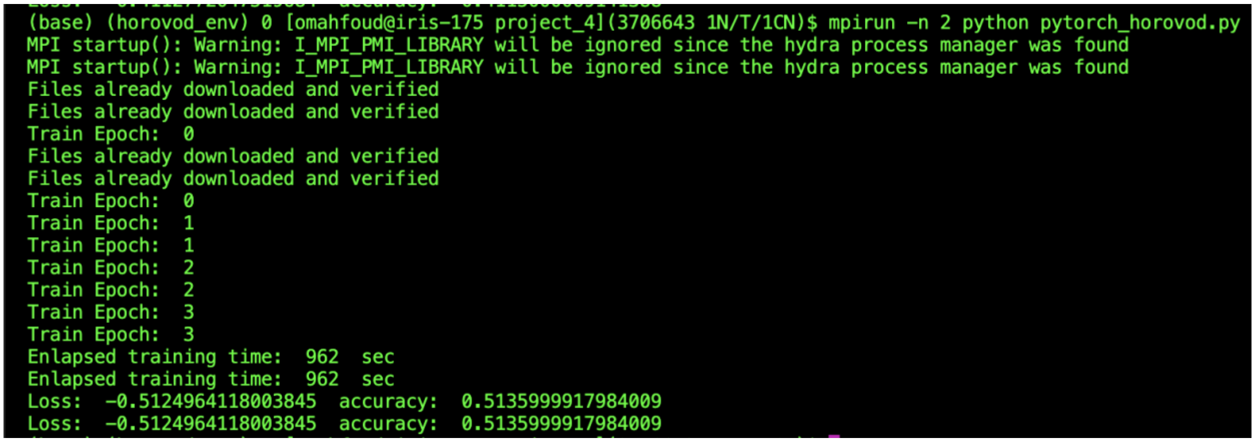
## 5. Pytorch Examples

Running PyTorch2 code with 1 GPU



```
(base) (horovod_env) 1 [omahfoud@iris-175 project_4](3706643 1N/T/1CN)$ mpirun -n 1 python pytorch_horovod.py
MPI startup(): Warning: I_MPI_PMI_LIBRARY will be ignored since the hydra process manager was found
Files already downloaded and verified
Files already downloaded and verified
Train Epoch:  0
Train Epoch:  1
Train Epoch:  2
Train Epoch:  3
Enlapsed training time:  572  sec
Loss:  -0.4112772047519684  accuracy:  0.4115000069141388
```

Running PyTorch2 code with 2 GPUs



```
(base) (horovod_env) 0 [omahfoud@iris-175 project_4](3706643 1N/T/1CN)$ mpirun -n 2 python pytorch_horovod.py
MPI startup(): Warning: I_MPI_PMI_LIBRARY will be ignored since the hydra process manager was found
MPI startup(): Warning: I_MPI_PMI_LIBRARY will be ignored since the hydra process manager was found
Files already downloaded and verified
Files already downloaded and verified
Train Epoch:  0
Files already downloaded and verified
Files already downloaded and verified
Train Epoch:  0
Train Epoch:  1
Train Epoch:  1
Train Epoch:  2
Train Epoch:  2
Train Epoch:  3
Train Epoch:  3
Enlapsed training time:  962  sec
Enlapsed training time:  962  sec
Loss:  -0.5124964118003845  accuracy:  0.5135999917984009
Loss:  -0.5124964118003845  accuracy:  0.5135999917984009
```

For some reason we don’t see improvements as seen for the TensorFlow version.