Testing GPUDirect RDMA on DGX1 Systems

Ahmed Shamsul Arefin, PhD
IT Advisor (HPC Systems)
schelp@csiro.au
CSIRO INFORMATION MANAGEMENT & TECHNOLOGY
www.csiro.au

ABSTRACT
In this work, we present a test deployment of a peer-to-peer remote direct memory access (RDMA) technology called GPUDirect introduced by NVIDIA. We deployed it on two DGX1 systems which demonstrates performance improvement on GPGPU accelerated HPC-based Machine Learning applications, such as TensorFlow.

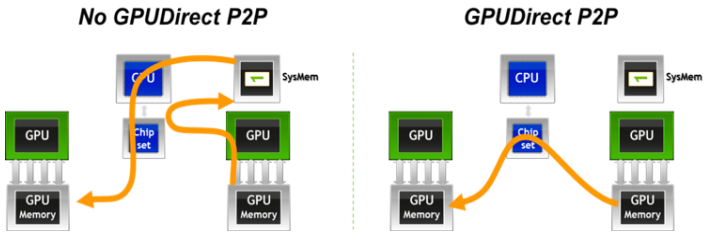
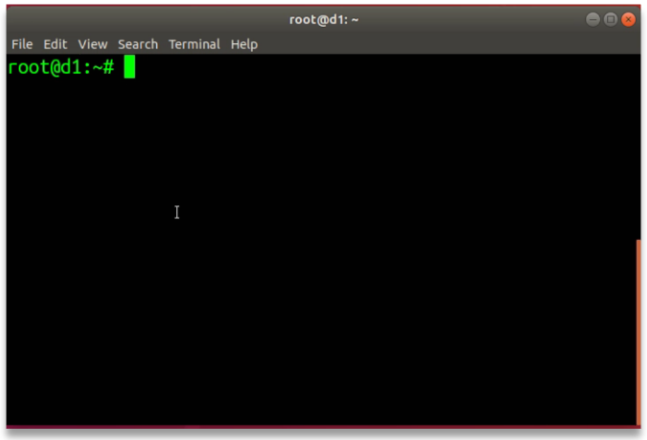


Figure 1: P2P GPUDirect Configurations (Image credit: VMWare)

We deployed GPUDirect on two test nodes: d1 and d2, with fresh installation of the Ubuntu 18.04. The setup was tested using OSU Benchmarks (P2P and Latency) and Tensorflow with resnet50 model with GPU and MPI enabled. The testing has been deployed as per the instructions in the MLNX GPUDirect User Manual.

The following five software packages were used and configured

- MLNX OFED 4.6.1
- NVIDIA CUDA 10.1
- NVIDIA GDRCopy
- Package NV_Peer_mem
- MVAPICH with GDR 2.3.1



OUTCOMES
We created a directory called "cloud" that is NFS shared between the both d1 and d2 DGX1 systems, where downloaded the OSU P2P, Latency benchmarks and the TensorFlow tests.

```
user@d1:~/cloud$ cat hosts
d1
d2
```

TENSORFLOW ON RESNET50

```
$ python tf_cnn_benchmarks.py --model=resnet50
Step    Img/sec total_loss
1       images/sec: 246.1 +/- 0.0 (jitter = 0.0)        8.220
10      images/sec: 246.9 +/- 0.1 (jitter = 0.3)        7.880
….
100     images/sec: 247.0 +/- 0.0 (jitter = 0.2)        8.035
----------------------------------------------------------------
total images/sec: 246.93
----------------------------------------------------------------
```

We enabled the Horovod to support the TensorFlow with MPI. We also exported the following environment variables to setup the MPI with CUDA and GDR support.

```
export MV2_USE_CUDA=1
export MV2_SMP_USE_CMA=0
export MV2_USE_GDRCOPY=1
export MV2_GPUDIRECT_GDRCOPY_LIB=/usr/local/gdrcopy/lib64/libgdrapi.so
export MV2_IBA_HCA=mlx5_0
export MV2_SUPPORT_TENSOR_FLOW=1

$ mpirun -np 2 -f hosts python tf_cnn_benchmarks.py --model=resnet50 --variable_update=horovod
Step Img/sec total_loss
1 images/sec: 236.3 +/- 0.0 (jitter = 0.0) 8.217
10 images/sec: 236.2 +/- 0.4 (jitter = 1.5) 7.877
….
100 images/sec: 235.4 +/- 0.3 (jitter = 1.1) 7.989
----------------------------------------------------------------
total images/sec: 470.77
----------------------------------------------------------------
```

The TensorFlow with MPI and GDR was much faster to execute and the total number of images processed was doubled per second.

CONCLUSION AND FUTURE WORKS
We demonstrated the installation procedure of the latest version of **GPUDirect technology**. It takes times to create a test environment and the relevant user manuals are often outdated. This work should ease the sysadmins who wish to deploy the latest GPUDirect on a GPU-enabled cluster system. We tested it on two DGX1 systems as a proof concept, but aim to deploy with in our production cluster and update the work.