PREREQUIED for Benchmarking Running on system:-

1. <https://us.download.nvidia.com/XFree86/Linux-x86_64/525.105.17/NVIDIA-Linux-x86_64-525.105.17.run>
2. <https://developer.nvidia.com/networking/hpc-x>
3. <https://developer.nvidia.com/cuda-toolkit-archive>
4. <https://developer.nvidia.com/rdp/cudnn-download>
5. <https://github.com/forresti/osu-micro-benchmarks.git>

Step1- NVIDIA DRIVER , CUDA-12.0 and Cudnn Installation

1. #cd /home/user/
2. #mkdir –p Package-Download
3. #cd Package-Download
4. <https://us.download.nvidia.com/XFree86/Linux-x86_64/525.105.17/NVIDIA-Linux-x86_64-525.105.17.run>
5. https://developer.download.nvidia.com/compute/cuda/12.0.0/local\_installers/cuda\_12.0.0\_525.60.13\_linux.run
6. https://developer.nvidia.com/downloads/compute/cudnn/secure/8.9.0/local\_installers/12.x/cudnn-linux-x86\_64-8.9.0.131\_cuda12-archive.tar.xz
7. #chmod +x NVIDIA-Linux-x86\_64-525.105.17.run
8. # chmod +x cuda\_12.0.0\_525.60.13\_linux.run
9. #tar –xvf cudnn-linux-x86\_64-8.9.0.131\_cuda12-archive.tar.xz
10. #vim /etc/modprobe.d/blacklist-nouveau.conf

blacklist nouveau

options nouveau modeset=0

save & exit

1. # dracut –f
2. #reboot
3. # telinit 3
4. #./ NVIDIA-Linux-x86\_64-525.105.17.run ## after installation of nvidia driver
5. #vim /etc/profile.d/cuda-12.0.sh

export PATH=/usr/local/cuda-12.0/bin${PATH:+:${PATH}}

export LD\_LIBRARY\_PATH=/usr/local/cuda-12.0/lib64${LD\_LIBRARY\_PATH:+: ${LD\_LIBRARY\_PATH}}

save & exit

1. #./ cuda\_12.0.0\_525.60.13\_linux.run # ./cuda\_11.7.0\_515.43.04\_linux.run --toolkitpath=/home/abcd/cuda\_test1
2. #source /etc/profile.d/cuda.sh
3. #nvcc –version ## check the cuda version
4. #nvidia-smi ## check the nvidia version
5. # cp cudnn-\*-archive/include/cudnn\*.h /usr/local/cuda-12.0/include
6. # cp -P cudnn-\*-archive/lib/libcudnn\* /usr/local/cuda-12.0/lib64
7. # chmod a+r /usr/local/cuda/include/cudnn\*.h /usr/local/cuda-12.0/lib64/libcudnn\*
8. # yum install libcudnn8-devel-8.x.x.x-1.cudaX.Y
9. #yum install libcudnn8-samples-8.x.x.x-1.cudaX.Y
10. #cp -r /usr/src/cudnn\_samples\_v8/ $HOME
11. #cd $HOME/cudnn\_samples\_v8/mnistCUDNN
12. #make clean && make
13. #./mnistCUDNN ##If cuDNN is properly installed and running on your Linux system, you will see a message similar to the following:

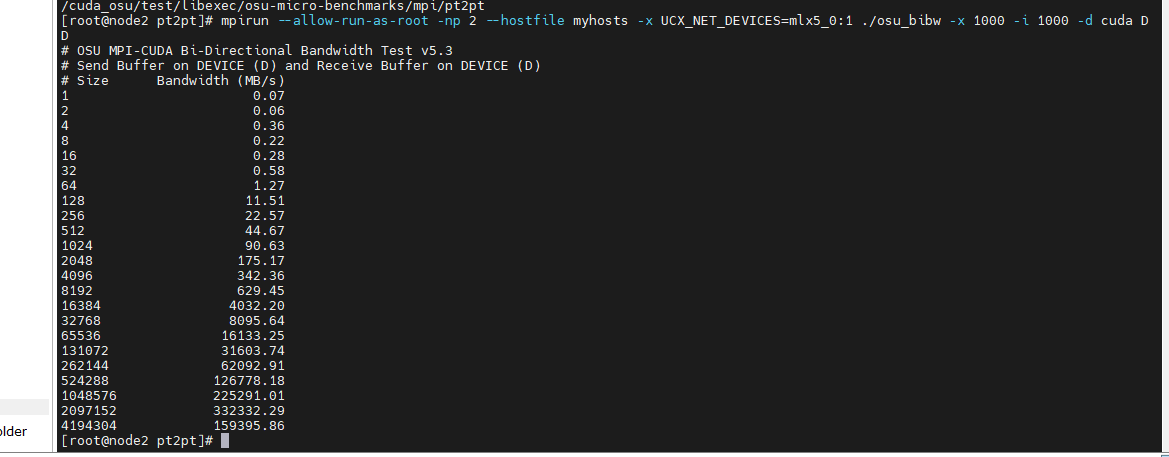
Test passed!

Step 2:- HPC-X and OSU Installation with cuda step

1. <http://www.mellanox.com/page/hpcx_eula?mrequest=downloads&mtype=hpc&mver=hpc-x&mname=v2.14/hpcx-v2.14-gcc-MLNX_OFED_LINUX-5-redhat7-cuda12-gdrcopy2-nccl2.17-x86_64.tbz>
2. #cd /home/user/Package-Download/ hpcx-v2.14-gcc-MLNX\_OFED\_LINUX-5-redhat7-cuda12-gdrcopy2-nccl2.17-x86\_64/
3. # export HPCX\_HOME=$PWD
4. # source $HPCX\_HOME/hpcx-init.sh
5. # hpcx\_load
6. # env | grep HPCX
7. #which mpicc

/home/user/Package-Download/hpcx-v2.14-gcc-MLNX\_OFED\_LINUX-5-redhat7-cuda12-gdrcopy2-nccl2.17-x86\_64/ompi/bin/mpicc

1. #git clone <https://github.com/forresti/osu-micro-benchmarks.git>
2. # cd osu-micro-benchmarks
3. #yum install numactl-devel.x86\_64 numactl-libs.i686 numactl-devel.i686
4. #yum install libcudaart
5. # autoreconf -f -i
6. # ./configure CC=mpicc CXX=mpicxx --enable-cuda --with-cuda=/usr/local/cuda-12.0 --prefix=/cuda\_osu/test
7. For C #./configure CC=mpicc CXX=mpiccxx CFLAGS=-I/opt/osu-micro-benchmarks-7.1-1/c/util/ --prefix=/app/osu-7.1
8. #./configure CC=/opt/tt/hpcx-v2.14-gcc-MLNX\_OFED\_LINUX-5-redhat7-cuda12-gdrcopy2-nccl2.17-x86\_64/ompi/bin/mpicc --prefix=/cudaosutest1/osu --enable-cuda=/usr/local/cuda-12.0
9. # make -j32
10. # make –j32 install
11. #cd /cudaosutest1/osu
12. #mpirun --allow-run-as-root -np 2 -x UCX\_NET\_DEVICES=ib0 ./osu\_bibw -x 1000 -i 1000
13. # cd /cuda\_osu/test/
14. # cd libexec/
15. #cd /cuda\_osu/test/libexec/osu-micro-benchmarks/mpi/pt2pt
16. # mpirun --allow-run-as-root -np 2 --hostfile myhosts -x UCX\_NET\_DEVICES=mlx5\_0:1 ./osu\_bibw -x 1000 -i 1000 -d cuda D D
17. mpirun --allow-run-as-root -np 2 --host node2:64 -x UCX\_NET\_DEVICES=mlx5\_0:1 ./osu\_bibw -x 1000 -i 1000



24 vim README.txt

25 export HPCX\_HOME=$PWD

26 source $HPCX\_HOME/hpcx-init.sh

27 hpcx\_load

28 env | grep HPCX

29 cd /home/ram/opt/osu-micro-benchmarks-7.1-1/

30 ls

31 ./configure CC=mpicc CXX=mpicxx --enable-cuda --with-cuda=/usr/local/cuda-12.0 --prefix=/home/ram/app

32 make -j32

33 make -j32 install

34 cd ..

35 ls

36 cd ..

37 ls

38 cd app/libexec/osu-micro-benchmarks/mpi/

39 cd pt2pt/

40 which mpicc

41 mpirun --allow-run-as-root -np 2 -x UCX\_NET\_DEVICES=ib0 ./osu\_bibw -x 1000 -i 1000 -d cuda D D

Usage: osu\_bw [options]

Options:

-b, --buffer-num Use different buffers to perform data transfer (default single)

Options: single, multiple

-m, --message-size [MIN:]MAX set the minimum and/or the maximum message size to MIN and/or MAX

bytes respectively. Examples:

-m 128 // min = default, max = 128

-m 2:128 // min = 2, max = 128

-m 2: // min = 2, max = default

-M, --mem-limit SIZE set per process maximum memory consumption to SIZE bytes

(default 536870912)

-i, --iterations ITER set iterations per message size to ITER (default 1000 for small

messages, 100 for large messages)

-x, --warmup ITER set number of warmup iterations to skip before timing (default 200)

-W, --window-size SIZE set number of messages to send before synchronization (default 64)

-c, --validation Enable or disable validation. Disabled by default.

-u, --validation-warmup ITR Set number of warmup iterations to skip before timing when validation is enabled (default 5)

-D, --ddt [TYPE]:[ARGS] Enable DDT support

-D cont //Contiguous

-D vect:[stride]:[block\_length] //Vector

-D indx:[ddt file path] //Index

-G, --graph tty,png,pdf graph output of per iteration values.

-T, --type [TYPE] Set MPI\_TYPE <all,mpi\_char,mpi\_int,mpi\_float>. Default:MPI\_CHAR.

-h, --help print this help

mpirun --allow-run-as-root -np 2 --hostfile hostfile --pernode --map-by ppr:1:node -x UCX\_NET\_DEVICES=mlx5\_0:1 osu\_bibw -x 1000 -i 1000

mpirun --allow-run-as-root -np 2 --hostfile hostfile --map-by ppr:1:node -x UCX\_NET\_DEVICES=mlx5\_0:1 osu\_bibw -x 1000 -i 1000

[root@node2 ~]# mpirun --allow-run-as-root -np 2 --hostfile /tmp/hosts -bind-to core --report-bindings --pernode -mca coll\_fca\_enable 0 -mca coll\_hcoll\_enable 0 -x UCX\_TLS=rc\_x,self,shm -mca btl\_openib\_if\_include mlx5\_0:1 -x UCX\_NET\_DEVICES=mlx5\_0:1 /app/new/v5.8/osu-micro-benchmarks-5.8/mpi/pt2pt/osu\_bw -i 1000

**ERROR:-**

[node1:03871] 1 more process has sent help message help-orte-rmaps-base.txt / deprecated

[node1:03871] Set MCA parameter "orte\_base\_help\_aggregate" to 0 to see all help / error messages

The following command line options and corresponding MCA parameter have

been deprecated and replaced as follows:

Command line options:

Deprecated: --bynode, -bynode

Replacement: --map-by node

Equivalent MCA parameter:

Deprecated: rmaps\_base\_bynode

Replacement: rmaps\_base\_mapping\_policy=node

The deprecated forms \*will\* disappear in a future version of Open MPI.

Please update to the new syntax.

**Running command**

#mpirun --allow-run-as-root -np 2 --hostfile hostfile -pernode --map-by node -x UCX\_NET\_DEVICES=mlx5\_0:1 osu\_bibw -x 1000 -i 1000

# mpirun --allow-run-as-root -np 2 --hostfile hostfile -npernode 1 -x UCX\_NET\_DEVICES=mlx5\_0:1 osu\_bibw -x 1000 -i 1000

**MCA Parameter check:-**

**#** ompi\_info --param all all

**#** ompi\_info --param btl all

**# mpirun --allow-run-as-root -np 2 --host node1,node2 -x 2UCX\_NET\_DEVICES=mlx5\_0:1 osu\_bibw -x 100 -i 100**

**# mpirun --allow-run-as-root -np 2 --mca rmaps\_base\_schedule\_policy slot --host node1,node2 osu\_bw -i 1000**

**# mpirun --allow-run-as-root -np 2 -mca ras\_gridengine\_verbose 1000 --host node1,node2 osu\_bw -i 1000 t\_connect**

**#mpirun --allow-run-as-root -np 2 --host node1,node2 osu\_bibw -i 100**

**#mpirun -np 32 -mca coll\_hcoll\_enable 1 -x coll\_hcoll\_np=0 -x HCOLL\_MAIN\_IB=<device\_name>:<port\_num>**

**#**