Mellanox GPUDirect RDMA User Manual

Rev 1.2

www.mellanox.com

NOTE:
THIS HARDWARE, SOFTWARE OR TEST SUITE PRODUCT (“PRODUCT(S)”) AND ITS RELATED DOCUMENTATION ARE PROVIDED BY MELLANOX TECHNOLOGIES “AS-IS” WITH ALL FAULTS OF ANY KIND AND SOLELY FOR THE PURPOSE OF AIDING THE CUSTOMER IN TESTING APPLICATIONS THAT USE THE PRODUCTS IN DESIGNATED SOLUTIONS. THE CUSTOMER'S MANUFACTURING TEST ENVIRONMENT HAS NOT MET THE STANDARDS SET BY MELLANOX TECHNOLOGIES TO FULLY QUALIFY THE PRODUCT(S) AND/OR THE SYSTEM USING IT. THEREFORE, MELLANOX TECHNOLOGIES CANNOT AND DOES NOT GUARANTEE OR WARRANT THAT THE PRODUCTS WILL OPERATE WITH THE HIGHEST QUALITY. ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT ARE DISCLAIMED. IN NO EVENT SHALL MELLANOX BE LIABLE TO CUSTOMER OR ANY THIRD PARTIES FOR ANY DIRECT, INDIRECT, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES OF ANY KIND (INCLUDING, BUT NOT LIMITED TO, PAYMENT FOR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY FROM THE USE OF THE PRODUCT(S) AND RELATED DOCUMENTATION EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Mellanox
TECHNOLOGIES

Mellanox Technologies
350 Oakmead Parkway Suite 100
Sunnyvale, CA 94085
U.S.A.
www.mellanox.com
Tel: (408) 970-3400
Fax: (408) 970-3403

Document Number: MLNX-15-3878

Table of Contents

List of Tables

Document Revision History

Table 1 - Document Revision History

Release	Date	Description
1.2	September, 2015	Updated the following sections: • Section 1.1, “System Requirements”, on page 6 • Section 2, “Installing GPUDirect RDMA”, on page 8 • Section 3.1, “Running GPUDirect RDMA with MVAPICH-GDR 2.1”, on page 9 • Section 3.2, “Running GPUDirect RDMA with OpenMPI 1.10.0”, on page 10
1.1	December 18, 2014	Updated Section 3.2, “Running GPUDirect RDMA with MVAPICH-GDR 2.0b”, on page 6 - Added how to enable RoCE communication.
1.0	May 19, 2014	Initial release

1 Overview

GPUDirect RDMA is an API between IB CORE and peer memory clients, such as NVIDIA Kepler class GPU's. It provides access the HCA to read/write peer memory data buffers, as a result it allows RDMA-based applications to use the peer device computing power with the RDMA interconnect without the need to copy data to host memory. This capability is supported with Mellanox ConnectX®-3 VPI and later or Connect-IB® InfiniBand adapters. It will also work seemlessly using RoCE technology with the Mellanox ConnectX®-3 and later VPI adapters.

1.1 System Requirements

The platform and server requirements for GPUDirect RDMA are detailed in the following table:

Table 2 - GPUDirect RDMA System Requirements

Platform	Type and Version
HCAs	• Mellanox ConnectX®-3 • Mellanox ConnectX®-3 Pro • Mellanox Connect-IB® • Mellanox ConnectX®-4 • NVIDIA® Tesla™ K-Series (K10, K20, K40, K80) GPU
Software/Plugins	• MLNX_OFED v2.1-x.x.x or later www.mellanox.com -> Products -> Software - > InfiniBand/VPI Drivers -> Linux SW/Drivers • Plugin module to enable GPUDirect RDMA www.mellanox.com -> Products -> Software - > InfiniBand/VPI Drivers -> GPUDirect RDMA (on the left navigation pane) • NVIDIA Driver http://www.nvidia.com/Download/index.aspx?lang=en-us • NVIDIA CUDA Runtime and Toolkit https://developer.nvidia.com/cuda-downloads NVIDIA Documentation http://docs.nvidia.com/cuda/index.html#getting-started-guides

1.2 Important Notes

- Once the NVIDIA software components are installed, it is important to check that the GPUDirect kernel module is properly loaded on each of the compute systems where you plan to run the job that requires the GPUDirect RDMA feature.

  To check:

  ```
  service nv_peer_mem status
  ```

  Or for some other flavors of Linux:

  ```
  lsmod | grep nv_peer_mem
  ```

  Usually this kernel module is set to load by default by the system startup service. If not loaded, GPUDirect RDMA would not work, which would result in very high latency for message communications.

  One you start the module by either:

  ```
  service nv_peer_mem start
  ```

Or for some other flavors of Linux:

```
modprobe nv_peer_mem
```

- To achieve the best performance for GPUDirect RDMA, it is required that both the HCA and the GPU be physically located on the same PCIe IO root complex.

  To find out about the system architecture, either review the system manual, or run `"lspci -tv |grep NVIDIA”`.

2 Installing GPUDirect RDMA

Please ensure that you have installed MLNX_OFED before trying to install GPUDirect RDMA. MLNX_OFED can be downloaded from:
www.mellanox.com -> Products -> Software - > InfiniBand/VPI Drivers -> Linux SW/Drivers

➤ ***To install GPUDirect RDMA for OpenMPI (excluding Ubuntu):***

Step 1. Unzip the package.

```
untar  nvidia_peer_memory-1.0-1.tar.gz
```

Step 2. Change the working directory to be `nvidia_peer_memory`.

```
cd nvidia_peer_memory-1.0-0/
```

Step 3. Display the content of the README file and follow the installation instructions.

```
cat README.txt
```

Note: On SLES OSes add `"--nodeps"`.

➤ ***To install GPUDirect RDMA for OpenMPI on Ubuntu:***

Copy the tarball to a temporary directory.

```
tar xzf <tarball>
cd <extracted directory>
dpkg-buildpackage -us -uc
dpkg -i <path to generated deb files>
```

Example:

```
dpkg -i nvidia-peer-memory_1.0-0_all.deb
dpkg -i nvidia-peer-memory-dkms_1.0-0_all.deb
```

Please make sure this kernel module is installed and loaded on each GPU InfiniBand compute nodes.

➤ ***To install GPUDirect RDMA for MVAPICH2:***

Step 1. Download gdrcopy library from https://github.com/NVIDIA/gdrcopy/archive/master.zip and build it.

```
cd /opt/mvapich2/gdr/2.1/cuda7.0/gnu
unzip master.zip
cd/opt/mvapich2/gdr/2.1/cuda7.0/gnu/gdrcopy-master
make CUDA=/usr/local/cuda-7.0 all
```

Step 2. Make sure gdrcopy is installed on all compute nodes and load the module on each GPU node.

```
cd/opt/mvapich2/gdr/2.1/cuda7.0/gnu/gdrcopy-master
./insmod.sh
```

3 Benchmark Tests

3.1 Running GPUDirect RDMA with MVAPICH-GDR 2.1

MVAPICH2 takes advantage of the new GPUDirect RDMA technology for inter-node data movement on NVIDIA GPUs clusters with Mellanox InfiniBand interconnect.

MVAPICH-GDR v2.1, can be downloaded from:
http://mvapich.cse.ohio-state.edu/download/

GPUDirect RDMA can be tested by running the micro-benchmarks from Ohio State University (OSU). Below is an example of running one of the OSU benchmark, which is already bundled with MVAPICH2-GDR v2.1, with GPUDirect RDMA.

```
[mpirun -np 2 host1 host2 -genv MV2_CPU_MAPPING=0 -genv MV2_USE_CUDA=1 -genv MV2_USE_GPUDIRECT=1 /
opt/mvapich2/gdr/2.1/cuda7.0/gnu/libexec/mvapich2/osu_bw -d cuda D D
# OSU MPI-CUDA Bandwidth Test
# Send Buffer on DEVICE (D) and Receive Buffer on DEVICE (D)
# Size      Bandwidth (MB/s)
...
2097152              6372.60
4194304              6388.63
```

Please note that MV2_CPU_MAPPING=<core number> has to be a core number from the same socket that shares the same PCI slot with the GPU.

The MV2_GPUDIRECT_LIMIT is used to tune the hybrid design that uses pipelining and GPU-Direct RDMA for maximum performance while overcoming P2P bandwidth bottlenecks seen on modern systems. GPUDirect RDMA is used only for messages with size less than or equal to this limit.

Here is a list of runtime parameters that can be used for process-to-rail binding in case the system has multi-rail configuration:

```
export MV2_USE_CUDA=1 export MV2_USE_GPUDIRECT=1
export MV2_RAIL_SHARING_POLICY=FIXED_MAPPING
export MV2_PROCESS_TO_RAIL_MAPPING=mlx5_0:mlx5_1
export MV2_RAIL_SHARING_LARGE_MSG_THRESHOLD=1G
export MV2_CPU_BINDING_LEVEL=SOCKET
export MV2_CPU_BINDING_POLICY=SCATTER
```

Additional tuning parameters related to CUDA and GPUDirect RDMA (such as MV2_CUDA_-BLOCK_SIZE) can be found in the MVAPICH2 user guideline.

Below is an example of enabling RoCE communication.

```
mpirun -np 2 host1 host2 -genv MV2_USE_RoCE=1 -genv MV2_DEFAULT_GID_INDEX=2 -genv MV2_DEFAULT_SER-
VICE_LEVEL=3 -genv MV2_USE_CUDA=1 MV2_USE_GPUDIRECT=1 /opt/mvapich2/gdr/2.1/cuda7.0/gnu/libexec/
mvapich2/osu_bw -d cuda D D
```

Where:

Parameter	Description
MV2_USE_RoCE=1	Enables RoCE communication.

Parameter	Description
MV2_DEFAULT_GID_INDEX=<gid index>	Selects the non-default GID index using MV2_DEFAULT_GID_INDEX since all VLAN interfaces appear as additional GID indexes (starting from 1) on the InfiniBand HCA side of the RoCE adapter. You can select a non-default GID index using run-time parameter MV2_DEFAULT_GID_INDEX(11.84) and RoCE priority service level using MV2_DEFAULT_SERVICE_LEVEL
MV2_DEFAULT_SERVICE_LEVEL=<service_level>	Selects RoCE priority service level using MV2_DEFAULT_SERVICE_LEVEL

3.2 Running GPUDirect RDMA with OpenMPI 1.10.0

The GPUDirect RDMA support is available on OpenMPI 1.10.0. Unlike MVAPICH2-GDR which is available in the RPM format, one can download the source code for OpenMPI and compile using flags below to enable GPUDirect RDMA support:

```
% ./configure --prefix=/path/to/openmpi-1.10.0_cuda7.0 \
--with-wrapper-ldflags=-Wl,-rpath,/lib --disable-vt --enable-orterun-prefix-by-default \
-disable-io-romio --enable-picky \
--with-cuda=/usr/local/cuda-7.0
% make; make install
```

The OSU benchmarks are CUDA-enabled benchmarks that can downloaded from http://mvapich.cse.ohio-state.edu/benchmarks.

When building the OSU benchmarks, you must verify that the proper flags are set to enable the CUDA part of the tests, otherwise the tests will only run using the host memory instead which is the default.

Additionally, make sure that the MPI libraries, OpenMPI is installed prior to compiling the benchmarks.

```
export PATH=/path/to/openmpi-1.10.0_cuda7.0/bin:$PATH
./configure CC=mpicc -prefix=/path/to/osu-benchmarks  \
--enable-cuda --with-cuda=/usr/local/cuda-7.0
make
make install
```

To run the OpenMPI that uses the flag that enables GPUDirect RDMA:

```
%  mpirun -mca btl_openib_want_cuda_gdr 1 -np 2 -npernode 1 -mca btl_openib_if_include mlx-
5_0:1 -bind-to-core -cpu-set 19 -x CUDA_VISIBLE_DEVICES=0 /path/to/osu-benchmarks/osu_latency
-d cuda D D
# OSU MPI-CUDA Latency Test
# Send Buffer on DEVICE (D) and Receive Buffer on DEVICE (D)
# Size          Latency (us)
0                       1.08
1                       3.83
2                       3.83
4                       3.84
8                       3.83
16                      3.83
32                      3.82
64                      3.80
...
```

Please note that -cpu-set=<core number> has to be a core number from the same socket that shares the same PCI slot with the GPU.

> If the flag for GPUDirect RDMA is not enabled, it would result in much higher latency for the above.

By default in OpenMPI 1.10.0, the GPUDirect RDMA will work for message sizes between 0 to 30KB. For messages above that limit, it will be switched to use asynchronous copies through the host memory instead. Sometimes, better application performance can be seen by adjusting that limit. Here is an example of increasing to adjust the switch over point to above 64KB:

```
-mca btl_openib_cuda_rdma_limit 65537
```