Appendix – NVIDIA SHARP Integration

Table of contents

List of Figures

NVIDIA Scalable Hierarchical Aggregation and Reduction Protocol (SHARP)™

NVIDIA SHARP is a technology that improves the performance of MPI operation by offloading collective operations from the CPU and dispatching to the switch network, and eliminating the need to send data multiple times between endpoints. This approach decreases the amount of data traversing the network as aggregation nodes are reached, and dramatically reduces the MPI operation time.

NVIDIA SHARP software is based on:

- Hardware capabilities in Switch-IB™ 2
- Hierarchical communication algorithms (HCOL) library into which NVIDIA SHARP capabilities are integrated
- NVIDIA SHARP daemons, running on the compute nodes
- NVIDIA SHARP Aggregation Manager, running on UFM

1. These components should be installed from HPCX or MLNX_OFED packages on compute nodes. Installation details can be found in SHARP Deployment Guide.

NVIDIA SHARP Aggregation Manager

Aggregation Manager (AM) is a system management component used for system level configuration and management of the switch-based reduction capabilities. It is used to set up the NVIDIA SHARP trees, and to manage the use of these entities.

AM is responsible for:

- NVIDIA SHARP resource discovery
- Creating topology aware NVIDIA SHARP trees
- Configuring NVIDIA SHARP switch capabilities
- Managing NVIDIA SHARP resources
- Assigning NVIDIA SHARP resource upon request

- Freeing NVIDIA SHARP resources upon job termination

AM is configured by a topology file created by Subnet Manager (SM): subnet.lst. The file includes information about switches and HCAs.

NVIDIA SHARP AM Prerequisites

In order for UFM to run NVIDIA SHARP AM, the following conditions should be met:

- Managed InfiniBand fabric must include at least one of the following Switch-IB 2 switches with minimal firmware version of 15.1300.0126:
    - CS7500
    - CS7510
    - CS7520
    - MSB7790
    - MSB7800
- NVIDIA SHARP software capability should be enabled for all Switch-IB 2 switches in the fabric (a dedicated logical port #37, for NVIDIA SHARP packets transmission, should be enabled and should be visible via UFM).
- UFM OpenSM should be running to discover the fabric topology.

NVIDIA SHARP AM is tightly dependent on OpenSM as it uses the topology discovered by OpenSM.

- NVIDIA SHARP AM should be enabled in UFM configuration by running:

```
[Sharp]
sharp_enabled = true
```

NVIDIA SHARP AM Configuration

By default, when running NVIDIA SHARP AM by UFM, there is no need to run further configuration. To modify the configuration of NVIDIA SHAPR AM, you can edit the following NVIDIA SHARP AM configuration file:/opt/ufm/files/conf/sharp/sharp_am.cfg.

Running NVIDIA SHARP AM in UFM

To run NVIDIA SHARP AM within UFM, do the following:

1. Make sure that the root GUID configuration file (root_guid.conf) exists in conf/opensm. This file is required for activating NVIDIA SHARP AM.

2. Enable NVIDIA SHARP in conf/opensm/opensm.conf OpenSM configuration file by running "ib sm sharp enable" or by setting the sharp_enabled parameter to 2:

   ```
   # SHArP support
   # 0: Ignore SHArP - No SHArP support
   # 1: Disable SHArP - Disable SHArP on all supporting switches
   # 2: Enable SHArP - Enable SHArP on all supporting switches
   sharp_enabled 2
   ```

3. Make sure that port #6126 (on which NVIDIA SHARP AM is communicating with NVIDIA SHARP daemons) is not being used by any other application. If the port is being used, you can change it by modifying **smx_sock_port** parameter in the NVIDIA SHARP AM configuration file: conf/sharp2/sharp_am.cfg or via the command "ib sharp port".

4. Enable NVIDIA SHARP AM in conf/gv.cfg UFM configuration file by running the command "ib sharp enable" or by setting the sharp_enabled parameter to true (it is false by default):

   ```
   [Sharp]
   sharp_enabled = true
   ```

5. (Optional) Enable NVIDIA SHARP allocation in conf/gv.cfg UFM configuration file by setting the sharp_allocation_enabled parameter to true (it is false by default):

```
[Sharp]
sharp_allocation_enabled = true
```

Note

If the field sharp_enabled, and sharp_allocation_enabled are both set as true in gv.cfg, UFM sends an allocation (reservation) request to NVIDIA SHARP Aggregation Manager (AM) to allocate a list of GUIDs to the specified PKey when a new “Set GUIDs for PKey” REST API is called. If an empty list of GUIDs is sent, a PKEY deallocation request is sent to the SHARP AM.

NVIDIA SHARP allocations (reservations) allow SHARP users to run jobs on top of these resource (port GUID) allocations for the specified PKey. For more information, please refer to the *UFM REST API Guide* under Actions REST API PKey GUIDs Set/Update PKey GUIDs.

Operating NVIDIA SHARP AM with UFM

If NVIDIA SHARP AM is enabled, running UFM will run NVIDIA SHARP AM, and stopping UFM will stop NVIDIA SHARP AM.

To

➢ **start UFM with NVIDIA SHARP AM (enabled):**

```
/etc/init.d/ufmd start
```

The same command applies to HA, using /etc/init.d/ufmha.

Upon startup of UFM or SHARP Aggregation Manager, UFM will resend all existing persistent allocation to SHARP AM.

➢ **To stop UFM with NVIDIA SHARP AM (enabled):**

```
/etc/init.d/ufmd stop
```

➢ **To stop only NVIDIA SHARP AM while leaving UFM running:**

```
/etc/init.d/ufmd sharp_stop
```

➢ **To start only NVIDIA SHARP AM while UFM is already running:**

```
/etc/init.d/ufmd sharp_start
```

Upon startup of UFM or SHARP Aggregation Manager, UFM will resend all existing persistent allocation to SHARP AM.

➢ **To restart only NVIDIA SHARP AM while UFM is running:**

```
/etc/init.d/ufmd sharp_restart
```

Upon startup of UFM or SHARP Aggregation Manager, UFM will resend all existing persistent allocation to SHARP AM.

➢ **To display NVIDIA SHARP AM status while UFM is running:**

```
/etc/init.d/ufmd sharp_status
```

Monitoring NVIDIA SHARP AM by UFMHealth

UFMHealth monitors SHARP AM and verifies that NVIDIA SHARP AM is always running. When UFMHealth detects that NVIDIA SHARP AM is down, it will try to re-start it, and will trigger an event to the UFM to notify it that NVIDIA SHARP AM is down.

Managing NVIDIA SHARP AM by UFM High Availability (HA)

In case of a UFM HA failover or takeover, NVIDIA SHARP AM will be started on the new master node using the same configuration that was used prior to the failover/takeover.

NVIDIA SHARP AM Logs

NVIDIA SHARP AM log file (sharp_am.log) at /opt/ufm/files/log.

NVIDIA SHARP AM log files are rotated by UFM logrotate mechanism.

NVIDIA SHARP AM Version

NVIDIA SHARP AM version can be found at /opt/ufm/sharp/share/doc/SHARP_VERSION.