Scalable Hierarchical Aggregation Protocol (SHARP)
Deployment Guide

Rev 3.0
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.
# Table of Contents

Document Revision History.................................................................................................................. 5

1 SHARP Overview............................................................................................................................. 6
   1.1 Packages ................................................................................................................................ 6
   1.2 Prerequisites ........................................................................................................................... 6
   1.3 Supported OS and Platforms .................................................................................................. 7

2 Downloading Packages .................................................................................................................. 8

3 Setting up SHARP Environment.................................................................................................... 8

4 Configuring Subnet Manager......................................................................................................... 9

5 Configuring Aggregation Manager .............................................................................................. 10

6 Running SHARP Deamons............................................................................................................ 11
   6.1 sharp_am Registration as a Service on the SM Server and its Starting ................................ 11
   6.2 sharpd Registration as Service on all Compute Nodes and its Starting ........................... 11
   6.3 Removing Daemons ............................................................................................................. 12

7 Running OpenMPI with SHARP ................................................................................................... 13

8 SHARP Benchmark Script ............................................................................................................ 17
   8.1 Prerequisites ......................................................................................................................... 17

9 Job Scheduler Integration ............................................................................................................. 18
   9.1 Running SHARPD daemon in Managed Mode ................................................................. 18
List of Tables

Table 1: Document Revision History ....................................................................................................... 5
# Document Revision History

**Table 1: Document Revision History**

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0</td>
<td>February 01st, 2017</td>
<td>• Added section:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <a href="#">Job Scheduler Integration</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Updated the following sections:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <a href="#">Configuring Subnet Manager</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <a href="#">Configuring Aggregation Manager</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <a href="#">Running SHARP Deamons</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <a href="#">sharpd Registration as Service on all Compute Nodes and its Starting</a></td>
</tr>
<tr>
<td>2.0</td>
<td>October 30th, 2016</td>
<td>• Added section:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <a href="#">Setting up SHARP Environment</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Updated the following sections:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <a href="#">Packages</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <a href="#">Running SHARP Deamons</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <a href="#">Running OpenMPI with SHARP</a></td>
</tr>
<tr>
<td>1.0</td>
<td>August 15, 2016</td>
<td>Initial version of this document</td>
</tr>
</tbody>
</table>
1 SHARP Overview

Mellanox Scalable Hierarchical Aggregation Protocol (SHARP™) technology improves upon the performance of MPI operation, by offloading collective operations from the CPU to the switch network, and eliminating the need to send data multiple times between endpoints. This innovative approach decreases the amount of data traversing the network as aggregation nodes are reached, and dramatically reduces the MPI operations time. Implementing collective communication algorithms in the network also has additional benefits, such as freeing up valuable CPU resources for computation rather than using them to process communication.

1.1 Packages

<table>
<thead>
<tr>
<th>Package</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLNX OFED</td>
<td>4.0-x.x.x</td>
</tr>
<tr>
<td>HPC-X</td>
<td>1.8.x</td>
</tr>
<tr>
<td>UFM (Aggregation Manager only)</td>
<td>5.8</td>
</tr>
</tbody>
</table>

1.2 Prerequisites

<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Externally managed Switch-IB 2</td>
<td>Firmware version: 15.1200.0102 or later</td>
</tr>
<tr>
<td>MLNX OS</td>
<td>3.6.2002</td>
</tr>
<tr>
<td>Subnet Manager</td>
<td>4.7 (MLNX OFED 3.3-x.x.x or UFM 5.6) or later</td>
</tr>
<tr>
<td></td>
<td>For hypercube topology 4.8.1 (MLNX OFED 4.0-x.x.x or UFM 5.8).</td>
</tr>
</tbody>
</table>
# Supported OS and Platforms

<table>
<thead>
<tr>
<th>Distro</th>
<th>Platform</th>
<th>Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>RHEL 6.1</td>
<td>x86-64</td>
<td>2.6.32-131.0.15</td>
</tr>
<tr>
<td>RHEL 6.2</td>
<td>x86-64</td>
<td>2.6.32-220</td>
</tr>
<tr>
<td>RHEL 6.3</td>
<td>x86-64</td>
<td>2.6.32-279</td>
</tr>
<tr>
<td>RHEL 6.4</td>
<td>x86-64</td>
<td>2.6.32-358</td>
</tr>
<tr>
<td>RHEL 6.5</td>
<td>x86-64</td>
<td>2.6.32-431</td>
</tr>
<tr>
<td>RHEL 7.0</td>
<td>x86-64</td>
<td>3.10.0-123</td>
</tr>
<tr>
<td>RHEL 7.2</td>
<td>x86-64</td>
<td>3.10.0-327</td>
</tr>
<tr>
<td>RHEL 7.2 ppcle</td>
<td></td>
<td>3.10.0-327</td>
</tr>
<tr>
<td>Fedora14</td>
<td>x86-64</td>
<td>2.6.35.6-45</td>
</tr>
<tr>
<td>Fedora16</td>
<td>x86-64</td>
<td>3.1.0-7</td>
</tr>
<tr>
<td>Fedora17</td>
<td>x86-64</td>
<td>3.3.4-5</td>
</tr>
<tr>
<td>Fedora18</td>
<td>x86-64</td>
<td>3.6.10-4</td>
</tr>
<tr>
<td>Fedora24</td>
<td>x86-64</td>
<td>4.5.5-300</td>
</tr>
<tr>
<td>SLES 11 SP1</td>
<td>x86-64</td>
<td>2.6.32.12-0.7</td>
</tr>
<tr>
<td>SLES 11 SP2</td>
<td>x86-64</td>
<td>3.0.13-0.27</td>
</tr>
<tr>
<td>SLES 11 SP3</td>
<td>x86-64</td>
<td>3.0.76-0.11</td>
</tr>
<tr>
<td>Ubuntu12.04</td>
<td>x86-64</td>
<td>3.2.0-37</td>
</tr>
<tr>
<td>Ubuntu13.10</td>
<td>x86-64</td>
<td>3.11.0-12</td>
</tr>
<tr>
<td>Ubuntu14.4</td>
<td>x86-64</td>
<td>3.13.0-24</td>
</tr>
<tr>
<td>Ubuntu14.4 ppc64le</td>
<td></td>
<td>3.13.0-32</td>
</tr>
<tr>
<td>Ubuntu15.10</td>
<td>x86-64</td>
<td>4.2.0-16</td>
</tr>
<tr>
<td>CentOS6.3</td>
<td>x86-64</td>
<td>2.6.32-279</td>
</tr>
<tr>
<td>CentOS6.0</td>
<td>x86-64</td>
<td>2.6.32-71</td>
</tr>
</tbody>
</table>
2 Downloading Packages

Download the HPC-X packages from the Mellanox site:
http://www.mellanox.com/page/products_dyn?product_family=189&mtag=hpc-x

3 Setting up SHARP Environment

SHARP binary distribution is available as part of HPC-X, MLNX_OFED or UFM 5.8 packages. UFM 5.8 package includes only the Aggregation Manager.

- In case of HPC-X package, please refer to HPC-X User Manual for installation and configuration procedures

  This Deployment Guide and examples refer to the following environment variables HPCX_SHARP_DIR, OMPI_HOME and assumes that HPC-X installation is in a shared folder accessible from all compute nodes.

- In case of MLNX_OFED distribution or custom installation, you have to set the HPCX_SHARP_DIR environment variable to point to the directory in which it was installed (/opt/mellanox/sharp is a default directory for SHARP in MLNX_OFED package). OMPI_HOME should point to the MPI installation folder.

- In case of using the Aggregation Manager from the UFM distribution, you have to enable SHARP support in UFM. For further information, refer to the UFM User Manual. The rest of the SHARP components should be installed from either the HPC-X or MLNX_OFED packages.

Make sure the following are set prior to configuring the setup:

- SHARP configuration files must be created in the same location (please refer sections 5 and 6). Make sure that you have write permission to HPCX_SHARP_DIR/.

- Aggregation Manager and SHARP daemons are trusted entities and should run by a privilege user (root user).

- Aggregation Manager is a central entity running on a dedicated server with SM. This dedicated server cannot serve as a compute node and host SHARP daemon.

- SHARP daemons and Aggregation Manager communicate over TCP/IP which should be configured before running SHARP. Please check firewall settings.

- Run ibdiagnet and check the Switch-IB firmware installed. See section Prerequisites for the supported switch firmware version.

  # ibdiagnet

  See “Nodes Information” section in /var/tmp/ibdiagnet2/ibdiagnet2.log

- Using UD Mcast feature requires IPoIB interface enabled in compute servers.
4 Configuring Subnet Manager

Subnet Manager should be active prior to activation of Aggregation Manager (AM). SHARP functionality should be enabled in Switch-IB 2 switches (it is disabled by default)

➢ To enable SHARP functionality on Switch-IB 2 based switches:
1. Edit the opensm.conf file.
2. Set the parameter “sharp_enabled” to “2”.
3. Run OpenSM with the configuration file.

```
# opensm -F <opensm configuration file> -B
```

4. Verify that the Aggregation Nodes were activated by the OpenSM, run "ibnetdiscover".

For example:

```
vendid=0x0
devid=0xcf09
sysimguid=0x7cfe900300a5a2a0
caguid=0x7cfe900300a5a2a8
Ca 1 "H-7cfe900300a5a2a8"
#"Mellanox Technologies Aggregation Node" and
"[1](7cfe900300a5a2a8) "S-7cfe900300a5a2a0"
[37] # lid 256 lmc 0 "MF0;sharp2:MSB7800/U1" lid 512 4xFDR
```

➢ To configure hypercube support in Subnet Manager:
1. Make sure OpenSM version 4.8.1 is installed to support SHARP in hypercube fabric.
2. Configure OpenSM to use DOR routing engine.

```
routing_engie dor
```

3. Configure OpenSM to create hypercube coordinates file.

```
dor_hyper_cube_mode TRUE
```

When set to TRUE, OpenSM generates hypercube coordinates file that is used by the AM (opensm-dor-coordinates in the dump files directory).
5 Configuring Aggregation Manager

Aggregation Manager (AM) is a central entity running on a dedicated server along with the Subnet Manager.

If you use AM from the UFM package, please refer to the UFM User Manual for further information.

➢ To configure AM in HPC-X or MLNX_OFED packages:

1. Create a configuration directory for the future SHARP configuration file.

   ```
   # mkdir $HPCX_SHARP_DIR/conf
   ```

2. Configure switch ranking for fat-tree topology

   - Copy the `root_guids.cfg` file if used for configuration of Subnet Manager to $HPCX_SHARP_DIR/conf/root_guid.cfg.
   
   Otherwise,

   - Identify the root switches of the fabric and create a file with the node GUIDs of the root switches of the fabric.

   Each line in the file should contain a single node GUID in hexadecimal format. The file should be located at: $HPCX_SHARP_DIR/conf/root_guid.cfg

   For example, if there are two root switches with node GUIDs 0x0002c90000000001 and 0x0002c90000000008, the file should be as follows:

   ```
   0x0002c90000000001
   0x0002c90000000008
   ```

3. Create the `sharp_am.cfg` file:

   - For fat-tree topology

     ```
     # cat > $HPCX_SHARP_DIR/conf/sharp_am.cfg << EOF
     root_guids_file $HPCX_SHARP_DIR/conf/root_guid.cfg
     ib_port_guid <PortGUID of the relevant HCA port or 0x0>
     EOF
     ```

   - For hypercube topology

     ```
     # cat > $HPCX_SHARP_DIR/conf/sharp_am.cfg << EOF
     topology_type hypercube
     hyper_cube_coordinates_file /var/log/opensm-dor-coordinates
     ib_port_guid <PortGUID of the relevant HCA port or 0x0>
     EOF
     ```
6 Running SHARP Deamons

The SHARP software, i.e. SHARP daemon (sharpd) should be executed on every compute node, and the Aggregation Manager daemon (sharp_am) should be executed on a dedicated server along with Subnet Manager.

**NOTE:** The sharpd and sharp_am commands must be executed as root user.

- sharpd on all compute nodes
- sharp_am on SM node only

➢ To setup the daemons he following script should be used:

```
$HPCX_SHARP_DIR/sbin/sharp_daemons_setup.sh
```

Usage: `sharp_daemons_setup.sh` <-s> <-p SHArP location dir> <-r> <-d daemon> <-m>

- `s` - Setup SHArP daemon
- `p` - Path to alternative SHArP location dir
- `r` - Remove SHArP daemon
- `d` - Daemon name (sharpd or sharp_am)
- `m` - Add monit capability for daemon control

6.1 sharp_am Registration as a Service on the SM Server and its Starting

1. Run as root the following:
   - without monit support
     ```
     # $HPCX_SHARP_DIR/sbin/sharp_daemons_setup.sh -s -d sharp_am
     ```
   - with monit support
     ```
     # $HPCX_SHARP_DIR/sbin/sharp_daemons_setup.sh -s -d sharp_am -m
     ```

     Daemon's log location is: /var/log/sharp_am.log

2. Start sharp_am as root.
   ```
   # service sharp_am start
   ```

6.2 sharpd Registration as Service on all Compute Nodes and its Starting

The procedure, described below, needs pdsh package. If you do not have pdsh, please use any other parallel execution tool and refer to the command below as an example.

1. SHARP daemons require explicit configuration for the used IB port.
   Create `$HPCX_SHARP_DIR/conf/sharpd.cfg` and add the following lines:
   ```
   ib_dev <hca:port>
   ```

2. Run as root the following:
   - without monit support
     ```
     # pdsh -w <hostlist> $HPCX_SHARP_DIR/sbin/sharp_daemons_setup.sh -s -d sharpd
     ```
   - with monit support
     ```
     # pdsh -w <hostlist> $HPCX_SHARP_DIR/sbin/sharp_daemons_setup.sh -s -d sharpd -m
     ```
Daemon's log location: /var/log/sharpd.log

3. Start sharpd daemons as root.

```
# pdsh -w <hostlist> service sharpd start
```

### 6.3 Removing Daemons

- To remove sharp_am, run on AM host:

```
# $HPCX_SHARP_DIR/sbin/sharp_daemons_setup.sh -r -d sharp_am
```

- To remove sharpd, run:

```
# pdsh -w <hostlist> $HPCX_SHARP_DIR/sbin/sharp_daemons_setup.sh -r -d sharpd
```
Running OpenMPI with SHARP

The following basic flags should be used in mpirun command line to enable SHARP protocol in HCOLL middleware. For the rest of flags please refer to SHARP Release Notes.

<table>
<thead>
<tr>
<th>FLAG</th>
<th>Values</th>
</tr>
</thead>
</table>
| HCOLL_ENABLE_SHARP          | Default: 0  
Possible values:  
• 0 – Do not use SHARP (default)  
• 1 - probe SHARP availability and use it  
• 2 - Force to use SHARP  
• 3 - Force to use SHARP for all MPI communicators  
• 4 - Force to use SHARP for all MPI communicators and for all supported collectives(BARRIER, Allreduce) |
| SHARP_COLL_LOG_LEVEL        | Default: 2  
SHARP coll logging level. Messages with a level higher or equal to the selected will be printed.  
Possible values:  
• 0 - fatal  
• 1 - error  
• 2 - warn  
• 3 - info  
• 4 - debug  
• 5 - trace |
| SHARP_COLL_JOB_QUOTA_PAYLOAD_PER_OST | Default: 128 (Max: 256)  
Maximum payload per OST quota request. value 0 mean allocate default value. |

For example:

```
% $OMPI_HOME/bin/mpirun --display-map --bind-to-core --map-by=node -H host01,host02,host03 -np 3 -mca pml yalla -mca bt1_openib_warn_default_gid_prefix 0 -mca rmaps_dist_device mlx5_0:1 -mca rmaps_base_mapping_policy dist:span -x MXM_RDMA_PORTS=mlx5_0:1 -x HCOLL_MAIN_IB=mlx5_0:1 -x MXM_ASYNC_INTERVAL=1800s -x HCOLL_ENABLE_SHARP=1 -x SHARP_COLL_JOB_QUOTA_PAYLOAD_PER_OST=128 <PATH/osu_allreduce> -i 10000 -x 1000 -f -m 256
```

The following HCOLL flags can be used when running SHARP collective with mpirun utility:

<table>
<thead>
<tr>
<th>FLAG</th>
<th>Values</th>
</tr>
</thead>
</table>
| HCOLL_SHARP_NP              | Default: 2  
Number of nodes(node leaders) threshold in communicator to create SHARP group and use SHARP collectives |
<table>
<thead>
<tr>
<th>FLAG</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCOLL_SHARP_UPROGRESS_NUM_POLLS</td>
<td>Default: 999 Number of unsuccessful polling loops in libsharp coll for blocking collective wait before calling user progress (HCOLL, OMPI).</td>
</tr>
<tr>
<td>HCOLL_BCOL_P2P_ALLREDUCE_SHARP_MAX</td>
<td>Default : 256 Maximum allreduce size run through SHARP. Message size greater than above will fallback to non-SHARP based algorithms (multicast based or non-multicast based)</td>
</tr>
<tr>
<td>SHARP_COLL_MAX_PAYLOAD_SIZE</td>
<td>Default : 256 (Max) Maximum payload size of sharp collective request Collective requests for larger than this size will be pipelined.</td>
</tr>
<tr>
<td>SHARP_COLL_JOB_QUOTA_PAYLOAD_PER_OST</td>
<td>Default : 128 (Max: 256) Maximum payload per OST quota request. value 0 mean allocate default value.</td>
</tr>
</tbody>
</table>
| SHARP_COLL_GROUP_RESOURCE_POLICY                             | Default : 1 SHARP Job resource sharing policy between the groups (communicators) Values:  
  1 - equal  
  2 - take_all by first group  
  3 - User input percent using SHARP_COLL_USER_GROUP_QUOTA_PERCENT |
<p>| SHARP_COLL_USER_GROUP_QUOTA_PERCENT                          | % of job quota to be allocated for each sharp group.                                                                                                                                                                                                                  |
| SHARP_COLL_JOB_QUOTA_OSTS                                   | Default : 0 Maximum job (per tree) OST quota request. value 0 mean allocate default quota.                                                                                                                                                                          |
| SHARP_COLL_JOB_QUOTA_MAX_GROUPS                              | Default: 0 Maximum no. of groups (comms) quota request. Value 0 means allocate default value.                                                                                                                                                                      |
| SHARP_COLL_JOB_QUOTA_MAX_QPS_PER_PORT                       | Maximum QPs/port quota request. Value 0 mean allocate default value.                                                                                                                                                                                             |
| SHARP_COLL_PIPELINE_DEPTH                                   | Default : 8 Size of fragmentation pipeline for larger collective payload                                                                                                                                                                                           |</p>
<table>
<thead>
<tr>
<th>FLAG</th>
<th>Values</th>
</tr>
</thead>
</table>
| SHARP_COLL_STATS_FILE                         | Default = ""

Destination to send statistics to. Possible values are:
- stdout - print to standard output.
- stderr - print to standard error.
- file:<filename> - save to a file (%h: host, %p: pid, %t: time, %u: user, %e: exe) |

| SHARP_COLL_STATS_TRIGGER                      | Default : exit

Trigger to dump statistics:
- Exit - dump just before program exits.
- signal:<signo> - dump when process is signaled (Not fully supported) |

| SHARP_COLL_STATS_DUMP_MODE                    | Default : 1

Stats dump modes
1 - dump per process stats
2 - dump accumulative (per job) stats

**NOTE:** For accumulative mode(2), its user responsibility to call sharp_coll_dump_stats() when OOB is still active |

| SHARP_COLL_ENABLE_GROUP_TRIM                  | Default: 1

Enables sharp group trim feature |

| SHARP_COLL_ENABLE_MCAST_TARGET               | Default: 1

Enables MCAST target on sharp collective ops. |

| SHARP_COLL_MCAST_TARGET_GROUP_SIZE_THRESHOLD | Default: 2

Group size threshold to enable mcast target |

| SHARP_COLL_POLL_BATCH                        | Default: 4

Defines the number of CQ completions to poll on at once. Maximum: 16 |

| SHARP_COLL_ERROR_CHECK_INTERVAL              | Default: 180000

Interval, in milli second, indicates the time between the error checks. 
"If you set the interval as 0, error check is not performed" |

Example of SHARP run with HCOLL
% $OMPI_HOME/bin/mpirun --bind-to core --map-by node -hostfile /tmp/hostfile -np 4 -mca pml yalla -mca btl_openib_warn_default_gid_prefix 0 -mca
rmaps_dist_device mlx5_0:1 -mca rmaps_base_mapping_policy dist:span -x MXM_RDMA_PORTS=mlx5_0:1 -x HCOLL_MAIN_IB=mlx5_0:1 -x MXM_ASYNC_INTERVAL=1800s -x MXM_LOG_LEVEL=ERROR -x HCOLL_ML_DISABLE_REDUCE=1 -x HCOLL_ENABLE_MCAST_ALL=1 -x HCOLL_MCAST_DP=1 -x LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${HPCX_SHARP_DIR}/lib -x LD_PRELOAD=${HPCX_SHARP_DIR}/lib/libsharpm.so:${HPCX_SHARP_DIR}/lib/libsharpm_coll.so -x HCOLL_ENABLE_SHARPM=2 -x SHARP_COLL_LOG_LEVEL=3 -x SHARP_COLL_GROUP_RESOURCE_POLICY=1 -x SHARP_COLL_MAX_PAYLOAD_SIZE=256 -x HCOLL_SHARP_UPROGRESS_MAX_POLLS=999 -x HCOLL_BCOL_P2P_ALLREDUCE_SHARP_MAX=4096 -x SHARP_COLL_PIPELINE_DEPTH=32 -x SHARP_COLL_JOB_QUOTA_OSTS=32 -x SHARP_COLL_JOB_QUOTA_MAX_GROUPS=4 -x SHARP_COLL_JOB_QUOTA_PAYLOAD_PER_OST=256 taskset -c 1 numactl --membind=0 <PATH/osu_allreduce> -i 100 -x 100 -f -m 4096:4096

**NOTE:** For the complete list of SHARP_COLL tuning options, run the `sharp_coll_dump_config` utility.

`$HPCX_SHARP_DIR/bin/sharp_coll_dump_config -f`
8  **SHARP Benchmark Script**

SHARP distribution provides a test script which executes OSU (allreduce, barrier) benchmark running with and without SHARP. To run the SHARP benchmark script the following prerequisites are required:

- ssh
- pdsh
- environment-modules.x86_64

You can find this script at `$HPCX_SHARP_DIR/sbin/sharp_benchmark.sh`. This script should be launched from a host running SM and Aggregation Manager. It receives a list of compute host from SLURM allocation or from “hostlist” environment variable. “hostlist” is comma separated list. Also it requires hca environment variables to be supplied. It runs OSU all reduce and OSU barrier benchmarks with and without SHARP.

Example:

```
# hca=mlx5_0:1 hostlist=host1,host2,host3
$HPCX_SHARP_DIR/sbin/sharp_benchmark.sh
```
9 Job Scheduler Integration

9.1 Running SHARPD daemon in Managed Mode

When running the daemon in a managed mode, it expects communication from the prolog/epilog scripts of the Job Scheduler (JS). The prolog/epilog scripts should invoke the "sharp_job_quota" executable to communicate with the SHARP.

To run the SHARPD in managed mode, use the “managed_mode” option (default: 0 – run in “unmanaged” mode).

JS can set/unset upper limit for SHARP resources (e.g. OSTs, groups and etc.) allowed for a particular user/job via `sharp_job_quota` using the "set" and "remove" commands.

Usage

```
sharp_job_quota [OPTIONS]
```

**sharp_job_quota option**

<table>
<thead>
<tr>
<th>Option</th>
<th>Required/Optional</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-t, --operation</td>
<td>Required</td>
<td>set / remove</td>
<td>Sets or removes quota</td>
</tr>
<tr>
<td>-i, --allocation-id</td>
<td>Required</td>
<td>Unique numeric 64-bit ID</td>
<td>This is the scheduler id for the job. No other job in the system at the same time can have the same id</td>
</tr>
<tr>
<td>-u, --uid</td>
<td>Optional</td>
<td>Numeric</td>
<td>UID of the user allowed to run the job</td>
</tr>
<tr>
<td>-n, --user_name</td>
<td>Optional</td>
<td>string</td>
<td>Name of the user allowed to run the job</td>
</tr>
<tr>
<td>--coll_job_quota_max_groups</td>
<td>Optional</td>
<td>Numeric value: 0..256</td>
<td>Maximum number of SHARP groups (communicators) allowed. Default value: 0. 0 means there is not limit for the job. It can ask for any number.</td>
</tr>
<tr>
<td>--coll_job_quota_max_qps_per_port</td>
<td>Optional</td>
<td>Numeric value: 0..256</td>
<td>Maximum QPs/port allowed. Default value: 0. 0 means there is not limit for the job. It can ask for any number.</td>
</tr>
<tr>
<td>Required/Optional</td>
<td>Arguments</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>--coll_job_quota_max_payload_per_ost</td>
<td>Optional</td>
<td>Numeric value: 0..256</td>
<td>Maximum payload per OST allowed. Default value: 256</td>
</tr>
<tr>
<td>--coll_job_quota_max_osts</td>
<td>Optional</td>
<td>Numeric value: 0..512</td>
<td>Indicates the maximum number of OSTs allowed for job per collective operation. Default value: 0. 0 means there is not limit for the job. It can ask for any number.</td>
</tr>
</tbody>
</table>

**Important Notes:**
- The executable needs to run with the same user as the SD (root).
- When using the “set” operation either the uid or the user_name must be provided
- Regardless of the job quota set in prolog, the AM can allocate less resources than requested or decline the request

**Examples**

```
# sharp_job_quota --operation set --user_name jobrunner -allocation_ud 2017
# sharp_job_quota --operation remove --allocation_id 2017
```

**SLURM Examples**

```
#sharp_job_quota --operation set --uid $SLURM_JOB_UID --allocation_id $SLURM_JOB_ID
#sharp_job_quota --operation remove --allocation_id $SLURM_JOB_ID
```