Connect. Accelerate. Outperform.™

MellanoX Messaging Library User Manual

Rev 2.1

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

Mellanox Technologies, Ltd.
Beit Mellanox
PO Box 586 Yokneam 20692
Israel
www.mellanox.com
Tel: +972 (0)74 723 7200
Fax: +972 (0)4 959 3245

List of Tables

Document Revision History

Table 1 - Document Revision History

Document Revision	Date	Description

2.1	February 2014	• Updated the following sections:
		  • Section 2.1, "Compiling Open MPI with MXM," on page 7
		  • Section 2.2, "Enabling MXM in Open MPI," on page 9
		  • Section 2.4, "Configuring Multi-Rail Support," on page 9
		  • Section 2.6, "Configuring MXM over Different Transports," on page 10
		• Added the following section:
		  • Section 3.1, "mxm_dump_config," on page 11
		  • Section 3.2, "mxm_perftest," on page 11

2.0	August 2013	• Updated the following sections:
		  • Section 2.1, "Compiling Open MPI with MXM," on page 7
		  • Section 2.3, "Tuning MXM Settings," on page 9
		  • Section 2.4, "Configuring Multi-Rail Support," on page 9
		  • Section 2.5, "Configuring MXM over the Ethernet Fabric," on page 10
		• Added the following section:
		  • Section 2.6, "Configuring MXM over Different Transports," on page 10

1.5	December 2012	• Updated the following sections:
		  • Section 2.1, "Compiling Open MPI with MXM," on page 7
		  • Section 2.3, "Tuning MXM Settings," on page 9
		• Added the following section:
		  • Section 2.4, "Configuring Multi-Rail Support," on page 9
		  • Section 2.5, "Configuring MXM over the Ethernet Fabric," on page 10

1.1	July 2012	Initial release

1 MellanoX Messaging Library

1.1 Overview

MellanoX Messaging (MXM) library provides enhancements to parallel communication libraries by fully utilizing the underlying networking infrastructure provided by Mellanox HCA/switch hardware. This includes a variety of enhancements that take advantage of Mellanox networking hardware including:

- Multiple transport support including RC, DC and UD
- Proper management of HCA resources and memory structures
- Efficient memory registration
- One-sided communication semantics
- Connection management
- Receive side tag matching
- Intra-node shared memory communication

These enhancements significantly increase the scalability and performance of message communications in the network, alleviating bottlenecks within the parallel communication libraries.

The latest MXM software can be downloaded from the Mellanox website.

1.2 System Requirements

- Mellanox OFED 2.0-3.0.0 and later
- Open MPI 1.6.5 or later

  To download the Open MPI v1.6.5 which contains special patches added by Mellanox, please refer to the Mellanox website.
- Open MPI v1.7.4 or later

  To Open MPI v1.7.4, go to: http://www.open-mpi.org/software/ompi/v1.7/

2 Configuring MXM

2.1 Compiling Open MPI with MXM

Step 1. Install MXM from:

- an RPM

```
% rpm -ihv mxm-x.y.z-1.x86_64.rpm
```

- a tarball

```
% tar jxf mxm-x.y.z.tar.bz
```

MXM will be installed automatically in the `/opt/mellanox/mxm` folder.

Step 2. Enter Open MPI source directory and run:

```
% cd $OMPI_HOME
% ./configure --with-mxm=/opt/mellanox/mxm <... other configure parameters...>
% make all && make install
```

Older versions of MLNX_OFED come with pre-installed older MXM and Open MPI versions. To use MXM v2.1 and higher with older MLNX_OFED versions, please uninstall any old MXM version prior to installing MXM v2.1.

Table 2 - MLNX_OFED and MXM Versions

MLNX_OFED Version	MXM Version
v1.5.3-3.1.0 and v2.0-3.0.0	MXM v1.x and Open MPI compiled with MXM v1.x
v2.0-3.0.0 and higher	MXM v2.x and Open MPI compiled with MXM v2.x

To check the version of MXM installed on your host, run:

```
% rpm -qi mxm
```

➢ ***To upgrade MLNX_OFED v1.5.3-3.1.0 or later with a newer MXM:***

Step 1. Remove MXM.

```
# rpm -e mxm
```

Step 2. Remove the pre-compiled Open MPI.

```
# rpm -e mlnx-openmpi_gcc
```

Step 3. Install the new MXM and compile the Open MPI with it.

To run Open MPI without MXM, run:

```
% mpirun -mca mtl ^mxm <...>
```

When upgrading to MXM v2.1, Open MPI compiled with the previous versions of the MXM should be recompiled with MXM v2.1.

2.2 Enabling MXM in Open MPI

MXM Rev 2.1 is automatically selected by Open MPI (up to v1.6) when the Number of Processes (NP) is higher or equal to 128.

➢ ***To activate MXM for any NP, run:***

```
% mpirun -mca mtl_mxm_np 0 <...other mpirun parameters ...>
```

From Open MPI v1.7.x, MXM is selected when the number of processes is higher or equal to 0. i.e. by default.

2.3 Tuning MXM Settings

The default MXM settings are already optimized. To check the available MXM parameters and their default values, run the `/opt/mellanox/mxm/bin/mxm_dump_config -f` utility which is part of the MXM RPM.

MXM parameters can be modified in one of the following methods:

- Modifying the default MXM parameters value as part of the mpirun:

```
% mpirun -x MXM_UD_RX_MAX_BUFFS=128000 <...>
```

- Modifying the default MXM parameters value from SHELL:

```
% export MXM_UD_RX_MAX_BUFFS=128000
% mpirun <...>
```

2.4 Configuring Multi-Rail Support

Multi-Rail support enables the user to use more than one of the active ports on the card, by making a better use of the resources. It provides a combined throughput among the used ports.

Multi-Rail support in MXM v2.1 allows different processes on the same host to use different active ports. Every process can only use one port (as opposed to MXM v1.5).

➢ ***To configure dual rail support:***

- Specify the list of ports you would like to use to enable multi rail support.

```
-x MXM_RDMA_PORTS=cardName:portNum
```

or

```
-x MXM_IB_PORTS=cardName:portNum
```

For example:

```
-x MXM_IB_PORTS=mlx5_0:1
```

It is also possible to use several HCAs and ports during the run (separated by a comma):

```
-x MXM_IB_PORTS=mlx5_0:1,mlx5_1:1
```

MXM will bind a process to one of the HCA ports from the given ports list according to the MXM_IB_MAP_MODE parameter (for load balancing).

Possible values for MXM_IB_MAP_MODE are:

- first - [Default] Maps the first suitable HCA port to all processes
- affinity - Distributes the HCA ports evenly among processes based on CPU affinity
- nearest - Tries to find the nearest HCA port based on CPU affinity

You may also use an asterisk (*) and a question mark (?) to choose the HCA and the port you would like to use.

- * - use all active cards/ports that are available
- ? - use the first active card/port that is available

  For example:

```
-x MXM_IB_PORTS=*:?
```

  will take all the active HCAs and the first active port on each of them.

2.5 Configuring MXM over the Ethernet Fabric

➢ ***To configure MXM over the Ethernet fabric:***

Step 1. Make sure the Ethernet port is active.

```
% ibv_devinfo
```

ibv_devinfo displays the list of cards and ports in the system. Please make sure (in the ibv_devinfo output) that the desired port has Ethernet at the `link_layer` field and that its state is `PORT_ACTIVE`.

Step 2. Specify the ports you would like to use, if there is a non Ethernet active port in the card.

```
-x MXM_RDMA_PORTS=mlx4_0:1
```

or

```
-x MXM_IB_PORTS=mlx4_0:1
```

2.6 Configuring MXM over Different Transports

MXM v2.1 supports the following transports.

- Intra node communication via Shared Memory with KNEM support
- Unreliable Datagram (UD)
- Reliable Connected (RC)
- SELF transport - a single process communicates with itself
- Dynamically Connected Transport (DC) (at beta level)

  Note: DC is supported on Connect-IB® HCAs with MLNX_OFED v2.1-1.0.0 and higher.

  To use DC set the following:

  - in the command line:

```
% mpirun -x MXM_TLS=self,shm,dc
```

  - from the SHELL:

```
% export MXM_TLS=self,shm,dc
```

By default the transports (TLS) used are: `MXM_TLS=self,shm,rc,ud`

3 MXM Utilities

3.1 mxm_dump_config

Enables viewing of all the environment parameters that MXM uses.

To see all the parameters, run: `/opt/mellanox/mxm/bin/mxm_dump_config -f`.

For further information, please run: `/opt/mellanox/mxm/bin/mxm_dump_config -help`

Environment parameters can be set by using the “export” command.
For example, to set the MXM_TLS environment parameter, run:

```
% export MXM_TLS=<...>
```

3.2 mxm_perftest

A server-client based application which is designed to test MXM's performance and sanity checks on MXM.

To run it, two terminals are required to be opened, one on the server side and one on the client side.

The working flow is as follow:

1. The server listens to the request coming from the client.
2. Once a connection is established between them, MXM sends and receives messages between the two sides according to what the client requested.
3. The results of the communications are printed out.

For further information, please run: `/opt/mellanox/mxm/bin/mxm_perftest -help`.

Example:

- From the server side run: `/opt/mellanox/mxm/bin/mxm_perftest`
- from the client side run:
  `/opt/mellanox/mxm/bin/mxm_perftest <server_host_name> -t send_lat`

Among other parameters, you can specify the test you would like to run, the message size and the number of iterations to run.