Intel® Xeon Phi™ Coprocessor

Intel® Manycore Platform Software Stack (Intel® MPSS)

Cluster Setup Guide

Document Number: 328613-001US

Revision: 1.1

World Wide Web: http://www.intel.com

Disclaimer and Legal Information

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm%20

Intel, the Intel logo, Intel Xeon, Intel Xeon Phi, Intel Many Integrated Core Software, Intel Composer XE, and Intel® MPI Library are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Revision History

Document Number	Revision Number	Description	Revision Date
328613-001	1.0	Initial release	February 2013
328613-001	1.1	Added new Sections 3.3, 6.3.2.1, and 8.7. Deleted Section 3.1. Revised text in current Sections 3.2, 3.4, 3.5, 4, 5, 6.3.2, 6.4, 8.6, and 8.8.	April 2013

Contents

List of Figures

List of Tables

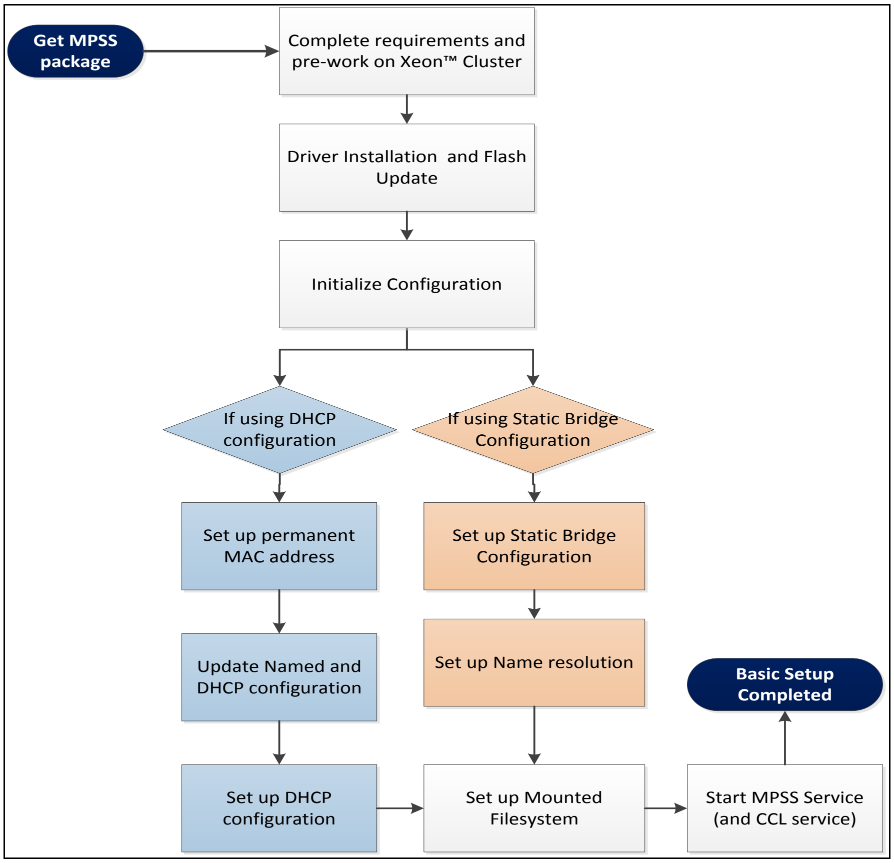
1 Introduction

This document is designed to outline the basic steps necessary to configure Intel® Xeon Phi™ coprocessors, based on Intel® Many Integrated Core (Intel® MIC) Architecture, in a wide variety of cluster environments. Some examples in this document may not apply to specific cluster environments, but were created in a general way so that they may be adapted. The following items are covered:

- Setting up an environment that enables communication between coprocessors in the cluster to enable the user to run the Intel® Message Passing Interface (Intel® MPI) application via Static Bridge or DHCP configuration.
- Enabling users to connect to and from the Intel® Xeon Phi™ coprocessor by using SSH without a password.
- Enabling administrators to mount the home file system.

The following flowchart shows the set up process that is covered in this document:

Figure 1 Setup Process Flowchart


Get MPSS package
Complete requirements and pre-work on Xeon™ Cluster
Driver Installation and Flash Update
Initialize Configuration
If using DHCP configuration
If using Static Bridge Configuration
Set up permanent MAC address
Set up Static Bridge Configuration
Update Named and DHCP configuration
Set up Name resolution
Basic Setup Completed
Set up DHCP configuration
Set up Mounted Filesystem
Start MPSS Service (and CCL service)

More features and functionalities are available for additional configuration but will not be covered in this document. See the Intel® MPSS Boot Configuration Guide document (MPSS_Boot_Config_Guide.pdf) for a detailed reference.

2 Requirements

Ensure the following requirements are met:

1. Administrative access privileges.
2. Verify that the kernel version of the Operating System installed in the cluster is supported for Intel® Manycore Platform Software Stack (Intel® MPSS). If not, then it will be necessary to rebuild the kmod*.rpms (not covered in this document).
3. Using Intel® MPSS Gold Update 3 driver.
4. Intel® Xeon Phi™ coprocessors are installed on the Intel® Xeon® host systems of the cluster, and are visible through lspci:

   ```
   user_prompt> lspci | grep –i co-processor
   ```

5. Ethernet connection between host systems is available.
6. Intel® ComposerXE and Intel® MPI packages (download from Intel® Premier) are installed and sourced.

3 Pre-work

3.1 RSA/DSA Key Creation

RSA/DSA keys for root and user (when available) should be created prior to Intel® MPSS installation. This will allow the mpssd service to automatically grab the keys and copy them to the coprocessor when the service is started.

3.2 IP Assignment (for Static Bridge only)

In order to get the Intel® Xeon Phi™ coprocessor to communicate with cards within the compute node as well as cards outside the compute node, all assigned IP for the coprocessor have to be set to the same subnet as the rest of the cluster. There are various options for implementing this. The setup described here uses a staggered approach, with the nodes IP as seen in Table 1, below.

NOTE: <*hostname*>-mic0 and <*hostname*>-mic1 are names for the Intel® Xeon Phi™ coprocessors.

Table 1 IP Assignment

Hostname	IP
node1	192.168.0.1
node1-mic0	192.168.0.2
node1-mic1	192.168.0.3
node2	192.168.0.4
node2-mic0	192.168.0.5
node2-mic1	192.168.0.6
node3	192.168.0.7
node3-mic0	192.168.0.8
node3-mic1	192.168.0.9
node4	192.168.0.10
node4-mic0	192.168.0.11
node4-mic1	192.168.0.12

3.3 Gather MAC Address List (for DHCP only)

Do this only if you have already configured the coprocessors to use DHCP configuration before (using Gold update 2) and want to keep the MAC address:

Collect all of the MAC addresses and their equivalent hostnames for the coprocessors to a file. This can be done by simply gathering the information from the DHCP server or using parallel shell and grep for the MAC address information from /etc/sysconfig/mic/mic?.conf .

The final result for the list should look like the following:

```
node1-mic0      ca:7a:05:e1:52:65
node1-mic1      76:f2:07:ff:00:3c
node2-mic0      fe:fc:cd:15:95:8a
node2-mic1      72:0c:4c:b7:94:86
```

3.4 Bridged Network Configuration

To prevent the need for routing daemon such as Quagga (not covered in this document), the compute nodes should deploy bridged networking. This will be required when using either Set up Static Bridge Configuration or Set up DHCP Configuration. An example of the bridged configuration for the compute nodes looks like the following:

ifcfg-br0 file:

```
DEVICE=br0
TYPE=Bridge
ONBOOT=yes
DELAY=0
NM_CONTROLLED="no"
MTU=9000
BOOTPROTO=dhcp
NOZEROCONF=yes
```

ifcfg-eth0 (assuming this is the main Ethernet used) file:

```
DEVICE=eth0
ONBOOT=yes
BRIDGE=br0
MTU=9000
```

NOTE: Make sure the MTU size is uniform across the switch, host, and card. Failure to do so will prevent the file transfer, even when ping pong test seems to work properly.

3.5 OpenFabrics Enterprise Distribution (OFED*) Installation

It is required to have the OFED* version 1.5.4.1 InfiniBand* drivers installed; otherwise OFED-MIC will fail. Do not use the InfiniBand* Driver supplied by Red Hat Enterprise Linux*. See the Intel® MPSS Readme for more information on how to enable InfiniBand* connection for Intel® True Scale Fabric or Mellanox* fabric.

4 Driver Installation

This section covers the method on how to install the Intel® Manycore Platform Software Stack (Intel® MPSS).

1. Download Intel® MPSS rpms.
2. Ensure all previous Intel® MPSS installations (if any exist) are removed via “rpm -e” or “yum erase”. **Do not UPGRADE the Intel® MPSS installation**.
   There are two options to remove all previous Intel® MPSS rpms:
   a. Option 1:
   ```
   user_prompt> rpm –e `rpm –qa | grep intel-mic`
   ```
   b. Option 2:
   ```
   user_prompt> yum erase intel-mic*
   ```
3. There are two options to install Intel® MPSS if the cluster is using RHEL* 6.1 or 6.2:
   a. Copy all rpms to a folder, and run the following command:
   ```
   user_prompt> rpm -ivh *.rpm
   ```
   b. Set up local yum repository to include the directory, and run the following command:
   ```
   user_prompt> yum install intel-mic*
   ```
4. Do not start the Intel® MPSS service directly after installation. To ensure this, do the following:
   ```
   user_prompt> chkconfig mpss off
   ```
   if installing OFED-MIC also run:
   ```
   user_prompt> chkconfig ofed-mic off
   ```
   Perform steps 5 and 6 below to update the flash and configure Intel® MPSS.
5. Update the Flash (see section 5).
6. Configure the Intel® MPSS service (see section 6).
7. Start the Intel® MPSS service (see section 7).

5 Flash Update

Before starting, make sure the following requirements are met:

1. The intel-mic-flash rpm is installed.
2. The coprocessor **stepping** is identified by running (optional):
   ```
   user_prompt> /opt/intel/mic/bin/micinfo | grep –i \
   "coprocessor stepping"
   ```
3. Check to make sure Intel® MPSS service is not running:
   ```
   user_prompt> service mpss status
   ```
   This should return: `mpss is stopped`
4. Make sure the required configuration has been initialized:
   ```
   user_prompt> micctrl --initdefaults
   ```

NOTE: If Intel® MPSS service has already been started for any reason, execute the following steps prior to doing the flash update:

1. `user_prompt> service ofed-mic status`

   If service is running: `ibp_server and ibscif loaded`
   If service is stopped: `ibp_server and/or ibscif not loaded`
2. If OFED-MIC service is running, stop the service:
   ```
   user_prompt> service ofed-mic stop
   ```
3. Reset the Intel® Xeon Phi™ coprocessor card:
   ```
   user_prompt> micctrl –r
   ```
4. Wait until the Intel® Xeon Phi™ coprocessor card has entered the 'Ready' state:
   ```
   user_prompt> micctrl -w
   ```

On a single host with two Intel® Xeon Phi^{TM} coprocessors, the firmware flash can be either updated one by one or all at the same time:

1. Updating one by one:
   ```
       a. user_prompt> /opt/intel/mic/bin/micflash -update \
   /opt/intel/mic/flash/EXT_HP2_SMC_Bootloader_1_8_4326.css_ab \
   -device 0 -noreboot
       b. user_prompt> /opt/intel/mic/bin/micflash -update \
   /opt/intel/mic/flash/EXT_HP2_SMC_Bootloader_1_8_4326.css_ab \
   -device 1 -noreboot
       c. user_prompt> /opt/intel/mic/bin/micflash -update \
          -device 0 -noreboot
       d. user_prompt> /opt/intel/mic/bin/micflash -update \
          -device 1 -noreboot
   ```

```
    e. user_prompt> reboot
```

2. Updating all at the same time:

```
    a. user_prompt> /opt/intel/mic/bin/micflash -update \
/opt/intel/mic/flash/EXT_HP2_SMC_Bootloader_1_8_4326.css_ab \
-device all -noreboot
    b. user_prompt> /opt/intel/mic/bin/micflash -update \
       -device all -noreboot
    c. user_prompt> reboot
```

In a cluster, it is recommended to use a tool like the open source Parallel Distributed Shell (PDSH), found in http://sourceforge.net/projects/pdsh, to update all nodes at the same time, although any parallel utilities can be used. The generic command usage for pdsh is:

```
user_prompt> pdsh -w <node range>  <command>
```

For the cluster used as example the steps are:

1. `user_prompt> pdsh -w node[1-4] 'micctrl -r'`

2.
   ```
   user_prompt> pdsh -w node[1-4] \
   '/opt/intel/mic/bin/micflash -update \
   /opt/intel/mic/flash/EXT_HP2_SMC_Bootloader_1_8_4326.css_ab \
   -device all -noreboot'
   ```

3.
   ```
   user_prompt> pdsh -w node[1-4] \
   '/opt/intel/mic/bin/micflash -update -device all -silent \
   -log /root/micflash_update_log.txt -noreboot'
   ```
4. Check whether the update was successful:
   ```
   user_prompt> pdsh -w node[1-4] \
   'cat /root/ micflash_update_log.txt | grep -i failed'
   ```
   If all return, “ssh exited with exit code 1”, then all updates were successful.
5. If the update was not successful, do the following:
   a. Ssh to the failed node.
   b. Look at the update log to determine which coprocessor failed to update.
   c. Re-execute the micflash update for that particular coprocessor.
   Example: if the log shows mic1 failed flash update, do the following:
   ```
   user_prompt> /opt/intel/mic/bin/micflash -update \
   /opt/intel/mic/flash/EXT_HP2_B0_0372-1.rom.smc \
   -device 1 -noreboot
   ```
6. Repeat the Flash/SMC/Bootloader update if it fails; otherwise, reboot the system:
   ```
   user_prompt> pdsh -w node[1-4] Reboot
   ```

6 Configuring the Intel® Xeon Phi™ Coprocessor

6.1 Initialize Configuration

Configuration should already be initialized during the flash update process in the previous chapter. However, if the node is reimaged or a new coprocessor is added to the server, the configuration must be reinitialized:

```
user_prompt> micctrl --initdefaults
```

This will generate the required *default.conf* and *micX.conf* files for all detected Intel® Xeon Phi™ coprocessors.

6.2 Set up Static Bridge Configuration

This section will cover how to set up the Static Bridge configuration. Skip this section if you are setting up the DHCP configuration.

6.2.1 Modify the Intel® MPSS configuration

There are two ways to set up the configuration. The first (and recommended) option is to use **micctrl** to set up the environment:

1. Add the coprocessors to the preexisting bridge:

   ```
   user_prompt> micctrl --addbridge=br0 --type=External \
   --ip=$HostIP --netbits=24  mic0 mic1
   ```

   where $HostIP is the IP address of the br0.

2. Set up the IP for the coprocessor:

   ```
   user_prompt> micctrl --network=static --bridge=br0 \
   --ip=$MIC0IP:$MIC1IP --netbits=24 mic0 mic1
   ```

   where $MIC0IP is the IP for mic0, and $MIC1IP is the IP for mic1.

3. Set the MTU size for the coprocessor:

   ```
   user_prompt> echo "MTU=$MTUSIZE" >>  \
   /opt/intel/mic/filesystem/mic0/etc/sysconfig/network/ifcf
   g-mic0
   ```

```
user_prompt> echo "MTU=$MTUSIZE" >> \
/opt/intel/mic/filesystem/mic1/etc/sysconfig/network/ifcf
g-mic0
```

where $MTUSIZE value should be identical to the bridge’s MTU size

4. Additional step for SUSE® Linux Enterprise Server:
   Make sure to restart the network so the compute node’s network stack can reflect the changes.

```
user_prompt> service network restart
```

Alternatively, system administrators can directly modify the configuration files located in the */etc/sysconfig/mic/* directory:

1. Add the following line in the *default.conf* file, so that it has the bridge setup:
   Bridge $Bridgename External $Bridge_or_Host_IP $Netbits $MTUSize

   where:

   a. $Bridgename can be the host’s preexisting bridge name (usually set as br0).
   b. $Bridge_or_Host_IP should have the same value as what is assigned to the network bridge of choice.
   c. $Netbits value should be equivalent to the bridge’s netmask.
   d. $MTUSize value should be identical to the bridge’s MTU size.

   Example line for default.conf for node with bridge br0 with IP 192.168.0.1:

```
Bridge br0 External 192.168.0.1 24 9000
```

2. Modify the Network configuration line in the micX.conf to make sure the Static Pair configuration is changed to Static Bridge:
   Network StaticBridge $Bridgename $MICxIPAddr $Y_N_Update_etc_host

   where:

   a. $Bridgename is the bridge that is defined in the default.conf.
   b. $MICxIPAddr is the IP address for MICX (mic0, mic1,…).
   c. $Y_N_Update_etc_host is to decide whether to update the /etc/hosts.

   Example for mic0 with IP of 192.168.0.2 and choose to update the /etc/hosts:

   Original line:

```
Network StaticPair 172.31.1.1 172.31.1.254 yes 24
```

   Replace with:

```
Network StaticBridge br0 192.168.0.2 yes
```

3. Execute: `micctrl --resetconfig`
4. Set the MTU size for the coprocessor:

```
user_prompt> echo "MTU=$MTUSIZE" >>  \
/opt/intel/mic/filesystem/mic0/etc/sysconfig/network/ifcf
g-mic0
```

```
user_prompt> echo "MTU=$MTUSIZE" >> \
/opt/intel/mic/filesystem/mic1/etc/sysconfig/network/ifcf
g-mic0
```

where $MTUSIZE value should be identical to the bridge's MTU size

5. Additional step for SUSE® Linux Enterprise Server:

   Make sure to restart the network so the compute node's network stack can reflect the changes.

   ```
   user_prompt> service network restart
   ```

6.2.2 Set up Name Resolution for Static Bridge Configuration

6.2.2.1 Modify the host's /etc/hosts

All compute nodes have to be able to identify each other as well as the Intel® Xeon Phi™ coprocessors. A simple way to ensure this is to populate the entire compute node's */etc/hosts* files with all available cards' IPs as shown below:

```
127.0.0.1                localhost

192.168.0.254            CChead.cluster CChead
master.cluster master
#Xeon Servers
192.168.0.1      node1
192.168.0.4      node2
192.168.0.7      node3
192.168.0.10     node4
#MICs
192.168.0.2      node1-mic0
192.168.0.3      node1-mic1
192.168.0.5      node2-mic0
192.168.0.6      node2-mic1
192.168.0.8      node3-mic0
192.168.0.9      node3-mic1
192.168.0.11     node4-mic0
192.168.0.12     node4-mic1
```

6.2.2.2 Modify the coprocessors' /etc/hosts

Modify the following */etc/hosts* located on the host:

/opt/intel/mic/filesystem/mic0/etc/hosts (for mic0)
/opt/intel/mic/filesystem/mic1/etc/hosts (for mic1)

Example for the coprocessors' /etc/hosts (for mic0):

```
127.0.0.1       node1-mic0 mic0 localhost.localdomain
localhost
::1             node1-mic0 mic0 localhost.localdomain
localhost
192.168.0.1     host node1
192.168.0.3     mic1 node1-mic1
192.168.0.4     node2
192.168.0.5     node2-mic0
192.168.0.6     node2-mic1
192.168.0.7     node3
192.168.0.8     node3-mic0
192.168.0.9     node3-mic1
192.168.0.10    node4
192.168.0.11    node4-mic0
192.168.0.12    node4-mic1
```

6.3 Set up DHCP Configuration

This section will cover how to set up the DHCP configuration. Skip this section if setting up the Static Bridge configuration. Permanent MAC address is required to make sure the DHCP IP for the coprocessor will never change. System administrators have the option to use the default DHCP leases, however it will not be recommended since name resolution cannot be set definitively.

6.3.1 Modify the Intel® MPSS Configuration

There are two ways to set up the configuration. The first (and recommended) option is to use **micctrl** to set up the environment:

1. Add the coprocessors to the preexisting bridge:

   ```
   user_prompt> micctrl --addbridge=br0 --type=External \
   --ip=dhcp mic0 mic1
   ```

2. Set up DHCP for the coprocessor:

   ```
   user_prompt> micctrl --network=dhcp --bridge=br0 mic0 \
   mic1
   ```

3. Set the MTU size for the coprocessor:

   ```
   user_prompt> echo "MTU=$MTUSIZE" >>  \
   /opt/intel/mic/filesystem/mic0/etc/sysconfig/network/ifcf
   g-mic0
   ```

   ```
   user_prompt> echo "MTU=$MTUSIZE" >> \
   /opt/intel/mic/filesystem/mic1/etc/sysconfig/network/ifcf
   g-mic0
   ```

   where $MTUSIZE value should be identical to the bridge's MTU size

Alternatively, system administrators can directly modify the configuration files located in the */etc/sysconfig/mic/* directory:

1. Add a line in default.conf with the bridge configuration:
   Bridge $Bridgename External dhcp 24 9000

   Example:

   ```
   Bridge br0 External dhcp 24 9000
   ```

2. Modify the Network configuration line in the micX.conf to change the Static Pair configuration so that it will be redirected to the bridge instead:
   Network Bridge $Bridgename

   where:
   $Bridgename is the bridge that is defined in the default.conf

   Example:

   ```
   Network Bridge br0
   ```

4. Execute the `resetconfig` command to make sure the configuration is reconfigured based on the changes. Usage:

   ```
   user_prompt> micctrl --resetconfig
   ```

5. Set the MTU size for the coprocessor:

   ```
   user_prompt> echo "MTU=$MTUSIZE" >>  \
   /opt/intel/mic/filesystem/mic0/etc/sysconfig/network/ifcf
   g-mic0
   ```

   ```
   user_prompt> echo "MTU=$MTUSIZE" >> \
   /opt/intel/mic/filesystem/mic1/etc/sysconfig/network/ifcf
   g-mic0
   ```

   where $MTUSIZE value should be identical to the bridge's MTU size

6.3.2 Set Permanent MAC Address for the Coprocessor

Beginning with Gold Update 3, MAC address assignment for every coprocessor should be unique and permanent.

6.3.2.1 Using Pre-set MAC addresses

If you have already set up DHCP configuration before and wish to keep the assigned MAC address, complete the following steps:

1. Use the list generated in 3.3 above.

   Example of the list in a text file:

   ```
   node1-mic0      ca:7a:05:e1:52:65
   node1-mic1      76:f2:07:ff:00:3c
   node2-mic0      fe:fc:cd:15:95:8a
   node2-mic1      72:0c:4c:b7:94:86
   ```

2. Add the following line to /etc/sysconfig/mic/mic?.conf:
   "MacAddrs $MICHOSTMacAddress $MicMacAddress"

   where:

   $MICHOSTMacAddress is the MAC address for host to coprocessor. This is only used for static pair configuration, so any valid MAC address value can be applied. The best route to go is to use the Host MAC value that can be acquired from running `micctrl --config`.

   $MICMacAddress is the MAC address that the coprocessor will be using. Assigning this is important to make sure that the DHCP server knows which IP it should assign to the coprocessor.

   For example, to set the config do the following:

   echo "MacAddrs $MICHOSTMacAddress $MicMacAddress " >>
   /etc/sysconfig/mic/mic0.conf

3. Once MacAddrs is set, execute the `resetconfig` command to ensure the MAC address update for the coprocessors is reflected on the card as well.

   ```
   user_prompt> micctrl --resetconfig
   ```

6.3.3 Modify the DHCP Configuration File

Add the list of DHCP configuration for the coprocessors to the dhcp configuration file. DHCP configuration should already be available for the compute node. Configuration format should be similar to the basic DHCP setup for compute nodes. For example:

```
host mnode01-mic0.mynetwork {
  hardware ethernet  ca:7a:05:e1:52:65;
  fixed-address 192.168.0.2;
}

host mnode01-mic1. mynetwork {
  hardware ethernet 76:f2:07:ff:00:3c;
  fixed-address 192.168.0.3;
}

host mnode02-mic0. mynetwork {
  hardware ethernet fe:fc:cd:15:95:8a;
  fixed-address 192.168.0.5;
}

host mnode02-mic1. mynetwork {
  hardware ethernet 72:0c:4c:b7:94:86;
  fixed-address 192.168.0.6;
}
```

6.3.4 Modify the Name Resolution Configuration File

For every coprocessor, make sure the name resolution is set. This will remove the need to guess or dig through the message log to figure out what IP a particular coprocessor ended up getting. Setting up the name resolution is the same as how it is done for regular compute nodes.

For example:

```
node01-mic0   IN A 192.168.0.2
node01-mic1   IN A 192.168.0.3
node02-mic0   IN A 192.168.0.5
node02-mic1   IN A 192.168.0.6
```

6.4 Set up Mounted File System

1. Make sure the filesystem that needs to be mounted is accessible from the coprocessor and properly exported.
2. Edit the */opt/intel/mic/filesystem/mic0/etc/fstab* (and */opt/intel/mic/filesystem/mic1/etc/fstab* when mic1 is available) to ensure the mounted */home* filesystem is listed. For example:

```
192.168.0.254:/home /home nfs     rsize=8192,wsize=8192,nolock   0 0
```

6.5 Intel® MIC User Access

Running Intel® MPI requires user access. Intel® MPSS service will automatically set up the passwordless authentication if the rsa key for the user has been created before `service MPSS start` is invoked.

Alternatively, users can be added later by specifying the `useradd` option that is provided by the micctrl tool.

7 Starting Intel® MPSS Service

At this point, all necessary configurations should be set up and Intel® MPSS service is ready to be started.

Start the Intel® MPSS service:

```
user_prompt> service MPSS start
```

7.1 Starting OFED-MIC Service

Do this only if OFED* 1.5.4.1 and ofed-mic rpms are installed.

1. Restart the openibd service.
2. Start OFED-MIC service:

   ```
   user_prompt> service ofed-mic start
   ```

7.2 Ensure Services Are Running After Reboot

NOTE: The following should only be applied if the compute node is not being reimaged on every reboot, and all configurations are set.

To automatically run Intel® MPSS service:

```
user_prompt> chkconfig MPSS on
```

To automatically run OFED-MIC service:

```
user_prompt> chkconfig ofed-mic on
```

8 Sample Configuration and Scripts

8.1 ifcfg-br0

```
DEVICE=br0
TYPE=Bridge
ONBOOT=yes
DELAY=0
NM_CONTROLLED="no"
MTU=9000
BOOTPROTO=dhcp
NOZEROCONF=yes
```

8.2 ifcfg-eth0

```
DEVICE=eth0
ONBOOT=yes
BRIDGE=br0
MTU=9000
```

8.3 Hostlist

```
192.168.0.1      node1
192.168.0.2      node1-mic0
192.168.0.3      node1-mic1
192.168.0.5      node2
192.168.0.6      node2-mic0
192.168.0.7      node2-mic1
192.168.0.9      node3
192.168.0.10     node3-mic0
192.168.0.11     node3-mic1
192.168.0.13     node4
192.168.0.14     node4-mic0
192.168.0.15     node4-mic1
```

8.4 OFED* 1.5.4.1 Custom Configuration Answer

```
kernel-ib=y
core=y
mthca=y
mlx4=y
mlx4_en=y
ipoib=y
sdp=y
srp=y
srpt=y
rds=y
kernel-ib-devel=y
libibverbs=y
libibverbs-devel=y
libibverbs-devel-static=y
libibverbs-utils=y
libibverbs-debuginfo=y
libmthca=y
libmthca-devel-static=y
libmthca-debuginfo=y
libmlx4=y
libmlx4-devel=y
libmlx4-debuginfo=y
libcxgb3=n
libcxgb3-devel=n
libcxgb3-debuginfo=n
libcxgb4=n
libcxgb4-devel=n
libcxgb4-debuginfo=n
libnes=n
libnes-devel-static=n
libnes-debuginfo=n
libipathverbs=y
libipathverbs-devel=y
libipathverbs-debuginfo=y
libibcm=y
libibcm-devel=y
libibcm-debuginfo=y
libibumad=y
libibumad-devel=y
libibumad-static=y
libibumad-debuginfo=y
libibmad=y
libibmad-devel=y
libibmad-static=y
libibmad-debuginfo=y
```

```
ibsim=y
ibsim-debuginfo=y
ibacm=y
librdmacm=y
librdmacm-utils=y
librdmacm-devel=y
librdmacm-debuginfo=y
libsdp=y
libsdp-devel=y
libsdp-debuginfo=y
opensm=n
opensm-libs=n
opensm-devel=n
opensm-debuginfo=n
opensm-static=n
compat-dapl=y
compat-dapl-devel=y
dapl=y
dapl-devel=y
dapl-devel-static=y
dapl-utils=y
dapl-debuginfo=y
perftest=y
mstflint=y
sdpnetstat=y
srptools=y
rds-tools=y
rds-devel=y
ibutils=y
infiniband-diags=y
qperf=y
qperf-debuginfo=y
ofed-docs=y
ofed-scripts=y
infinipath-psm=y
infinipath-psm-devel=y
mpi-selector=y
mvapich_gcc=y
mvapich2_gcc=y
openmpi_gcc=y
mpitests_mvapich_gcc=y
mpitests_mvapich2_gcc=y
mpitests_openmpi_gcc=y
build32=0
prefix=/usr
mvapich2_conf_impl=ofa
mvapich2_conf_romio=1
mvapich2_conf_shared_libs=1
```

```
mvapich2_conf_ckpt=0
mvapich2_conf_vcluster=small
```

8.5 Pre-config Script for Static Bridge Configuration

```
/*
* Copyright (C) 2013 Intel Corporation
*
* Intel makes no warranty of any kind regarding this code. This
* code is licensed on an "AS IS" basis and Intel will not provide
* any support, assistance, installation, training,  or other
* services. Intel may not provide any updates, enhancements or
* extensions to this code. Intel specifically disclaims any
* warranty of merchantability, non-infringement, fitness for any
* particular purpose, or any other warranty. Intel disclaims all
* liability, including liability for infringement of any
* proprietary rights, relating to use of the code. No license,
* express or implied, by estoppel or otherwise, to any
* intellectual property rights is granted herein.
*/

rm -rf /etc/sysconfig/network-scripts/ifcfg-eth0
cp -f /donotremove/ifcfg-eth0 /etc/sysconfig/network-scripts/
cp -f /donotremove/ifcfg-br0 /etc/sysconfig/network-scripts/
micctrl --initdefaults
service network restart
sleep 5

Value=`ifconfig br0 | grep "inet addr" | gawk -F: '{print $2}' |
gawk '{print $1}' | awk '{print substr($0,11,4)}'`
HostIP=`ifconfig br0 | grep "inet addr" | gawk -F: '{print $2}' |
gawk '{print $1}'`
IPQuad=`ifconfig br0 | grep "inet addr" | gawk -F: '{print $2}' |
gawk '{print $1}' | awk '{print substr($0,0,9)}'`
newvalue1=`expr $Value + 1`
newvalue2=`expr $Value + 2`
micctrl --addbridge=br0 --type=External --ip=$HostIP mic0 mic1
micctrl --network=static --bridge=br0 --
ip=$IPQuad.$newvalue1:$IPQuad.$newvalue2 mic0 mic1
echo "MTU=9000" >>
/opt/intel/mic/filesystem/mic0/etc/sysconfig/network/ifcfg-mic0
echo "MTU=9000" >>
/opt/intel/mic/filesystem/mic1/etc/sysconfig/network/ifcfg-mic0
service mpss start
```

8.6 Pre-config Script for DHCP for New Setup

```
/*
* Copyright (C) 2013 Intel Corporation
*
* Intel makes no warranty of any kind regarding this code. This
* code is licensed on an "AS IS" basis and Intel will not provide
* any support, assistance, installation, training,  or other
* services. Intel may not provide any updates, enhancements or
* extensions to this code. Intel specifically disclaims any
* warranty of merchantability, non-infringement, fitness for any
* particular purpose, or any other warranty. Intel disclaims all
* liability, including liability for infringement of any
* proprietary rights, relating to use of the code. No license,
* express or implied, by estoppel or otherwise, to any
* intellectual property rights is granted herein.
*/
rm -rf /etc/sysconfig/network-scripts/ifcfg-eth0
cp -f /donotremove/ifcfg-eth0 /etc/sysconfig/network-scripts/
cp -f /donotremove/ifcfg-br0 /etc/sysconfig/network-scripts/
micctrl --initdefaults
service network restart
sleep 5
echo "master:/home /home nfs     rsize=8192,wsize=8192,nolock,
intr    0 0" >> /opt/intel/mic/filesystem/mic0/etc/fstab
echo "master:/home /home nfs     size=8192,wsize=8192,nolock,
intr    0 0" >> /opt/intel/mic/filesystem/mic1/etc/fstab
micctrl --resetconfig
echo "MTU=9000" >>
/opt/intel/mic/filesystem/mic0/etc/sysconfig/network/ifcfg-mic0
echo "MTU=9000" >>
/opt/intel/mic/filesystem/mic1/etc/sysconfig/network/ifcfg-mic0
service mpss start
service openibd restart
service ofed-mic start
```

8.7 Pre-config Script for DHCP for Pre-existing Setup

```
/*
* Copyright (C) 2013 Intel Corporation
*
* Intel makes no warranty of any kind regarding this code. This
* code is licensed on an "AS IS" basis and Intel will not provide
* any support, assistance, installation, training,  or other
* services. Intel may not provide any updates, enhancements or
* extensions to this code. Intel specifically disclaims any
* warranty of merchantability, non-infringement, fitness for any
* particular purpose, or any other warranty. Intel disclaims all
* liability, including liability for infringement of any
* proprietary rights, relating to use of the code. No license,
* express or implied, by estoppel or otherwise, to any
* intellectual property rights is granted herein.
*/
rm -rf /etc/sysconfig/network-scripts/ifcfg-eth0
cp -f /root/ifcfg-eth0 /etc/sysconfig/network-scripts/
cp -f /root/ifcfg-br0 /etc/sysconfig/network-scripts/
micctrl --initdefaults
service network restart
sleep 5
micctrl --addbridge=br0 --type=External --ip=dhcp mic0 mic1
micctrl --network=dhcp --bridge=br0  mic0 mic1
hostname=`hostname`
mic0hostmac=`micctrl --config mic0 | grep -i host | grep -i mac |
gawk '{print $3}'`
mic1hostmac=`micctrl --config mic1 | grep -i host | grep -i mac |
gawk '{print $3}'`
mic0mac=`cat /donotremove/mnodemac.txt | grep -i $hostname-mic0 |
awk '{print $2}'`
mic1mac=`cat /donotremove/mnodemac.txt | grep -i $hostname-mic1 |
awk '{print $2}'`
echo "MacAddrs $mic0hostmac $mic0mac" >>
/etc/sysconfig/mic/mic0.conf
echo "MacAddrs $mic1hostmac $mic1mac" >>
/etc/sysconfig/mic/mic1.conf
micctrl --resetconfig
micctrl -w
echo "master:/home /home nfs
rsize=8192,wsize=8192,nolock,intr       0 0" >>
/opt/intel/mic/filesystem/mic0/etc/fstab
echo "master:/home /home nfs
rsize=8192,wsize=8192,nolock,intr       0 0" >>
/opt/intel/mic/filesystem/mic1/etc/fstab
```

```
echo "MTU=9000" >>
/opt/intel/mic/filesystem/mic0/etc/sysconfig/network/ifcfg-mic0
echo "MTU=9000" >>
/opt/intel/mic/filesystem/mic1/etc/sysconfig/network/ifcfg-mic0
service mpss start
micctrl -w
service openibd restart
service ofed-mic start
```

8.8 Upload Intel® MPI binary sample (optional)

Use this only if the necessary binaries are not available in the card as part of the mounted filesystem.

```
/*
* Copyright (C) 2013 Intel Corporation
*
* Intel makes no warranty of any kind regarding this code. This
* code is licensed on an "AS IS" basis and Intel will not provide
* any support, assistance, installation, training,  or other
* services. Intel may not provide any updates, enhancements or
* extensions to this code. Intel specifically disclaims any
* warranty of merchantability, non-infringement, fitness for any
* particular purpose, or any other warranty. Intel disclaims all
* liability, including liability for infringement of any
* proprietary rights, relating to use of the code. No license,
* express or implied, by estoppel or otherwise, to any
* intellectual property rights is granted herein.
*/

#!/bin/sh
# Copy the pmi_proxy binary and libmpi.so to each MIC card in
# prep for running an MPI program

MPI_DIR=/opt/intel/impi/4.1.0.024

for ip in 192.168.0.2 192.168.0.3
 do
 {
   scp ${MPI_DIR}/mic/bin/pmi_proxy  ${ip}:/bin/pmi_proxy
   scp ${MPI_DIR}/mic/lib/libmpi.so.4.1  ${ip}:/lib64/libmpi.so.4
   scp ${MPI_DIR}/mic/bin/IMB-MPI1  ${ip}:/bin/IMB-MPI1
   scp ${MPI_DIR}/mic/lib/libmpigf.so.4.1
${ip}:/lib64/libmpigf.so.4
   scp ${MPI_DIR}/mic/lib/libmpigc4.so.4.1
${ip}:/lib64/libmpigc4.so.4
   # Finally copy the actual MIC binary you'd like to run
 }
 done
```