Notes regarding Hardware Rebuilds using Rocky Linux 9.3 - 4/18/2024

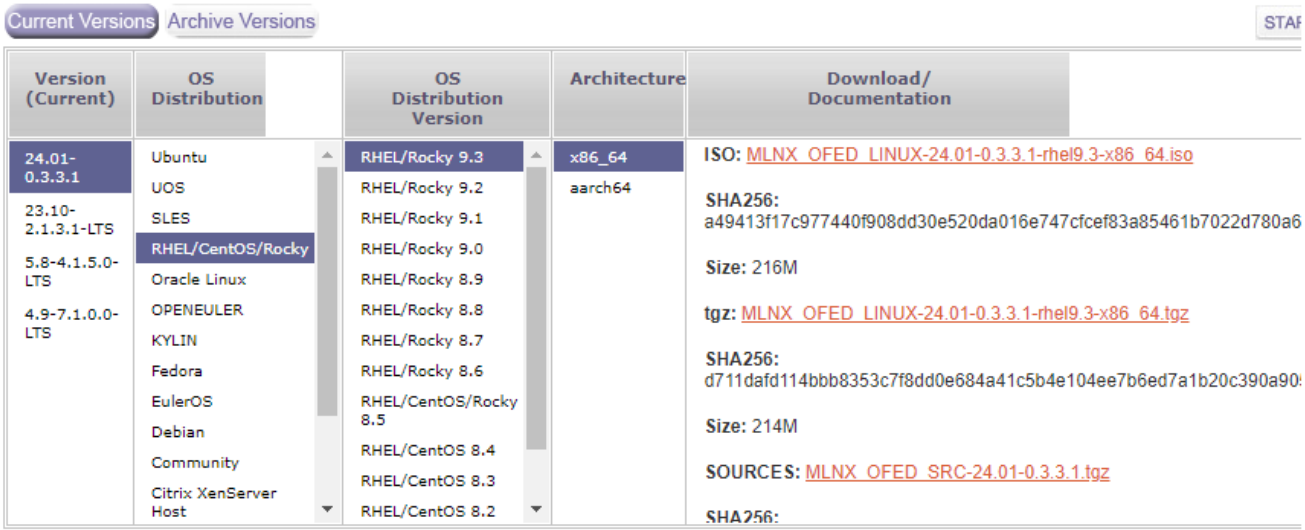
1) Install the OS
   a) https://rockylinux.org/download/
   b) Make sure you select the DVD version – it has most of the packages
   c) Use Rufus to burn the ISO to a thumb drive
2) During the setup menu, please select the following packages from the Software Selection Menu
   a) Debugging Tools
   b) FTP Server
   c) Performance Tools
   d) Remote Desktop Clients
   e) Remote Management for Linux
   f) Console Internet Tools
   g) Development Tools
   h) .NET Development
   i) Graphical Admin Tools
   j) RPM Development Tools
   k) System Tools
3) Set up the root password. Click allow SSH login with password
4) Create sfi as the additional admin account during the install procedure.
   a) Leave full name empty
   b) User name: sfi
   c) Click make this user administrator
5) Set the hostname under network & hostname menu
   a) Bottom left: Hostname should be sm-node-## (replace ## with the sm-node number you're working on)
6) Select the drive (not USB key) as installation destination
   a) Click the drive and then select done in the top left
   b) It may complain about insufficient space. If so, follow steps below
   c) Click reclaim space
   d) Click delete all
   e) Click reclaim space again

Add sfi account to sudoers file

1) Open a terminal
2) sudo visudo /etc/sudoers
3) Add a line under root ALL=(ALL) ALL
   a) sfi ALL=NOPASSWD: ALL
4) Since we granted sfi admin privileges, it also gets added to the wheel group. Give this group NOPASSWD permission
   a) Find the line %wheel = ALL=(ALL) ALL and put hash in front so that it is commented out
   b) Remove hash (#) from %wheel = ALL=NOPASSWD: ALL

Install additional packages needed for MLNX_OFED

1) sudo yum install perl-sigtrap
2) sudo yum install createrepo
3) sudo yum install kernel-rpm-macros
1) sudo yum install tk gcc-gfortran
2) sudo yum install apr-util – needed for CTSD
3) sudo yum install numactl-devel – needed for Intel IMB
4) Download the latest MLNX_OFED TGZ file for the Rocky Linux version you installed



5) Extract the TGZ file: tar zxvf MLNX_OFED_LINUX-24.01-0.3.3.1-rhel9.3-x86_64.tgz
6) Change to the MLNX_OFED Directory and execute the install script
   a) sudo ./mlnxofedinstall --add-kernel-support
7) At the end of the install, it will ask you to issue the following commands:
   a) sudo dracut -f
   b) sudo /etc/init.d/openibd restart
   c) I usually just reboot the server

Installing OpenMPI 5.0.2 - 3/18/2024 updated 3/19/2024

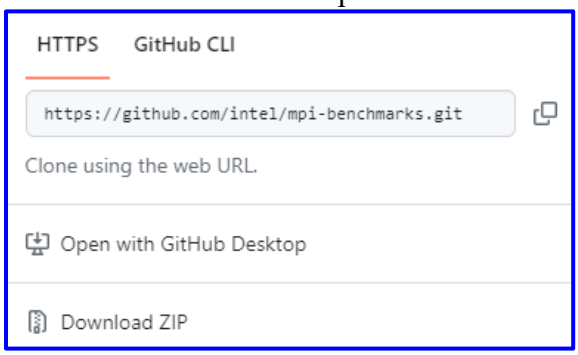
1) https://docs.open-mpi.org/en/v5.0.x/installing-open-mpi/quickstart.html
2) Open a terminal and type cd Downloads
3) Type wget "https://download.open-mpi.org/release/open-mpi/v5.0/openmpi-5.0.2.tar.gz" (include parentheses)
4) tar -xvf openmpi-5.0.2.tar.gz
5) cd openmpi-5.0.2
6) ./configure –with -ucx
7) make all
8) sudo make install

Installing Intel IMB

Installing Intel IMB - 10/6/2024

1) Go to Intel IMB: https://github.com/intel/mpi-benchmarks
2) Next click on the Code button which will bring you to a download.

3) Click on the Download Zip and save it to the directory /home/sfi/Downloads



4) Issue the command unzip mpi-benchmarks-master.zip
5) Go into the folder mpi-benchmarks-master
6) Go into src_c folder and make all
   a) Copy the resulting executable IMB-MPI1 to /usr/local/bin
      i) sudo cp IMB-MPI1 /usr/local/bin/
      ii) Note that MPI1 is the letter I and then 1
7) Additional tests to show everything needed is working.
   a) echo $PATH
   b) mpirun --version ➔ Open MPI 5.0.2
   c) mpicc -show ➔ this will show the path and specifically the path for gcc compiler
      i) gcc -I/usr/local/include -L/usr/local/lib -Wl,-rpath -Wl,/usr/local/lib -Wl,--enable-new-dtags -lmpi
8) If you make a **mistake** and need to reinstall
   a) cd mpi-benchmarks-master/src_c
      i) sudo make clean
      ii) sudo make uninstall
   b) cd openmpi-5.0.2
      i) make clean
   c) Delete the original folder where you installed Open MPI (not the downloaded tarball)
9) For installation help, see https://www.intel.com/content/www/us/en/docs/mpi-library/user-guide-benchmarks/2021-2/overview.html

Configuring the system prior to running the Perl Script

1) SELinux is on by default.
   a) You must disable it. Edit the file /etc/selinux/config and set SELINUX=disabled
   b) sudo vim config
   c) SELINUX=enforcing -> SELINUX=disabled

2) You must disable the firewalld service. Issue the command sequence
   a) sudo systemctl stop firewalld
   b) sudo systemctl mask firewalld
   c) sudo systemctl status firewalld
   d) You can restart the firewalld service by running the commands “sudo systemctl unmask firewalld” and then “sudo systemctl start firewalld”

3) Creating folders and files
   a) Create a data directory: sudo mkdir /data
   b) Create the file master_node_list and put this file in the /data directory. This file contains all the possible hosts you may run against. The file includes only host names, and not domains.
      i) For example: sm-node-01 sm-node-02 sm-node-03 sm-node-04 (one per line).
   c) Create the file current-cluster in the user’s home directory.
      i) This file contains a subset of the host names from the file /data/master_node_list that you wish to test against.
   d) Create the file mpi-hosts and put this in user’s home directory.
      i) This file contains the number of instances you want the MPI script to run.
         sm-node-01-ib
         sm-node-01-ib
         sm-node-01-ib
         sm-node-01-ib
         sm-node-02-ib
         sm-node-02-ib
         sm-node-02-ib
         sm-node-02-ib

Configuring persistent static IP addresses for IPoIB

1) cd /etc/NetworkManager/system-connections
2) nmcli con edit (this brings up an interactive network connection configuration tool in the command line)
3) enter the connection type as infiniband
4) Enter the command: goto infiniband

```
[sfi@sm-node-20 system-connections]$ nmcli con edit
Valid connection types: 6lowpan, 802-11-olpc-mesh (olpc-mesh), 802-11-wireless (
wifi), 802-3-ethernet (ethernet), adsl, bluetooth, bond, bridge, cdma, dummy, ge
neric, gsm, infiniband, ip-tunnel, loopback, macsec, macvlan, ovs-bridge, ovs-dp
dk, ovs-interface, ovs-patch, ovs-port, pppoe, team, tun, veth, vlan, vpn, vrf,
vxlan, wifi-p2p, wimax, wireguard, wpan, bond-slave, bridge-slave, team-slave
Enter connection type: infiniband
```

5) Set the mac address to the hardware address of the IB port you are going to use/test
   a) Find the hardware address by entering ip addr in a separate terminal

```
4: ibp65s0f0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 2044 qdisc mq state DOWN g
roup default qlen 256
    link/infiniband 00:00:10:49:fe:80:00:00:00:00:00:00:98:03:9b:03:00:bd:49:4c
brd 00:ff:ff:ff:ff:12:40:1b:ff:ff:00:00:00:00:00:00:ff:ff:ff:ff
```

   b) You will need to copy all the hex numbers (and colons) in between link/infiniband up to brd
   c) Back in the nmcli command line tool enter "**goto infiniband**" without the quotes. This will change the command prompt to "nmcli infiniband"
   d) Enter "**set mac-address**" followed by pasting in your 20 byte hardware address
6) Enter the command **back** – this gets you back out of the infiniband configuration
7) Enter command **goto ipv4**
   a) Set addresses to whatever static IP address you want, for example: **set ipv4.addresses 192.168.XXX.YYY/24**. The server and the DUT must have unique IP addresses.
8) Set method manual. Example: **set ipv4.method manual**
9) Enter the command **back** (to back out of the ipv4 configuration)
10) Save and the quit

Final Steps to configure IPoIB

1) sudo vi infiniband.nmconnection
   a) Change the id=infiniband to id="your interface name" (for example ibp65s0f0)
2) Your.nmconnection file will be named infiniband.nmconnection
   a) Rename it using sudo mv infiniband.nmconnection followed by whatever you want to rename it.
   b) For example: sudo mv infiniband.nmconnection ibp65s0f0.nmconnection
   c) You should rename it to match the interface name that the system has assigned when you type ip addr

Final Steps before running the Interop Scripts

5) Modify /etc/hosts - add lines for all the hosts you want to test
   a) The 10.20 addresses are for the system ethernet ports
   b) The 192.168 addresses are for the IB ports you assigned addresses in the steps above

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.20.1.101     sm-node-01
10.20.1.102     sm-node-02
10.20.1.103     sm-node-03
10.20.1.104     sm-node-04

192.168.30.101  sm-node-01-ib
192.168.30.102  sm-node-02-ib
192.168.30.103  sm-node-03-ib
192.168.30.104  sm-node-04-ib
```

SSH Key Exchange

It is best not to use root account. All the systems must be set up with at least one identical user account. The account must be able to ssh to all systems from the system which launches the Open MPI tests.

1) Connect the IB cards directly or through an IB switch and start opensm
2) Type the following command and accept all the defaults
   a) sudo ssh-keygen
3) Copy the file generated in .ssh to the other machines you want to share with. Below we assume you want nodes sm-node-01 and sm-node-02 and their IB nodes to share the keys
   a) sudo ssh-copy-id -i .ssh/id_rsa.pub sfi@sm-node-01
   b) sudo ssh-copy-id -i .ssh/id_rsa.pub sfi@sm-node-01-ib
   c) sudo ssh-copy-id -i .ssh/id_rsa.pub sfi@sm-node-02
   d) sudo ssh-copy-id -i .ssh/id_rsa.pub sfi@sm-node-02-ib
4) If you have more than the two servers as in this example, then you would follow steps (2) and (3) (need to verify this) for all the servers you want to use.
   a) There is also a script to automate this process which is easier if you have 3 or more servers.

exchange-keys.sh

5) Test each one of your servers with the command ssh root@sm-node-* so your keys get registered

Executing the Perl Script

1) Copy the perl script from the home directory of an existing working interop server
2) **Command syntax**: perl ibta-mpi_script<date>.pl openmpi b 14 “someFolderName”
3) perl ibta-mpi_script_2024-03-20 openmpi b 25 “Nvidia/DUT-43-337”
   a) There are two test plans (a and b) – you can ignore this and just use b which is Intel IMB
   b) Speed is 14 (FDR), 25 (EDR), 50 (HDR) etc.

ibta-mpi_script_2024-04-02.pl

Nvidia Utilities

Get device Info

1) mst start – this is a Mellanox Service
2) mst status – this will show the device info and path

```
[root@sm-node-20 ~]# mst status
MST modules:
------------
    MST PCI module is not loaded
    MST PCI configuration module loaded

MST devices:
------------
/dev/mst/mt4129_pciconf0         - PCI configuration cycles access.
                                   domain:bus:dev.fn=0000:41:00.0 addr.reg=88 data.reg=92 cr_bar.gw_offset=-1
                                   Chip revision is: 00
```

3) mlxlink -d /dev/mst/mt4129_pciconf0 -p 1 -m > mlxlink.txt 2>&1
   a) This will provide all the stats about the device
   b) This is how you redirect both stdout and stderr to a file
      i) > filename.txt 2>&1