InfiniBand Trade Association

RDMA over Converged Ethernet (RoCE) Interoperability

Method of Implementation (MOI)

This is confidential material. The document discusses the methods currently used by the IBTA to ensure that devices interoperate. It should not be shared with non-IBTA members.

Table of Contents

Contents

Revision History

Revision	Date	Author	Comments
1.0.00	2017-09-15	Llolsten Kaonga	• Initial version of the RoCE Interop MOI update
			•

Introduction

This document describes the RDMA over Converged Ethernet (RoCE) interoperability (RoCE Interop) testing procedure for a RoCE fabric. The procedure uses the Intel® MPI Benchmarks (IMB) to test the point-to-point and fabric-wide operations for a variety of message sizes. The procedure utilizes Open MPI for the message passing protocol. The IMB tests include PingPong, Gather, Sendrecv, Scatter, Allreduce, Alltoall, Allgatherv. For a more complete list of the benchmarks, please check the Intergrators' List.

The procedure described in this document is used at all InfiniBand® Trade Association (IBTA) Plugfests. These plugfests are held twice a year, in spring, and fall.

Return to Top

System Configuration

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 launched the Open MPI tests. This means that the ssh host keys must already be cached. Please see the section below for details on ssk-key exchange.

1. CentOS 6.8 or newer. Please only have one RoCE card in each server. If you have more than one card, the IP address assignments may be assigned incorrectly, and it can take quite a bit of time to resolve this. Make sure you configure the host name. Change the *localhost.localdomain* under the network set up to something like *sm-node-** where '*' is the node number.
2. OFED 3.18 or newer. It is recommended that you have the same version of OFED installed in the same file system location on all systems.
3. Open MPI 1.10.2 or newer. You must have the same version of Open MPI in the same file system location on all systems.
4. Intel® IMB_2018 or newer. To download, you need to scroll down the page, and click the Intel MPI Benchmarks GitHub Repository button then click the "Clone or Download" drop down to download the latest version.

Open MPI Installation

1. cd to the location where you unpacked the Open MPI download
2. Invoke the command ***./configure***
3. Invoke the command ***make all install***

You can verify the version of Open MPI with the command '***mpirun –version***' or '***ompi_info –version***'

mpirun command

As of Open MPI 1.8.2, you must include the mpirun command option "--allow-run-as-root". Without this option, mpirun will abort when you attempt to run Open MPI as root.

Intel® IMB Installation

1. cd to the '*src*' location directory of the location where you unpacked the Intel MPI Benchmark tarball.
2. Open the *make_ict* file and change line 3 from ***CC = mpiicc*** to ***CC = mpicc***.
3. While still in the '*src*' directory, invoke the command '***make all***'. If you get the error:
   "mpiCC: error while loading shared libraries …"
   a. Run the command "*ldconfig"*
   b. Run *make all* again
4. Copy the IMB-MPI1 file which has just been built to the directory ***/usr/local/bin***.

Lockable Memory Limits

The lockable memory limits in each system must be set to allow unlimited locked memory per process. This means that you must edit the ***/etc/security/limits.conf*** file so the following two lines read thus:

* hard memlock unlimited ← the (*) is not a bullet! It is part of the entry in the file
* soft memlock unlimited ← the (*) is not a bullet! It is part of the entry in the file

SELinux

SELinux is on by default in RHEL based 7.x distributions. You must disable it. Edit the file ***/etc/selinux/config*** and set *SELINUX=disabled*

Disable the Firewall Daemon

You must disable the ***firewall*** service in RHEL based 7.x distributions. Issue the command sequence

1. systemctl stop firewalld
2. systemctl mask firewalld
3. systemctl status firewalld

The last command just checks to confirm that the firewalld daemon is really disabled. You should see something similar to the following:

```
● firewalld.service
   Loaded: masked (/dev/null)
   Active: inactive (dead) since Mon 2016-05-09 10:12:03 EDT; 1h 1min ago
 Main PID: 1075 (code=exited, status=0/SUCCESS)

May 09 10:01:56 sm-node-7 systemd[1]: Starting firewalld - dynamic firewall daemon...
May 09 10:01:56 sm-node-7 systemd[1]: Started firewalld - dynamic firewall daemon.
May 09 10:12:02 sm-node-7 systemd[1]: Stopping firewalld - dynamic firewall daemon...
May 09 10:12:03 sm-node-7 systemd[1]: Stopped firewalld - dynamic firewall daemon.
May 09 10:47:37 sm-node-7 systemd[1]: Cannot add dependency job for unit firewalld.service, ignoring: Unit firewalld.service is masked.
May 09 10:49:30 sm-node-7 systemd[1]: Cannot add dependency job for unit firewalld.service, ignoring: Unit firewalld.service is masked.
May 09 10:54:57 sm-node-7 systemd[1]: Cannot add dependency job for unit firewalld.service, ignoring: Unit firewalld.service is masked.
May 09 10:55:01 sm-node-7 systemd[1]: Cannot add dependency job for unit firewalld.service, ignoring: Unit firewalld.service is masked.
May 09 11:13:22 sm-node-7 systemd[1]: Cannot add dependency job for unit firewalld.service, ignoring: Unit firewalld.service is masked.
```

Failure to disable the firewalld service will result in the Open MPI script will returning a “Broken Pipe” error. Since you are stopping the firewalld daemon, it may be best to set up your test system on an isolated subnet, one that is not connected to the internet or any other network.

You can restart the firewalld service at any time by running the commands “*systemctl unmask firewalld*” and then “*systemctl start firewalld*”

iptables

Some older versions of Linux still use iptables as a firewall. Do one of the following to resolve it:

1. Configure iptables to pass traffic
    a. iptables -I INPUT 1 -I <interface_name> -p tcp -m tcp -j ACCEPT
    b. iptables -I OUTPUT 1 -o <interface_name> -p tcp -m tcp -j ACCEPT

    <interface_name> is the name of the network interface as shown in *ip addr* command

2. Stop the iptables service and stop it from starting on reboot, repeat for IPv6.
    a. service iptables stop
    b. chkconfig –levels 345 iptables off
    c. service ip6tables stop
    d. chkconfig –levels 345 ip6tables off

Create Open MPI Hostfile

Create the file *mpi-hosts-ce* and put this in the user's home directory. This file contains the hostnames of all the nodes that you want included in your Open MPI/IMB job. So, it contains a series of entries like this (not numbered, and one per line)

1. sm-node-1-ce
2. sm-node-2-ce
3. sm-node-3-ce
4. sm-node-4-ce

Configure IPoIB

Add ***/etc/sysconfig/network-scripts/ifcfg-eth**** and configure it using the MAC address of the RoCE card you plan to use in the server. You may need to avoid quotes as some operating systems (eg RH 6.4) are not very tolerant of them. You can use IP addresses reserved for private internets (RFC1918) such as 192.168.30.x or 10.0.0.x, where 'x' is the last entry of the corresponding Ethernet IP address of the server assigned by the DHCP server. Please make sure you use the correct MAC address for the RNIC.

Note: It is important to create the file /etc/sysconfig/network-scripts/ifcfg.eth* or update an existing ifcfg.eth* file with the correct MAC address *before* you insert the new RoCE card. If you miss this, then do it as soon as possible after you install the new RoCE card. This is done to ensure that the server has the correct IP address for the RoCE card. Failure to follow this action may result in the server providing conflicting IP information or no IP information at all. The system may also re-assign an IP address you may have wanted to use to some other device on your server. Once you get these errors, it can take a long time to resolve.

Modify /etc/hosts

You need to add lines for all the hosts you want to test, assuming you do not have DNS

Example hosts file for a network without DNS. You would have similar entries like the last two below for each server.

```
127.0.0.1       localhost localhost.localdomain localhost4 localhost
::1             localhost.localdomain6  localhost6
10.20.0.211     sm-node-1
10.20.0.212     sm-node-2
192.168.5.211   sm-node-1-ce
192.168.5.212   sm-node-2-ce
```

Configure Host Name

This assumes there is no DNS.

1. CentOS 6.x
    a. Use the utility system-config-network
    b. Go to the DNS tab and change the name from *localhost.localdomain* to a host name and domain name
    c. Save your changes and restart the network with the command */etc/init.d/network restart*
2. CentOS 7.x
    a. You can change the *localhost.localdomain* to a host name and domain name during the OS installation when you configure eth0, or
    b. Go to **Applications→System Tools→Settings→Details**. Change the *Device Name* under **Overview**

3. SLES 1x
    a. You can change the *localhost.localdomain* to a host name and domain name during the OS installation.
    b. Using YaST access the Network Settings applet to change system hostname

SSH Key Exchange

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. This means that the ssh host keys must already be cached.

1. Connect the RNICs to the fabric
2. Type the following command and accept all the defaults
    a. ssh-keygen
3. Do a copy to the other machine you want to share with. Below we assume you want nodes sm-node-1 and sm-node-2 and their RoCE interfaces to share the keys
    a. ssh-copy-id -i .ssh/id_rsa.pub root@sm-node-1
    b. ssh-copy-id -i .ssh/id_rsa.pub root@sm-node-1-ce
    c. ssh-copy-id -i .ssh/id_rsa.pub root@sm-node-2
    d. ssh-copy-id -i .ssh/id_rsa.pub root@sm-node-2-ce

    Repeat (a) – (d) for each server in your fabric.
4. Test each one of your servers with the command *ssh root@sm-node-** so your keys get registered

Return to Top

Switch Configuration

If you are using a managed Mellanox InfiniBand switch, you can change the flow control, link speed and other fields via the command line or through the web interface. We show how to change the flow control and link speed using the web interface first and then how to do the same thing from the serial .

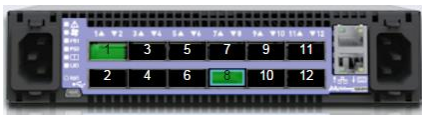
Web Interface

Get the switch's IP address (via Tera Term, or PuTTY for example), and login to the switch's web interface.

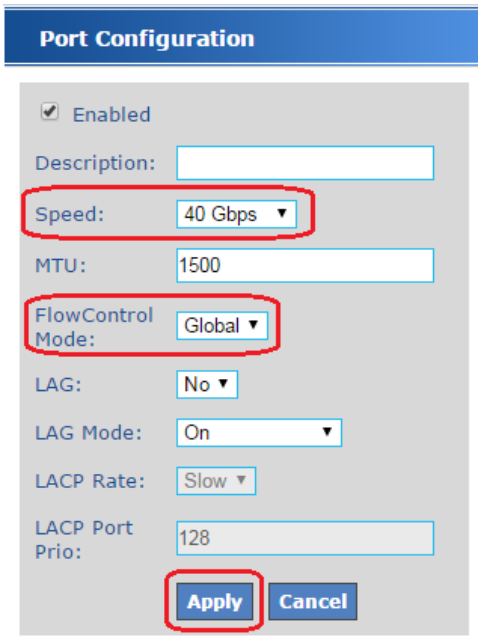
1. Open the web browser and connect to the switch by entering the IP address assigned to the switch.
2. Log in

3. Click on the **Ports** tab.

4. On the image of the switch, click the port you would like to configure.



5. Scroll down to the Port Configuration section of the page and change the link speed via the interface as shown in the screen below. Change the **FlowControl Mode** to **Global**, then click **Apply**. Set MTU to 9000. We choose 9000 but it only has to be large enough for 4096 + OS padding.



6. Repeat steps 4 and 5 for all the ports you will need for testing, including the ones used by control cables.

Make sure that you save your changes – the save button is in the top right corner. If you do not save your changes, then the settings will need to be re-enabled after you reboot the switch. These steps apply to all managed switches.

Command Line

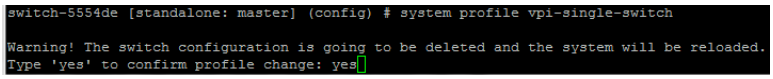
As indicated above, you can change all these fields via the console command line if you connect to the switch through the console (Tera Term or putty). Please remember that for the MSN2700, you need to set the correct baud rate (to 115200). This is somewhere around page 29 of the link. After selecting the correct COM port and connecting to the switch, do the following (for the *speed* command, see page 143 of the MLNX-OS User Manual)

1. Login to the switch
2. *en*[able]
3. *con*[figure] *t*[erminal]
4. *speed 1000* [Mbps]
   "*no speed*" sets the interface to its default speed
5. *interface ethernet 1/*<port num> *flowcontrol receive|send on|off force*

MSX6036G

The first 8 ports on the MSX6036G switch are, by default, set to Ethernet mode. You can easily switch all the ports to InfiniBand and/or switch the first 8 ports to Ethernet if they are in IB mode.

1. Follow the first three steps in the Command Line section above
2. Invoke the command *system profile vpi-single-switch*
3. Type "*yes*" on the next prompt and hit Enter



4. The switch will change the first 8 ports to Ethernet and the rest to IB and reboot.
5. You can switch all the ports back to (non-default) IB mode with the command *system profile ib-single-switch*

Return to Top

RNIC Configuration

OFED 4.8-1 will already contain drivers for Broadcom and Cavium (QLogic). If you are using OFED 4.8-1 or later you will be able to skip these instructions. If a specific vendor's device drivers are not installed with OFED, then please see the vendor specific sections below for instructions on how to install the drivers.

Broadcom RNICs Configuration

This section explains how to install Broadcom RNICs drivers if they are not included in OFED.

1. Unpack the driver package (tar -xvzf *.gz)
2. cd to package folder
3. Run the command ./install.sh
4. Reboot the server
5. Execute the driver load script from the package folder
    a. ./load.sh
6. Set the RNIC speed
    a. ethtool -s eth* speed 40000/50000

For ***sfi_perf***, please call it with the option "-I 96". This means that you would execute the command "***sfi_perf -I 96***". This changes the maximum inline data size to 96 bytes, the maximum supported by Broadcom.

RDMA perftest settings: for example: ib_write_bw

1. Default settings work for Broadcom to Broadcom
2. The default GID for Mellanox RNICs is RoCE v2. Broadcom does not support this, so the default settings will not work with Broadcom to Mellanox connections.

Return to Top

Cavium RNICs Configuration

This section explains how to install Cavium RNICs drivers if they are not included in OFED. This covers the installation and usage of the following Cavium RoCE cards

1. 25 Gb Intelligent Ethernet Adapters:
    a. QL45211HLCU-CK/SP/BK
    b. QL45212HLCU-CK/SP/BK
2. 40 Gb Intelligent Ethernet Adapters:
    a. QL45411HLSR-CK/SP/BK
    b. QL45412HLSR-CK/SP/BK
    c. QL45411HLCU-CK/SP/BK
    d. QL45412HLCU-CK/SP/BK
3. 100 Gb Intelligent Ethernet Adapters:
    a. QL45611HLSR-CK/SP/BK
    b. QL45611HLCU-CK/SP/BK
4. Driver Installation –
    a. Unzip the driver package
    b. cd into the package folder and **read** the README file
    c. Run the *make install* command
        i. If the command fails with *missing /usr/src/compat-rdma-3.18/compat/config.h file*
            1. Run the *configure* script located in /usr/src/compat-rdma-3.18/ directory
        ii. Run *make install* from the package directory again
    d. cd into *libqedr-<version>* directory
        i. **Read** the README file
        ii. The prerequisites of the README should not be applicable if you already have OFED installed since the libibverbs library is part of the OFED installation
        iii. Run the command

            ./configure --prefix=/usr --libdir=${exec_orefix}/lib64 --sysconfdir=/etc

        iv. Run the command *make install*

            This is the same as running the command *make libqedr_install* from the parent directory

        v. The loading section of README shows the commands to load the driver modules for the cards (multiple options)
            1. Systemctl start rdma.service
            2. /etc/init.d/rdma start
            3. To manually load the drivers if above two steps fail, do the following:
                a. insmod qedr.ko
                b. If you get an error, restart the server
                c. The command in (a) must be done every time the server is restarted.

5. Firewall Settings

   At this point, you can run ibstat and similar commands but TCP socket connections are unlikely to be established between the cards. There may possibly be a firewall issue which may or may not arise for all systems. Do the following to resolve it:

   a. iptables -I INPUT 1 -I <interface_name> -p tcp -m tcp -j ACCEPT
   b. iptables -I OUTPUT 1 -o <interface_name> -p tcp -m tcp -j ACCEPT

      <interface_name> is the name of the network interface as shown in *ip addr* command

6. Testing RDMA Connectivity
   a. Cables which support the speeds you are testing must be connected to port 1 on each Cavium RoCE card to a switch or you connect the cards directly using port 1 on each card.
   b. Run the following command on the server: *ib_write_bw -q 4 -s 102400 -report_gbits -d qedr0*
   c. Run the following command on the client: *ib_write_bw -q 4 -s 102400 -report_gbits -d qedr0 <server IP address>*
   d. Options explanation:
      i. -q == number of queue pairs to be used in the connection
      ii. -s == message size
      iii. -report_gbits defines how results should be displayed
      iv. -d == device to use in the connection. Please note that the 2-port Cavium cards report as two separate devices so one must be defined in the connection
   e. These settings should return the correct throughput speeds based on the supported bandwidth of the hardware.

Return to Top

Huawei RNICs Setup

This following instructions from Huawei explain how to load Huawei RNICs (SmartIO) drivers. The current drivers only support SuSE11SP3 and OFED 3.18.

1. User space
    a. Run *echo "driver smartio" > /etc/libibverbs.d/smartio.driver*
    b. *cp driver/linux/usr/*.so /user/lib64*
    c. *cp driver/linux/ofed/user/* /usr/lib64*
2. Kernel space
    a. The following modules should already be loaded by OFED
        compat.ko
        ib_addr.ko
        ib_core.ko
        ib_uverbs.ko
        ib_mad.ko
        ib_sa.ko
        ib_umad.ko
        ib_cm.ko
        iw_cm.ko

    b. You must insert the following modules manually
        insmod ib_ucm.ko
        insmod rdma_cm.ko
        insmod rdma_ucm.ko

    c. Insert the Huawei hi1822 driver
        insmod smartio_en.ko svc_mode=3,3,3,3
        insmod smartio_roce.ko g_roce_mode=1
        You should see four RDMA devices: acn_roce_0, acn_roce_1, acn_roce_2, acn_roce_3
3. Checking firmware version
    Run the command: */hinicshell hinic0* to check the firmware version
4. Updating firmware
    a. Locate the firmware file (e.g. smartio_roce_std_2_100G_perf.bin)
    b. Load the driver and stop the flow *before* firmware update
        ./hinicadm updatefw -I ethX -f update_filename
    c. Use the command "sft_reset_chip" in hinicshell and reboot the server to allow it to use the updated firmware.

Return to Top

Mellanox RNICs Setup

This section explains how to set up the Mellanox network interface cards for RoCE. This section assumes that the xCA/RNIC's firmware is up to date. Please consult appropriate documentation on how to update device firmware.

You can run the RoCE Interop tests with the Mellanox RoCE cards or you can configure Mellanox's ConnectX cards to run in RoCE mode. You can do this in two ways. We illustrate both methods with ConnectX-3 and ConnectX-4 cards.

ConnectX-3

1. Connect the port you would like to use on the card to a RoCE switch. In this case, the port will automatically configure itself to run in Ethernet mode.
2. Run the ***connect_port_config*** command.

```
[root@sm-node-3 ~]# connectx_port_config

ConnectX PCI devices :
|---------------------------|
| 1             0000:04:00.0 |
|---------------------------|

Before port change:
auto (ib)
auto (ib)

|---------------------------|
| Possible port modes:      |
| 1: Infiniband             |
| 2: Ethernet               |
| 3: AutoSense              |
|---------------------------|
Select mode for port 1 (1,2,3): 1
Select mode for port 2 (1,2,3): 2

After port change:
ib
eth
[root@sm-node-3 ~]# connectx_port_config

ConnectX PCI devices :
|---------------------------|
| 1             0000:04:00.0 |
|---------------------------|

Before port change:
ib
eth

|---------------------------|
| Possible port modes:      |
| 1: Infiniband             |
| 2: Ethernet               |
| 3: AutoSense              |
|---------------------------|
Select mode for port 1 (1,2,3):
```

In the screen shot shown above, the card was initially configured with both ports running in IB mode. This is the default mode. We then changed port 2 to run in Ethernet mode and confirmed the change with the second ***connect_port_config*** command call. If the command hangs, just reboot the server and when it comes back up, the port will be in Ethernet mode.

ConnectX-4

1. Connect the port you would like to use on the card to a RoCE switch. In this case, the port will automatically configure itself to run in Ethernet mode.
2. You cannot use ***connectx_port_config*** with these cards. So, follow the steps below instead:
   a. Start MST with the command ***mst start***
   b. Get the vendor_part_id with ibv_devinfo

```
[root@sm-node-16 ~]# ibv_devinfo
hca_id: mlx5_1
        transport:                      InfiniBand (0)
        fw_ver:                         12.18.2000
        node_guid:                      248a:0703:0040:9f69
        sys_image_guid:                 248a:0703:0040:9f68
        vendor_id:                      0x02c9
        vendor_part_id:                 4115
```

   c. Use the command ***mlxconfig*** to query the host about ConnectX-4 adapters. The screen output is long. We have only picked the relevant parts below. In this case, both ports are configured as IB

```
[root@sm-node-16 ~]# mlxconfig -d /dev/mst/mt4115_pciconf0 q

Device #1:
----------

Device type:    ConnectX4
PCI device:     /dev/mst/mt4115_pciconf0

Configurations:                                 Next Boot
         NON_PREFETCHABLE_PF_BAR                False(0)
         NUM_OF_VFS                             0
```

```
         LINK_TYPE_P1                           IB(1)
         LINK_TYPE_P2                           IB(1)
```

   d. Change one of the ports (LINK_TYPE_P2) to Ethernet, we leave the other port unchanged. You must confirm the changes for the command to complete

```
[root@sm-node-16 ~]# mlxconfig -d /dev/mst/mt4115_pciconf0 set LINK_TYPE_P1=1 LINK_TYPE_P2=2

Device #1:
----------

Device type:    ConnectX4
PCI device:     /dev/mst/mt4115_pciconf0

Configurations:                                 Next Boot       New
        LINK_TYPE_P1                            IB(1)           IB(1)
        LINK_TYPE_P2                            IB(1)           ETH(2)

 Apply new Configuration? ? (y/n) [n] : y
Applying... Done!
-I- Please reboot machine to load new configurations.
```

   e. Reboot the server for the changes to take effect.

Return to Top

Test Scenarios

Interoperability test scenarios vary at each event depending on the devices registered for testing. Please refer to the Integrators' List to get some idea of the different test scenarios. The list provides scenarios from previous Plugfest events.

Test Procedure

Please ensure that you have a link to all nodes. If there is no link, try re-inserting the cable; you may also need to reboot the systems. The test should normally take no more than 5 minutes to complete. If the test takes longer than 10 minutes, you may stop it with **Ctrl+C**.

1. Reset the error counters (NOTE: a cable must be connected in order to perform this)
    a. ethtool -i [INTERFACE] – Note the name of the driver
    b. ifdown [INTERFACE]
    c. modprobe -r [DRIVER] ← You may have to stop openibd before this command
    d. modprobe [DRIVER] ← if you stopped openibd, then restart it after this command
    e. Wait for the cable to link before issuing the next command
    f. ifup [INTERFACE]
2. Check the status of the connection
    a. ibstatus – make note of the rate (link Width and Expected Speed on the spreadsheet)
3. Run the MPI test
    a. mpirun --allow-run-as-root --mca btl openib,self,sm --mca pml ob1 --mca btl_openib_gid_index 0 --mca btl_openib_receive_queues P,65536,120,64,32 --mca btl_openib_cpc_include rdmacm -hostfile mpi-hosts-ce /usr/local/bin/IMB-MPI1 | tee -a $filname.txt
4. Check the status of the error counters
    a. ethtool -S eth3 | grep -i -e error -e errs -e roce_drop -e dropped -e out_of_range -e oversize -e err -e unsupported -e undersize -e fragments -e jabbers -e phy_bits -e phy_corrected_bits -e link_down_events | tee -a $filname.txt
    b. All of the counters should return 0.
5. Conditions for passing the Interoperability tests:

Conditions for passing Interoperability Testing	
Link Width	Link width is @ expected width - i.e. 1x,4x, etc
Link Speed	Link speed is @ expected speed - e.g. 100 GbE
Symbol Errors	There must be no errors during the MPI Run
Link and Port Errors	There must be no errors during the MPI Run
MPI Test	The MPI test must run to completion without error

6. This is a sample of the ethtool output.

```
rx_errors: 0
tx_errors: 0
rx_dropped: 0
tx_dropped: 0
rx_length_errors: 0
rx_over_errors: 0
rx_crc_errors: 0
rx_frame_errors: 0
rx_fifo_errors: 0
rx_missed_errors: 0
tx_aborted_errors: 0
tx_carrier_errors: 0
tx_fifo_errors: 0
tx_heartbeat_errors: 0
tx_window_errors: 0
vport_rx_dropped: 0
vport_tx_dropped: 0
rx_in_range_length_error: 0
rx_out_range_length_error: 0
```

Miscellaneous Information

1) Information about the difference between using “ifup/ifdown” & “ifconfig up/down”
   a) https://access.redhat.com/solutions/27166

Known Issues

1. If there is no link between an RNIC and a switch or another RNIC, and reboots or re-sitting the cables fails, then you may need to re-sit the RNIC(s). It is also always a good idea to try the link with known good cables first.
2. The ssh-key exchange process tends to be error prone. Make sure that IPoIB and the host files are set correctly before you attempt to do the ssh-key exchange.

Return to Top