Configuring Infiniband for AIX

Infiniband is an interconnect technology that breaks through the bandwidth and fanout limitations of PCI bus by switching from traditional shared bus architecture to a switched fabric architecture. It is a switched fabric I/O technology that ties together servers, storage devices, and network devices. Instead of sending data in parallel, which is what PCI does, Infiniband sends data in serial and can carry multiple channels of data at the same time in a multiplexing signal.

IBM® AIX® 610 supports Infiniband hardware and various protocols that run over Infiniband. This article shows how to configure Infiniband and set up IP over Infiniband interface (IPoIB) in AIX. Also, this article explains how to use RDS (Reliable Datagram Sockets), a protocol (similar to UDP) designed to work over Infiniband to send and receive data using sockets.

Configuring Infiniband

Internet Protocol (IP) packets can be sent over an Infiniband (IB) network interface by using IP over IB (IPoIB). IPoIB encapsulates the IP packets into IB packets and sends using the IB interface. In order to use IPoIB, you must install and configure the ICM driver and at least one IB device in the system. The following steps are required to configure an IB device and then configure IPoIB using the ICM.

1. Before configuring Infiniband, you need to check to see if the IB device, for instance Infiniband HCA (Infiniband Host Channel Adapter), is configured and is in “Available” state on your AIX box. To check the status, do the following:

```
#  lsdev -Cc adapter | grep "host channel"
iba0 Available  InfiniBand host channel adapter
```

2.

3. or

```
# lsdev -Cc adapter | grep "HCA"
iba0
                         Available 01-00 PCIE Dual Port HCA (b3157862)
```

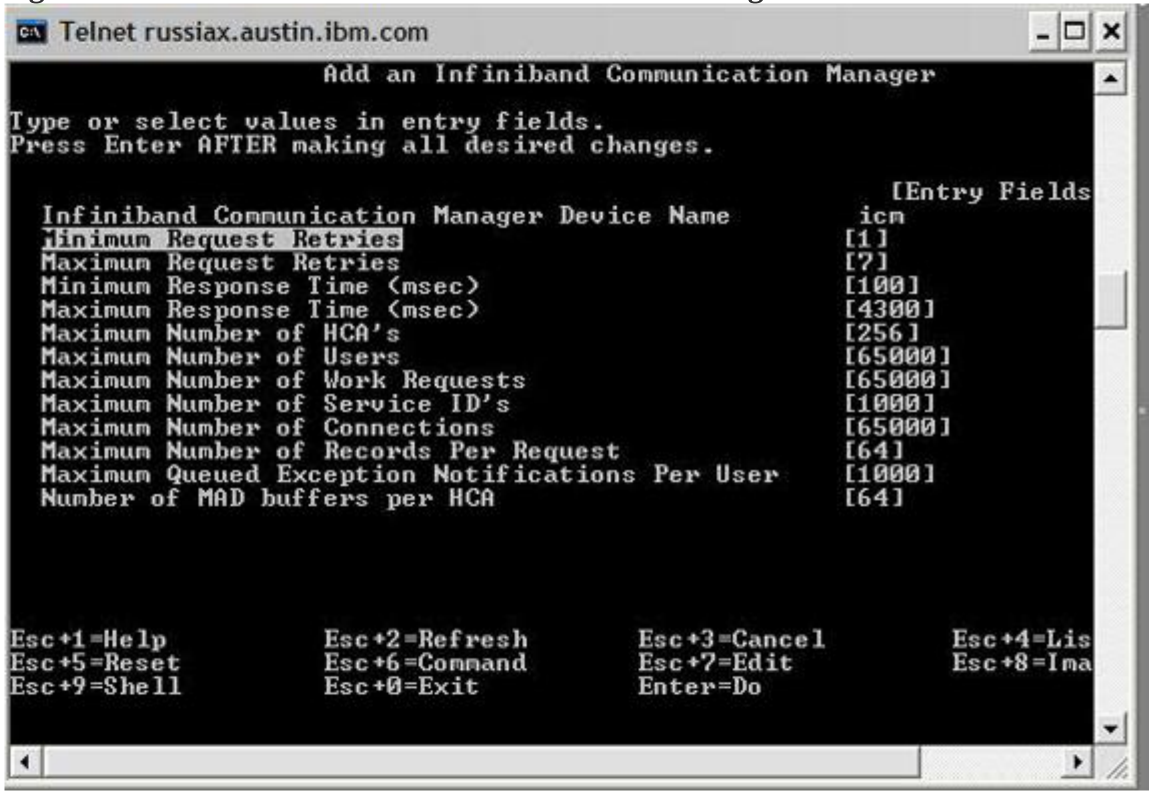
4.

5. Configure ICM (Infiniband Communication Manager). To configure ICM, do the following:

   smit icm -> Add an Infiniband Communication Manager -> Add an Infiniband Communication Manager à select **ICM** (as the ‘Name of IB Communication manager to

Add') and you will see the screen as shown in Figure 1.

Figure 1. Configure ICM



```
                    Add an Infiniband Communication Manager

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                          [Entry Fields
  Infiniband Communication Manager Device Name             icm
  Minimum Request Retries                                 [1]
  Maximum Request Retries                                 [7]
  Minimum Response Time (msec)                            [100]
  Maximum Response Time (msec)                            [4300]
  Maximum Number of HCA's                                 [256]
  Maximum Number of Users                                 [65000]
  Maximum Number of Work Requests                         [65000]
  Maximum Number of Service ID's                          [1000]
  Maximum Number of Connections                           [65000]
  Maximum Number of Records Per Request                   [64]
  Maximum Queued Exception Notifications Per User         [1000]
  Number of MAD buffers per HCA                           [64]


Esc+1=Help           Esc+2=Refresh        Esc+3=Cancel        Esc+4=Lis
Esc+5=Reset          Esc+6=Command        Esc+7=Edit          Esc+8=Ima
Esc+9=Shell          Esc+0=Exit           Enter=Do
```

Click **Enter** to use the default values for each of the fields. The next screen will show 'Command: OK' and 'icm Available'. ICM configuration is done.

To check if ICM has been configured, do

```
# lsdev -l icm
icm Available  Infiniband Communication Manager
```

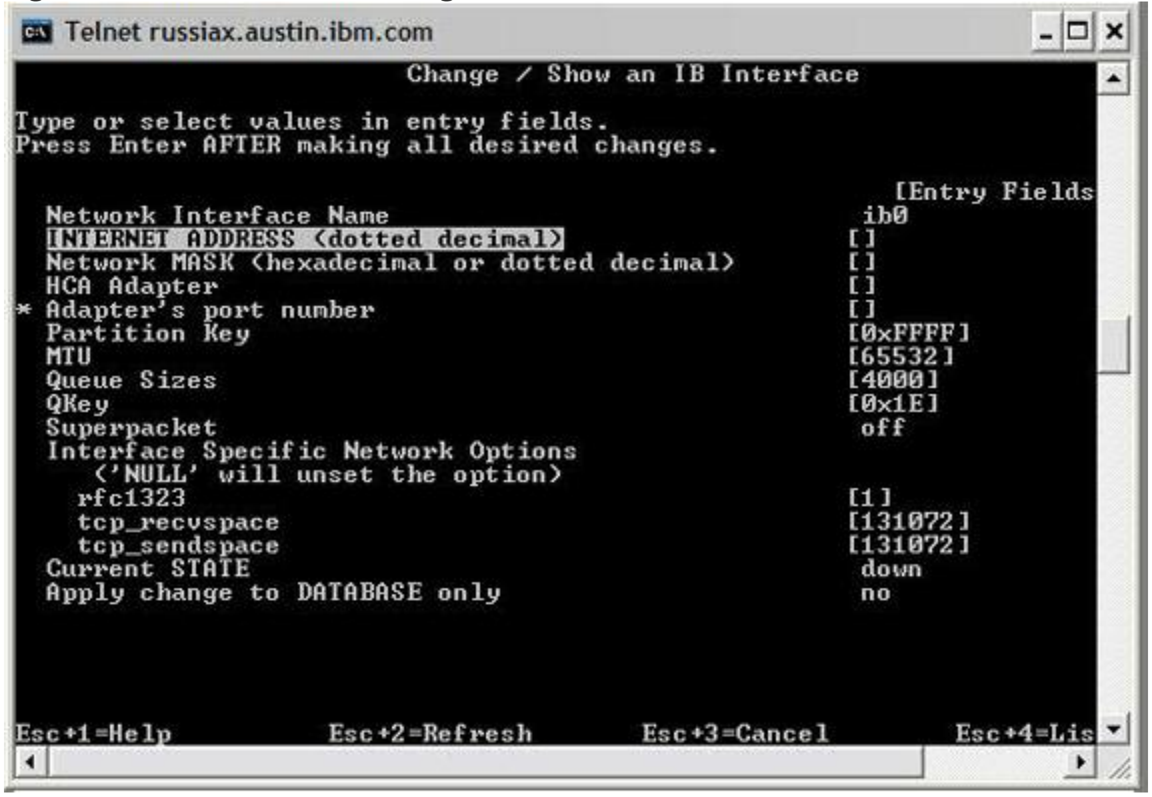
6. Configure Infiniband Network interface. IB interface can be configured using command line interface or SMIT user interface.

   To configure IB interface using SMIT do:

   smit inet -> Change / Show Characteristics of a Network interface -> select ib0 (IP over

Infiniband Network Interface). You will see the screen as shown in Figure 2.

Figure2. Configure IB interface



```
                         Change / Show an IB Interface

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                          [Entry Fields
  Network Interface Name                                   ib0
  INTERNET ADDRESS (dotted decimal)                       []
  Network MASK (hexadecimal or dotted decimal)            []
  HCA Adapter                                             []
* Adapter's port number                                   []
  Partition Key                                           [0xFFFF]
  MTU                                                     [65532]
  Queue Sizes                                             [4000]
  QKey                                                    [0x1E]
  Superpacket                                              off
  Interface Specific Network Options
     ('NULL' will unset the option)
    rfc1323                                               [1]
    tcp_recvspace                                         [131072]
    tcp_sendspace                                         [131072]
  Current STATE                                            down
  Apply change to DATABASE only                            no



Esc+1=Help          Esc+2=Refresh       Esc+3=Cancel        Esc+4=Lis
```

Enter the values for the following fields:

- Internet Address (for example, 1.2.3.92)
- Network mask (for example, 255.255.255.0)
- HCA adapter (the one we configured in step 1, iba0)
- Adpater's port number . There are two ports, Port 1 and Port 2. Use the command `ibstat` to check which port is Active. If both are active, select that you would like to use, as per your network configuration.
- Current state – up

Use the default values for the remaining fields.

The next screen shows 'Command: OK' and 'ib0 changed'. IB interface configuration is done.

To check the IB interface status, run the `ifconfig` command.

```
# ifconfig ib0
ib0: flags=e3a0063<UP,BROADCAST,NOTRAILERS,RUNNING,ALLCAST,MULTICAST,GROUPRT>

inet 1.2.3.92 netmask 0xffffff00 broadcast 1.2.3.255

tcp_sendspace 131072 tcp_recvspace 131072 rfc1323 1
```

Configure the IB interface using the command line interface.

Step 1 and step 2 are the same as above. For step 3, do the following:

```
# /usr/sbin/mkiba  -a 1.2.3.92  -i ib0 -A iba0  -p 2  -P  0xFFFF  -S up
 -m  255.255.255.0 -M 2044 ib0 changed
```

The syntax for the `mkiba` command is:

```
 /usr/bin/mkiba {-a address -i interface -A ib_adapter -p ib_port [-P P_KEY]
          [-m subnet_mask]
  [-S state] [ -M mtu ] [ -q queue_pair_size ] [ -Q Q_KEY ] [-k superpacket]
}
```

where:

-a address	IP address of the interface specified by –I (must be in dotted decimal notation)
-i interface	Interface to associate with the -a IP address
-A ib_adapter	IB adapter associated to the interface
-p ib_port	IB port associated with the IB adapter. (Defaults to 1)
-P p_key	Partition key associated with the IB port. Key (Please note that once configured partition key cannot be changed. The user must obtain the Partition key from the network administrator before configuring).
-m subnet_mask	Subnetwork mask (dotted decimal or 0x notation)
-S state	down,up,detach : The state of the ib interface.
-M ib_mtu	HCA MTU required
-q srq_size	Send and Receive queue sizes
-Q Q_KEY	Q_Key associated with the multicast group

-k superpacket	Superpacket feature on or off

Keep the following in mind:

- The –k option for superpacket is available from AIX 61B and 53N releases onwards; the lower releases do not contain the superpacket feature. Also, when this feature is enabled, it gives a good performance boost. It allows TCP/IP to send 64KB datagrams to the interface, which can increase performance. Note that this feature is supported only in AIX from an AIX host to AIX host, as long as interfaces at both the hosts are enabled with this feature.
- The –M option for HCA MTU size. AIX supports 4K physical MTU if the switch and the adapter support it. The interface for the first time, expects the user to create the broadcast multicast group in the switch. If the group is not there, always a 2K multicast group will be created by default. So if you have a 4K physical MTU supported adapter and switch, and you don't create the broadcast group in the switch, the interface will lower the MTU to 2K by creating a multicast group of 2K.

Run `ifconfig` to check the IB interface status.

```
# ifconfig ib0
ib0: flags=e3a0063<UP,BROADCAST,NOTRAILERS,RUNNING,ALLCAST,MULTICAST,GROUPRT>
    inet 1.2.3.92 netmask 0xffffff00 broadcast 1.2.3.255
     tcp_sendspace 131072 tcp_recvspace 131072 rfc1323 1
```

You are done! IB interface is configured.

To verify that all is working well, configure two nodes using the steps indicated above and run ping between the two notes. If ping works, IB is configured properly.

Configuring RDS

RDS uses the IB network interface for communication. Thus, IpoIB and IB network interface should be configured in order to use RDS protocol.

Before loading the RDS driver on the systems that you would like to communicate with using RDS, check if the IB network interfaces on those systems are able to ping each other.

Loading RDS

Run the following command to load RDS:

```
# bypassctrl load rds
```

If you receive the error `Exec format error ..`, the IB interface is not configured. See Configuring Infiniband and configure the IB interface and then try to load RDS using `bypassctrl`.

If RDS is already loaded, the error `/usr/lib/drivers/rds already loaded` displays.

To check that the RDS driver was loaded successfully, run the following:

```
# genkex | grep rds
         47e1000    53770 /usr/lib/drivers/rds
```

If a socket is created to use RDS protocol and returns the error `socket: Addr family not supported by protocol`, the RDS driver is not loaded and you need to load it. Also, note that on a reboot, the RDS driver gets unloaded and thus needs to be reloaded using the `bypassctrl` utility after every reboot.

rdsctrl utility

Once RDS is loaded, use the `rdsctrl` (/usr/sbin/rdsctrl) utility to get the RDS statistics for modifying the tuneable parameters and for diagnostics.

The `# rdsctrl stats` command displays various RDS statistics.

The statistics can be reset using

```
# rdsctrl stats reset .
```

Tuning parameters

The following RDS parameters can be tuned after RDS is loaded, but before any RDS application is run. To set any parameter, use the syntax:

```
# rdsctrl set <tunable parameter>=<value to be set>
```

- The `rds_sendspace` parameter refers to the high-water mark of the per-flow send buffer. (There may be multiple flows per socket.)

  The default value is 524288 bytes (512KB). The value is set using the command:

```
  # rdsctrl set rds_sendspace=<value in bytes>
```

- `rds_recvspace` refers to the per-flow high-water mark of the per-socket receive-buffer. For every additional flow to this socket, the receive high-water mark is bumped up by this value.

  The default value is 524288 bytes (512 KB). The value is set using the command:

```
# rdsctrl set rds_recvspace=<value in bytes>
```

  For good RDS streaming performance, the `rds_sendspace` and `rds_recvspace` parameters must be at least four times the largest RDS sendmsg size. RDS sends an ACK for each 4 messages received and if the `rds_recvspace` is not at least 4 times the message size, the throughput will be very low.

- `rds_mclustsize` refers to the size of the individual memory cluster, which is also the message fragment size. The default size is 16384 bytes (16KB). The value, always a multiple of 4096, is set using the command: `# rdsctrl set rds_mclustsize=<multiple of 4096, in bytes>`

  The `rds_mclustsize` value must be the same on all machines (nodes) in the cluster. Changing this value also has performance implications.

The current values that are set for the tuneable parameters can be retrieved using the command:

```
   # rdsctrl get <tunable parameter>
```

If this is run without any tuneable parameter, it gives the entire list of tuneable parameters

`# rdsctrl get` provides the list of tuneable parameters with their current values.

```
# rdsctrl get
 rds_conn_block_limit = 100
            rds_acksz = 180
            rds_txqsz = 1024
            rds_rxqsz = 1024
                   rds_mclustsize = 16384
                     rds_recvspace = 524288
                     rds_sendspace = 524288
```

Data-structure dumps

Various RDS structures can be dumped for troubleshooting purposes. The command to use is `# rdsctrl dump <structure>`

`<structure>` can be any one of the following:

- IBC (the details of the IB Reliable Connection)
- sendcb (the flow details)
- pcb (the RDS socket PCB details)