xCAT 2 InfiniBand Support

06/08/2010, 03:22:40 PM

Contents

Table of Contents

1. IB Interface Configuration....4
   1.1. Get sample scripts....4
   1.2. Modify the /etc/hosts....4
   1.3. Update networks table with IB sub-network....4
   1.4. Use rsh on AIX systems....5
   1.5. Update /etc/resolv.conf....5
   1.6. Setup name server on management node....5
   1.7. Check the IB network....5
   1.8. Prepare for IB drivers/libraries....6
   1.9. Modify the <profile>.otherpkgs.pkglist....6
   1.10. Update the xCAT postscripts table....6
   1.11. Start to install the nodes or update the nodes for IB configuration....6
   1.12. Check the result of IB configuration....7
2. xdsh support for IB switch....8
   2.1. Create IB switch configuration file....8
   2.2. Update /etc/hosts....9
   2.3. Use rsh for AIX....9
   2.4. Define IB switch as a node....9
   2.5. Setup ssh connection....9
   2.6. Run the test commands....9
3. Sample Scripts....10
   3.1. Annotatelog....10
      3.1.1. Description....10
      3.1.2. Supported log file....10
      3.1.3. Syntax....10
      3.1.4. Examples....11
   3.2. getGuids 11
      3.2.1. Description....11
      3.2.2. Syntax....11
      3.2.3. Examples....12
   3.3. configCEC....12

3.3.1. Description....12
3.3.2. Syntax....13
3.3.3. Examples....14
3.4. HealthCheck....15
3.4.1. Description....15
3.4.2. Syntax....18
3.4.3. Examples....18
4. IB Monitoring....19
4.1. Install RMC and xCAT-rmc packages on mn....20
4.2. Install predefined conditions, sensors and responses....20
4.3. Enable remote logging....20
4.4. Start the monitoring....20
Appendix....21

1. IB Interface Configuration

XCAT provides two sample postscripts configiba.1port and configiba.2ports to config IB secondary adapter. These two scripts can run on both AIX and Linux managed nodes.

There are two ways to configure IB interfaces, either together with node installation or using the updatenode command to update the node if the node has install systems. Most of the configuration steps for the two ways are the same, user can follow them exactly as following:

1.1. Get sample scripts

The two scripts are stored in /opt/xcat/share/xcat/ib/scripts. Since each IB adapter have two ports, if there is only one port is available per adapter, user need to manually copy configiba.1port to /install/postscript as /install/postscript/configiba. If two ports are both available per adapter, user needs to manually copy configiba.2ports to /install/postscript also, named as configiba.

```
cp /opt/xcat/share/xcat/ib/scripts/configiba.1port
/install/postscript/configiba
```

1.2. Modify the /etc/hosts

The IP address entries for IB interfaces in /etc/hosts on xCAT management nodes should have the node short hostname and the unique IB interface name in them. The format should be <ip_address_for_this_ib_interface node_short_hostname-ib_interfacename>.

For example:

```
xcat01 is the node short hostname, xcat01-ib0, xcat01-ib1, xcat01-
ib2, etc. are the IP names for the IB interfaces on xcat01.
```

For AIX, ml0 interface is also required to be setup together with IB interfaces. It follows the same name conversion with IB interfaces.

Following is an example of /etc/hosts for AIX,

```
192.168.0.10    xcat01-ib0
192.168.1.10    xcat01-ib1
192.168.2.10    xcat01-ib2
192.168.3.10    xcat01-ib3
192.168.4.10    xcat01-ml0
```

1.3. Update networks table with IB sub-network

For example:

```
chtab net=192.168.1.0 networks.netname=ib0
networks.mask=255.255.255.0 networks.mgtifname=ib0
chtab net=192.168.2.0 networks.netname=ib1
networks.mask=255.255.255.0 networks.mgtifname=ib1
chtab net=192.168.3.0 networks.netname=ib2
networks.mask=255.255.255.0 networks.mgtifname=ib2
```

```
chtab net=192.168.4.0 networks.netname=ib3
networks.mask=255.255.255.0 networks.mgtifname=ib3
chtab net=192.168.5.0 networks.netname=ib4
networks.mask=255.255.255.0 networks.mgtifname=ib4
```

Note: Attributes gateway, dhcpserver, tftpserver, and nameservers in networks table are not necessary to assign, since the xCAT management work is still running on ethernet.

1.4. Use rsh on AIX systems

On AIX, change the default connection between management nodes and compute nodes from ssh to rsh:

```
chtab key=useSSHonAIX site.value=no
```

1.5. Update /etc/resolv.conf

If the computer node have already been installed and are runnning, make sure /etc/resolv.conf is available on the compute node before running updatenode, since configiba script will connect to name server to resolve IP address for the IB interfaces. If not, define /etc/resolv.conf on compute node or use rcp to copy resolv.conf from management node to the compute node. Following is an example of /etc/resolv.conf:

```
nameserver 192.168.0.13
domain ppd.pok.ibm.com
search ppd.pok.ibm.com
```

Note: 192.168.0.13 is the name server address which could provide the IP addresses for IB interfaces on compute nodes.

1.6. Setup name server on management node

Put IB interface entries in /etc/hosts into DNS and restart the DNS:

For Linux Management Nodes:

```
makedns
service named restart
```

For AIX Management Nodes:

```
makedns
stopsrc -s named
startsrc -s named
lssrc -s named
```

1.7. Check the IB network

Check if DNS resolving of the IB network has been setup successfully on management node . If not, redo the steps in 1.5

```
nslookup xcat01-ib0
nslookup xcat01-ib1
```

1.8. Prepare for IB drivers/libraries

For AIX, the IB drivers/libraries have been installed in the system. So this step is only for RHEL and SLES.

The required packages for RHEL and SLES have been listed in appendix.

1)For RHEL, the drivers/libraries are shipped in RHEL release CD/DVD.

2)For SLES10, the drivers/libraries are shipped in SLES10 SP2 AS.

3)For SLES11, the drivers/libraries are shipped in SLES11 release CD/DVD.

After got the packages from CD/DVD, put them under /install/post/otherpkgs/<os>/<arch> directory where <os> and <arch> can be found in the nodetype table.

1.9. Modify the <profile>.otherpkgs.pkglist

Add rpm names (without version number) into /install/custom/install/<ostype>/<profile>.otherpkgs.pkglist

where <profile> is defined in the nodetype table.

<ostype> is the operating system name without the version number.

The following os types are recognized by xCAT.

```
centos
fedora
rh
sles
windows
```

1.10. Update the xCAT postscripts table

```
chdef xcat01 postscripts.postscripts,=configiba
```

Note:
1.Please keep this order for these two scripts, since configiba depends on otherpkgs to install IB driver/library.
2.Since perl is not default installed on diskless nodes. Postscripts configiba which is written in perl will failed. User needs to add perl in diskless image as workaround.

1.11. Add perl packages(Optional)

For diskless boot on Linux, perl packages have been removed to reduce the space size, but configiba is written in perl, you have to add perl packages for diskless boot.

Remove the following line in file /opt/xcat/share/xcat/netboot/rh/compute.exlist to add perl pckages for diskless boot on Linux:

```
./usr/lib/perl5*
```

Note: This is a workaround in xCAT 2.4. xCAT will rewrite configiba in bash which don't require perl.

1.12. Start to install the nodes or update the nodes for IB configuration

Now all the preparation work for IB configuration has been done, user can either use the updatenode command to update the nodes if systems on compute nodes have been installed

```
updatenode xcat01 otherpkgs,configiba
```

or continue with the node installation process,

To diskless boot Linux nodes, you have to put the IB device driver packages into diskless image before node installation, for more details, please check section 4.3.1 “Generate the stateless image for compute node” in doc xCAT2pLinux.pdf.

```
nodeset xcat01 netboot
rnetboot xcat01
```

To install diskful Linux nodes:

```
nodeset xcat01 install
rnetboot xcat01
```

To diskless boot AIX nodes:

```
nimnodeset xcat01 netboot
rnetboot xcat01
```

To install diskful AIX nodes:

```
nimnodeset xcat01 install
rnetboot xcat01
```

Note: In the sample postscript, the netmask is set to default value: 255.255.255.0 and gateway is set to "X.X.255.254". If the IB interface name is not a simple combination of short hostname and ibX or netmask and gateway does not meet the user's requirement, then modify the sample script , like in the example below:

The short hostname of the compute node is xcat01-en, and the IB interface name is xcat01-ib0, xcat01-ib1, etc. The user should modify the /install/postscript/configiba as follows:

```
my $hostname = "$ENV{NODE}-$nic";
```

to

```
my $fullname = `echo $ENV{NODE} | cut -c 1-11`;
chomp($fullname);
my $hostname = "$fullname-$nic";
```

It is assumed every node has two IB adapters, if only one adapter is available on each node, modify the /install/postscript/configiba as following:

```
my @nums = (0..3);
```

to

```
my @nums = (0..1);
```

1.13.Check the result of IB configuration

Use ping test from management node to the IB interfaces on compute nodes to see if IB adapter works or not.

```
ping xcat01-ib0
```

2. xdsh support for IB switch

2.1. Create IB switch configuration file

A new switch configuration file on management node is introduced to allow the xdsh command to setup ssh, that is transfer the ssh keys to the IB device. The device configuration file is located in /var/opt/xcat/<DevicePath>/config.

The <DevicePath> is parsed by xdsh from the attribute value of the "--devicetype" flag or the environment variable "DEVICETYPE" which is input to the xdsh call.

For example:
If the devicetype for Qlogic switch is "IBSwitch::Qlogic" then the device configuration file must be be found in the following directory:
/var/opt/xcat/IBSwitch/Qlogic/config

The following is an example of a device configuration file:

```
# Qlogic switch device configuration
[main]
ssh-setup-command=sshKey add
[xdsh]
pre-command=NULL
post-command=showLastRetcode -brief
```

Below is the explanation of the file attributes:
ssh-setup-comand

Specify the ssh key appending command supported by device specified. If this entry is not provided, xCAT uses default ways for HMC and IVM-managed devices to write ssh keys of Management Nodes.

pre-command
Specify the pre-execution commands before remote command. For example, users might want to export some environment variables before executing real commands. If the value of this entry is assigned "NULL", it means no pre-execution commands are needed.
For example, the Qlogic Switch does not support environment variable, the 'pre-command' is assigned with "NULL" to disable environment variables usage.
If no entry is provided, the default behavior is to export the environment variables that are normally exported by xdsh when running remote commands.

post-command
Specify the built-in command provided by device specified to show the last command execution result. For example, the Qlogic Switch provides "showLastRetcode -brief" to display a numeric return code of last command execution.
If the value of this entry is assigned "NULL", it means no post-command is used.
If no entry is provided, the default behavior to run "echo $?" used to dump return code of last command execution.

2.2. Update /etc/hosts

In the /etc/hosts,

```
9.114.47.172    ibswitch
```

2.3. Use rsh for AIX

The default remote shell on the AIX management node is rsh. Changing the site useSSHonAIX attribute=yes will change the default to ssh for xdsh.

```
chtab key=useSSHonAIX site.value=yes
```

2.4. Define IB switch as a node

Define IB switch as a node, this is required by xdsh which only support the input as a node.

```
mkdef –t node –o ibswitch groups=all nodetype=switch
```

2.5. Setup ssh connection

You can use xdsh to configure ssh login to the IB device by running the following. Note you must use the correct userid for your device. After this configuration is complete, you will be able to login to the device without a password.

```
xdsh ibswitch -K -l admin --devicetype IBSwitch::Qlogic
```

```
Enter the password for the userid on the node where the ssh keys will
be updated.

/usr/bin/ssh setup is complete.
return code = 0
```

2.6. Run the test commands

After setup of the mmands on IB switches from the management nssh keys for the login, the admin can run the coode using xdsh.

Interactive commands like List on IB switch are not supported by xdsh. An error message will print out if user inputs an interactive command.

Below is an example of using xdsh to list the valid commands on the device.

```
/opt/xcat/bin/xdsh ibswitch -l admin --devicetype IBSwitch::Qlogic
fwVersion
```

Or:

```
export DEVICETYPE=IBSwitch::Qlogic && /opt/xcat/bin/xdsh ibswitch -l
admin fwVersion
```

3. Sample Scripts

3.1. Annotatelog

3.1.1. Description

annotatelog is a sample script to parse the QLogic log entries in log files on the xCAT Management Node output by subnet manager, IB node, chassis, FRU(Field-Replaceable Unit) or a particular node. This script is supported on AIX and Linux management nodes.

3.1.2. Supported log file

/var/log/messages is supported to be analyzed by annotatelog. But from xCAT's view the log to analyze should be xCAT consolidated log, which means this log file should come from xCAT syslog/errorlog monitoring mechanism, such as /var/log/xCAT/errorlog/[xCAT Management Nodes] file.

3.1.3. Syntax

The syntax of the annotatelog command will be:

annotatelog -f log_file [-s start_time] [-e end_time]
{ [-i -g guid_file -l link_file] [-S] [-c] [-u]| [-A -g guid_file -l link_file]}
{[-n node_list -g guid_file] [-E]}
[-h]

-f log_file

Specifies a log file fullpath name to analyze; must be xCAT consolidated log got from Qlogic HSM or ESM.

-s start_time

Specifies the start time for analysis, where the **start_time** variable has the format ddmmyyhh:mm:ss (day, month, year, hour, minute, and second), if it is not specified, annotatelog will parse the log file from the beginning.

-e end_time

Specifies the end time for analysis, where the **end_time** variable has the format ddmmyyhh:mm:ss (day, month, year, hour, minute, and second), if it is not specified, annotatelog will parse the log file to the end.

-l link_file

Specifies a link file fullpath name, which concatenates all '/var/opt/iba/analysis/baseline/fabric*links' files from all fabric management nodes.

-g guid_file

Specifies a guid file fullpath name, which has a list of GUIDs as obtained from the "getGuids" script.

-E

Annotate with node ERRLOG_ON and ERRLOG_OFF information. This can help determine if a disappearance was caused by a node disappearing. It is for AIX nodes only and should be used with -n or -i flag

-S

Sort the log entries by subnet manager only.

-i

Sort the log entries by IB node only.

-c

Sort the log entries by chassis only.

-u

Sort the log entries by FRU only.

-A

Output the combination of -i, -S, -c and -u. It should be used with -g and -l flags.

-n node_list

Specifies a comma-separated list of xCAT Managed Node host names, IP addresses to look up in log entries, it should be used with -g flag.

-h

Display usage information.

3.1.4. Examples

1). Sort the log entries by subnet manager only.

```
./annotatelog -f /var/log/messages -S
```

2). Sort the log entries by chassis only.

```
./annotatelog -f /var/log/messages -c
```

3.2. getGuids

3.2.1. Description

getGuids is a sample script to get GUIDs for Infiniband Galaxy HCAs (Host Channel Adapter) and their ports from xCAT Management Nodes. It needs to be run on the xCAT Management Node. It will use a xdsh call to all the xCAT Managed Nodes to get the information about the IB devices. It uses the ibstat command on AIX system or ibv_devinfo command on Linux system to get the information about the IB devices.

3.2.2. Syntax

The syntax of the getGuids command will be:

getGuids [-h] [-f output_file]

-f output_file

Specifies a file full path name that is used to save the GUIDs output.

-h

Display usage information.

3.2.3. Examples

1). xcat05 is an AIX compute node defined in xCAT management node, run getGuid to get guid of xcat05

```
./getGuids -f guid_file
```

3.3. configCEC

3.3.1. Description

The configCECs script is written in ksh, and used to create a full system partition for each CECs Managed by the HMC. It will use ssh to login the HMC with the hscroot userid in order to rename the CECs based on a certain pattern specified through command line and create full partition for all the CECs.

Since for the large HPC environment the user usually does not use many nodes that are not Power6 575 nodes, so we only support Power6 575 servers in this script. If the users wants to do LPAR setup for other servers, they needs to modify this sample script manually.

To specify the name format to be used for the CEC/LPAR/Profile, this script uses the same logic that the 'date' command uses for specifying how to output the date. There are 4 field descriptors that the script will recognize:

%F = the frame number of the frame that the CEC is in
%N = the relative node number of the CEC in the frame
%C = the cage number of the CEC in the frame
%S = the serial number of the CEC

For example if you want the CEC name to be 'airbus_f<frame#>n<node#>_SN<serial#>', then the format to use would be 'airbus_f%Fn%N_SN%S'

The way the script finds the CECs on the HMC is to issue the 'lssyscfg -r frame' command to find all the frames and then issues the 'lssyscfg -r cage' command for each frame to list the contents of each cage position in a given frame. It then starts looking for CECs starting at cage 1 and going through to the last cage. The first CEC found in a frame is assumed to be node 1, the second node found is node two and so on. The script then will assign each CEC a frame number, a node number, a cage number and the Serial number of the CEC which can be used in naming the CEC/LPAR/Profile. If no frames/cages/CECs are found on this HMC, an error message will be displayed. xCAT command rspconfig could be used to setup ssh remote shell from the xCAT Management Node to the HMCs without prompting for the hscroot password; otherwise the user has to type in the password manually for many times. And if the user wants to use the frame number in the name of the CEC or LPAR then the frame number must be set on the frames through HMC Web GUI or HMC command line before issuing this script.

This script supports three resource allocate_types to create the full system partition; they are always_all, always_list and conditional. The default method is always_all. always_all indicates to always use the 'all resources' LPAR flag; always_list indicates to always explicitly list the devices in the LPAR; and conditional indicates to use the 'all resources' LPAR flag if not --exclude_hw is found, otherwise use an explicit list for the hardware.

As default, this script will assign all the resources to the full system partition, but if the allocate_type is always_list or conditional, then the user could use --exclude_hw flag to exclude those devices that can not be assigned or not supported by the operating system from assignment. The supported hardware names or 'device_id's to exclude are RIO and 10G, RIO indicates Galaxy 1 HCA used for RIO connection; 10G indicates 2-port 10G integrated adapter.

Actually, this script will not change the CECs/LPARs directly but creates one or two scripts (Rename_cecs, Build_lpars) in /tmp directory on xCAT MN that will do the changes once the user executes them. The /tmp/Rename_cecs should be run first and then the /tmp/Build_lpars. The reason why we do it this way is to have the user see exactly what HMC commands would be executed and also have a better chance to fine tune the commands if it is needed.

Warning: this script will configure all the CECs that managed by this hmc passed in. Please do check the contents in Build_lpars before run it. Remove the commands in Build_lpars that related to the CECs you dodn't want to do any change.

3.3.2. Syntax

configCECs -H hmc_list [-c cec_format] [-l lpar_format] [-p profile_format]
[--frame_pad_len len_number] [--node_pad_len len_number]
[--cage_pad_len len_number]
[--allocate_type always_all | always_list | conditional]
[--exclude_hw]
[-h]

-H hmc_list

Specifies a comma-separated list of HMC host names, IP addresses to configure CECs on.

-c cec_format

Specifies the naming format for CEC, the default format is f%Fn%N_SN%S.

-l lpar_format

Specifies the naming format for LPAR, the default format is f%Fn%N.

-p profile_format

Specifies the naming format for profile, the default format is the same with lpar_format.

--frame_pad_len len_number

Specifies the number of digits used for the frame numbers, it will be zero filled if needed. The default value is no padding.

--node_pad_len len_number

Specifies the number of digits used for the node numbers, it will be zero filled if needed. The default value is no padding.

--cage_pad_len len_number

Specifies the number of digits used for the cage numbers, it will be zero filled if needed. The default value is no padding.

--allocate_type

Specifies the allocation method that is used to allocate resources to full system partition. The supported allocation methods are always_all, always_list and conditional. The default method is always_all. always_all indicates to always use the 'all resources' LPAR flag; always_list indicates to always explicitly list the devices in the LPAR; and conditional indicates to use the 'all resources' LPAR flag if not --exclude_hw is found, otherwise use an explicit list for the hardware.

--exclude_hw

Specifies a comma-separated list of hardware names or 'device id's that do not need to assign. The supported hardware names are RIO and 10G, RIO indicates Galaxy 1 HCA

used for RIO connection; 10G indicates 2-port 10G integrated adapter. It can only be used with --allocate_type is always_list or conditional.

-h Display usage information.

3.3.3. Examples

1). If c98m6hmc01 manage one CEC, config the CEC with name Server_f1n1_SN0262672 as a single node f1n1:

```
./configCECs -H c98m6hmc01 -c Server_f%Fn%N_SN%S
/tmp/Build_lpars
```

3.4. HealthCheck

3.4.1. Description

This script is used to check the system health for both AIX and Linux Managed Nodes on Power6 platforms. It will use xdsh to access the target nodes, and check the status for processor clock speed, IB interfaces, memory and large page configuration. If xdsh is unreachable, an error message will be given.

1. Processor clock speed check

This script will use xdsh command to access the target nodes, and run "/usr/pmapi/tools/pmcycles -M" command on the AIX MNs or "cat /proc/cpuinfo" command on Linux MNs to list the actual processor clock speed in MHz. Compare this actual speed with the minimal value that user specified in command line with -p flag, if it is smaller than the minimal value, a warning message will be given out to indicate the unexpected low frequency.

2. IB interface status check by llstatus

In LoadLeveler cluster environment, all the nodes are sharing the same cluster information. So we only need to xdsh to one of these nodes, and run LoadLeveler command "/usr/lpp/LoadL/full/bin/llstatus -a" on AIX or "/opt/ibmll/LoadL/full/bin/llstatus -a" on Linux nodes to list the IB interface status. If the status is not "READY", a warning message related to its nodename and IB port will be given out. This check process needs the "llstatus" command existed on the MNs, if it does not exist, an error message will be output.

3. IB interface status check by lsrsrc

This script will use xdsh command to access the target nodes, and run "/usr/bin/lsrsrc IBM.NetworkInterface Name OpState" command on AIX or Linux MNs to list the IB interface status for each node. If the "OpState" value is not "1", a warning message related to its nodename and IB port will be given out.

4. Memory check

This script will use xdsh command to access the target nodes, and run "/usr/bin/vmstat" command on AIX MNs or "cat /proc/meminfo" commands on Linux MNs to list the total memory information. If the total memory is smaller than the minimal value specified by the user in GB, a warning message will be given out with the node name and its real total memory account.

5. Free large page check

This script will use xdsh command to access the target nodes, and run "/usr/bin/vmstat -l" command on AIX MNs or "cat /proc/meminfo" commands on Linux MNs to list the free large page information. If the free large page number is smaller than the minimal value specified by the user, a warning message will be given out with the node name and its real free large page number.

6. Check HCA status

This script will use xdsh command to access the target nodes. For AIX nodes, we use command ibstat -v | egrep "IB PORT.*INFO|Port State:|Physical Port" to get the HCA status of Logical Port State, Physical Port State, Physical Port Physical State, Physical Port Speed and Physical Port Width. The expected values are "Logical Port State: Active", "Physical Port State: Active", "Physical Port Physical State: Link Up", "Physical Port Width: 4X". If the actual value is not the same as expected one, a warning message will be given out.

This is an example of the output of ibstat command:

```
c890f11ec01:/ # ibstat -v | egrep "IB PORT.*INFO|Port State:|Physical Port"
 IB PORT 1 INFORMATION (iba0)
Logical Port State:                Active
Physical Port State:               Active
Physical Port Physical State:          Link Up
Physical Port Speed:                2.5G
Physical Port Width:                4X
 IB PORT 2 INFORMATION (iba0)
Logical Port State:                Active
Physical Port State:               Active
Physical Port Physical State:          Link Up
Physical Port Speed:                2.5G
Physical Port Width:                4X
```

For Linux nodes, we use command ibv_devinfo -v | egrep "ehca|port:|state: |width:| speed:" to get the HCA status of port state, active_width, active_speed and phys_state. The expected values are "port state: PORT_ACTIVE", "active_width: 4X", "phys_state: LINK_UP". If the actual value is not the same as expected one, a warning message will be given out.

This is an example of the output of ibv_devinfo command:

```
c890f11ec05:~ # ibv_devinfo -v | egrep "ehca|port:|state:|width:|speed:"
```

```
hca_id: ehca0
        port:   1
                state:              PORT_ACTIVE (4)
                active_width:         4X (2)
                active_speed:         2.5 Gbps (1)
                phys_state:         LINK_UP (5)
        port:   2
                state:              PORT_ACTIVE (4)
                active_width:         4X (2)
                active_speed:         2.5 Gbps (1)
                phys_state:         LINK_UP (5)
```

But for "Physical Port Speed" on AIX nodes or "active_speed" on Linux nodes, since SDR and DDR adapters will use the different speeds, SDR is 2.5G and DDR is 5.0G, so the user needs to specify this "Speed" by flag "--speed", for example:

healthCheck -M -H --speed 2.5

If "--speed" is not specified with "-H" flag, healthCheck script will list the actual value of "Physical Port Speed" gotten from ibstat command for each HCAs, so that it is easy for the user to use "grep" command to find the speed value he/she wants.

The output format is <node_name>:<interface_name>:< Physical Port Speed >:<speed_value>, for example:

c890f11ec01.ppd.pok.ibm.com: ib0: Physical Port Speed: 2.5G

c890f11ec01.ppd.pok.ibm.com: ib1: Physical Port Speed: 2.5G

c890f11ec02.ppd.pok.ibm.com: ib0: Physical Port Speed: 5.0G

c890f11ec02.ppd.pok.ibm.com: ib1: Physical Port Speed: 5.0G

Since the output of ibstat or ibv_devinfo is identified by HCA name and port number, so we will use the mapping table below to map the HCA name and port number to its interface name. Please see the table below:

Interface Name	Adapter Name	Port Number
ib0	iba0/ehca0	1
ib1	iba0/ehca0	2
ib2	iba1/ehca1	1
ib3	iba1/ehca1	2

.....

For "Physical Port Width" on AIX nodes or "active_width" on Linux nodes, since it could be 4X or 12X, so the user needs to specify this "width" by flag "--width", for example:

healthCheck -M -H --width 4X

If "--width" is not specified, healthCheck script will list the actual value of "Physical Port Width" gotten from ibstat command for each HCAs, so that it is easy for the user to use "grep" command to find the speed value he/she wants.

The output format is <node_name>:<interface_name>:< Physical Port Width >:<width_value>, for example:

c890f11ec01.ppd.pok.ibm.com: ib0: Physical Port Width: 4X

c890f11ec01.ppd.pok.ibm.com: ib1: Physical Port Width: 4X

c890f11ec02.ppd.pok.ibm.com: ib0: Physical Port Width: 4X

For the ports that are not used by the target nodes, the user could use –ignore flag to exclude them from HCA status check. If the user does not specify these "unused port" with --ignore flag, healthCheck script will check all HCA check items for all interfaces, and return the warning message to for the failed ones.

The user could use grep piped into wc -l to get the total number of "unused port".

3.4.2. Syntax

healthCheck { [-n node_list] [-M]}

{[-p min_clock_speed] [-i method] [-m min_memory]

[-l min_freelp] [-H [--speed speed --ignore interface_list –width width]]}

[-h]

-M Check status for all the Managed Nodes that are defined on this MN.

-n node_list

Specifies a comma-separated list of node host names, IP addresses for health check.

-p min_clock_speed

Specifies the minimal processor clock speed in MHz for processor monitor.

-i method

Specifies the method to do Infiniband interface status check, the supported

check methods are LL and RSCT.

-m min_memory

Specifies the minimal total memory in MB.

-l min_freelp

Specifies the minimal free large page number.

-H Check the status for HCAs.

--speed speed

Specifies the physical port speed in G bps, it should be used with -H flag.

--ignore interface_list

Specifies a comma-separated list of interface name to ignore from HCA status check, such as ib0,ib1. It should be used with -H flag.

--width width

Specifies the physical port width, such as 4X or 12X. It should be used with -H flag.

-h Display usage information.

3.4.3. Examples

1). Check IB interface status of one node:

./healthCheck -n xcat04 -i RSCT

Output log is being written to "/var/log/xcat/healthCheck.log".

Checking health for AIX nodes: xcat04...

Checking IB interface status using command /usr/bin/lsrsrc for nodes: xcat04...

IB interfaces of all nodes are normal.

2). Check processor clock speed for one node:

./healthCheck -n xcat04 -p 500

Output log is being written to "/var/log/xcat/healthCheck.log".

Checking health for AIX nodes: xcat04...

Checking processor clock speed for nodes: xcat04...

The processor clock speed of all nodes is normal.

3). Check memory usage for one node:

./healthCheck -n xcat04 -m 500

Output log is being written to "/var/log/xcat/healthCheck.log".

Checking health for AIX nodes: xcat04...

Checking memory for nodes xcat04...

Memory size of all nodes are normal.

4. IB Monitoring

xCAT has the capability to monitor, through IBM's Resource Monitoring and Control (RMC) subsystem, the errors or information in the syslog logged by IB switches and the subnet manager.

RMC is part of the IBM's Reliable Scalable Cluster Technology (RSCT) that provides a comprehensive clustering environment for AIX and Linux. The RMC subsystem and the core resource managers that ship with RSCT enable you to monitor various resources of your system and create automated responses to changing conditions of those resources. RMC also allows you to create your own conditions (monitors), responses (actions) and sensors (resources). rmcmon is xCAT's monitoring plug-in module for RMC. It's responsible for automatically setting up RMC monitoring domain for RMC and creates predefined conditions, responses and sensor on the management node, the service node and the nodes.

To monitor IB, RMC will leverage the remote syslog capability of the switches and subnet manager.

4.1. Install RMC and xCAT-rmc packages on mn

Please refer to section `1 "Intall RMC on mn"` and `2 "Install xCAT-rmc on mn"` of chapter 2.4 in this document https://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2-Monitoring.pdf for how to setup RMC monitoring on management node. You do not have to install RMC on the compute node if you just want to monitor IB logs.

4.2. Install predefined conditions, sensors and responses

```
moncfg rmcmon
```

This will get all predefined conditions, sensors and responses installed.
To verify, run command:

```
lscondition
```

One of the condition is called `IBSwitchLog`.

```
lssensor
```

One of the sensor is called `IBSwitchLogSensor`

4.3. Enable remote logging

Configure the switches and the subnet manager to send all logs to the management node.

4.4. Start the monitoring

```
startcondresp IBSwitchLog EmailRootAnyTime
```

With this condition-response association, any logs that's are level local6.info and above will be caught and sent to the root's mail box. This may generate a lot of mails for root.

You can customize the condition to filter out certain logs and send them to root. For example:

```
chcondition -e "String =? 'error'" IBSwitchLog
```

It will only send the logs that contain the word 'error' to the root.
To make it more efficient, you can customize the sensor instead. First run the following command to check the attributes of the sensor.

```
lssensor IBSwitchLogSensor
```

The output looks like this:

```
Name = IBSwitchLogSensor
ActivePeerDomain =
Command = /opt/xcat/sbin/rmcmon/monaixsyslog -p local6.info
ConfigChanged = 0
ControlFlags = 0
Description =
ErrorExitValue = 1
ErrorMessage =
ExitValue = 0
Float32 =
Float64 =
Int32 =
Int64 =
MonitorStatus = 0
NodeNameList = {xcat20RRmn.cluster.net}
RefreshInterval = 0
SavedData =
SD =
String =
TimeCommandRun = Wed Dec 31 19:00:00 2008
Uint32 =
Uint64 =
UserName = root
```

You can change the `Command` attribute to only process severity level 'warning' and above for IB logs. Since `Command` cannot be changed once the sensor is defined, you have to create a new sensor. For AIX, run the following command:

```
mksensor -i 60 -e 1 -c 0 IBSwitchWarn
         "/opt/xcat/sbin/rmcmon/monaixsyslog -p local6.warn"
```

For Linux, replace the string `monaixsyslog` with `monerrorlog`.
Now change the condition to use this new sensor:

```
chcondition -s "Name='IBSwitchWarn'" IBSwitchLog
```

Appendix

Driver/Library	Corresponding rpms in RHEL5.3	
openib	*openib-*. el5.noarch.rpm*	
libib	**32bit**	*libibcm-*.el5.ppc.rpm* *libibcm-devel-*.el5.ppc.rpm* *libibcm-static-*.el5.ppc.rpm* *libibcommon-*.el5.ppc.rpm*

		libibcommon-devel-.el5.ppc.rpm* *libibcommon-static-*.el5.ppc.rpm* *libibmad-*.el5.ppc.rpm* *libibmad-devel-*.el5.ppc.rpm* *libibmad-static-*.el5.ppc.rpm* *libibumad-*.el5.ppc.rpm* *libibumad-static-*.el5.ppc.rpm* *libibumad-devel-*.el5.ppc.rpm* *libibverbs-*.el5.ppc.rpm* *libibverbs-devel-*.el5.ppc.rpm* *libibverbs-static-*.el5.ppc.rpm* *libibverbs-utils-*.el5.ppc.rpm*
	64bit	*libibcm-*.el5.ppc64.rpm* *libibcm-devel-*.el5.ppc64.rpm* *libibcm-static-*.el5.ppc64.rpm* *libibcommon-*.el5.ppc64.rpm* *libibcommon-devel-*.el5.ppc64.rpm* *libibcommon-static-*.el5.ppc64.rpm* *libibmad-*.el5.ppc64.rpm* *libibmad-devel-*.el5.ppc64.rpm* *libibmad-static-*.el5.ppc64.rpm* *libibumad-*.el5.ppc64.rpm* *libibumad-devel-*.el5.ppc64.rpm* *libibumad-static-*.el5.ppc64.rpm* *libibverbs-*.el5.ppc64.rpm* *libibverbs-devel-*.el5.ppc64.rpm* *libibverbs-static-*.el5.ppc64.rpm* *libibverbs-utils(it is used to ship ibv_* commands and depends on 32bit IB libraries) 64bit rpm is not available in RedHatEL5.3. Please install 32bit IB libraries also if user needs both ibv_* commands and the 64bit libraries.*
libehca (for Galaxy1/ Galaxy2 support)	**32bit**	*libehca-*.el5.ppc.rpm* *libehca-static-*.el5.ppc.rpm*
	64bit	*libehca-*.el5.ppc64.rpm* *libehca-static-*.el5.ppc64.rpm*

libmthca (for Mellanox InfiniHost support)	**32bit**	*libmthca-*.el5.ppc.rpm* *libmthca-static-*.el5.ppc.rpm*
	64bit	*libmthca-*.el5.ppc64.rpm* *libmthca-static-*.el5.ppc64.rpm*
libmlx4 (for Mellanox ConnectX support)	**32bit**	*libmlx4-*.el5.ppc.rpm* *libmlx4-static-*.el5.ppc.rpm*
	64bit	*libmlx4-*.el5.ppc64.rpm* *libmlx4-static-*.el5.ppc64.rpm*

RedHatEL5.3 only ships 32bit libibverbs-utils(it is used to ship ibv_* commands) package in CDs/DVD, which depends on 32bit IB libraries, so it will fail to be installed if only 64bit libraries exist on the system. For the user who needs both these IB commands and the 64bit libraries, please install both 32bit and 64bit library packages.

Driver/Library	Corresponding rpms in RHEL5.4	
openib	*openib-*. el5.noarch.rpm*	
libib	**32bit**	*libibcm-*.el5.ppc.rpm* *libibcm-devel-*.el5.ppc.rpm* *libibcm-static-*.el5.ppc.rpm* *libibcommon-*.el5.ppc.rpm* *libibcommon-devel-*.el5.ppc.rpm* *libibcommon-static-*.el5.ppc.rpm* *libibmad-*.el5.ppc.rpm* *libibmad-devel-*.el5.ppc.rpm* *libibmad-static-*.el5.ppc.rpm* *libibumad-*.el5.ppc.rpm* *libibumad-static-*.el5.ppc.rpm* *libibumad-devel-*.el5.ppc.rpm* *libibverbs-*.el5.ppc.rpm* *libibverbs-devel-*.el5.ppc.rpm* *libibverbs-static-*.el5.ppc.rpm* *libibverbs-utils-*.el5.ppc.rpm*
	64bit	*libibcm-*.el5.ppc64.rpm*

		libibcm-devel-.el5.ppc64.rpm* *libibcm-static-*.el5.ppc64.rpm* *libibcommon-*.el5.ppc64.rpm* *libibcommon-devel-*.el5.ppc64.rpm* *libibcommon-static-*.el5.ppc64.rpm* *libibmad-*.el5.ppc64.rpm* *libibmad-devel-*.el5.ppc64.rpm* *libibmad-static-*.el5.ppc64.rpm* *libibumad-*.el5.ppc64.rpm* *libibumad-devel-*.el5.ppc64.rpm* *libibumad-static-*.el5.ppc64.rpm* *libibverbs-*.el5.ppc64.rpm* *libibverbs-devel-*.el5.ppc64.rpm* *libibverbs-static-*.el5.ppc64.rpm* *libibverbs-utils(it is used to ship ibv_* commands and depends on 32bit IB libraries) 64bit rpm is not available in RedHatEL5.4. Please install 32bit IB libraries also if user needs both ibv_* commands and the 64bit libraries.*
libehca (for Galaxy1/ Galaxy2 support)	**32bit**	*libehca-*.el5.ppc.rpm* *libehca-static-*.el5.ppc.rpm*
	64bit	*libehca-*.el5.ppc64.rpm* *libehca-static-*.el5.ppc64.rpm*
libmthca (for Mellanox InfiniHost support)	**32bit**	*libmthca-*.el5.ppc.rpm* *libmthca-static-*.el5.ppc.rpm*
	64bit	*libmthca-*.el5.ppc64.rpm* *libmthca-static-*.el5.ppc64.rpm*
libmlx4 (for Mellanox ConnectX support)	**32bit**	*libmlx4-*.el5.ppc.rpm* *libmlx4-static-*.el5.ppc.rpm*
	64bit	*libmlx4-*.el5.ppc64.rpm* *libmlx4-static-*.el5.ppc64.rpm*

RedHatEL5.4 only ships 32bit libibverbs-utils(it is used to ship ibv_* commands) package in CDs/DVD, which depends on 32bit IB libraries, so it will fail to be installed if only 64bit libraries exist on the system. For the user who needs both these IB commands and the 64bit libraries, please install both 32bit and 64bit library packages.

Platforms	Driver/Library
SLES11	*ofed-*.ppc64.rpm* *ofed-kmp-default-*.ppc64.rpm* *ofed-kmp-ppc64-*.ppc64.rpm* *opensm-*.ppc64.rpm* *opensm-32bit-*.ppc64.rpm* *libcxgb3-rdmav2-*.ppc64.rpm* *libcxgb3-rdmav2-32bit-*.ppc64.rpm* *libehca-rdmav2-*.ppc64.rpm* *libehca-rdmav2-32bit-*.ppc64.rpm* *libibcm-*.ppc64.rpm* *libibcm-32bit-*.ppc64.rpm* *libibcommon1-*.ppc64.rpm* *libibcommon1-32bit-*.ppc64.rpm* *libibmad1-*.ppc64.rpm* *libibmad1-32bit-*.ppc64.rpm* *libibumad1-*.ppc64.rpm* *libibumad1-32bit-*.ppc64.rpm* *libibverbs-*.ppc64.rpm* *libibverbs-32bit-*.ppc64.rpm* *libibverbs-devel-*.ppc64.rpm* *libibverbs-devel-32bit-*.ppc64.rpm* *libipathverbs-*.ppc64.rpm* *libipathverbs-32bit-*.ppc64.rpm* *libmlx4-rdmav2-*.ppc64.rpm* *libmlx4-rdmav2-32bit-*.ppc64.rpm* *libmthca-rdmav2-*.ppc64.rpm* *libmthca-rdmav2-32bit-*.ppc64.rpm* *librdmacm-*.ppc64.rpm* *librdmacm-32bit-*.ppc64.rpm* *libsdp-*.ppc64.rpm* *libsdp-32bit-*.ppc64.rpm*

	mpi-selector-.ppc64.rpm* *mstflint-*.ppc64.rpm* *glibc-devel-*.ppc64.rpm* *glibc-devel-32bit-*.ppc64.rpm* *linux-kernel-headers-*.noarch.rpm* *kernel-default-*.ppc64.rpm* *kernel-default-base-*.ppc64.rpm* *(Note: libibverbs-devel-*.ppc64.rpm and libibverbs-devel-32bit-*.ppc64.rpm are in SLES 11 SDK ISO)*
SLES10	*libcxgb3-64bit-*.ppc.rpm* *libcxgb3-devel-*.ppc.rpm* *libcxgb3-devel-64bit-*.ppc.rpm* *libehca-*.ppc.rpm* *libehca-64bit-*.ppc.rpm* *libehca-devel-*.ppc.rpm* *libehca-devel-64bit-*.ppc.rpm* *libibcm-*.ppc.rpm* *libibcm-64bit-*.ppc.rpm* *libibcm-devel-*.ppc.rpm* *libibcm-devel-64bit-*.ppc.rpm* *libibcommon-*.ppc.rpm* *libibcommon-64bit-*.ppc.rpm* *libibcommon-devel-*.ppc.rpm* *libibcommon-devel-64bit-*.ppc.rpm* *libibmad-*.ppc.rpm* *libibmad-64bit-*.ppc.rpm* *libibmad-devel-*.ppc.rpm* *libibmad-devel-64bit-*.ppc.rpm* *libibumad-*.ppc.rpm* *libibumad-64bit-*.ppc.rpm* *libibumad-devel-*.ppc.rpm* *libibumad-devel-64bit-*.ppc.rpm* *libibverbs-*.ppc.rpm* *libibverbs-64bit-*.ppc.rpm* *libibverbs-devel-*.ppc.rpm*

	libibverbs-devel-64bit-.ppc.rpm* *libipathverbs-*.ppc.rpm* *libipathverbs-64bit-*.ppc.rpm* *libipathverbs-devel-*.ppc.rpm* *libipathverbs-devel-64bit-*.ppc.rpm* *libmlx4-*.ppc.rpm* *libmlx4-64bit-*.ppc.rpm* *libmlx4-devel-*.ppc.rpm* *libmlx4-devel-64bit-*.ppc.rpm* *libmthca-*.ppc.rpm* *libmthca-64bit-*.ppc.rpm* *libmthca-devel-*.ppc.rpm* *libmthca-devel-64bit-*.ppc.rpm* *librdmacm-1.0.6-*.ppc.rpm* *librdmacm-64bit-*.ppc.rpm* *librdmacm-devel-*.ppc.rpm* *librdmacm-devel-64bit-*.ppc.rpm* *libsdp-*.ppc.rpm* *libsdp-64bit-*.ppc.rpm* *libsdp-devel-*.ppc.rpm* *libsdp-devel-64bit-*.ppc.rpm* *mpi-selector-*.ppc.rpm* *mstflint-*.ppc.rpm* *mvapich2-*.ppc.rpm* *mvapich2-64bit-*.ppc.rpm* *mvapich2-devel-*.ppc.rpm* *mvapich2-devel-64bit-*.ppc.rpm* *ofed-1.3-*.ppc.rpm* *ofed-cxgb3-NIC-kmp-ppc64-*.ppc.rpm* *ofed-doc-*.ppc.rpm* *ofed-kmp-ppc64-*.ppc.rpm* *open-iscsi-*.ppc.rpm* *opensm-*.ppc.rpm* *opensm-64bit-*.ppc.rpm* *opensm-devel-*.ppc.rpm*

	opensm-devel-64bit-.ppc.rpm* *perftest-*.ppc.rpm* *qlvnictools-*.ppc.rpm* *rds-tools-*.ppc.rpm* *release-notes-as-*.ppc.rpm* *ruby-*.ppc.rpm* *sdpnetstat-*.ppc.rpm* *srptools-*.ppc.rpm* *tvflash-*.ppc.rpm*