XCAT 2 on AIX

Booting AIX diskless nodes

2010／9／12, AM 08:48:30

1.0 Overview

This "How-To" describes how AIX diskless nodes can be deployed and updated using xCAT and AIX/NIM commands.

NIM (Network Installation Management) is an AIX tool that enables a cluster administrator to centrally manage the installation and configuration of AIX and optional software on machines within a networked environment. This document assumes you are somewhat familiar with NIM. For more information about NIM, see the IBM AIX Installation Guide and Reference. (http://www-03.ibm.com/servers/aix/library/index.html)

The process described below is one basic set of steps that may be used to boot AIX diskless nodes and is not meant to be a comprehensive guide of all the available xCAT or NIM options.

Before starting this process it is assumed you have completed the following.

- An AIX system has been installed to use as an xCAT management node.
- xCAT and prerequisite software has been installed on the management node.
- The xCAT management node has been configured.
- One or more LPARs have already been created using the HMC interfaces.
- The cluster management network has been set up. (The Ethernet network that will be used to do the network installation of the cluster nodes.)

2.0 Deploying AIX diskless nodes using xCAT

2.1 Create a diskless image

In order to boot a diskless AIX node using xCAT and NIM you must create an xCAT *osimage* definition as well as several NIM resources.

The **mknimimage** will handle all the NIM setup as well as the creation of the xCAT *osimage* definition. It will not attempt to reinstall or reconfigure NIM if that process has already been completed. See the **mknimimage man** page for additional details.

Note: For various reasons it is recommended that you make sure that the primary hostname of the management node is the interface that you will be using to install the nodes. If you do this before you configure NIM then NIM will automatically use it to define the NIM primary network. This will mean that you will not have to

create any additional NIM network definitions and could avoid additional complications.

If you wish to install and configure NIM manually you can run the AIX **nim_master_setup** command (Ex. "*nim_master_setup -a mk_resource=no -a device=<source directory>*").

By default, the **mknimimage** command will create the NIM resources in subdirectories of /install. Some of the NIM resources are quite large (1-2G) so it may be necessary to increase the files size limit.

For example, to set the file size limit to "unlimited" for the user "root" you could run the following command.

/usr/bin/chuser fsize=-1 root

There are several NIM resources that must be created in order to deploy a diskless node. The main resource is the NIM SPOT (Shared Product Object Tree). An AIX diskless image is essentially a SPOT. It provides a **/usr** file system for diskless nodes and a root directory whose contents will be used for the initial diskless nodes root directory. It also provides network boot support.

When you run the command you must provide a source for the installable images. This could be the AIX product media, a directory containing the AIX images, or the name of an existing NIM *lpp_source* resource. You must also provide a name for the *osimage* you wish to create. This name will be used for the NIM SPOT resource that is created as well as the name of the xCAT *osimage* definition. The naming convention for the other NIM resources that are created is the *osimage* name followed by the NIM resource type, (ex. " *61cosi_lpp_source*").

Stateful or stateless?

You can choose to have your diskless nodes be either "stateful" or "stateless". If you want a "stateful" node you must use a NIM "root" resource. If you want a "stateless" node you must use a NIM "shared_root" resource.

A "stateful" diskless node preserves its state in individual mounted root filesystems. When the node is shut down and rebooted, any information that was written to a root filesystem will be available

A "stateless " diskless node uses a mounted root filesystem that is shared with other nodes. When it writes to the root directory the information is actually written to memory. If the node is shut down and rebooted any data that was written is lost. Any node-specific information must be re-established when the node is booted.

The advantage of stateless nodes is that there is much less network traffic and fewer resources used which is especially important in a large cluster environment.

For more information regarding the NIM "root" and "shared_root" resource refer to the NIM documentation.

If you wish to set up stateless cluster nodes you must use the "-r" option when you run the **mknimimage** command. The default behavior would be to set up stateful nodes.

For example, to create a stateless-diskless osimage called "*61cosi*" using the software contained in the */myimages* directory you could issue the following command.

mknimimage -V -r -t diskless -s /myimages 61cosi

(Note that this operation could take a while to complete!)

Caution: Do not interrupt (kill) a NIM process while it is creating a SPOT resource.

Starting with xCAT version 2.5 you can also use the "-D" option to specify that a dump resource should be created. See the section called "ISCSI dump support" below for details.

Note: To populate the */myimages* directory you could copy the software from the AIX product media using the AIX gencopy command. For example you could run "*gencopy -U -X -d /dev/cd0 -t /myimages all*".

The **mknimimage** command will display a summary of what was created when it completes. For example:

Object name: 61cosi
imagetype=NIM
lpp_source=61cosi_lpp_source
nimtype=diskless
osname=AIX
paging=61cosi_paging
shared_root=61cosi_shared_root
spot=61cosi

The NIM resources will be created in a subdirectory of */install/nim* by default. You can use the "-l" option to specify a different location.

You can also specify alternate or additional resources on the command line using the "attr=value" option, ("<nim resource type>=<resource name>"). For example, if you want to include a "*resolv_conf*" resource named "*61cosi_resolv_conf"* you could run the command as follows.

mknimimage -V -r -t diskless -s /myimages 61cosi resolv_conf=61cosi_resolv_conf

Any additional NIM resources specified on the command line must be previously created using NIM interfaces. (Which means NIM must already have been configured previously.)

Note: Another alternative is to run **mknimimage** without the additional resources and then simply add them to the xCAT *osimage* definition later. You can add or change the *osimage* definition at any time. When you initialize and install the nodes xCAT will use whatever resources are specified in the *osimage* definition.

The xCAT *osimage* definition can be listed using the **lsdef** command, modified using the **chdef** command and removed using the **rmnimimage** command. See the **man** pages for details.

To get details for the NIM resource definitions use the AIX **lsnim** command. For example, if the name of your SPOT resource is "*61cosi*" then you could get the details by running:

lsnim -l 61cosi

To see the actual contents of a NIM resource use "*nim -o showres <resource name>*". For example, to get a list of the software installed in your SPOT you could run:

nim -o showres 61cosi

2.2 Update the image (SPOT)

You must install any additional software you need and make any required customizations to the image before you boot the nodes.

2.2.1 Update options

There are basically three types of updates you can do to a SPOT.

2.2.1.1 Add or update software

The SPOT created in the previous step should be considered the basic minimal diskless AIX operating system image. It does not contain all the software that would normally be installed as part of AIX, if you were installing a standalone system from the AIX product media. (The "nim -o showres ..." command mentioned above will display what software is contained in the SPOT.)

You can use the **mknimimage** "-u" option to update software in the diskless image (SPOT).

To do this you can update your xCAT *osimage* definition with any "installp_bundles", "otherpkgs" you wish to use and then run the **mknimimage -u** command.

Use the **chdef** command to update the osimage definition. For example:

chdef -t osimage -o 61cosi installp_bundle="hpcbnd,labbnd"

Once the *osimage* definition is updated you can run **mknimimage**.

mknimimage -u 61cosi

Note: You can also specify "installp_bundle" and "otherpkgs" on the command line. However in this case you wouldn't have a record of what software was added to the SPOT.

2.2.1.2 Update system configuration files.

You can also add other configuration files such as /etc/passwd etc. These files will then be available to every node that boots with this image.

This can be done manually or by setting the “synclists” attribute of the *osimage* definition to point to a synclists file. This file contains a list of configuration files etc. that you wish to have copied into the SPOT.

For more information on using the synchronization file function see the document called “How to sync files in xCAT” (http://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2SyncFilesHowTo.pdf)

Use the **chdef** command to update the *osimage* definition. For example:

chdef -t osimage -o 61cosi synclists=”/u/test/mysyncfiles”

Once the *osimage* definition is updated you can run **mknimimage**.

mknimimage -u 61cosi

Note: You can do BOTH the software updates and configuration file updates at the same time with one call to **mknimimage**.

2.2.1.3 Run commands in the SPOT using chroot.

Starting with xCAT 2.5 and AIX 6.1.6 the **xcatchroot** command can be used to modify the SPOT using the **chroot** command.

The **xcatchroot** command will take care of any of the required setup so that the command you provide will be able to run in the spot **chroot** environment. It will also mount the lpp_source resource listed in the *osimage* definition so that you can access additional software that you may wish to install.

For example, to set the root password to "cluster" in the spot so that when the diskless node boots it will have a root password set you could run a command similar to the following.

xcatchroot -i 61cosi "/usr/bin/echo root:cluster | /usr/bin/chpasswd -c"

See the **xcatchroot** man page for more details.

Caution:

Be very careful when using **chroot** on a SPOT. It is easy to get the SPOT into an unusable state! It may be adviseable to make a copy of the SPOT before you try to run any commands that have an uncertain outcome.

When you are done updating a NIM spot resource you should always run the NIM check operation on the spot.

nim -Fo check 61cosi

For more information on updating a diskless image see the document called “Updating AIX cluster nodes” (http://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2onAIXUpdates.pdf)

See the section titled “Updating diskless nodes”

2.2.2 Adding required software

You will have to install *openssl* and *openssh* along with several additional requisite software packages.

The basic process is:

- Copy the required software to the lpp_source resource that you used to create your SPOT.
- Create NIM installp_bundle resources
- Check the lpp_source and bundle files
- Install the software in the SPOT.

2.2.2.1 Copy the software.

You will have to update the SPOT with additional software required for xCAT.

The required software is specified in the sample bundle file discussed below. The **installp** filesets should be available from the AIX product media. The prerequisite rpms are available in the dep-aix-<version>.tar.gz tar file that you downloaded from the xCAT download page.

The required software must be copied to the NIM lpp_source resource that is being used for this OS image. The easiest way to do this is to use the “nim -o update” command.

For example, assume the dep-aix* .tar.gz file has been copied and unwrapped in the /tmp/images directory and that the name of the NIM lpp_source resource is “61cosi_lpp_source”.

In more recent versions of the dep-aix* tar file the software will be found in subdirectories corresponding to the level of AIX you are using. (ex. ./dep-aix/5.3, ./dep-aix/6.1).

Note: Typically all the rpms are copied to the lpp_source resource even though they are not all used when installing a compute node.

For example, to copy all the rpms from the dep-aix package you could run the following command.

nim -o update -a packages=all -a source=/tmp/images/dep-aix/6.1 61cosi_lpp_source

The NIM command will find the correct directories and update the lpp_source resource.

2.2.2.2 Create NIM installp_bundle resources

To get all this additional software installed we need a way to tell NIM to include it in the installation. To facilitate this, xCAT provides sample NIM installp bundle files.

(Always make sure that the contents of the bundle files you use are the packages you want to install and that they are all in the appropriate lpp_source directory.)

Starting with xCAT version 2.4.3 there will be a set of bundle files to use for installing a compute node. They are in "*/opt/xcat/share/xcat/installp_bundles*". There is a version corresponding to the different AIX OS levels. (xCATaixCN53.bnd, xCATaixCN61.bnd etc.) Just use the one that corresponds to the version of AIX you are running.

Note: For earlier versions of xCAT the sample bundle files are shipped as part of the xCAT tarball file.

To use the bundle file you need to define it as a NIM resources and add it to the xCAT *osimage* definition.

Copy the bundle file (say xCATaixCN61.bnd) to a location where it can be defined as a NIM resource, for example "/install/nim/installp_bundle".

To define the NIM resource you can run the following command.

nim -o define -t installp_bundle -a server=master -a location= /install/nim/installp_bundle/xCATaixCN61.bnd xCATaixCN61

To add this bundle resource to your xCAT osimage definition run:

chdef -t osimage -o 610SNimage installp_bundle="xCATaixSN61"

2.2.2.3 Check the osimage (optional)

This command is available in xCAT 2.4.3 and beyond.

To avoid potential problems when installing a node it is adviseable to verify that all the software that you wish to install has been copied to the appropriate NIM lpp_source directory.

Any software that is specified in the "otherpkgs" or the "installp_bundle" attributes of the osimage definition must be available in the lpp_source directories.

Also, if your bundle files include rpm entries that use a wildcard (*) you must make sure the lpp_source directory does not contain multiple packages that will match that entry. (NIM will attempt to install multiple version of the same package and produce an error!)

To find the location of the lpp_source directories run the "lsnim -l <lpp_source_name>" command. For example:

lsnim -l 610image_lpp_source

If the location of your lpp_source resource is "/install/nim/lpp_source/610image_lpp_source/" then you would find rpm packages in "/install/nim/lpp_source/610image_lpp_source/RPMS/ppc" and you would find your installp and emgr packages in "/install/nim/lpp_source/610image_lpp_source/installp/ppc".

To find the location of the installp_bundle resource files you can use the NIM "lsnim -l" command. For example,

lsnim -l xCATaixSSH

Starting with xCAT version 2.4.3 you can use the xCAT **chkosimage** command to do this checking. For example:

chkosimage -V 61cosi

See the **chkosimage** man page for details.

2.2.2.4 Install the software into the SPOT.

You can install and update software in a NIM SPOT resource by using NIM commands directly or you can use the xCAT **mknimimage** command.

If you wish to use the NIM support directly you can check the NIM documentation for information on the NIM "cust" and "maint" operations.

In this example the xCAT **mknimimage** command will be used.

Run the **mknimimage** command to update the SPOT using the information saved in the xCAT "61cosi" osimage definition.

mknimimage -u 61cosi

See the **mknimimage** man page for more options that are available for updating diskless images (SPOT resources).

Note: You cannot update a SPOT that is currently allocated. To check to see if the SPOT is allocated you could run the following command.

lsnim -l <spot name>

2.2.3 Set up statelite support (for diskless-stateless nodes only)

This support is available in xCAT version 2.5 and beyond.

The xCAT *statelite* support for AIX provides the ability to "overlay" specific files or directories over the standard diskless-stateless support.

There is a complete description of the statelite support in (http://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2onAIX.pdf).

See the section titled "Using AIX statelite support".

To set up the *statelite* support you must:

1. fill in one or more to the *statelite* tables in the xCAT database.
2. Run the "**mknimimage -u**" command which will use that information to modify the SPOT resource.

Note: You could also fill in the *statelite* tables before initially running the **mknimimage** to create the *osimage*. (Rather than doing the setup later with the "-u" option.)

2.3 Define xCAT networks

Create a network definition for each network that contains cluster nodes. You will need a name for the network and values for the following attributes.

net	The network address.
mask	The network mask.
gateway	The network gateway.

This "How-To" assumes that all the cluster node management interfaces and the xCAT management node interface are on the same network. You can use the xCAT **mkdef** command to define the network.

For example:

mkdef -t network -o net1 net=9.114.113.0 mask=255.255.255.224 gateway=9.114.113.254

Note: NIM also requires network definitions. When NIM was configured in an earlier step the default NIM master network definition was created. The NIM definition should match the one you create for xCAT. If multiple cluster subnets are needed then you will need an xCAT and NIM network definition for each one. A future xCAT enhancement will simplify this by automatically creating NIM network definitions based on the xCAT definitions.

2.4 Create additional NIM network definitions (optional)

For the processs described in this document we are assuming that the xCAT management node and the LPARs are all on the same network.

However, depending on your specific situation, you may need to create additional NIM network and route definitions.

NIM network definitions represent the networks used in the NIM environment. When you configure NIM, the primary network associated with the NIM master is automatically defined. You need to define additional networks only if there are nodes that reside on other local area networks or subnets. If the physical network is changed in any way, the NIM network definitions need to be modified.

To create the NIM network definitions corresponding to the xCAT network definitions you can use the xCAT **xcat2nim** command.

For example, to create the NIM definitions corresponding to the xCAT "clstr_net" network you could run the following command.

xcat2nim -V -t network -o clstr_net

Manual method

The following is an example of how to define a new NIM network using the NIM command line interface.

Step 1

Create a NIM network definition. Assume the NIM name for the new network is "clstr_net", the network address is "10.0.0.0", the network mask is "255.0.0.0", and the default gateway is "10.0.0.247".

nim -o define -t ent -a net_addr=10.0.0.0 -a snm=255.0.0.0 -a routing1='default 10.0.0.247' clstr_net

Step 2

Create a new interface entry for the NIM "master" definition. Assume that the next available interface index is "2" and the hostname of the NIM master is "xcataixmn". This must be the hostname of the management node interface that is connected to the "clstr_net" network.

nim -o change -a if2='clstr_net xcataixmn 0' -a cable_type2=N/A master

Step 3

Create routing information so that NIM knows how to get from one network to the other. Assume the next available routing index is "2", and the IP address of the NIM master on the "master_net" network is "8.124.37.24". Assume the IP address on the NIM master on the "clstr_net" network is " 10.0.0.241". This command will set the route from "master_net" to "clstr_net" to be " 10.0.0.241" and it will set the route from "clstr_net" to "master_net" to be "8.124.37.24".

nim -o change -a routing2='master_net 10.0.0.241 8.124.37.24' clstr_net

Step 4

Verify the definitions by running the following commands.

lsnim -l master

lsnim -l master_net

lsnim -l clstr_net

See the NIM documentation for details on creating additional network and route definitions. (*IBM AIX Installation Guide and Reference*. http://www-03.ibm.com/servers/aix/library/index.html)

2.5 Define the HMC as an xCAT node

The xCAT hardware control support requires that the hardware control point for the nodes also be defined as a cluster node.

The following command will create an xCAT node definition for an HMC with a host name of "*hmc01*". The *groups, nodetype, mgt, username*, and *password* attributes **must** be set.

mkdef –t node –o hmc01 groups="all" nodetype=hmc mgt=hmc username=hscroot password=def456

2.6 Discover the LPARs managed by the HMC

This step assumes that the LPARs were already created using the standard HMC interfaces.

Use the xCAT **rscan** command to gather the LPAR information. In this example we will use the "-z" option to create a stanza file that contains the information gathered by **rscan** as well as some default values that could be used for the node definitions.

To write the stanza format output of **rscan** to a file called "*mystanzafile"* run the following command.

rscan –z hmc01 > mystanzafile

This file can then be checked and modified as needed. For example you may need to add a different name for the node definition or add additional attributes and values etc.

Note: The stanza file will contain stanzas for objects other than the LPARs. This information must also be defined in the xCAT database. It is not necessary to modify the non-LPAR stanzas in any way.

2.7 Define xCAT cluster nodes

The information gathered by the **rscan** command can be used to create xCAT node definitions.

Since we have put all the node information in a stanza file we can now pass the contents of the file to the **mkdef** command to add the definitions to the database.

cat mystanzafile | mkdef –z

You can use the xCAT **lsdef** command to check the definitions (ex. "*lsdef –l node01*"). After the node has been defined you can use the **chdef** command to make any additional updates to the definitions, if needed.

2.8 Add IP addresses and hostnames to /etc/hosts

Make sure all node hostnames are added to /etc/hosts. Refer to the section titled "Add cluster nodes to the /etc/hosts file" in the following document for details. (http://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2onAIX.pdf)

2.9 Create and define additional logical partitions (optional)

You can use the xCAT **mkvm** command to create additional logical partitions for diskless nodes in some cases.

This command can be used to create new partitions based on an existing partition or it can replicate the partitions from a source CEC to a destination CEC.

The first form of the **mkvm** command creates new partition(s) with the same profile/ resources as the partition specified on the command line. The starting numeric partition number and the *noderange* for the newly created partitions must also be provided. The LHEA port numbers and the HCA index numbers will be automatically increased if they are defined in the source partition.

The second form of this command duplicates all the partitions from the specified source CEC to the destination CEC. The source and destination CECs can be managed by different HMCs.

The nodes in the *noderange* must already be defined in the xCAT database. The "mgt" attribute of the node definitions must be set to "hmc".

For example, to create a set of new nodes. (The "groups" attribute is required.)

mkdef -t node -o clstrn04-clstrn10 groups=all,aixnodes mgt=hmc

To create several new partitions based on the partition for node clstrn01.

mkvm -V clstn01 -i 4 -n clstrn04-clstrn10

See the **mkvm** man page for more details.

2.10 Gather MAC information for the boot adapters.

Use the xCAT **getmacs** command to gather adapter information from the nodes. This command will return the MAC information for each Ethernet adapter available on the target node. The command can be used to either display the results or write the information directly to the database. If there are multiple adapters the first one will be written to the database and used as the install adapter for that node.

The command can also be used to do a ping test on the adapter interfaces to determine which ones could be used to perform the network boot. In this case the first adapter that can be successfully used to ping the server will be written to the database.

Before running **getmacs** you must first run the **makeconservercf** command. You need to run **makeconservercf** any time you add new nodes to the cluster.

makeconservercf

To retrieve the MAC address for all the nodes in the group "*aixnodes*" and write the first adapter MAC to the xCAT database you could issue the following command.

getmacs aixnodes

To display all adapter information but not write anything to the database.

getmacs -d aixnodes

To retrieve the MAC address and do a ping test to determine which adapter MAC to use for the node you could issue the following command. (The ping operation may take a while to complete.)

getmacs -d aixnodes -S 10.14.0.2 -G 10.14.0.2 -C 10.14.0.4

The output would be similar to the following.

 # Type Location Code MAC Address Full Path Name Ping Result Device Type

ent U9125.F2A.024C362-V6-C2-T1 fef9dfb7c602 /vdevice/l-lan@30000002 successful virtual

ent U9125.F2A.024C362-V6-C3-T1 fef9dfb7c603 /vdevice/l-lan@30000003 unsuccessful virtual

From this result you can see that "*fef9dfb7c602*" should be used for this nodes MAC address.

For more information on using the **getmacs** command see the man page.

To manually add a MAC value to the node definition you can use the **chdef** command. For example:

chdef -t node node01 mac=fef9dfb7c603

2.11 Define xCAT groups (optional)

XCAT supports both *static* and *dynamic* node groups. See the section titled "xCAT node group support" in the "xCAT2 Top Doc" document for details on using xCAT groups. (http://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2top.pdf)

2.12 Set up post boot scripts (optional)

xCAT supports the running of customization scripts on the nodes when they are installed. For diskless nodes these scripts are run when the /etc/inittab file is processed during the node boot up.

This support includes:

- The running of a set of default customization scripts that are required by xCAT.

  You can see what scripts xCAT will run by default by looking at the "xcatdefaults" entry in the xCAT "postscripts" database table. (I.e. Run "tabdump postscripts".). You can change the default setting by using the xCAT **chtab** or **tabedit** command. The scripts are contained in the /install/postscripts directory on the xCAT management node.

- The optional running of customization scripts provided by xCAT.

  There is a set of xCAT customization scripts provided in the /install/postscripts directory that can be used to perform optional tasks such as additional adapter configuration. (See the "configiba" script for example.)

- The optional running of user-provided customization scripts.

  To have your script run on the nodes:

  1. Put a copy of your script in /install/postscripts on the xCAT management node. (Make sure it is executable.)
  2. Set the "postscripts" attribute of the node definition to include the comma separated list of the scripts that you want to be executed on the nodes. The order of the scripts in the list determines the order in which they will be run. For example, if you want to have your two scripts called "foo" and "bar" run on node "node01" you could use the **chdef** command as follows.

     chdef -t node -o node01 -p postscripts=foo,bar

     (The "-p" means to add these to whatever is already set.)

Note: The customization scripts are run during the boot process (out of /etc/inittab).

2.13 Set up prescripts (optional)

This support will be available in xCAT 2.5 and beyond.

The xCAT *prescript* support is provided to to run user-provided scripts during the node initialization process. These scripts can be used to help set up specific environments on the servers that handle the cluster node deployment. The scripts will run on the install server for the nodes. (Either the management node or a service node.) A different set of scripts may be specified for each node if desired.

One or more user-provided prescripts may be specified to be run either at the beginning or the end of node initialization. The node initialization on AIX is done either by the **nimnodeset** command (for diskfull nodes) or the **mkdsklsnode** command (for diskless nodes.)

For more information about using the xCAT prescript support refer to the des cription earlier in this documant and also the "xCAT2 Top Doc", (

http://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2top.pdf)

2.14 Verify the node definitions

Verify that the node definitions include the required information.

To get a listing of the node definitions you can use the **lsdef** command. For example, to display the definitions of all nodes in the group "aixnodes" you could run the following command.

lsdef -t node -l -o aixnodes

The output for one diskless node might look something like the following:

```
Object name: clstrn02
   cons=hmc
   groups=lpar,all
   hcp=clstrhmc01
   hostnames=clstrn02.mycluster.com
   id=2
   ip=10.1.3.2
   mac=001a64f9bfc9
   mgt=hmc
   nodetype=lpar,osi
   os=AIX
   parent=clstrf1fsp03-9125-F2A-SN024C352
   pprofile=compute
```

Most of these attributes should have been filled in automatically by xCAT.

Note: xCAT supports many different cluster environments and the attributes that may be required in a node definition will vary. For diskless nodes the node definition should include at least the attributes listed in the above example.

2.15 Initialize the AIX/NIM diskless nodes

You can set up NIM to support a diskless boot of nodes by using the xCAT **mkdsklsnode** command. This command uses information from the xCAT database and default values to run the appropriate NIM commands.

For example, to set up all the nodes in the group "*aixnodes*" to boot using the SPOT (COSI) named "*61cosi*" you could issue the following command.

mkdsklsnode -i 61cosi aixnodes

The command will define and initialize the NIM machines. It will also set the "*provmethod*" attribute in the xCAT node definitions to "*61cosi "*.

Starting with xCAT version 2.5 you can also to specify the "*configdump"* attribute to specify the type of system do to configure. See the section called "ISCSI dump support" below for details.

To verify that NIM has allocated the required resources for a node and that the node is ready for a network boot you can run the "**lsnim –l**" command. For example, to check node "*node01*" you could run the following command.

lsnim -l node01

2.16 Verifying the node initialization before booting (optional)

Once the **mkdsklsnode** command completes you can check several things to see if it has been initialized correctly.

- The /etc/bootptab and /etc/exports files.
- The <nodename>.info file in the /tftpboot directory.
- The NIM node definition for the node. ("*lsnim -l node01")*

2.17 Open a remote console (optional)

You can open a remote console to monitor the boot progress using the xCAT **rcons** command. This command requires that you have **conserver** installed and configured.

If you wish to monitor a network installation you must run rcons before initiating a network boot.

To configure conserver run:

makeconservercf

To start a console:

rcons node01

Note: You must always run makeconservercf after you define new cluster nodes.

2.18 Initiate a network boot

Initiate a remote network boot request using the xCAT **rnetboot** command. For example, to initiate a network boot of all nodes in the group "*aixnodes*" you could issue the following command.

rnetboot aixnodes

Starting with xCAT version 2.5 you can also specify the "-I" option to specify the iscsi boot option. ("*rnetboot -I aixnodes*") See the section called "ISCSI dump support" below for details.

Note: If you receive timeout errors from the **rnetboot** command, you may need to increase the default 60-second timeout to a larger value by setting ppctimeout in the site table:

chdef -t site -o clustersite ppctimeout=180

2.19 Verify the deployment

- You can use the AIX **lsnim** command to see the state of the NIM installation for a particular node, by running the following command on the NIM master:

  lsnim -l <clientname>

- Retry and troubleshooting tips:
  - For p6 lpars, it may be helpful to bring up the HMC web interface in a browser and watch the lpar status and reference codes as the node boots.
  - Verify network connections
  - If the **rnetboot** returns "unsuccessful" for a node, verify that bootp and tftp is configured and running properly.
    - For bootp, view */etc/bootptab* to make sure an entry exists for the node.
    - For dhcp, view /var/lib/dhcp/db/dhcpd.leases to make sure an entry exists for the node.
    - Verify that the information in /tftpboot/<node>.info is correct.
    - Stop and restart inetd:
      stopsrc -s inetd
      startsrc -s inetd
    - Stop and restart tftp:
      stopsrc -s tftp
      startsrc -s tftp
    -
  - Verify NFS is running properly and mounts can be performed with this NFS server:
    - View */etc/exports* for correct mount information.
    - Run the **showmount** and **exportfs** commands.
    - Stop and restart the NFS and related daemons:
      stopsrc -g nfs
      startsrc -g nfs
    - Attempt to mount a file system from another system on the network.
  - If the **rnetboot** operation is successful, but **lsnim** shows that the node is stuck at one of the netboot phases, you may need to redo your NIM definitions.

3.0 ISCSI dump support

Starting with xCAT version 2.5 you can configure remote system dump support for diskless nodes. To set up diskless dump support you must define the NIM dump

resource and allocate the resource to the diskless node. The dump resource appears as an iSCSI disk to the NIM client and can only be used to configure the primary dump device. You can only configure firmware-assisted system dumps on primary dump devices.

When a dump resource is allocated to a client node, NIM creates a subdirectory identified by the client's name for the client's exclusive use. After initialization, the client uses this directory to store any dump images it creates. For example, "/install/nim/dump/61dskls_dump/node05".

See the AIX/NIM documentation for more information on the diskless (thin server) dump support.

3.1 Prerequisites

The xCAT support for diskless system dump has the following prerequisites.

- xCAT version 2.5 or higher
- Hardware: POWER6 or later
- Operating system version: AIX 6.1.6 or above
- Required firmware: pfw350
- Software prereqs: devices.tmiscw (Available from the AIX Expansion Pack)

3.2 Configuring diskless dump

The basic process for configuring diskless dump support is as follows:

1. Install prerequisite software on the xCAT management node

   The AIX ISCSI dump support requires the devices.tmiscw fileset. This is currently available from the AIX Expansion Pack.

   Use the standard AIX interfaces to install the software. (SMIT, installp, geninstall etc.) (devices.tmiscw.rte)

   Note: This software must also be installed on any xCAT AIX service nodes that are being used.

2. Create a NIM dump resource

   If you are creating a new diskless *osimage* then you can have the dump resource created automatically when you run **mknimimage**. See the **mknimimage** man page for more details.

   For example,

   mknimimage -V -r -D -t diskless -s 61rte_lpp_source 61dskls max_dumps=1

   The "-D" option specifies that a dump resource should be created.

If you already have created a diskless *osimage* then you can just create the NIM dump resource manually and add the name to the xCAT *osimage* definition.

To create a NIM dump resource you could run a command similar to the folowing:

nim -o define -t dump -a server=master -a
location=/install/nim/61dskls_dump
-a max_dumps=1 61dskls_dump

To add this to the *61dskls* osimage definition you could run the following:

chdef -t osimage -o 61dskls dump=61dskls_dump

3. Define and initialize the diskless client node

   Use the xCAT **mkdsklsnode** command to define and initialize a NIM diskless machine.

   For example:

   mkdsklsnode -V -i 61dskls compute15 configdump=selective

   See the **mkdsklsnode** man page for more information on diskless dump options.

4. Boot the node

   Use the xCAT **rnetboot** command to boot the diskless node.

   rnetboot compute15

3.3 Initiating a diskless dump

System dumps are normally initiated when a fatal system error occurs, however, you can also initiate a system dump using the AIX **sysdumpstart** command.

For example, to initiate a system dump on a diskless node called "compute15" you could run the following:

xdsh compute15 "sysdumpstart -p"

The dump will take 2 or more hours to complete.

The dump file will be created in the NIM dump resource directory on the NIM master for the node.

For example,
"*/install/nim/dump/61dskls_dump/clstrn05/dump.2010.06.15.10:47:34.BZ*".

3.4 NIM snap and dump operations

Use the NIM "snap" operation to gather system configuration information. ("*nim –o snap –a snap_flags=<value> nodename*")

For example:

nim –o snap compute15

The snap file (ex. snap.pax.2010.02.17.11:47:38.Z) will be saved in the dump resource directory.

Use the NIM "**showdump**" operation to see what dump files are available for a node. You must run the **nim** command on the NIM master that is being used for the node. In an xCAT clauster this would either be the management node or an AIX service node. You can use the **xdsh** command to run the nim command on the service node.

nim -o showdump compute15

3.5 Reading a system dump file

Use the AIX **kdb** command to read dump files.

For example:

dmpuncompress dumpname.BZ
kdb ./dumpname /unix

The AIX **dmpuncompress** command restores the original dump files that were compressed at dump time. The compressed dump file is removed and replaced by an expanded copy. The expanded file has the same name as the compressed version, but without the .BZ

Note the "/unix" should be the one contained in the SPOT (ex. "*/install/nim/spot/<spotname>/usr//lib/boot/unix_64*")

See the **kdb** documentation for more details on how to use kdb with dump files.

4.0 Special Cases

4.1 Using other NIM resources.

When you run the **mknimimage** command to create a new xCAT *osimage* definition it will create default NIM resources and add their names to the *osimage* definition. It is also possible to specify additional or different NIM resources to use for the *osimage*. To do this you can use the "attr=val [attr=val ...]" option. These "attribute equals value" pairs are used to specify NIM resource types and names to use when creating the the xCAT *osimage* definition. The "attr" must be a NIM resource type and the "val" must be the name of a **previously defined** NIM resource of that type, (ie. "<nim_resource_type>=<resource_name>").

For example, to create a diskless image and include *tmp* and *home* resources you could issue the command as follows. This assumes that the *mytmp* and *myhome* NIM resources have already been created by using NIM commands directly.

mknimimage -t diskless -s /dev/cd0 611cosi tmp=mytmp home=myhome

These resources will be added to the xCAT *osimage* definition. When you initialize a node using this definition the **mkdsklsnode** command will include all the resources when running the "nim -o dkls_init" operation.

See the NIM documentation for more information on supported diskless resources.

4.2 Booting a "dataless" node.

AIX NIM includes support for "dataless" systems as well as "diskless". NIM defines a dataless machine as one that has some local disk space that could be used for paging space and optionally the /tmp and /home. If you wish to use dataless machines you can create an xCAT *osimage* definition for them with the **mknimimage** command. When creating the *osimage*, use the "-t" option to specify a type of "dataless".

For example, to create an *osimage* definition for "dataless" nodes you could run the command as follows.

mknimimage -s /dev/cd0 -t dataless 53cosi

When the node is initialized to use this image the **mkdsklsnode** command will run the "nim -o dtls_init .." operation.

See the NIM documentation for more information on the NIM support for dataless systems.

4.3 Specifying additional values for the NIM node initialization.

When you run the **mkdsklsnode** command to initialize diskless nodes the command will run the required NIM commands using some default values. If you wish to use different values you can specify them on the **mkdsklsnode** command line using the

"attr=val [attr=val ...]" option. See the **mkdsklsnode man** page for the details of what attributes and values are supported.

For example, when **mkdsklsnode** defines the diskless node there are default values set for the "speed"(100) and "duplex"(full) network settings. If you wish to specify a different value for "speed" you could run the command as follows.

mkdsklsnode -i myosimage mynode speed=1000

5.0 Cleanup

The NIM definitions and resources that are created by xCAT commands are not automatically removed. It is therefore up to the system administrator to do some clean up of unused NIM definitions and resources from time to time. (The NIM lpp_source and SPOT resources are quite large.) There are xCAT commands that can be used to assist in this process.

5.1 Removing NIM machine definitions

Use the xCAT **rmdsklsnode** command to remove all NIM machine definitions that were created for the specified xCAT nodes. This command will not remove the xCAT node definitions.

For example, to remove the NIM machine definition corresponding to the xCAT node named "node01" you could run the command as follows.

rmdsklsnode node01

The previous example assumes that the NIM machine definition is the same name as the xCAT node name. If you had used the "-n" option when you created the NIM machine definitions with **mkdsklsnode** then the NIM machine names would be a combination of the xCAT node name and the *osimage* name used to initialize the NIM machine. To remove these definitions you must provide the **rmdsklsnode** command with the name of the *osimage* that was used.

For example, to remove the NIM machine definition associated with the xCAT node named "node2" and the *osimage* named "61spot" you could run the following command.

rmdsklsnode -i 61spot node02

If the NIM machine is currently running or the machine definition was left in a bad state you can use the **rmdsklsnode** "-f" (force) option. This will stop the node and deallocate any resources it is using so the machine definition can be removed.

The **rmdsklsnode** command is intended to make it easier to clean up NIM machine definitions that were created by xCAT. You can also use the AIX **nim** command directly. See the AIX/NIM documentation for details.

5.2 Removing NIM resources

Use the xCAT **rmnimimage** command to remove all the NIM resources associated with a given xCAT *osimage* definition. The command will only remove a NIM resource if it is not allocated to a node. You should always clean up the NIM node definitions before attempting to remove the NIM resources. The command will also remove the xCAT *osimage* definition that is specified on the command line.

For example, to remove the "61spot" *osimage* definition along with all the associated NIM resources run the following command.

rmnimimage -x 61spot

If necessary, you can also remove the NIM definitions directly by using NIM commands. See the AIX/NIM documentation for details.

6.0 Notes

6.1 Terminology

image

The term "image" is used extensively in this document. The precise meaning of an "image" will vary depending on the context in which the term is being used. In general you can think of an image as the basic operating system image as well as other resources etc. that are needed to boot a node. In most cases in this document we will be referring to an image as either an xCAT *osimage* definition or an AIX/NIM diskless image (called a SPOT or COSI).

osimage - This is an xCAT object that can be used to describe an operation system image. This definition can contain various types of information depending on what will be installed on the node and how it will be installed. The image definition is not node specific and can be used to deploy multiple nodes. It contains all the information that will be needed by the underlying xCAT and NIM support to deploy the node.

COSI - A Common Operating System Image is the name used by AIX/NIM to refer to a SPOT resource. From the NIM perspective this would be an AIX diskless image.

diskless node

The operating system is not stored on local disk. For AIX systems this means the file systems are mounted from a NIM server. NIM also supports the concept of a *dataless* system which has some limited disk space that can be used for certain file systems. See the "Special Cases" section below for information on using additional NIM features.

diskful node

For AIX systems this means that the node has local disk storage that is used for the operating system. Diskfull AIX nodes are typically installed using the NIM **rte** or **mksysb** type install methods.

6.2 NIM diskless resources

The following list describes the required and optional resources that are managed by NIM to support diskless and dataless clients.

boot

Defined as a network boot image for NIM clients. The **boot** resource is managed automatically by NIM and is never explicitly allocated or deallocated by users.

SPOT

Defined as a directory structure that contains the AIX run-time files common to all machines. These files are referred to as the **usr** parts of the fileset. The **SPOT** resource is mounted as the **/usr** file system on diskless and dataless clients.

Contains the **root** parts of filesets. The **root** part of a fileset is the set of files that may be used to configure the software for a particular machine. These **root** files are stored in special directories in the **SPOT**, and they are used to populate the root directories of diskless and dataless clients during initialization.

The network boot images used to boot clients are constructed from software installed in the **SPOT**.

A **SPOT** resource is required for both diskless and dataless clients.

root

Defined as a parent directory for client "/" (**root**) directories. The client root directory in the **root** resource is mounted as the "/" (**root**) file system on the client.

When the resources for a client are initialized, the client **root** directory is populated with configuration files. These configuration files are copied from the **SPOT** resource that has been allocated to the same machine.

A **root** resource is required for both diskless and dataless clients.

This resource is managed automatically by NIM.

dump

Defined as a parent directory for client dump files. The client dump file in the **dump** resource is mounted as the dump device for the client.

A **dump** resource is required for both diskless and dataless clients.

This resource is managed automatically by NIM.

paging

Defined as a parent directory for client paging files. The client paging file in the **paging** resource is mounted as the paging device for the client.

A **paging** resource is required for diskless clients and optional for dataless clients.

This resource is managed automatically by NIM.

home

Defined as a parent directory for client **/home** directories. The client directory in the **home** resource is mounted as the **/home** file system on the client.

A **home** resource is optional for both diskless and dataless clients.

shared_home

Defined as a **/home** directory shared by clients. All clients that use a **shared_home** resource will mount the same directory as the **/home** file system.

A **shared_home** resource is optional for both diskless and dataless clients.

tmp

Defined as a parent directory for client **/tmp** directories. The client directory in the **tmp** resource is mounted as the **/tmp** file system on the client.

A **tmp** resource is optional for both diskless and dataless clients.

resolv_conf

This resource is a file that contains nameserver IP addresses and a network domain name.

It is copied to the **/etc/resolv.conf** file in the client's root directory.

A **resolv_conf** resource is optional for both diskless and dataless clients.

The AIX/NIM resources for diskless/dataless machines will remain allocated and the node will remain initialized until they are specifically unallocated and uninitialized.

6.3 NIM Commands

6.3.1 COSI commands

AIX provides commands that may be used to manage the SPOT (or COSI) resource. Refer to the AIX man pages for further details.

- **mkcosi** Create a COSI (SPOT) for a thin server (diskless or dataless client) to mount and use.

- **chcosi** Manages a Common Operating System Image (COSI).
- **cpcosi** Create a copy of a COSI (SPOT).
- **lscosi** List the properties of a COSI (SPOT).
- **rmcosi** Remove a COSI (SPOT) from the NIM environment.

6.3.2 Thin server commands

AIX provides several commands that can be used to manage diskless (also called thin server) nodes. See the AIX man pages for further details.

- **mkts** Create a thin server and all necessary resources.
- **lsts** List the status and software content of a thin server.
- **swts** Switch a thin server to a different COSI.
- **dbts** Perform a debug boot on a thin server.
- **rmts** Remove a thin server from the NIM environment.