xCAT 2 on AIX

Updating AIX cluster nodes

Date: 10/20/2009

1.0 Overview

There are various techniques that can be used to update the nodes of an xCAT cluster. This document describes some of the basic support that is provided for AIX nodes.

2.0 Updating diskfull nodes

2.1.1 Using the nimnodecust command

The xCAT **nimnodecust** command can be used to customize AIX/NIM standalone machines. This command uses underlying NIM support to perform the remote customization of AIX diskfull (standalone) nodes.

The software packages and/or updates that you wish to install on the nodes must be copied to the appropriate directory locations in the NIM lpp_source resource that you are using for the nodes you wish to update.

The easiest way to do this is to use the "nim -o update" command. For example, assume all the required software has been copied and unwrapped in the /tmp/images directory.

To add all the packages to the lpp_source resource named "*610SNimage_lpp_source*" you could run the following command:

```
nim -o update -a packages=all -a source=/tmp/images  610SNimage_lpp_source
```

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

When using the **nimnodecust** command the packages you wish to install on the nodes may be specified with either a comma-separated list of package names or by a comma-separated list of installp_bundle names.

For example, To install the installp package "openssh.base.server" on an xCAT node named "node01" assuming the software has been copied to the "610SNimage_lpp_source" lpp_source resource you could run the following command.

```
nimnodecust -s  610SNimage_lpp_source -p openssh.base.server node01
```

For more details on using the **nimnodecust** command see the corresponding man page.

2.1.2 Using the xdsh method

Another method for updating a diskfull node would be to mount a directory containing the updates on the node and use the **xdsh** command to run the appropriate **installp** or **rpm** command.

For example:
To mount a directory you could run something like-

```
xdsh <nodename>  "mount <servername>:/my-inst-images /mnt"
```

To install an **installp** fileset-

```
xdsh <nodename>  "installp –agQX -d /mnt <fileset name>"
```

To unmount the directory-

```
xdsh <nodename>  "umount /mnt"
```

3.0 Using the rolling update support

The **rollupdate** command creates and submits scheduler jobs that will notify xCAT to shutdown a group of nodes, run optional out-of-band commands from the xCAT management node, and reboot the nodes. Currently, only LoadLeveler is supported as a job scheduler with **rollupdate**.

Input to the **rollupdate** command is passed in as stanza data through STDIN. Information such as the sets of nodes that will be updated, the name of the job scheduler, a template for generating job command files, and other control data are required. See /opt/xcat/share/xcat/rollupdate/rollupdate.input.sample for stanza keywords, usage, and examples.

The **rollupdate** command will use the input data to determine each set of nodes that will be managed together as an update group. For each update group, a job scheduler command file is created and submitted. When the group of nodes becomes available and the scheduler runs the job, the job will send a message to the xCAT daemon on the management node to begin the update process for all the nodes in the update group. The nodes will be stopped by the job scheduler (for LoadLeveler, the nodes are drained), an operating system shutdown command will be sent to each node, out-of-band operations can be run on the management node, and the nodes are powered back on.

The **rollupdate** command assumes that, if the update is to include rebooting stateless nodes to a new operating system image, the image has been created and tested, and that all relevant xCAT commands have been run for the nodes such that the new image will be loaded when xCAT reboots the nodes.

See the **rollupdate** man page for usage details.

4.0 Updating AIX diskless nodes

This section describes how AIX diskless nodes can be updated using xCAT and AIX/NIM commands. It covers both the switching of the node to a completely different image or updated the current image. It is not meant to be an exhaustive presentation of all options that are available to xCAT/AIX system administrators.

To update an AIX diskless node with new or additional software you must modify the NIM SPOT resource (operating system image) that the node is using and then reboot the node with the new SPOT. You cannot install software on a running diskless node directly.

Since you cannot modify a SPOT while a node is using it, you have basically two options. You can either stop all the nodes and then update the existing OS image, or, you can create a new updated image to use to boot the nodes.

Stopping the nodes to do the updates means the nodes will be unusable for some period of time and there will be no easy way to return to the previous image if necessary. For these reasons the procedure described in this “How-To” will focus on creating a new image and rebooting the nodes with that image. The new image could be a completely new operating system image or it could be a copy of the the existing image that you can update as needed.

4.1 Create a new image

4.1.1 Create a new image from different source

In this case we create a new xCAT *osimage* definition with a new set of resources by running the xCAT **mknimimage** command with the source for the new resources. This is the same way you created the original xCAT *osimage* definition for the node.

When you run the command you must provide a source for the installable images. This can be the location of the source code or the name of another NIM *lpp_source* resource. You must also provide a name for the image you wish to create. This name will be used for the NIM SPOT resource definition as well as the xCAT *osimage* definition.

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

For example, to create a diskless image called "*61dskls*" using the AIX installation images in the */my-install-images* directory as the source you could issue the following command.

```
mknimimage -t diskless -s /my-install-images 61dskls
```

(This operation could take a while to complete!)

The command will create new NIM lpp_source and SPOT resources. It will also create dump, paging, and root resources if needed. A new xCAT *osimage* definition will also be created, (called "*61dskls"),* which will contain the names of these resources.

You could also use the name of an existing NIM lpp_source resource as the source of a new *osimage* definition. For example, you could use a resource created for a previous *osimage* called *61dskls_lpp* to create a whole new *osimage* called *61dskls_updt* as follows.

```
mknimimage -t diskless -s 61dskls_lpp 61dskls_updt
```

The **mknimimage** command will display the contents of the new xCAT *osimage* definition when it completes.

This new image can now be updated and used to boot the node.

4.1.2 Copy an existing image

You can use the **mknimimage** command to create a copy of an image. For example, if the name of the currently running image is *61dskls* and you want make a copy of it to update, you could run the following command.

```
mknimimage -t diskless -i 61dskls 61dskls_updt
```

If an "-i " value is provided then all the resources from the xCAT *osimage* definition (*61dskls*) will be used in the new *osimage* definition except the SPOT resource. The new SPOT resource will be copied from the one specified in the original definition and renamed using the new *osimage* name provided (*61dskls_updt)*. A new xCAT *osimage* definition will also be created, called "*61dskls_updt",* which will contain the names of these resources.

This new image can now be updated and used to boot the node.

4.2 Update the image (optional)

Updating a diskless node with fixes or additional software involves updating the SPOT that is being used to boot the node.

There are two basic types of updates you can make to a SPOT:

1. Install (or update) additional **installp** file sets or **rpm** packages.
2. Add or modify specific files, (such as /etc/inittab).

These two processes can be done manually or by using the **mknimimage** command with the "-u" option.

> **Note:** You should not attempt to update a SPOT resource that is currently allocated to a node. If you need to update an allocated SPOT either you can shut down the nodes and deallocate the SPOT resource first or you can make a copy of the SPOT and update that. To check to see if the SPOT is allocated you could run the following command.
>
> ***lsnim -l <spot name>***
>
> To shut down the nodes you can use **xdsh** to run "shutdown -F &" on the nodes. You can use the xCAT **rmdsklsnode** command to deallocate the nodes resources and remove the node from the NIM database. This command will not remove the node from the xCAT database.

4.2.1 Install additonal software

You can use the xCAT "**mknimimage -u**" command to install both **installp** file sets and **rpm** packages in a SPOT resource.

Before running the **mknimimage** command you must add the new filesets and/or RPMs to the *lpp_source* resource used to create the SPOT. If we assume the *lpp_source* location for *61dskls* is */install/nim/lpp_source/61dskls_lpp_*source . The **installp** packages would go in: */install/nim/lpp_source/61dskls_lpp_*source*/installp/ppc* and the RPM packages would go in: */install/nim/lpp_source/61dskls_lpp_*source*/RPMS/ppc.*

The easiest way to copy the software to the correct locations is to use the "**nim -o update** .." command. Just provide the directory that contains your software and the NIM lpp_source resource name. (ie. "61dskls_lpp_source").

If your new packages are in /tmp/myimages then you could run:

nim -o update -a packages=all -a source=/tmp/myimages 61dskls_lpp_source

Note: If you do not use this command to update the lpp_source then make sure you update the .toc file by running "inutoc .".

Once the lpp_source has been updated you can use the **mknimimage** command to install the updates in the SPOT resource for this xCAT osimage.

There are two ways to use the update feature. You can update the xCAT osimage definition and run the **mknimimage** command with no "installp_bundle", "otherpkgs", or "synclists" command line values OR you could do a more ad hoc update by providing one or more of the "installp_bundle", "otherpkgs", or "synclists" values on the command line.

If you do not provide any "installp_bundle", "otherpkgs", or "synclists" command line values then the information for updating the SPOT will come from the osimage definition only. This has the advantage of keeping a record of any changes that were made to the SPOT. To update the osimage definition you can use the **chdef** command to set the "installp_bundle", "otherpkgs", or "synclists" attribute values. Use this approach if you want the update information to be saved and used the next time you need to run **mkdsklsnode** with this image.

The "installp_bundle" value can be a comma separated list of (previously define) NIM installp_bundle resource names. The "otherpkgs" value can be a comma separated list of **installp** filesets and/or **rpm** package names. The rpm names must be preceded by "R:", (ex. R:foo.rpm). The "synclists" value is described below.

For example, to add these values to an xCAT osimage called "my61dskls" you could run a command similar to the following:

chdef -t osimage -o my61dskls installp_bundle="mybndlres1,mybndlres2" otherpkgs="openssh.base,R:popt-1.7-2.aix5.1.ppc.rpm"

Once the osimage definition is updated you can use the mknimimage command to apply those updates to the SPOT associated with that osimage.

mknimimage -u my61dskls

If you do provide one or more of the "installp_bundle", "otherpkgs", or "synclists" values on the command line then the **mknimimage** command will use those values only. The xCAT osimage definition will not be used or updated in this case.

In this case you would run the **mknimimage** command similar to the following.

mknimimage -u my61dskls installp_bundle="mybndlres1,mybndlres2" otherpkgs="openssh.base,R:popt-1.7-2.aix5.1.ppc.rpm"

The difference here is that the information the osimage definition is not used and this information is not saved.

Any additional software that is needed can be installed in a similar manner.

> **Note**: When installing software into a SPOT the pre and post install scripts for a particular software package will not run any code that will impact your running system, (like restarting daemons etc.). The script will check to see if it's installing into a SPOT and it will not run that code.

You can also specify **installp** flags on the **mknimimage** command line by setting the “installp_flags” attribute to the value you want to be used. The default flags, if not specified, are "agQX".

For example, to specify different flags you could run the command as follows.

mknimimage -u my61dskls installp_flags=”agcQX”

4.2.2 Add or modify files in a SPOT

You can update files in the SPOT manually or by using the xCAT **mknimimage** command.

4.2.2.1 To update manually

The root file system for a diskless node will be created by copying the “*inst_root”* directory contained in the SPOT. In the SPOT we created for this example the “*inst_root*” directory would be:

/install/nim/spot/61dskls/usr/lpp/bos/inst_root/

If you need to update the */etc/inittab* file in the SPOT that will be used on the diskless nodes you could edit:

/install/nim/spot/61dskls/usr/lpp/bos/inst_root/etc/inittab

You can also copy specific files into the *inst_root* directory so they will be available when the nodes boot. For example, you could copy a script called *myscript* to */install/nim/spot/61cosi/usr/lpp/bos/inst_root/opt/foo/myscript* and then add an entry to */etc/inittab* so that it would be run when the node boots.

All the diskless nodes that are booted using this SPOT will get a copy of *inst_root* as the initial root directory.

4.2.2.2 To update using mknimimage

XCAT supports the concept of a *synclists* file. This is a file that can be used to specify what configuration files need to be updated (synchronized). In the *synclists* file, each line is an entry which describes the location of the source files and the destination location for the files.

To use the **mknimimage** command to update files in the SPOT you must create a *synclists* file and pass the full path name to the command. One advantage of using the *synclists* file is that you have a record of what was done for a particular *osimage* and the update can be repeated easily if needed.

The basic entry format looks like following:

```
path_of_src_file1  -> path_of_dst_file1
```

path_of_src_file1 path_of_src_file2 ... -> path_of_dst_directory

The path_of_src_file* should be the full path of the source file on the Management Node. The path_of_dst_file* should be the full path of the destination file on target node. The path_of_dst_directory should be the full path of the destination directory.

The following formats are supported:

(1) sync file /etc/file1 to the file /etc/file1 on the node with the same file name. Note with one file, full path to file must be provide.

/etc/file1 -> /etc/file1

(2) sync file /etc/file2 to the file /etc/file3 on the node (with different file name)

/etc/file2 -> /etc/file3

(3) sync file /etc/file2 to the file /etc/tmp/file3 on the node with different file name, different directory) If the directory does not exist, it will be created.

/etc/file2 -> /etc/tmp/file3

(4) sync the multiple files /etc/file1, /etc/file2, /etc/file3, ... to the directory /tmp/etc (/tmp/etc must be a directory when multiple files are sync'd at one time). If the directory does not exist, xdcp will create it.

/etc/file1 /etc/file2 /etc/file3 -> /tmp/etc

Once the *synclists* file is create you can either add it to the xCAT *osimage* definition or specify it on the **mknimimage** command line.

To add it to an osimage definition you could run a command similar to:

chdef -t osimage -o 61dskls synclists="/full-path/mysyncfile"

You could then run mknimimage as follows:

mknimimage -u 61dskls

(Do not specify "installp_bundle", "otherpkgs", or "synclists" on the command line.)

Or, you could do a one time update by specifying the synclists file on the command line as follows:

mknimimage -u 61dskls synclists="/full-path/mysyncfile"

4.3 Verify the new image (optional)

To display the xCAT image definition run the xCAT **lsdef** command.

lsdef -t osimage -l -o 61dskls

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

lsnim -l 61dskls

To see the actual contents of a 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  61dskls
```

4.4 Re-initialize the NIM diskless nodes

You can re-initialize your diskless nodes to boot with the new or updated SPOT by running the **mkdsklsnode** command.

There are two basic situations where you would need to re-initialize a NIM diskless machine.

1. When you want to switch a node to a new image.
2. When you want to do the initialization for a new image while the node is still running. (This avoids having the node be down while the initialization step is completing.)

In the first situation you want to switch the nodes to use a new or updated image. If the diskless node is currently running you can use the "-f" (force) option of the **mkdsklsnode** command. With this option the **mkdsklsnode** command will stop the running node, deallocate the resources and do the NIM re-initialization with the new image. In this case the node would be unavailable during the initialization as well as the time for the node reboot.

Note: The NIM support for re-initialization take 3-4 minutes and is done sequentially.

For example, to switch the node named "*node29*" to a new image named "*611spot*" you could run the following command.

```
mkdsklsnode -f -i 611spot node29
```

The name of the image ("*611spot*") is the xCAT *osimage* name which is also the name of the SPOT resource that was created for this *osimage* definition.

In the second scenario we want to initialize an xCAT diskless node while the node continues running. To do this we need to create an alternate NIM machine definition for the same xCAT cluster node.

Creating alternate NIM machine definitions is possible because the NIM name for a machine definition does not have to be the hostname of the node. This allows you to have multiple NIM machine definitions for the same node. Since all the NIM initialization of the alternate machine definition can be done while the node is running, the downtime for the node is reduced to the time it takes to reboot.

For example, to initialize the xCAT node named "*node42*" to use the xCAT *osimage* named "*61dskls*" you could run the following command.

```
mkdsklsnode -n -i 61dskls node42
```

The naming convention for the new NIM machine name is "<xcat_node_name>_<image_name>", (Ex."*node42_61dskls*"). You could continue to create alternate machine definitions for each new image you wish to use for the node. The last NIM machine name that is initialized will determine what the node will use for the next boot.

> **Debug tip**: If you have forgotten which machine name you last initialized with NIM, and want to verify which image will actually be loaded on the next boot, NIM creates a /tftpboot/<*hostname*>.info file that contains mount information for the SPOT and other resources. You can check what will be mounted for the next boot of the node.

Using the "-n" option will save time but it will also leave you with multiple alternate NIM machine definitions for the same node. If you wish to do go back to the "normal" naming convention, (using the xCAT node name as the NIM machine name), you could run the **mkdsklsnode** command for the same node without the "-n" option. For example, in the previous example you got a NIM machine definition called "node42_61dskls". If you wish to switch back to a NIM machine name of "node42" for the next update you could run **mkdsklsnode** as follows. (You may need the "-f" (force) option if the "node42" definition already exists.)

```
mkdsklsnode -f -i 611dskls node42
```

4.5 Verify node readiness (optional)

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
```

In preparation for the network boot the NIM "dkls_init" operation configures bootp/dhcp. At this point you can verify that the /etc/bootptab file for bootp or /var/lib/dhcp/db/dhcpd.leases file for dhcp has an entry for each node you wish to boot. Also, it is recommended that you stop and restart the inetd service to ensure the new *bootp/dhcp* configuration is loaded:

```
stopsrc -s inetd

startsrc -s inetd
```

4.6 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
```

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
```

5.0 Using the updatenode command

You can use the **updatenode** command on AIX to run customization scripts and to synchronize files on diskfull (standalone) type nodes.

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

6.0 Getting software and firmware levels

6.1 Using the sinv command

The sinv command is designed to check the configuration of the nodes in a cluster. The command takes as input command line flags, and one or more templates which will be compared against the output of the xdsh command, designated to be run by the -c or -f flag, on the nodes in the noderange.

The nodes will then be grouped according to the template they match and a report returned to the administrator in the output file designated by the -o flag, or to stdout.

sinv supports checking the output from the rinv or xdsh command.

The sinv command is an xCAT Distributed Shell Utility.
See the man pages for sinv & rinv for more details.