XCAT 2 on AIX

AIX support Overview

10/19/10, PM 12:49:35

1.0 Overview

This document provides an overview of some of the xCAT support that is provided for the AIX operating system.

2.0 Installing xCAT and prerequisite Software

2.1 Set up an AIX system to use as an xCAT Management Node

- Follow AIX documentation and procedures to install and configure the base AIX operating system. (Typically by using the product media.)
- Apply the latest AIX software updates and fixes as needed.
- Make sure the OS version installed on the management node is greater than or equal to the OS versions you wish to install on the cluster nodes.

2.2 Install AIX prerequisite software

To install the additional AIX software you have a choice of several different interfaces provided by AIX. Perhaps the easiest method is to use the SMIT (or "smitty") interface but you could also use the AIX **geninstall, installp, or rpm** commands if you like. Refer to the AIX documentation if you are not familiar with this support. (http://www-03.ibm.com/servers/aix/library/index.html)

Important Note
Since these are **installp** file sets you should run **/usr/sbin/updtvpkg** after installing to make sure that the RPM reflection of what was installed by **installp** is updated. This makes it possible for RPM packages with a dependencies to recognize that the dependency is satisfied.

updtvpkg

2.2.1 openssl and openssh

The openssl and openssh installp filesets are now available on AIX product media. (Starting in AIX 6.1.3.)

You can check to see if these are installed by running the "**lslpp**" command. For example:

lslpp -l | grep open

If they are not installed then use the AIX product media and standard AIX tools to install them.

2.2.2 expect, tk, and tcl

This software is now shipped with AIX product media. (Starting in AIX 6.1.2.)

They are normally installed with AIX but in some cases you will have to install them manually from the AIX media.
Check if they are installed and install them if needed.

If they are not available on the AIX media then you can get them from the "AIX Toolbox for Linux Applications" (http://www-03.ibm.com/systems/power/software/aix/linux/toolbox/alpha.html)

2.2.3 devices.tmiscw (Optional)

If you plan to be using AIX diskless nodes and you wish to set up system dump support for those nodes then you will need the devices.tmiscw software installed on your management node. This software is available on the AIX Expansion Pack.

Install the software using standard AIX interfaces.

Note: The xCAT diskless dump support is available in xCAT version 2.5 and beyond. You will also need AIX 6.1.6 or greater for full support.

2.3 Create a new volume group for your /install directory (optional)

By default xCAT uses the /install directory to store various xCAT and NIM resources. XCAT will create /install as a subdirectory of the / (root) file system. In some cases /install may not contain enough space for your intended use.

To avoid this problem you could create a separate file system called /install on the management server to store the files that are to be used with xCAT and NIM. The size of this file system depends on your particular cluster.

The largest files that will be stored in /install subdirectories will be the NIM resources required for installing AIX nodes. The space required for a unique set of AIX operating system installation resources is approximately 2.0 GB. If you will need to manage several levels of OS images you should plan on at least 2G for each.

You can create the /install file system as part of the rootvg or in its own volume group. The following examples illustrate how to create the /install file system using

the root volume group. To create a 5 GB file system called /install you could issue the AIX **crfs** command:

```
crfs -v jfs2 -g rootvg -m /install -a size=5G -A yes
```

After you have created /install, you must mount it, as follows:

```
mount  /install
```

Note: You can use the AIX SMIT interfaces to create new volume groups and file systems etc. For example, to create a new file system you could use the SMIT fastpath ("crfs") to go directly to the correct SMIT panel. (Just type "*smit crfs".)*

2.4 Create a new volume group for your NIM dump resource directory (optional)

If you will be using NIM diskless dump resources you want to also consider creating a separate dump file system to store any system dumps that are initiated on the nodes. This will prevent the /install filesystem from running out of space. When you define your NIM dump resource you can use the new dump filesystem as the location.

You can use the AIX SMIT (or smitty) interfaces to create new volume groups and file systems etc. See the AIX documentation for details.

2.5 Download and install the prerequisite Open Source Software (OSS)

(Don't forget to run ***updtvpkg*** before installing rpms!)

- Download the latest dep-aix-*.tar.gz tar file from http://xcat.sourceforge.net/#download and copy it to a convenient location on your xCAT management node.
- Unwrap the tar file. For example:

  gunzip dep-aix-*.tar.gz
  tar –xvf dep-aix-*.tar

- Read the README file.
- Run the **instoss** script (contained in the tar file) to install the OSS packages. Please make sure the /opt and the other file systems have enough disk space to install these OSS packages before running the **instoss** script.

Note: The expect, tk and tcl rpms are no longer shipped by xCAT. They are now shipped with AIX and should have been installed automatically. If they are not then install them from the AIX media.

Note #2: For easier downloading without a web browser, you may want to download and install the **wget** tool from the AIX Toolkit for Linux.

2.6 Download and install the xCAT software.

Note: For various reasons it is recommended that you set the primary hostname of the management node to the interface that you will be using to install the nodes. If you do this before you install xCAT then xCAT will be able to set some cluster site default values automatically. It will also make it easier when configuring NIM. When setting the primary host name make sure the domain is included.

- Download the latest xCAT for AIX tar file from http://xcat.sourceforge.net/#download and copy it to a convenient location on your xCAT management node.
- Unwrap the xCAT tar file. For example,

  gunzip core-aix-*.tar.gz
  tar –xvf core-aix-*.tar

- Run the **instxcat** script (contained in the tar file) to install the xCAT software. The post processing provided by the xCAT packages will perform some basic xCAT configuration. (This includes initializing the SQLite database and starting **xcatd** daemon processes.) Note: xCAT software packages will install about 200MB files into /opt directory, make sure the /opt directory has enough disk space before running **instxcat** script.
- Execute the system profile file to set the xCAT paths. This file was updated during the xCAT post install processing. (". */etc/profile").* (**Note**: Make sure you don't have a .profile file that overwrites the "PATH" environment variables.)

2.7 Verify the xCAT installation.

- Run the "*lsdef –h*" to check if the xCAT daemon is working. (If you get a correct response then you should be Ok.)

- Check to see if the initial xCAT definitions have been created. For example, you can run "*lsdef -t site -l*" to get a listing of the default site definition. You should see output similar to the following.

  Setting the name of the site definition to 'clustersite'.

Object name: clustersite
domain=abc.foo.com
installdir=/install
tftpdir=/tftpboot
master=7.104.46.27
useSSHonAIX=yes
xcatdport=3001
xcatiport=3002

Important: The "domain" and "master" values are set automatically by xCAT when it is installed. To do this xCAT looks at the primary hostname of the management node.

For the "domain" attribute, if the management node hostname was set to a short hostname then the domain attribute would not be set by default. It is also possible that the domain would be set to a value other than the domain that is used for the cluster nodes. In either case you must manually set the domain value to the network domain that will be used for the cluster nodes. You can use the xCAT **chdef** command to modify the domain attribute of the cluster site definition.

For example:

chdef -t site domain=mycluster.com

The "master" attribute must be set to the hostname of the xCAT management node, as known by the nodes.

For example:

chdef -t site master=xcatmn

3.0 Additional configuration of the management node

3.1 Cluster network configuration notes

- The cluster network topology, naming conventions etc. should be carefully planned before beginning the cluster node deployment.
- XCAT requires an Ethernet network for installing and managing cluster nodes.
- Cluster nodes may be on different subnets.
- The cluster nodes must all have unique short host names to use in the xCAT node definitions.
- All cluster nodes must use the same domain name. The domain attribute must be set in the cluster site definition.

- The management node interfaces that will be used to manage the nodes should be configured before starting the xCAT deployment process.
- XCAT network definitions will have to be created for each unique subnet used in the cluster. (This will be described in one of the install documents listed below.)
- If you will be using the xCAT management node or a service node as a gateway remember to set "ipforwarding" to "1".

3.2 Choose the shell to use in the cluster (optional)

By default the xCAT support will automatically set up **ssh** on all AIX cluster nodes. If you wish to use r**sh** you should modify the cluster site definition. To use r**sh** you would have to set the "useSSHonAIX=no". You can also specify a path for the **ssh** and **scp** commands by setting the "**rsh**" and "**rcp**" . If not set the default path would be "/usr/bin/ssh" and "/usr/bin/scp".

You will also have to make sure that the **openssl** an **openssh** software is installed on your nodes. This is covered in the cluster node installation documents listed below.

To change the shell you must change the value of the *useSSHonAIX* attribute in the cluster site definition. For example:

chdef -t site useSSHonAIX=no

Note: If, at some future point, you wish to check which shell is being used you can run **xdsh** to a node with the "-T" (trace) option. For example:

xdsh node01 -v -T date

Note: The default shell for xCAT 2.3 and beyond is **ssh**. In earlier versions of xCAT the default was **rsh**.

3.3 Configuring name resolution (optional)

Name resolution is required by xCAT. You can use a simple /etc/hosts mechanism or you can optionally set up a DNS name server. In either case you must start by setting up the /etc/hosts file.

If you do not set up DNS you may need to distribute new versions of the /etc/hosts file to all the cluster nodes whenever you add new nodes to the cluster.

3.3.1 Add cluster nodes to the /etc/hosts file

There are several ways to get entries for all the cluster nodes in the /etc/hosts file.

These include:

- Manually adding the entries.
- Running a custom script that uses some cluster naming convention to automate the adding of the node entries. (User-provided.)

- Using the xCAT **makehosts** command after the XCAT node definitions have been created.

If you are dealing with a large number of nodes this task can be quite tedious. The xCAT **makehosts** option may be useful in some cases. This process uses a regular expression to automatically determine the IP addresses and hostnames for a set of nodes. To use this method you must decide on appropriate naming conventions and IP address ranges for your nodes. This process may seem a bit complicated but once you get things set up it can save time and add structure to your cluster.

If you choose to use this process you will have to come back to this section after you have created the xCAT node definitions later in this process. You should read through this now and decide on naming conventions etc. for when you create your xCAT node definitions.

The basic process is:

- Decide on a node naming convention such that the node IP & long hostname can be determined from the node name.
- Include all the nodes in a node "group" definition.
- Set the group "ip" and "hostnames" attribute to a regular expression that can be used to derive the node IP and hostname.
- Run the **makehosts** command to add all the node information to the /etc/hosts file.

As an example, suppose we decide on a node naming convention that includes the hardware frame number, the CEC number and the partition number. (Say "clstrf01c01p01" etc.) Also, lets say that the IP addresses would look something like "100.1.1.1" where the second number is the frame number, the third is the CEC number and the forth is the partition number.

With this example we can define a regular expression that, given a node name, could be used to derive a corresponding IP address and long hostname.

To have this regular expression applied to each node you can make use of the xCAT node group support. Let's say that all your cluster nodes belong to the group "compute". I can add the following values to the "compute" group definition.

chdef -t group -o compute ip='|clstrf(\d+)c(\d+)p(\d+)|10.($1+0).($2+0).($3+0)|' hostnames='|(.*)|($1).cluster.com|'

This basically says that for any node in the "compute" group the "ip" can be derived by the regular expression *'|clstrf(\d+)c(\d+)p(\d+)|10.($1+0).($2+0).($3+0)|'* , and the hostname can be derived from the expression *|(.*)|($1).mycluster.com|'*.

So let's say that you have defined all your nodes using the xCAT support such as **rscan** or **mkvm** using the naming convention mentioned above. Now you could display the node definition as follows:

lsdef -l clstrf01c02p03

Since this node belongs to the "compute" group, when I display the definition it will use the regular expressions to derive the "ip" and "hostnames" values.

The output might look something like the following:

Object name: clstrf01c02p03
cons=hmc
groups=lpar,all,compute
hcp=clstrhmc01
hostnames=clstrf01c02p03.mycluster.com
id=1
ip=10.1.2.3
mac=001a64f9c009
mgt=hmc
nodetype=lpar,osi
os=AIX
parent=clstrf1fsp01-9125-F2A-SN024C332
postscripts=myscript
profile=MYimg

Now that all the nodes have an "ip" and "hostnames" value you can run the xCAT **makehosts** command to update /etc/hosts.

makehosts compute -l

3.3.2 Set up a DNS nameserver

To set up the management node as the DNS name server you must set the "domain", "nameservers" and "forwarders" attributes in the xCAT "site" definition.

For example, if the cluster domain is "mycluster.com", the IP address of the management node is "*100.0.0.41*" and the site DNS servers are "9.14.8.1,9.14.8.2" then you would run the following command.

chdef -t site domain= mycluster.com nameservers= 100.0.0.41 forwarders= 9.14.8.1,9.14.8.2

Edit "/etc/resolv.conf" to contain the cluster domain and nameserver. For example:

search mycluster.com
nameserver 100..0.0.41

Create xCAT network definitions for each of the cluster networks. (Your network and mask value need to be defined for **makedns** to be able to set up the correct ip range for the management node to serve.)

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.

You can use the xCAT **makenetworks** command to gather cluster network information and create xCAT network definitions. See the **makenetworks** man page for details. (This feature is available in xCAT 2.3 and beyond.)

You can also use the xCAT **mkdef** command to define the network.

For example:

mkdef -t network -o net1 net=9.114.113.224 mask=255.255.255.224 gateway=9.114.113.254

Run **makedns** to create the /etc/named.conf file and populate the /var/named directory with resolution files.

makedns

Start DNS:

startsrc -s named

3.4 Syslog setup

xCAT will automatically set up **syslog** on the management node and the cluster nodes when they are deployed (installed or booted). When **syslog** is set up on the nodes it will be configured to forward the logs to the management node.
If you do not wish to have **syslog** set up on the nodes you must remove the "syslog" script from the "xcatdefaults" entry in the xCAT "postscripts" table. You can change the "*xcatdefaults*" setting by using the xCAT **chtab** or **tabedit** command.

3.5 Add cluster resolv.conf file (optional)

The xCAT deployment code will automatically handle the creation of an /etc/resolv.conf file on all the cluster nodes. If you want xCAT to handle this you should make sure the "domain" and "nameservers" attributes of the "site" definition are set.

For example:

chdef -t site -o clustersite domain=mycluster.com nameservers= 100.240.0.1

3.6 Set cluster root password (optional)

You can have xCAT create an initial root password for the cluster nodes when they are deployed. To do this you must modify the xCAT "passwd" table.

You can use the **tabedit** command to add and entry to this table. For example:

tabedit passwd

You will need an entry with a *"key"* set to *"system",* a *"username"* set to *"root"* and the *"password"* attribute set to whatever string you want.

In xCAT version 2.5 and beyond you may add an encrypted password to the table. If the password is encrypted you must also set the "cryptmethod" attribute so that the password can be set correctly on the nodes.

You can change the passwords on the nodes at any time using **xdsh** and the AIX **chpasswd** command.

For example:

xdsh node01 'echo "root:mypw" | chpasswd -c'

3.7 Set up NTP (optional)

To enable the NTP services on the cluster, first configure NTP on the management node and start **ntpd**.

Next set the "*ntpservers"* attribute in the site table. Whatever time servers are listed in this attribute will be used by all the nodes that boot directly from the management node.

If your nodes have access to the internet you can use the global servers:

chdef -t site ntpservers= 0.north-america.pool.ntp.org,
1.northamerica.pool.ntp.org,2.north-america.pool.ntp.org,
3.northamerica.pool.ntp.org

If the nodes do not have a connection to the internet (or you just want them to get their time from the management node for another reason), you can use your management node as the NTP server. For example, if the name of you management node is "myMN" then you could run the following command.

chdef -t site ntpservers= myMN

3.8 Increase file size limit

Some of the AIX/NIM resources that are used to install nodes are quite large (1-2G) so it may be necessary to increase the file 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

3.9 Check the policy definitions.

When the xCAT software was installed it created several policy definitions. To list the definitions you can run:

lsdef -t policy -l

You may need to add additional policy definitions. For example, you will need a policy for the hostname that was used when xCAT was installed. To find out what this was you can run:

openssl x509 -text -in /etc/xcat/cert/server-cert.pem -noout|grep Subject:

So, for example, if the hostname is "myMN.foo.bar" then you can create a policy definition with the following command. (The policy names are numbers, just pick a number that is not yet used.)

mkdef -t policy -o 8 name= myMN.foo.bar rule=allow

3.10 Check system services

- **inetd**

  inetd includes services such as telnet, ftp, bootp/dhcp, and others. **Edit the /etc/inetd.conf file to turn on all services that are needed.** FTP and bootp/dhcp are required for System p node installations. Stop and restart the **inetd** service after any changes:

  stopsrc -s inetd
  startsrc -s inetd

- **NFS**

  NFS is required for all NIM installs. Ensure the NFS daemons are running:

  lssrc -g nfs

  If any NFS services are inoperative, you can stop and restart the entire group of services:

  stopsrc -g nfs
  startsrc -g nfs

  There are other system services that NFS depends on such as inetd, portmap, biod, and others.

- **TFTP**

  To check if the TFTP daemon is running.

  lssrc -a | grep tftpd

To stop and start tftp daeon.

stopsrc -s tftpd
startsrc -s tftpd

4.0 Terminology

Some basic terminology.

- **standalone –** An AIX node that has it's operating system installed on a local disk.
- **rte** install - A network installation method supported by NIM that uses a NIM lpp_source resource to install a standalone node.
- **mksysb** install - A network installation method supported by NIM that uses a system backup of one node (mksysb image) to install other standalone cluster nodes.
- **diskful** – For AIX systems this means that the node has local disk storage that is used for the operating system. (A standalone node.) Diskfull AIX nodes are typically installed using the NIM **rte** or **mksysb** install methods.
- **diskless** - The operating system is not stored on local disk. For AIX systems this means the file systems are mounted from a NIM server.
- **stateful** – A node that maintains its "state" after it has been shut down and rebooted. The node state is basically any node-specific information that has been configured on the node. For AIX diskless nodes this means that each node has its own NIM "root" resource that it can use to store node-specific information. Each node mounts its own root directory.
- **stateless** – A node that does NOT maintain its state after it has been shut down and rebooted. For AIX diskless nodes this means that the nodes all use the same NIM "shared_root" resource. Each node mounts the same root directory. Anything that is written to the local root directory is redirected to memory and is lost when the node is shut down. Node-specific information must be re-established when the node is booted.
- **statelite** - An AIX diskless stateless node that also has a small amount of persistent files and/or directories. The persistent files and/or directories are mounted on the nodes. This support is available for AIX nodes in xCAT version 2.5 and beyond.

5.0 xCAT on AIX documentation

5.1 Installing AIX standalone nodes (using NIM rte method)

http://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2onAIXinstall.pdf

5.2 Booting AIX diskless nodes (using stateless method)

http://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2onAIXDiskless.pdf

5.3 Cloning AIX nodes (install using AIX mksysb image)

http://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2onAIXmksysb.pdf

5.4 Using xCAT Service Nodes with AIX

http://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2onAIXServiceNodes.pdf

5.5 Updating AIX cluster nodes

http://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2onAIXUpdates.pdf

6.0 Using AIX statelite support

6.1 AIX statelite support overview

This feature is available in xCAT version 2.5 and beyond. It is included as "beta" support in the xCAT 2.5 release.

Note: The xCAT *statelite* implementation for AIX is not the same as the Linux implementation due to basic differences in the base operating systems and their deployment methods.

The xCAT support for AIX diskless nodes includes options for using either a NIM "root" or a "shared_root" resource. You can choose either one for a diskless node deployment.

If you choose the "*root*" resource then each node will get it's own unique mounted root file system. If the node is shut down and rebooted it will get the same root filesystem. Anything that the node wrote to it's root file system is preserved. This case is referred to as "stateful".

If you choose a "*shared_root*" resource then the nodes will share the same root filesystem. When an individual node writes to it's root file system it is actually writing to local memory (using the AIX STNFS support). If the node is shut down and rebooted it will get the same root filesystem it originally started with. In this case anything that the node wrote to it's root file system is lost. Any node-specific information or configuration will have to be redone. This case is referred to as "stateless".

The big advantage for using *stateless* nodes is that they can all share a common "*shared_root*" resource and that there is very little network traffic since the nodes all write to local memory.

For large scale cluster environments it may be advantageous to use a *stateless* implementation. However there will very likely be a need to have some subset of files or directories be persistent. There also may be a need to specify unique files or directories for each node.

The xCAT on AIX "statelite" implementation provides this type of support. It basically provides the ability to "overlay" specific files or directories over the standard diskless-stateless support.

The AIX stateless support is only available when using the diskless-stateless deployment method.

The *statelite* setup on the node occurs early in the boot process. Information that is provided in *statelite* tables will be used during the boot process.

Note: Statelite support must be used with caution especially when modifying system configurations files that are used early in the boot process. For, example if you try to use the /etc/objrepos as a statelite directory the diskless boot will hang. If you are not sure about how a system configuration file change will affect the system, you should try it on a test system before deploying the cluster nodes.

6.2 Statelite options

The xCAT for AIX statelite support includes the following three options:

1. **persistent** – Provide a mounted file or directory that is copied to the xCAT persistent location and then over-mounted read-write on the local file or directory. Anything written to that file or directory is preserved. For example, this could be used to preserve log or trace files or to provide node configuration data for the next time the node is booted. (Requires the *statelite* table to be filled out with a location for persistent storage – see below).

2. **rw** - Provide a file or directory for a node to use when booting, allow the node to write to the file, but on the next diskless boot the original (or latest) version of the file on the server will be used. (read-write – non-persistent)

3. **ro** – Provide files or directories that can be overmounted read-only on the local files or directories. The directory or file will be mounted on the node while the node is running (not just during the boot process), and overmounted on the local version of the file or directory. Changes made to this file or directory on the server will be immediately seen in this file or directory on the node. This option requires that the file or directory to be mounted must be available in one of the entries in the *litetree* table

The default option is "**rw**", which means if you leave the option attribute as blank, then the option will be treated as "rw" when the entry in *litefile* table is used.

The examples provided below illustrate how to use these options.

6.3 Statelite database tables

In order to specify the information needed for xCAT to do the statelite setup you must update one or more of the xCAT *statelite* database tables.

There are three xCAT database tables that may need to be updated to implement the xCAT statelite support.

The **statelite** table contains the location on an NFS server where a nodes persistent files and directories are stored. Any file marked persistent in the **litefile** table will be stored in the location specified in this table for that node.

Example:

Assume that the xCAT node group name is "*aixnodes*", that the server for the NFS mounted directory will be the management node (*xcatmn*), and that the persistent directory is */nodedata*.

In this case the table entry would look like the following:

```
#node,image,statemnt,comments,disable
"aixnodes",,"xcatmn:/nodedata/",,,
```

The "image" value is not currently used in this table.

In the *statelite* table, The "*node*" attribute can be filled in with either a node name or a group name. However, you must make sure a node is not included twice. This

could easily happen when using group names. There can be only one *statelite* persistent directory per node.

Export any directories that are listed in this table before attempting to boot the nodes. Use the export options appropriate to your environment. Make sure the nodes will be able to read-from and write-to the persistent directory.

Also, consider that the internal xCAT code must be able to create additional files and subirectories under the *statelite* directories, so make sure the permissions will allow this. (For example, set the permissions on the *statelite* directories to "755".)

The **litefile** table specifies the directories and files for the statelite setup along with the option to use to do the setup. If no option is provided then the default is "rw".

Example:

> The "image" value can be either the name of the xCAT *osimage* definition or "ALL" (which means all osimages.)
>
> ```
> #image,file,options,comments,disable
> "ALL","/mydata/","persistent",,
> "61cosi","/lppcfg","rw",,
> "61cosi","/etc/lppcfg","ro",,
> ```
>
> Note: A directory name should end with a "/".

Make sure the file and directory permissions are set the way you want them to appear on the nodes and that they are appropriate for the statelite option you are using.

Do not include BOTH a directory and subdirectory (or file) in the litefile table. For example, if you have an entry of "/foo/" then you should not also include "/foo/bar".

The **litetree** table controls where the initial content of the files in the **litefile** table come from, and the long term content of the "ro" files. The "*priority*" value indicates the search path to use when looking for a file or directory. If the file cannot be found in any of the *litetree* entries (or there are no entries), then the default will be to use the file contained in the *osimage* SPOT resource.

The *directory* value must be the location (hostname:path) of a directory that contains the file or directory exactly as specified in the *litefile* table. For example, if the *litefile* entry is "/etc/motd" and the *litetree* directory entry is "/myfiles" then the assumption is that the file would be located in "/myfiles/etc/motd".

Export any directories that are listed in this table before attempting to boot the nodes. Use the export options appropriate to your environment.

Example:

#priority,image,directory,comments,disable
"1","61cosi","xcatmn:/clstrnodedata/",,,
"2","61cosi","xcatsvr:/mydata/",,,

To update an xCAT database table you can use the **tabedit** command. ("tabedit <table-name>") See the **tabedit** man page for details.

Also see the examples described below.

6.4 Statelite user management

If you wish to set up something other than root user access to the statelite files or directories on the nodes you must also set up the new user and group IDs on the nodes. (For example, if you wish to set up a persistent logging directory to be written to by a different userid.)

One way to accomplish this is to create a set of configuration files, that include the user information, and include them in the SPOT resource that will be used to boot the node.

The list of configuration files you will need in order to get the exact same user information on the nodes is:

- /etc/passwd
- /etc/group
- /etc/security/passwd
- /etc/security/group
- /etc/security/user
- /etc/security/limits

You can create these files on the management nodes and copy them to the SPOT resource being used to boot the nodes. (ex. /install/nim/spot/<spot-name>/usr/lpp/bos/inst_root/etc)

Another option is to use the xCAT *synclists* support. With this support you can create a "synclists" file containing a list of all the extra configuration files you would like added to the SPOT resource. You can then update the SPOT using the "mknimimage -u ... ". For more information on using the synclists support see the following xCAT document. "How to sync files in xCAT" (http://xcat.svn.sourceforge.net/viewvc/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2SyncFilesHowTo.pdf)

6.5 Examples of how to use the xCAT for AIX statelite support

Refer to the xCAT documents that describe now to boot diskless AIX nodes (mentioned above) for details on the deployment process. The examples below describe additional steps that are needed to utilize the *statelite* support.

When the **mkdsklsnode** command is run during the deployment process it will use the information in the *statelite* tables to make modifications to the *osimage* SPOT resource to prepare for the *statelite* setup. When the node boot process begins an xCAT setup script is run to do the required *statelite* setup on the node.

You must add the required information to the *statelite* tables before you run **mkdsklsnode**.

The "**Result**" sections below indicate some of the internal structure that is used in the xCAT for AIX implementation. (This may be useful for debug purposes.)

6.5.1 Provide a persistent logging directory

Provide a unique persistent logging directory location for the cluster diskless-stateless nodes. (read-write-persistent directory)

Assume that you have an xCAT node group named "*aixstateless*", that the server for the NFS mounted directory will be the management node (*xcatmn*), and that the persistent directory is */nodedata*. Also assume the xCAT *osimage* name is "*61dskls*" and that the directory to store the logs in should be "*/logs*".

1. Modify the *statelite* table. (using **tabedit**)

   #node,image,statemnt,comments,disable
   "aixstateless",,"xcatmn:/nodedata",,

   This says that each node in the "*aixstateless*" node group will store their persistent data in the */nodedata* file system mounted from the management node.

   Make sure you export any directories that are listed in this table before attempting to boot the nodes.

2. Modify the *litefile* table. (using **tabedit**)

   #image,file,options,comments,disable
   "61dskls","/logs/","persistent",,

This says that any node using the "*61dskls*" osimage should have a */logs/* directory, and that it should be persistent. Notice that a directory name must end in a "/".

Result:
When the node (say node01) writes to it's local */logs* directory it really goes to the over-mounted */.statelite/persistent/node01/logs* directory. These directories were created during the *statelite* setup. The */nodedata* directory was then mounted from the management node. So when you write to the local */logs* directory you are really writing to the */nodedata/node01/logs* directory on the management node.

6.5.2 Provide a read-write configuration file

Provide the current version of a configuration file for the node to use when booting, allow the node to write to the file, but on the next install the latest version of the file on the server should be used. (read-write - non-persistent)

Assume that you want any node that uses the *osimage* named "*61dskls*" to boot using the configuration file called "*/etc/FScfg"*. The original version of the file should come from the server named *"FSserver.cluster.com"*.

1. Modify the *litefile* table. (using **tabedit**)

   ```
   #image,file,options,comments,disable
   "61dskls","/etc/FScfg",”rw”,,
   ```

   This says that any node using the "*61dskls*" *osimage* should have an */etc/FScfg* file, and that it should be read-write.

2. Modify the *litetree* table. (using **tabedit**)

   ```
   #priority,image,directory,comments,disable
   “1”,”61dskls”,”FSserver.cluster.com:/myfiles”,,,
   ```

   This says that the initial version of the file should be taken from "*FSserver.cluster.com:/myfiles"*

   Make sure you export any directories that are listed in this table before attempting to boot the nodes.

Result:
When the node is being booted an xCAT script mounts "*/myfiles"* from "*FSserver"* and copies "*/myfiles/etc/FScfg"* into the local */etc/FScfg.*

When the node writes to the file it is actually writing to local memory, (the normal stateless function), and the updates are not preserved for the next deployment. When

the node is re-deployed the */etc/FScfg* file from *FSserver* is again used for the initial value.

6.5.3 Provide a read-only configuration file

Provide the nodes with a unique configuration files to use when booting. The file should not be modified. (read-only)

The server for the NFS mounted directory is the management node (*xcatmn*). The *osimage* name is be "*61dskls*" and the configuration file is */etc/lppcfg*.

1. Modify the *litefile* table. (using **tabedit**)

   ```
   #image,file,options,comments,disable
   "61dskls","/etc/lppcfg",”ro”,,
   ```

   This says that any node using the "*61dskls*" osimage should have a */etc/lppcfg* file, and that it should be read-only.

2. Modify the *litetree* table. (using **tabedit**)

   ```
   #priority,image,directory,comments,disable
   “1”,”61dskls”, ”xcatmn:/statelite/”,,,
   “2”,”61dskls”, “xcatmn:/”,,,
   ```

   This says that any node using the "61dskls" osimage should look for the litefile names, first in the ”xcatmn:/statelite/” directory and, if not found, then look for it in ”xcatmn:/” (ie. Take the file from the management node.). If the file is not found in either of these places the default will be to take the one that exists in the SPOT "inst_root" location. If that doesn't exist then an empty file is created.

   Make sure you export any directories that are listed in this table before attempting to boot the nodes.

Result

When the *statelite* setup is being done xCAT will find the file using the entries in the *litetree* table. It will then mount the correct file or directory to the local xCAT *statelite* directories. Then the xCAT statelite directory is used to overmount the local *"/etc/lppcfg" file.*

When the node reads the local "*/etc/lppcfg* " file it is actually reading the file mounted from the server specified in the *litetree* table. Any change to the file located on the server will be seen immediately on the local node.

6.5.4 Using variables in table entries

In the previous examples it would also have been possible to specify unique files or directories for each node or set of nodes. To do this you could use variables from the xCAT database. When the *statelite* setup information is read, the variables in the table will be substituted with the actual values for that node. This would support having one statement cover multiple different nodes. It will also support having unique servers and locations for each node or group of nodes.

For example, a "*directory*" entry in the *litetree* table might look like the following:

$noderes.nfsserver:/mydir/$node

This would mean that the NFS server would be the value of "*$noderes.nfsserver"* for this node definition and the location would be *"/mydir/<nodename>"*.

You could also use the variable support for the "*statement*" attribute of the *statelite* table. For example,

```
#node,image,statemnt,comments,disable
"node01",,"$noderes.servicenode:/foo/$nodetype.profile",,
```

This would say to use the node's service node as the NFS server and "*/foo/<osimage_name>*" as the persistent directory.

7.0 References

- xCAT man pages: http://xcat.sf.net/man1/xcat.1.html
- xCAT DB table descriptions: http://xcat.sf.net/man5/xcatdb.5.html
- xCAT mailing list: http://xcat.org/mailman/listinfo/xcat-user
- xCAT bugs: https://sourceforge.net/tracker/?group_id=208749&atid=1006945
- xCAT feature requests: https://sourceforge.net/tracker/?group_id=208749&atid=1006948
- xCAT wiki: http://xcat.wiki.sourceforge.net/