Node Provisioning: How do I set up a root FS over Lustre?

How do I set up a root FS over Lustre?

This article describes how to configure a cluster so that the compute nodes mount their root file system over Lustre. It assumes that the Lustre file system has already been setup and that the client packages have been installed on the head node(s), as well as in the software images. If this is not the case, then please refer to the administrators manual for instruction on how to setup Lustre.

Setting up root over Lustre is similar to setting up root over NFS, which is also documented in the manual. This article will describe where root over Lustre is different. Note that this has only been tested on CentOS 6.

In addition to the Lustre client installation on the head node(s) and software image(s), the packages need to be installed into the node installer NFS root. This can be done by running: `rpm -r /cm/node-installer -i <Lustre client rpm files>`

Lustre needs to know what networks to use, which is set using module parameters for the lnet kernel module. On the head node and in the images, it can be set by putting the following line into /etc/modprobe.d/lustre.conf:

```
options lnet networks=o2ib(ib0)
```

For the installer environment, a similar line should be added to /cm/node-installer/etc/modprobe.conf.

The above tells Lustre to use the o2ib Lustre network, through the ib0 interface.

The next step is to modify `/etc/mkinitrd_cm.conf` in each software image that should be used for root over Lustre. After the auto-generated section add the following keyword, on a single line:

```
syncmodules
```

After that, force re-creation of the ramdisk for the software images. The added keyword triggers the Lustre kernel modules to be synchronized into the NFS root whenever the ramdisk is generated.

Similar to root over NFS, we need to setup the FSMounts so that there is a read-only root, and some tmpfs overlays for parts that should be writable. For example, we can add something like this:

```
device                                  mountpoint (key)        filesystem  mountoptions

--------------------------------------- ----------------------- ----------- -------------

10.11.12.13@o2ib:/images/default-image  /                       lustre      ro,_netdev

tmpfs                                   /tmp                    tmpfs       defaults

tmpfs                                   /var                    tmpfs       defaults

tmpfs                                   /etc                    tmpfs       defaults

tmpfs                                   /cm/local/apps/cmd/etc  tmpfs       defaults

tmpfs                                   /dev                    tmpfs       defaults

tmpfs                                   /root                   tmpfs       defaults
```

Note that these mounts are in addition to the already present mounts like `/home`, `/cm/shared`, etc.

To prevent strange issues with `rsync`, it is important that the provisioning nodes mount the software images off of the same Lustre file system. To do that add the appropriate FSMounts to those machines (possibly including the head node).

Node Provisioning: How do I set up a root FS over Lustre?

When using root over Lustre, the `/etc/mtab` file that is installed onto the node is initially created as a symbolic link to `/proc/mounts`. This is mainly to make sure the distribution's init scripts don't mount the tmpfs mounts again. A side effect of this is that tools like `df` will fail, because of how the Lustre root is setup. To solve this, put the following into the `/etc/rc.local` file of the software image:

```
#!/bin/bash

# This is part of the root on lustre implementation.
# To prevent the init script from remounting file systems
# already mounted by the installer, /etc/mtab is a symlink
# to /proc/mounts. We now restore that file and remove the
# lustre image mount so that tools like df don't complain.

if [ -L /etc/mtab ]; then
rm -f /etc/mtab
cat /proc/mounts | grep -vE '.+\ \/lustre\ lustre\ .+' > /etc/mtab
fi

touch /var/lock/subsys/local
```

During provisioning the `rsync` process compares the local file system with the software image. But since in this case those are largely (except for the tmpfs's) the same, this is rather pointless. Therefore, as an optimization, you can add all non-tmpfs sub-directories of the root to the full/update exclude lists.

Unique solution ID: #1113
Author: Teun
Last update: 2013-04-25 17:50