Hoffman2 System Infrastructure Proposal

HPC Group (Kejian, Prakashan)
Academic Technology Services
University of California, Los Angeles

December 11, 2007

Contents

1 Background

The standard procedure in installing OS in a beowulf type cluster is to install OS on one of the node and make that node a TFTP, NSF and DHCP server and use that node to install OS on any number of nodes using PXE Boot and configure them according to the instructions in a kickstart script. In this procedure each node's hard drive is formatted and each node is booted independently from its local harddrive. When there is a requirement of adding a new application which writes its shared libraries in /usr partition one needs to repeat the installation process either through "rpm installation" or through "yum installation" on all the compute nodes. Also when there is a kernel upgrade the same process has to be repeated on all the nodes. In the event of a compromise one has to go through all of the above process on all the nodes in order to rebuild the system.

2 Goals

1. Avoid the need to rebuild all machines when a headnode is compromised which may or may not have compromised the nodes that were accessible from the headnode.
2. Update the OS software or kernel on one of the node and the new image need to be pushed out to all compute nodes once the updated image is tested and made sure all applications run fine on it.
3. The above process must be scalable in terms of human resources and there should not be any loss in performance for the applications.
4. When there is a hardware failure on one of the nodes the rebuilt node should have exactly same image as it had before the failure in order to be identical to other nodes.
5. Ease of use by cluster admin
6. The tool to install OS should be independent of the distribution.
7. Provide additional security by mounting the system files as read only from a server that has a very tight firewall.
8. Provide Quality of Service to the users.
9. The tool should be open source

3 Issues

1. A Cluster with more than 16 nodes will take weeks or months to install or upgrade the OS.
2. Security Compromise is very often in university environment where the headnode has to be open to any computer in the world. Local root exploitation is very common in a system with hundreds of users. Usually when a user account is compromise, the hackers will be able to exploit the system with vulnerable kernels by running codes that will cause buffer overflow. Smart hackers usually installs rootkit, keyboard sniffing and other software to collect passwd and other information to hack more nodes.
3. In the above situation rebuilding a cluster with 100+ nodes is going to overwhelm the system staff.

4. Quality of Service: During the sytem rebuild time, if multiple identical images are available, another identical image need to be substituted during the rebuild process to provide uninterrupted service.

5. When there is a hardware failure on one of the node it is not very straight forward to reproduce the image before the failur using traditional install and yum-update method.

4 Diskless System using Perceus

We made an accessment on Perceus distributed as open source by LBNL. We found perceus to answer most of the issues mentioned above. The tool is easy to install and instructions are easy to follow as well. The number of additional pre-installed packages in order for perceus to run is very minimal.

4.1 How does Perceus work?

First install the required pre-reqisite packages through a yum installation procedure on one of the node with tight firewall and then download and install the Perceus software. Next step is to configure the private network where perceus service is available and then start the perceus daemon. Perceus basically combines the DHCP and tftp in one daemon. When a node reboots with PXE boot enabled in a network that is connected through a swith to the node that is running perceus, the new node will be able to lease a tftp and dhcp connection. It first boots in Perceus kernel, then it uses Perceus kernel to boot the target OS. After the target OS is up, the perceus kernel is removed from RAM.

5 Installation

We found Perceus is very easy to use and install. We actually made the Perceus work for CentOS 5.0. The only modification we have is to modify the CentOS repository:

```
[vnfs-centos5-base]
#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$ARCH&repo=os
name=CentOS-$releasever - Base
baseurl=http://mirror.centos.org/centos/$releasever/os/$ARCH/
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5
```

6 Cluster Management

You can modify and update the system in few commands as shown below. Perceus is very simple to use compared to rocks or other distribution.

We have tested perceus on CentOS, but it should work similar way with FC7 or Ubuntu because ther is no dependency on a particular distribution.

The following are the examples that a cluster admin might be interested in.

6.1 Add/Remove a user

```
1. Login to Perceus Appliance (which provides the image)
2. Mount the image:
      perceus vnfs mount centos-5.0.x86_64.stateless
```

```
3. chroot /mnt/centos-5.0.x86_64.stateless

4. useradd "options" to add newuser or userdel "options" to remove users (example)

5. Unmount the image
      perceus vnfs umount centos-5.0.x86_64.stateless

6. Sync to all child nodes (Once command syncronize all nodes)
perceus vnfs livesync centos-5.0.x86_64.stateless
```

6.2 Add/Remove an Package

```
1. Login to Perceus Appliance (which provides the image)
2. Mount the image:
      perceus vnfs mount centos-5.0.x86_64.stateless

3. chroot /mnt/centos-5.0.x86_64.stateless
   yum -y install PACKAGE_NAME
   or
   yum --installroot /mnt/centos-5.0.x86_64.stateless install PACKAGE_NAME
   or
   rpm --root /mnt/centos-5.0.x86_64.stateless -e PACKAGE_NAME

5. Unmount the image
      perceus vnfs umount centos-5.0.x86_64.stateless
6. Sync to all child nodes (Once command syncronize all nodes)
perceus vnfs livesync centos-5.0.x86_64.stateless
```

6.3 Install ib driver

```
1. Login to Perceus Appliance (which provides the image)
2. Mount the image:
      perceus vnfs mount centos-5.0.x86_64.stateless

3. chroot /mnt/centos-5.0.x86_64.stateless
      compile the driver, make the installation or do a rpm or yum install.

5. Unmount the image
      perceus vnfs umount centos-5.0.x86_64.stateless

6. Sync to all child nodes (Once command syncronize all nodes)
perceus vnfs livesync centos-5.0.x86_64.stateless
```

7 Performance

Perceus is only providing a way for you to boot a diskless to the RAM. Once it is there, it runs like regular system.

It maybe even faster because it does not use any swap. Local hard drive can be formatted to use swap and provide local scratch directory (to be tested).

8 Conclusion

We found perceus to be very easy and simple to build a diskles system and recommends for Hoffman2 cluster. It is tested on both 32 abnd 64 bit architecture. We were able to run an MPI job on a node running perceus image using IB interconnect. Perceus is the solution for building a diskless system for Hoffman2.