Remote Node Configuration

Building and Configuring Cluster Nodes with DHCP

Walt Akers, Jie Chen, Chip Watson, Ying Chen

May 24, 2001

TJNAF - Thomas Jefferson National Accelerator Facility

DOCUMENT DATE: Table of Contents generated: May 24, 2001 3:23 pm

TRADEMARKS: UNIX is a registered trademark of AT&T in the USA and other countries.

VxWorks is a register trademark of Wind River Systems.

The X Window System is a trademark of Massachusetts Institute of Technology.

OSF/Motif and Motif are trademarks of Open Software Foundation, Inc.

Ultrix and DEC are registered trademarks of Digital Equipment Corporation.

HPUX is a registered trademark of Hewlett Packard.

SURA/CEBAF: The Southeastern Universities Research Association (SURA) operates the Continuous Electron Beam Accelerator Facility (CEBAF) for the United States Department of Energy under contract DE-AC05-84ER40150.

DISCLAIMER: This report was prepared as an account of work sponsored by the United States government. Neither the United States nor the United States Department of Energy, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or any agency thereof. The view and opinions of the authors expressed herein do not necessarily state or reflect those of the United States government or any agency thereof.

Table of Contents

List of Figures

1. Strategies for Building Homogeneous Compute Nodes

Overview

As we begin the development of a large cluster, it will be necessary to rapidy configure many compute nodes. Because configuring many systems by hand is both time-consuming and tedious, we are developing a solution that will allow the system to automatically configure itself from a remote image available from a ‘master’ machine. The tools that we are using for this project are PXE, DHCP, enhanced TFTP, BPBatch and Kickstart.

What is PXE?

Intel’s Preboot Execution Environment is a standard defining the protocol that a remote node will use to boot from a DHCP server. On most newer machines, PXE is available as a boot option in the systems BIOS. PXE must be set as the primary boot device in order for the system to boot from a remote server.

What is DHCP?

The Dynamic Host Configuration Protocol (DHCP) is an Internet protocol for automating the configuration of computers that use TCP/IP. DHCP can be used to automatically assign IP addresses, to deliver TCP/IP stack configuration parameters such as the subnet mask and default router, and to provide other configuration information such as the addresses for printer, time and news servers.

What is TFTP?

The Trivial File Transfer Protocol is a very compact mechanism for transferring data between networked systems. By definition, the standard TFTP server provides a very small data window for transferring files. Since remote node-building will require the transfer of very large files, we have chosen to use an ‘enhanced TFTP’ server that expands the size of the data window from 512 bytes to 1408 bytes. This will greatly enhance performance during file transfer operations.

What is BPBatch?

BpBatch is a versatile remote-boot processor, that can be downloaded for free from the web. It can perform a large variety of actions on a computer at boot-time, prior to starting the operating system. Actions performed by BpBatch ranges from partitioning hard disk to authenticating users, including a graphical interface. The main feature of BpBatch is the partition cloning facility, which let you create an image of a computer's hard disk partition and then distribute and install this image on a cluster of PC.

What is Kickstart?

Kickstart is RedHat Linux’s system configuration program. In conjunction with Anaconda, it uses a configuration file (ks.cfg) to partition, format, and build a Linux system from the ground up. This package is part of the RedHat Linux distribution.

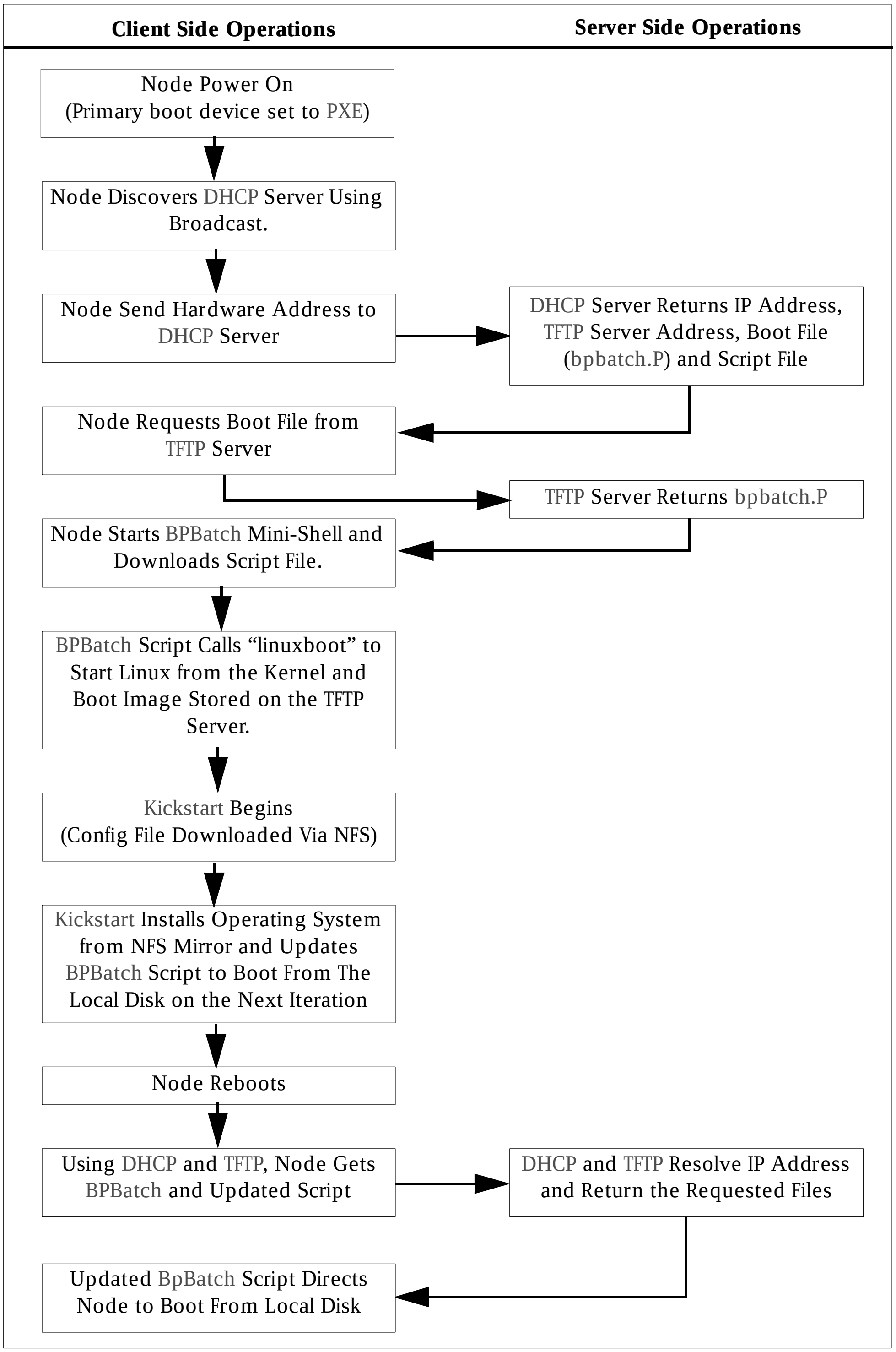
While Kickstart can load the operating system from a variety of sources (CD-ROM, NFS, etc) we will be using a locally installed mirror site to install the system.

2. The Remote Boot / Configuration Sequence

Sequence of Events

In order to understand why each product is being used, the diagram below illustrates how each of the products is employed to build a system from the ground up.

Figure 1: Remote Boot / Configuration Sequence



Client Side Operations
Server Side Operations
Node Power On (Primary boot device set to PXE)
Node Discovers DHCP Server Using Broadcast.
Node Send Hardware Address to DHCP Server
DHCP Server Returns IP Address, TFTP Server Address, Boot File (bpbatch.P) and Script File
Node Requests Boot File from TFTP Server
TFTP Server Returns bpbatch.P
Node Starts BPBatch Mini-Shell and Downloads Script File.
BPBatch Script Calls “linuxboot” to Start Linux from the Kernel and Boot Image Stored on the TFTP Server.
Kickstart Begins (Config File Downloaded Via NFS)
Kickstart Installs Operating System from NFS Mirror and Updates BPBatch Script to Boot From The Local Disk on the Next Iteration
Node Reboots
Using DHCP and TFTP, Node Gets BPBatch and Updated Script
DHCP and TFTP Resolve IP Address and Return the Requested Files
Updated BpBatch Script Directs Node to Boot From Local Disk

3. Configuring the DHCP Server

Obtain and Install DHCP

The first step in building the boot server is to get a copy of the DHCP distribution. For our cluster we used Version 3 of the ISC DHCP daemons. These binaries are typically part of the RedHat distribution, but can also be obtained from www.isc.org. Once you have installed DHCP, the next step is to create the configuration file: /etc/dhcpd.conf.

The DHCP Configuration File

The DHCP configuration file has many configuration options that are outlined at various websites, as well as in **The DHCP Handbook** by Droms and Lemon. Since the objective of all good programmers is to accomplish the task using the simplest approach, we will only use the most fundamental constructs within this configuration file.

Sample dhcp.conf

Figure 2: Sample /etc/dhcpd.conf configuration file

```
#
# DHCP configuration file. ISC DHCP server v3.0
#
# Suffering the ongoing modifications of Walt Akers
# May 24, 2001
#
# Global options
deny unknown-clients;

#
# Globally specify the domain name, DNS servers, subnet mask
# and routers.  These will be reloaded by DHCP nodes each time
# the system boots.
#
option domain-name "jlab.org";
option domain-name-servers 129.57.16.64, 129.57.32.100, 129.57.32.101;
option subnet-mask 255.255.252.0;
option routers 129.57.40.1;

# Since we are using static IP addresses, we will set the default
# lease time to inifinity
default-lease-time -1;

# Definition of PXE-specific options
# Code 1: Multicast IP address of bootfile
# Code 2: UDP port that client should monitor for MTFTP responses
# Code 3: UDP port that MTFTP servers are using to listen
#         for MTFTP requests
# Code 4: Number of seconds a client must listen for activity before
#         trying to start a new MTFTP transfer
# Code 5: Number of seconds a client must listen before trying to
#         restart a MTFTP transfer
option space PXE;
option PXE.mtftp-ip    code 1 = ip-address;
option PXE.mtftp-cport code 2 = unsigned integer 16;
option PXE.mtftp-sport code 3 = unsigned integer 16;
option PXE.mtftp-tmout code 4 = unsigned integer 8;
option PXE.mtftp-delay code 5 = unsigned integer 8;
```

Figure 2: Sample /etc/dhcpd.conf configuration file (continued)

```
#
# This block of definitions applies to all systems within
# this subnet
#
subnet 129.57.41.0 netmask 255.255.255.0
   {
   #
   # Add a group for the HPC Computing Cluster Nodes
   # to minimize the number of lines of code. The entries
   # specified in this group will apply to each host defined
   # below.
   #
   group
      {
      # This is the name of the boot file that will be
      # retrieved from the TFTP server for all members
      # of this group. Note: This boot file’s location is
      # relative to the /tftpboot directory on the TFTP server.
      filename "bpbatch.P";

      # PXE specific options
      class "pxeclients"
         {
         match if substring (option vendor-class-identifier, 0, 9) =
         "PXEClient";
         option vendor-class-identifier "PXEClient";

         # At least one of the vendor-specific option must
         # be set. We set the MCAST IP address to 0.0.0.0 to
         # be PXE compliant
         option PXE.mtftp-ip 0.0.0.0;
         vendor-option-space PXE;
         }

      #
      # Host Specifications:  Each host will contain a hardware
      # ethernet address, a corresponding internet address, and
      # “option-135” which specifies the bootfile that BPBatch
      # should use.  Note: The path to the boot file is relative
      # from the /tftpboot directory on the TFTP server.
      #
      host hpcdev02
         {
         hardware ethernet 0:30:48:11:4c:7d;
         fixed-address 129.57.41.128;
         option option-135 "boot/129.57.41.128/boot";
         }
      host hpcdev03
         {
         hardware ethernet 0:30:48:11:4c:7e;
         fixed-address 129.57.41.129;
         option option-135 “boot/129.57.41.129/boot”;
         }
      }
   }
```

Modifications to dhcpd.conf

Once you have created the file /etc/dhcpd.conf and have entered the basic information, it will be necessary to add an entry for each machine that you wish to remote boot using DHCP. In the sample file above, the entries for hpcdev02 and hpcdev03 should be replaced with entries for machines in your cluster. Once all of the systems have been added to /etc/dhcpd.conf, you are ready to start the DHCP server.

Starting the DHCP Server

Depending on which DHCP server you are using, some arguments may be required. If you are using version 3 of the ISC DHCP, all of the defaults are sufficient and it is not necessary to provide any additional arguments.

Prior to starting the server, a dhcpd.leases file must be created. See the documentation provided with this product for instructions on where to create this file and any additional configuration options.

Once the product has been configured in accordance with the documentation, add the following command to /etc/rc.d/rc.local to have the DHCP server automatically start each time the system is booted.

Figure 3: Start the DHCP Server in /etc/rc.d/rc.local

```
#
# Automatically start the DHCP server each time the system is booted.
#
/sbin/dhcpd
```

4. Configuring the TFTP Server

Which TFTP Server

The Trivial File Transfer Protocol has been around for a long time. Consequently there are many version and variations of it that are available today. The RedHat distribution comes with a vanilla TFTP server that will work well, however, the data size for each packet is limited to 512 bytes. This makes the protocol very inefficient.

More recent versions of the TFTP server, called enhanced TFTP, increase the data size for each packet to 1408 bytes, nearly tripling the file transfer performance. For this reason, we chose to use InCom's Enhanced TFTP server.

Obtain and Install TFTP

The InCom TFTP server is a free product that is available from the following URL:

http://cui.unige.ch/info/pc/remote-boot/soft/incomtftpdlx.tar.gz

This product is distributed in binary format and requires no special configuration. To install the product unzip and untar the distribution and copy the appropriate binary to the /sbin directory.

Starting the TFTP Server

Since the tftp daemon does not use a configuration file, there are a variety of options that must be specified on the command line. The documentation that accompanies the product will describe all of these in detail.

Once you have decided which options you're system will need, add a line to the rc.local file to automatically start the TFTP daemon each time the system is booted.

Figure 4: Start the TFTP Server in /etc/rc.d/rc.local

```
#
# Automatically start the InCom TFTP server each time the system
# is booted using the following arguments:
#
# -c 64 : Set the number of concurrent TFTP transfer to 64
# -d /tftpboot : Set current directory for daemon
# -h : Enable read ahead buffers
# -i 0 : Exit after num seconds of inactivity (0 = never)
# -l /var/log/tftp.log : log all messages to this file
# -r : restruct access to current directory
# -s 1408 59 : Set segment size to 1408 and port to 59
# -v 2 : Set message verbosity to 2 (high)
/sbin/tftpd -c 64 -d /tftpboot -h -i 0 -k 5 -l /var/log/tftp.log \
   -r -s 1408 59 -v 2 &
```

5. Configuring the BPBatch Loader

Obtain and Install BPBatch

The most recent version of BPBatch can be downloaded from: http://www.bpbatch.org. This website also provides in-depth documentation for BPBatch, as well as, guidance on how to develop scripts and how to use the product in conjunction with DHCP and TFTP.

Once you have downloaded, unzipped and untarred the distribution, install the following three files in the /tftpboot directory: bpbatch.P, bpbatch.ovl, and bpbatch.hlp. These are all of the files necessary to start the mini-shell on the remote client.

Create the Boot Script

Each system will require a boot script to be loaded by the BPBatch mini-shell. For simplicity we have chosen to call the file boot.bpb. The boot.bpb file is stored in the directory: /tftpboot/boot/xxx.xxx.xxx.xxx, where xxx.xxx.xxx.xxx is the IP address of the client machine.

While there are many commands that can be used in the BPBatch mini-shell, our script only contains the commands necessary to load the linux kernel and start a kick start install.

Note: The line wrapping on the linuxboot command is for readability in this document only. In the actual file this data should all be on the same line.

Figure 5: boot.bpb File to configure a node with Kickstart

```
#
# Turn off local caching to avoid disk space problems
#
set CacheNever=”On”

#
# Call linuxboot to start the build:
# The first argument is the kernel file name: vmlinuz
# The second argument are the command line parameters, in this
#  case it tells linux where to find the kickstart config file.
# The third argument is the name of the ram disk image.
# The kernel and ram disk files will be downloaded from the
# server using TFTP.
#
linuxboot "vmlinuz" \
"ks=nfs:hpcdev01:/tftpboot/boot/129.57.41.128/ks.cfg" \
"initrd.img"
```

Installing the Linux Kernel

In order to install RedHat linux, the kernel and ramdisk will need to be available for download via TFTP. From the RedHat distribution copy the following files to the /tftpboot directory:

Kernel: /redhat/redhat-7.1-en/os/i386/images/pxeboot/vmlinuz

Ram Disk: /redhat/redhat-7.1-en/os/i386/images/pxeboot/initrd.img

These files will be automatically downloaded from the TFTP server when they are needed.

6. Configuring the Kickstart Script

Create the Kickstart Configuration File

The next step is to create a kickstart configuration file that will load the operating system and the desired packages for each remote client. This file (named ks.cfg) should be installed in the same directory as the boot.bpb file. (In this example, xxx.xxx.xxx.xxx should be replaced with the IP address of the remote client)

Example: /tftpboot/boot/xxx.xxx.xxx.xxx/ks.cfg

The kickstart documentation defines all of the options that can be included in a kickstart configuration file. For illustrative purposes, we have provided a kickstart file with a basic server installation.

Figure 6: Sample Kickstart configuration file

```
# Set the default language to English...
lang en_US

# Configure the network using DHCP
network --bootproto dhcp

# Install the distribution using NFS
nfs --server mirror --dir /mirror/redhat/redhat-7.1-en/os/i386

# Setup hardware devices
device ethernet eepro100
keyboard "us"

# Partition a 20 GB disk
clearpart --all
part /boot --size 500
part / --size 4000
part /iodisk --size 14000
part swap --size 1000 --grow

# Install RedHat rather than upgrading
install

# Define other configuration options
mouse genericps/2
timezone America/New_York
xconfig --server "Mach64" --monitor "generic monitor"
rootpw --iscrypted s$d$sSssWaqd$XV/PWLsFPW.SA.BmWi1Cu1
auth --nisdomain HPCCLUSTER --nisserver hpcfs1 --useshadow
lilo --linear --location mbr

# Reboot following installation
reboot

# Install the predefined Server packages
%packages
@ Server
```

Additions to the Kickstart Configuration

Using the above configuration file, the remote node will be happy to boot, install linux, reboot and then install linux again... forever. In order to prevent the configuration process from continuing indefinitely, it will be necessary to add a few commands to the post installation section of the kickstart script.

This set of commands will mount the remote client's configuration directory from the server and will rewrite the boot.bpb file, causing the node to boot from the local hard drive on all subsequent iterations.

Figure 7: Updating the boot script using kickstart

```
#
# Commands in the %post section will be executed after
# the installation process has completed.
#
%post
# ------------------------------------------------------------------
# This set of commands will perform the following tasks:
# 1)   Mount the boot configuration directory for this node
#      (129.57.41.128) from the DHCP/TFTP server (hpcdev01)
# 2)   Make a backup of the existing boot.bpb file
# 3)   Create a new boot.bpb file that
#      a) Releases memory held by the boot prom (hidebootprom)
#      b) Boots the client node from its local hard drive
# 4)   The script will then unmount the configuration directory
#      and remove the mount point.
# ------------------------------------------------------------------
mkdir -p /rboot
mount hpcdev01:/tftpboot/boot/129.57.41.128 /rboot
cp -f /rboot/boot.bpb /rboot/boot.bak
echo "hidebootprom" > /rboot/boot.bpb
echo "hdboot" >> /rboot/boot.bpb
umount /rboot
rm -d /rboot
# ------------------------------------------------------------------
# This client should autmatically boot from the local hard drive
# on all subsequent start-ups.
# ------------------------------------------------------------------
```

Other Kickstart Operations

Following these commands in the Kickstart configuration file, any number of other post processing steps can be added. Please note, you should always update the boot.bpb file prior to performing any other post installation operations. In the event that one of these other operations causes the kickstart script to fail, you probably do not want the node to continue re-installing linux indefinitely --- or until you detect and correct the error.

7. Exporting File Systems

Why Export File Systems

In order for Kickstart to access files that are located on the server, it will be necessary to export them through NFS. There are several other strategies that may be used to allow these files to be accessed and updated, however, each of them has it’s benefits and drawbacks. Using NFS appears to be the most straightforward approach and, on a private network, it offers an adequate level of security.

Which File Systems to Export

At a minimum the /tftpboot directory must be exported in read-only mode to allow kickstart to mount the file system and obtain the ks.cfg file. Additionally, in order for the kickstart configuration script to update the boot.bpb file, each node’s configuration directory will need to be exported. Since these directories must be exported in read-write mode, we have chosen to limit there accessibility.

Modifying the exports File

The following entries should be added to the /etc/exports file to allow all necessary directories to be exported.

Figure 8: Modifications to the /etc/exports file

```
#
# Export /tftpboot in readonly mode globally
#
/tftpboot                           *(ro)

#
# Export individual configuration directories only to the
# node that requires them. You’ll want to change these entries
# to match your node names and IP addresses.
#
/tftpboot/boot/129.57.41.128        hpcdev02(rw,no_root_squash)
/tftpboot/boot/129.57.41.129        hpcdev03(rw,no_root_squash)
...
```