Setting up thin Clients At The Wendell Free Library

Robert Heller
Deepwoods Software
Wendell, MA, USA *

August 1, 2009

Abstract

One of the services that many public libraries provide is computer access, both for local applications and access to Internet services. The Wendell Free Library provides these services with thin[1] Linux workstations. These workstations provide a web browser (Firefox), for access to web sites of all sorts and also provide a number of local applications, including an office suite (OpenOffice), plus several other applications, including a movie player (MPlayer), an image editing program (GIMP), and a large collection of games.

Using thin clients reduces energy costs, creates a uniform environment that only needs to be updated in one place, on the server, and quieter operation. This article describes the setup and configuration of these thin clients at the Wendell Free Library and covers the step-by-step process of setting up the server and the client machines and includes the process from installing the base system on the server to configuring the PXE boot process to user authentication and management.

Contents

1 Introduction and Motivation **2**
1.1 Hardware 3

2 Server Operating System Software **3**
2.1 Daemon Programs Needed 3

3 Server File System Layout **4**
3.1 NFS Exports 4

[1] Thin clients are diskless computers, which get their operating system and application software from a file server on the local area network (LAN).

4 Booting the Diskless Clients **4**
4.1 PXEBoot 4
4.2 DHCPD and DNS setup 5
4.3 TFTPD 5
4.4 Kernel and Initrd 6

5 Diskless Clients Root file system RAMDISK **7**
5.1 Read-only root file system 7

6 Diskless Clients NFS Mounted file systems **7**

7 OpenLDAP: User Authentication **8**
7.1 Luma: Managing users and groups 8
7.2 PAM mkhomedir setting 8

8 Sendmail setup **8**

9 Firewall, Routing, and IP Masquerading **9**

10 CUPS Printer Sharing **9**

11 Handling the MS-Windows machines. **9**
11.1 Alternative Local Area Network for CWMARS 9
11.2 Samba Anonymous Printer Sharing 9

12 One "Thicker" client for burning CDs. **9**

13 Backups **10**

14 Acknowledgements **10**

1 Introduction and Motivation

Like the UBuntu-based LTSP thin clients, these thin clients are diskless. The main difference is the LTSP systems run little more than the X server process, with all of the client processes running on the server system. These thicker clients are full-fledged Linux workstations running the end-user programs directly on the client machines. These machines mount all of the basic Linux file systems as NFS file systems served by the server. This distributes both the CPU cycle resources and the memory resources amongst the client machines, rather than concentrating these resource demands on the server itself. The LTSP system has proved to be less than satisfactory, given the usage patterns, which includes heavy use of Flash-based games by the patrons. By using this style of thin client, there is both more memory as well as more CPU cycles (using faster, dedicated processors) providing more directly available compute resources for patron use.

1.1 Hardware

The client machines are little refurbished 2.4GHz P4 boxes we got from Geeks.com for about $50 each. They came with 256meg of memory and we add a 1gig DIMM to each.

The server is a Dell server with a Core2Duo processor, 2gig of RAM and 4 80gig SATA disks. We are using software RAID.

2 Server Operating System Software

The server is running CentOS[2]. This is a stable and long-term production grade operating system. With multi-media software installed from third-party repositories, the client machines have installed access to the end-user software[3] the patrons will make use of. CentOS is otherwise a stable and solid system with long term support, with ongoing upstream releases of essential bug and security updates.

2.1 Daemon Programs Needed

The server machine needs to provide a number of services for the client machines. These services include:

- NFS (Network File System) services. This service provides the file systems used by the client machines.
- DNS (Domain Name Service) services. This service maps between machine host names and IP addresses.
- DHCP (Dynamic Host Configuration Protocol) services. This service provides both the mapping from client machine hardware addresses and their IP addresses and also provides the core information needed for the client machines to get their kernel images and boot themselves up properly.
- TFTP (Trivial File Transfer Protocol) services. This service provides the transfer methodology to actually deliver the kernel and initial ramdisk images to the client machines.
- CUPS (Common UNIX Print Service). This service provides access to the shared printers that are available for patron use.
- SMTP (Simple Mail Transfer Protocol). This service allows for local communication[4]
- LDAP (Lightweight Directory Access Protocol) services. This service is used for user authentication.

[2] CentOS Release 5.3 at the time this article has been written.

[3] Such as Adobe Acrobat Reader, MPlayer, and the Shockwave Flash plugin.

[4] At present, the patrons don't make direct use of this service. It is mostly provided to allow for automated messaging between the client machines and the server machine.

3 Server File System Layout

The server has a number of separate file systems, as shown in this table:

Mount Point	Size	Description
/	1G	This is the server's root file system. It is not NFS exported.
/boot	1G	This is the server's boot file system. It is not NFS exported.
/nfsroot	2G	This is client root file system. It is NFS exported, read-only. The client machines actually has a ramdisk mounted at their root file system, with symlinks to this file system. The /var directory on this file system is copied to the root ramdisk mounted on the client machines.
/usr	6G	This is the /usr file system for both the server and the clients. It is mounted read-only on the client machines.
/var	1G	This is the /var file system for the server. Some of this file system is also mounted on the clients.
/home	50G	This is the file system containing the user home directories. It is NFS exported to the clients, which in turn mount it read-write.

The /nfsroot file system is basically a mirror of the server's root (/) and /var file systems, with some important changes. The /var file system is simplified. Primarily due to that many services running on the server instead of the clients. The settings in the /nfsroot's etc sub-directory are different, both because many services won't be running and because of the need to do things like mount selected NFS file systems earlier in the boot process than would normally be the case for a workstation with a local disk. This is covered in depth in Section 5.

3.1 NFS Exports

Several of the server's file systems are NFS exported to the client machines. This includes the client machines' root file system (read-only), the /usr file system (read-only), and the /var/lib/rpm directory (read-only). In addition, the /var/spool/mail[5] directory and the /home directories are exported (read-write).

4 Booting the Diskless Clients

4.1 PXEBoot

The diskless clients boot over the network using boot ROMs that implement the industry standard PXE (Pre-eXecution Environment). This environment uses the DHCP

[5] See below for Sendmail's mail delivery settings.

to get its initial boot images, the pxelinux.0 image that is part of the syslinux package. This image uses the TFTP protocol to download the kernel and initial ramdisk images and then uncompresses and starts the kernel, in much the same way as a normal local boot loader (such as LILO or Grub) would do. The initial ramdisk includes programs and scripts that set up the clients networking, mount the root file system's backing tree and then set up, mount, and initialize the client's root file system. Control is then passed to the init program and boot up continues in much the same way as for a "normal" disk-based workstation.

4.2 DHCPD and DNS setup

In order for the diskless clients to boot up, they need some basic configuration information. This information is provided initially by the DHCPD process running on the server. There is a group block containing common option definitions and each workstation has a DHCPD host block within the group block. Here is an example:

```
group {

  filename "/pxelinux.0";
  option root-path "server1.wendellfreelibrary.org:/nfsroot";
  next-server 192.168.1.254;

  host station1 {
    # New box adult station
    hardware ethernet 00:40:ca:7a:b4:b5;
    fixed-address 192.168.1.21;
    option host-name "station1.wendellfreelibrary.org";
  }

  # Additional host sections here

}
```

These blocks are in the server's /etc/dhcpd.conf file[6]. In addition, it is necessary to configure the DNS server as well[7].

4.3 TFTPD

The tftp server is started from xinetd[8], and is controlled by a configuration file in /etc/xinetd.d named tftp:

```
# default: off
# description: The tftp server serves files using the trivial file transfer \
#       protocol.  The tftp protocol is often used to boot diskless \
#       workstations, download configuration files to network-aware printers, \
```

[6] See the downloads section at the end of this article for a copy of the /etc/dhcpd.conf used by the server at the Wendell Free Library.

[7] Don't forget to adjust server's firewall settings to set the LAN's ethernet interface as "trusted". This is needed so that the server can accept various network connections from the clients for the various services they need.

[8] Make sure xinetd is installed and set to start. The base CentOS 5 install does not include xinetd nor does it start it by default!

```
#       and to start the installation process for some operating systems.
service tftp
{
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /tftpboot
        disable                 = no
        per_source              = 11
        cps                     = 100 2
        flags                   = IPv4
}
```

All of the boot configuration files live in the /tftpboot directory. You need to copy pxelinux.0 from the /usr/lib/syslinux/ directory, then copy the kernel and initrd images there (see the next section) and create a directory named "pxelinux.cfg", containing a single file named "default":

```
LABEL linux
        KERNEL vmlinuz-2.6.18-92.el5
        APPEND initrd=pxeboot-2.6.18-92.el5.img enforcing=0
```

4.4 Kernel and Initrd

The kernel used is the same kernel that installs with the operating system. The initial ramdisk is different. The initial ramdisk needs to be loaded up with all of the possible network drivers and the NFS file system modules. Busybox (I used, in this example, the 1.13.2 tarball) also needs to be built and installed. I've included the configuration file I generated with this article's downloads. I also wrote a greatly simplified and statically linked version of lspci, the source code of which is also included. I adapted a startup script that does the following:

- Auto detects the network device and installs its driver.
- Installs the NFS modules.
- Initializes the network (both lo and eth0). Eth0 is configured using DHCP.
- The NFS root file system is mounted read-only.
- A RAMDISK file system is mounted as the system root.
- All of the toplevel files and directories on the NFS mounted root file system are symlinked to the root RAMDISK file system.
- A selection of the files and directories at the root file system are specially handled, including root's home directory, and various mount points.
- The /var file system is copied to the RAMDISK.
- The /dev file system is initialized.

- /proc/sys/kernel/real-root-dev is initialized.
- And finally, the root file system is switched into place with the `pivot_root` function.
- Control is then passed to init and the system proceeds with the boot up process.

5 Diskless Clients Root file system RAMDISK

The diskless clients run with a root file system that is actually a RAMDISK. This RAMDISK is initialized with symbolic links to a read-only mounted NFS file system. Selected directories (/var and /root) are either copied during the boot process. Certain files are created either on the fly or are symbolic links, based on the client's host name. The special files and directories are created initially under /nfsroot/var/etc and consist of:

- A resolv.conf file is generated from information retrieved via DHCP.
- A client-specific version of /etc/modprobe.conf. This file contains module aliases for client-specific driver modules, generally for the sound card, but might also include other devices, such as special USB, Firewire, or other devices.
- A client-specific version of the /etc/sysconfig/ directory. Again anything specific to the client that would normally be configured in /etc/sysconfig/ goes here.

5.1 Read-only root file system

Mostly, /nfsroot is a copy of the server's own root and /var file systems. Some important differences include: Several scripts under /etc/init.d/ have to be modified and selected files under /etc are symlinked to files under /var/etc[9]. There are no local file systems, instead there are some read-only file systems, namely /usr and /var/lib/rpmdb[10] that have to be mounted **early**. The rc.sysinit needs to be modified to skip bothering with fsck'ing and to mount these file systems early. It is also important **NOT** to unmount the root file system during shutdown, this means that the halt and netfs scripts need to be modified to not unmount read-only NFS file systems. This is slightly tricky since the root file system is not actually the NFS file system, but a RAMDISK that has symbolic links to a NFS mounted file system. Other changes include turning off probing for new or changed hardware, since the file system where these changes would be recorded is a read-only file system.

6 Diskless Clients NFS Mounted file systems

The fstab file on the diskless clients looks like this:

[9] These files are "computed" at boot time.

[10] /var/lib/rpmdb does not have to be mounted early, but because it is marked read-only, it gets mounted when /usr gets mounted.

```
#none                    /                          tmpfs    defaults
0 0
2 none                   /dev/pts                   devpts   gid=5,mode=620
0 0
none                     /proc                      proc     defaults
0 0
none                     /dev/shm                   tmpfs    defaults
0 0
server1.wendellfreelibrary.org:/usr      /usr              nfs      ro,nolock      0 0
server1.wendellfreelibrary.org:/lib/rpm  /var/lib/rpm      nfs      ro,nolock      0 0
7 server1.wendellfreelibrary.org:/var/spool/mail /var/spool/mail nfs      rw             0 0
server1.wendellfreelibrary.org:/home     /home             nfs      rw             0 0
```

The two read-only file systems, /usr and /var/lib/rpm are mounted by rc.sysinit and not by the netfs script. The other file systems, /var/spool/mail and /home are mounted later.

7 OpenLDAP: User Authentication

User authentication is managed through OpenLDAP and slapd is run on the server. Extensive documentation on how to set up OpenLDAP and slapd and how to migrate to OpenLDAP exists on the CentOS and RedHat websites. A good place to start is on the CentOS web site: `http://www.centos.org/docs/5/html/Deployment_Guide-en-US/ch-ldap.html`.

7.1 Luma: Managing users and groups

I installed Luma (Version 2.4, built from source downloaded from SourceForge) to provide the librarian with an easy to use GUI program to manage user accounts. Most patrons will use the guest accounts provided, but a handful will want to have their own personal logins[11]. Luma is available from SourceForge: `http://luma.sourceforge.net/`.

7.2 PAM mkhomedir setting

I have enabled the PAM mkhomedir, so that new user's will have the home directory automatically created.

8 Sendmail setup

I have set up Sendmail on the server as a mail hub for the diskless machines and configured Sendmail on the client machines to pass local delivery to the server. This avoids possible problems with file locking on the mail spool.

[11] I will attempt to encourage regular patrons to move towards having their own logins, as this enhances security and privacy and provides additional benefits including the ability to maintain a 'fully furnished' home away from home.

9 Firewall, Routing, and IP Masquerading

The server at the Wendell Free Library also acts as a router to the public Internet through a pair of HughesNet satellite dishes[12] The server provides a fire wall and performs IP Masquerading between the local network and the public Internet.

10 CUPS Printer Sharing

The server also provides shared printer spools for the two wired network based printers. I set the cupsd.conf file on the server to "share" its printers with computers on the LAN and set up the clients to look for shared printers.

11 Handling the MS-Windows machines.

There are two MS-Windows machines on our network. One for patrons for the handful of programs patrons used that only run under MS-Windows (mostly kids learning games) and one for the librarians (mostly to run the special CWMARS circulation program).

11.1 Alternative Local Area Network for CWMARS

There is a MS-Windows PC is on an alternative Local Area Network[13]. This setup is to allow an alternative, protected network for CWMARS business, one that is logically isolated and secured from patron access. I set up aliases network interfaces on the server and on the two staff clients. This allows the server to provide print services and allows the two staff client machines to use vncviewer to remotely access this MS-Windows machine. This allows the librarians and selected volunteers to use the CWMARS cataloging and circulation software to manage inter-library loans and to update Wendell's section of the CWMARS catalog database.

11.2 Samba Anonymous Printer Sharing

Both the CWMARS MS-Windows and the one patron MS-Windows machine need access to the printers. Sine MS-Windows does not implement CUPS, the printers are shared using the Samba software package, which provides NETBIOS services. In this case, simple anonymous printer sharing is provided by the server.

12 One "Thicker" client for burning CDs.

One of the clients is a full fledged Linux system, with its own hard drive. This machine includes a CD-RW drive. I left this machine with a disk, since burning CD-Rs

[12] We plan to be moving to a new Internet provider using a 900MHz wireless link provided by an adjacent town.

[13] Same physical network, but in a different private IP address range.

involves time-critical disk I/O and I would doubt that NFS over a busy network connection would be fast enough to keep up with the CD burner. Other than booting off its local disk, it functions much like the diskless clients. It uses OpenLDAP for user authentication and auto-mounts user home directories.

13 Backups

Backups are always important and I have set up an automated, once a day backup. This uses the dump utility and dumps daily incremental backups to a USB external disk[14] connected to the server. Once a month, full (level 0) backups are performed on all of the drives. Once a week, level 1 backups are performed on selected drives, and the rest of the week, a Tower Of Hanoi sequence[15] of incremental backups are performed on the busiest drives (/home and /var).

A USB connected DVD burner is also connected to the server. Early each month, a manual backup of the previous month's backup will be made on DVD-Rs, which will be then stored off site.

14 Acknowledgements

I'd like to thank Christine Texiera for proofreading this document. She some useful comments and found a bunch of small errors. Thanks Christine!

[14] A 160gig SATA laptop drive in a USB external enclosure.
[15] Levels 3 2 5 4 7 6 9 8.