LINUX TERMINAL SERVER PROJECT (LTSP)

Nitesh Rijal
BE IT, NCIT
+977-9841458173
rijal.it@gmail.com

Pramod Bhandari
BE ELX, NCIT
+977-9849158004
catchpramod@gmail.com

ABSTRACT

This paper focuses on implementation of LTSP (Linux Terminal Server Project) to make complete use of old hardware to be used efficiently and effectively. This paper focuses its implementation in GNU/ Linux.

Keywords

LTSP, TFTP, ROM, PXE, X, chroot, DHCP, IP

General Terms

LTSP, X-server, Thin Client

1. INTRODUCTION

One of the key technologies included in most modern GNU/Linux operating systems is the Linux Terminal Server Project (LTSP) which allows us to boot many thin clients from a single server. For educational environments, LTSP lowers hardware costs by enabling the use of older or less powerful machines as thin clients, as well as reduced administration overhead by having only to install and maintain the software on the server. Thin clients are generally useful for 5 to 10 years, compared to 2 to 4 years for fat clients. When a workstation fails, it can simply be replaced without data loss or re-installation of the operating system.

Thin client computing has been around for a long time in the UNIX world. Although the implementation has evolved quite a bit, the concept has remained the same.

• The thin client only takes care of the basic functions like display, keyboard, mouse and sound.

• The server does the heavy weightlifting. All the applications run on the server, and they simply display on the thin client.

Because the thin clients have a limited number of tasks to manage, the hardware for the thin client can be small and cheap. The thin clients themselves are basically maintenance free. They last longer because they have no storage with moving parts like hard disks. If they break no data is lost since nothing is stored on the client itself. Simply swap the client with another one and go back to work. If your thin client is stolen or put in the trash, no data ends up in the hands of unauthorized people.

2. WORKING

2.1. How LTSP works

LTSP is a collection of software that turns a normal GNU/Linux installation into a terminal server. This allows low-powered, low-cost thin-clients (or legacy hardware already in possession) to be used as terminals to the thin-client server. It requires only a PXE capable network interface, which many thin-clients and PCs have built-in already. This means that you need absolutely no physical storage media (hard disk, compact-flash, etc.) for your thin-client to boot to LTSP.

The process of booting a thin-client to an LTSP server is as follows:

- Thin-clients boot via a protocol called PXE (Pre-eXecution Environment)
- PXE requests an IP address from a local DHCP server
- The DHCP server passes additional parameters to the thin-client and downloads a Linux file system image via TFTP into a RAM disk on the client itself.

- The thin-client then boots the downloaded GNU/Linux image, detects hardware, and connects to the LTSP server's X session (normally handled by LDM).
- From here, all operations such as authenticating your username and password, launching applications, and viewing websites are actually handled on the LTSP server rather than the thin-client. The LTSP server transfers all graphical information to the thin-client over the network. This allows very low powered thin-clients to utilize the power of the server for all operations. It also allows for large client deployments with reduced overall resource utilization, as 50thin-clients all running the popular OpenOffice suite under different sessions generally only require enough RAM for a single instance of OpenOffice (excluding per-user configuration which is minimal).

3. BENEFITS

3.1 Secure

Security has become a key challenge for administrators and LTSP both recognizes and handles this quite well. Often schools lack the specialized IT staff or time to lock and clean up computers. Operating systems with LTSP included, being GNU/Linux based operating systems, enjoy the security advantages of its Unix-like and open source heritages. This translates into higher quality code, and spyware and viruses are of minimal concern. In addition, it has a strict, proactive security policy which means that many common problems, such as open ports or misconfigured software, never make it into the released product. Finally, LTSP based systems are true multi-user operating systems, making it easy to allow users to complete their tasks without having a level of access that could compromise the system.

3.2 Maintenance

With administrators and especially school IT departments deploying and administering an increasing number of computers, it is difficult to find time to manage individual machines. LTSP thin client technology, makes deployment and maintenance simple and easy. The server runs all applications and contains all the data. All the regular maintenance (software updates, administration) takes place on the server. A single server is all that is needed to set up, manage and administrate an entire network.

3.3 It's Green

With the ongoing debate about climate change, questions are finally being asked and answered in the fields of IT, education and thin client technology in general. A recent study compared the energy and resource consumption of a regular PC and Thin Client setups.

You can find that study here: http://it.umsicht.fraunhofer.de/TCecology/-index_en.html

They found that thin clients use half the energy of traditional workstations, which not only helps on the cost savings (calculate that a 40 terminal thin client lab will save approximately $500-$800 per year), but is ecologically effective in avoiding electronic waste and high carbon emissions. Thin client production, assembly and logistics costs far less and requires less energy than traditional PC manufacturing. The recycling of old machinery also helps the environment, making LTSP a green solution to the environmental and power saving issues many IT managers face today.

3.4 Cost effective

With ever-increasing demands on school budgets, expensive technology is often last on the list. LTSP can help you offer what your students increasingly require from computer technology, without breaking the bank. GNU/Linux is and always will be free to acquire, use and modify, including the underlying LTSP structure that holds it all together.

Need to set up another machine? Or another 100? Just install them! With GNU/Linux you'll have no more expensive OS upgrades and licenses, and having specialized programs on only some computers will become a thing of the past.

LTSP can also help you save hardware costs, by allowing you to redeploy older machines as thin clients using LTSP technology.

3.5 Well supported

LTSP support is available from both the LTSP community. Many of the authors of the software included in LTSP, including the respective developers of the various LTSP GNU/Linux implementations themselves, can be contacted directly via mailing lists and IRC channels. There are many forms of support available, including mailing lists, Wiki websites, IRC channels, and bug trackers. There are also special support groups for using LTSP and GNU/Linux.

The official IRC support channel is found on ***freenode.org at #ltsp***

The official LTSP mailing list is found here: ***https://lists.sourceforge.net/lists/listinfo/ltsp-discuss***

3.6 Built for education, government and business

LTSP based distros (Ubuntu, SuSe, Nepalinux, etc.) come with translations for many languages and localization features that allow people from all over the world to enjoy their computing experience. Accessibility features strive to provide a pleasant, high-quality computing experience to disabled users. Most distributions have software and applications that runs at school or work. The LTSP server software allows administrators, IT managers and teachers to create a low cost computer lab so that users can have access to the opportunities that GNU/Linux and the Internet can provide.

4.HARDWARE REQUIREMENTS

A person setting up a LTSP thin client environment for the first time typically asks two questions:

- ***Will my existing machines work as terminals or, what should I buy to use as a terminal?***
- ***How big a server do I need?***

Chances are, hardware that you already have is more than sufficient for terminals. One of the great advantages of an LTSP Server is that you can set up a high quality lab of terminals for your students to use, by leveraging the machines you already have. As for servers, usually, it's very easy to turn any high-end single user desktop machine into a terminal server capable of handling many thin clients. We'll present some guidelines that should help in making the most of your resources.

There are four key areas to look at:

- **Server**
- **Thin client**
- **Network**
- **Boot method**

4.1 Server

An LTSP thin client network is quite scalable; a moderately powerful machine can serve several thin clients, and if you need to add more thin clients, you can either expand the capabilities of the existing server, or, simply add more servers.

Server sizing in an LTSP network is more art than science. Ask any LTSP administrator how big a server you need to use, and you'll likely be told "It depends". How big a server you need does depend largely on what it is you're planning on doing with your thin client network. Here are some common guidelines that should fit most "average" cases.

4.1.1 Memory

A GNU/Linux based operating system makes efficient use of memory. The usual formula that's used for adding memory to a thin client server is: ***256 + (192 * users) MB***

So, if your target is to have a server with 20 terminals, you'll need:

256 + (192 * 20) = 256 + 3840 = 4096 MB

So four Gigabytes of memory is a fair estimate for the server. Making sure you've got enough memory is the single most important thing you can do to help the performance of an LTSP thin client server. If you do not have enough memory in your server, you’ll find your server will have to use the hard drive as an overflow "virtual" memory. Hard drives are much slower than memory, so you’ll find things getting very slow if this happens. If you intend to make heavy use of graphics work in your curriculum, you may want to add even more, perhaps doubling the previous estimate.

4.1.2 Processors

How fast a processor you need is entirely dependent on what programs you plan to use. Interactive games require a bit more than say, a word processor. If you plan to use Java and Flash plug-ins in your web browser, these can consume a lot of processing power. For a "mixed" model, i.e. some people playing TuxMath, a few people browsing the web, and a few people typing in OpenOffice.org, a 2GHz or better processor should be able to adequately handle 20 people with some minor delays. A 3GHz processor would be better.

For larger networks, moving to an SMP (Symmetric Multi Processing), or multiple CPU server may be advantageous. If you plan to handle 30 or more clients, a newer dual-core Xenon server or dual-core Opteron will provide good results.

4.1.3 Disks

It’s advisable to use some form of RAID in the terminal servers. Besides saving your data when a single disk fails, it improves the performance (especially read performance, which is the most common type of file access). For people on a budget, setting up software RAID 1, with 2 SATA disks with NCQ (Native Command Queuing) will provide good results.

4.2 Thin clients

A lot of LTSP deployments are in classroom environments, and usually, in these situations, the primary goal is to re-use existing hardware that the school already owns. However, specifically designed thin clients can be used.

4.2.1 Older hardware

There are several things to consider when trying re-using existing hardware.

- **CPU**
- **Network**
- **Thin Client Ram**
- **Video Card**

4.2.2 CPU

For using the default, secure mode of LTSP, you’ll need to have a slightly faster CPU. Any 533 MHz or better CPU should provide acceptable performance. If you have slower clients, in the range of 233 MHz to 533 MHz, you may be able to use them, if you’re willing to reduce the security of your thin client network.

4.2.3 Network

A thin client boots over the network, using a small program called a network boot loader. This network boot loader is sometimes located on the card itself, or, for older cards without one, the user can provide one on a floppy or CD Rom which can be used to boot the thin client.

The common network boot loaders which can be used are:

PXE: This one is the most common, and many network cards and motherboards with built-in network cards support this. If you have one of these, you’ll be able to boot without any problems.

gPXE: The gPXE boot loader is a Free implementation of the PXE boot loader. It is the successor to Etherboot, and is maintained by the Etherboot project.

You can find out more information at: ***http://etherboot.org*** , and floppy and ROM images may be downloaded from: ***http://rom-o-matic.net***

Yaboot: For Macintosh PowerPC machines (iMac’s and later), you can use the built in Yaboot network boot.

4.2.4 Thin Client Ram

The bare minimum for a thin client to work is about 48MB, but it will be slow, so it is recommended to install at least 128MB Ram, with 256MB Ram if you can spare it. This will really help speed up thin clients.

4.2.5 Video Card

Typically, any video card that uses the PCI bus and has 32 MB or more of memory should make a reasonable client.

4.3 Network

If you have more then 20 users, it is recommended to use Gigabit Ethernet connected to a gigabit port on a switch for your LTSP servers if you plan to have a lot of graphically intensive programs running. Although normal usage ranges from 0.5 to 2mbit, clients can peak quite high (70mbit), especially when watching multimedia content. For office or business computing, with only moderate graphics usage, 100 Megabit networks should be fine.

4.4 Boot method

Getting the thin client to boot over the network can be accomplished in a variety of ways:

- **Boot ROM**
- **Local media**

4.4.1 Boot ROM

Depending on your network card, it may already contain a boot ROM, or you may be able to use an EPROM programmer to create your own. Check the hardware documentation for the network card in your thin client for details.

4.4.2 Local media

If your network card in the thin client doesn’t have a boot ROM built in, and you don’t have access to an EPROM burner, have no fear! Chances are, that old machine has a floppy drive, or CD-ROM in it. If so, then you can use local media to boot the thin client.

Floppy disk: Booting Etherboot from a floppy is an excellent way of booting an LTSP thin client that doesn’t have a boot ROM. Etherboot is loaded in the boot sector of the floppy. The boot code will be executed, the network card will be initialized, and the kernel will be loaded from the network server.

Hard disk: The hard disk can be used with LILO or GRUB, to load the Linux kernel and initrd. You can also load the Etherboot bootrom image from the hard disk and it will act like a bootrom.

CD-ROM: A bootable CD-ROM can be loaded either with a Linux kernel, or an Etherboot image.

USB Memory device: Just like a CD-ROM, Floppy disk and Hard disk, you can use a USB Memory device to boot an Etherboot module.

5. LTSP IN NEPAL

For a developing country like Nepal this technology can help to minimize the digital divide. The computers which are old and not used in city areas can be reused to provide computer education in rural areas. Since the cost of deploying a LTSP lab is very low, most of the schools can have this technology. A good example of this kind of project can be establishment of LTSP based e-library in some schools in rural parts of Nepal.

Currently FOSS Nepal along with Help Nepal Network (HeNN) is involved in implementation of LTSP e-library in rural parts of Nepal. Some of the instances are successful implementation in Myagdi, Dang, Bhaktapur and Banepa.

6. CONCLUSION

From this paper it is clear that LTSP is a very viable yet efficient option for large and small scale purposes. It comprises a fusion of security feature of Linux as well as the principle of e-learning through e-library.

This will be always a good option because we use Free and Open Source software.

7. ACKNOWLEDGEMENT

We would like to thank everybody involved directly or indirectly for this paper, especially Mr. Saroj Shakya for his support.

8. REFERENCES

a. www.ltsp.org
b. www.google.com
c. www.wikipedia.org
d. http://etherboot.org
e. http://it.umsicht.fraunhofer.de