Linux Terminal Server Project (LTSP) Practice Guide

Linux Terminal Server Project (LTSP) Practice Guide

Table of Contents

Author

Ben Green, ben@bristolwireless.net

(April 2007)

Audience

This guide is aimed at a technically oriented audience.

Summary

Organisations that use Linux Terminal Server Project (LTSP) to add thin-client support to Linux servers have speedier applications, easier application deployment, lower software cost and much lower administration cost.

LTSP is a FOSS (Free and Open Source Software) project. LTSP (Linux Terminal Server Project) is a set of tools for allowing low specification computers to connect to a central server. Users of these remote computers may log in to the server and use applications there, or run applications locally on the client.

LTSP is most easily applied to office systems. Shortcomings for other types of applications, such as sound programs, may be overcome by additional configuration, and by a number of new LTSP technologies like local device support.

LTSP can reduce cost and power use in situations where there are as few as two concurrent users on one system.

Advantages of LTSP over proprietary solutions include:

- Lower administration costs
- Lower hardware costs
- Lower power consumption
- Greater application speed
- It is a Free and Open Source solution

Use LTSP wherever there are no reasons not to. Increased set up and maintenance difficulties arise when particular legacy applications need supporting. LTSP is not currently appropriate where very high performance multimedia tasks are being undertaken such as:

- Live music performance
- Video editing

Migration costs may be the only factor limiting uptake. Deployment may be partial and also incremental.

This document deals with issues of deployment including software and hardware choices, maintaining services, legacy applications and calculating total costs for a migration.

1. Server Sizing

For system of less than 40 users a single dual cored server should be adequate, depending on the demands put upon it. Beyond that, some separation of services should be considered.

On older Intel systems the cost of switching tasks between different processors in SMP systems was high, meaning bad performance under load and unresponsive applications. Avoid these architectures.

- **Scale to your needs** – remember to pitch your systems at the maximum number of terminals you will need. Designing a system that will scale to an appropriate size means you can add more capacity.
- **Think about your applications load** – qualify and quantify the particular applications people will be running and consider policies that limit users or workstations to specific applications.
- **An appropriate desktop environment** (DE)– for users with limited application use, a lower impact DE can cut down considerably on memory use. Those suitable for LTSP include XFCE4, JWM, ICEWM and Flux Box.
- **Multi-processor systems are good** – SMP systems ensure that no one user can dominate the computer's resources, and no additional configuration is necessary to achieve this. The maximum speed any single threaded application can run is that of the largest single processing unit; hence each individual processor needs to be fairly fast to maintain responsive applications.
- **Work out your maximum memory constraints** – things move very slowly when swap space is being used. Make sure you have enough main memory to run all your applications without swapping. This demands experience of application use in a multi-user environment.

  http://wiki.ltsp.org/twiki/bin/view/Ltsp/ServerSizing

  http://virtualthreads.blogspot.com/2006/02/understanding-memory-usage-on-linux.html (how to properly measure memory use on Linux)

2. Client Level Quality of Service

- **No hoggers** – A robust LTSP system must ensure that no one can hog the application server's computer resources. A variety of Unix tools can be used to achieve this including:
  - Network traffic shaping on the internal network per client
  - Process limits per user
  - File system quotas per user
- **No problem on small systems** – The Linux kernel usually does a good job by itself, balancing demanding processes and clearing memory of used code. Dead processes may build up. Use disk swap to provide an area where these processes can reside outside of main memory.

  http://www.yolinux.com/TUTORIALS/LinuxTutorialQuotas.html (filesystem quotaing)

  http://gentoo-wiki.com/SECURITY_Limit_User_Processes (process limits)

3. Choose your Distribution

3.1 Considerations

- **Software available** – whatever software is available to the server will also be available to the clients. Choose an operating system that has the software you want your users to be able to use, or one that you are comfortable customizing applications for.
- **Support** – if you need external software support for your server systems, choose a distribution with a support structure you like.
- **LTSP integration** – LTSP-5.0 is integrated into distributions, rather than being a separate install. This system has the best support for local applications and uses the distribution's own package management system to build the client image. If you wish to use local applications, use a distribution supporting LTSP-5.0

  http://wiki.ltsp.org/twiki/bin/view/Ltsp/Ltsp5Status (status of LTSP-5 integration)

3.2 Recommendations

Debian is at the time of writing the only distribution with a record of high security and stability that contains a wide range of packages. The package count for the next stable release is getting on for 20,000. This wealth of software includes most good stable and actively maintained open source releases. For a flexible, sustainable and completely free solution for LTSP, Debian is highly recommended. LTSP-5.0 support is also good in Debian - and LTSP-4.2 works very well too.

An often cited disadvantage of Debian is that the stable releases are outdated at release because of their long release cycle - the incredible pace of Linux development means that users can miss out on innovations for some time. Issues of hardware support are not prevalent in LTSP environments.

For an off the peg educational solution, K12LTSP is also recommended. Note however that an easy installation does not necessarily mean an easily upgradable long term solution.

The Debian project with similar aims to K12LTSP is DebianEdu. There is an associated live CD project called SkoleLinux.

http://debian.org

http://wiki.debian.org/DebianEdu (and the English language Skolelinux page)

http://k12ltsp.org

4 Redundancy

Managing data and service redundancy are both relatively easy in a thin client system because data is located on few servers, or often just the one. Research appropriate strategies after ascertaining your user requirements.

- **Split application servers** – In a large LTSP system applications can be split between a number of servers. For example, each client may connect to a different application server based upon which server is least loaded. Home directories may be shared via a distributed file system.
- **Keep backups** – data redundancy for users is essential. Back up the system and

home directories via mirroring or a RAID set up.

- **TFTP, DHCP, NFS, X** – these are LTSP's essential services. The server(s) providing X and NFS will also be serving the applications. Separating the other two services is possible.

http://republico.estv.ipv.pt/~nmct/ltsp/ha/ (high availability through 2 application servers)

http://www1.uni-hamburg.de/RRZ/W.Buech/LTC-linux-kongress.pdf (another method of application load balancing through clusters).

http://www.linuxjournal.com/article/9097 (splitting all the LTSP services out as much as possible)

5. Clients

Higher transistor densities mean computing systems use less power. Thin clients don't need much processing power unless local applications are being used. High speed graphical interfaces are more crucial to user interactiveness than high processing speed. The newer the technology a thin client uses, the lower the power.

Recent systems based on the SiS SoC (system on a chip) have produced clients with excellent graphical capabilities and typical power usage of 5W. This figure is likely to become more common place, though around 20W is more typical for a thin client in the current market. For re-purposed desktop machine power usage will probably at least 40W.

- **Use low power** – with increasing power costs the sustainability of computer systems depends increasingly on their power consumption performance. Clients are available that run on as little as 5W each, and 15” monitors at 14W. Research and test in use before buying.
- **Check power requirements if you are re-using old hardware** – LTSP systems using old desktops converted to LTSP can reduce power consumption hugely over standalone systems. Remove hard drives and other unused peripherals to save the most power. Consider underclocking and fan removal to further reduce power use. Research and test power use.
- **Convert clients using PXE Ethernet cards** – second hand PXE based cards are a very cheap way to get thin clients booting, requiring only that they have a PCI slot.

6. Security

- **Separated networks** – application data passes over LTSP networks unencrypted. Separation of open and thin client networks reduces the chance of intrusion.
- **Physically secure the network** – protect users from compromising person/ personal data by ensuring the LTSP segment is not accessible to other users.
- **Local Data Manager (LDM)** – open networks might be vulnerable to snooping, use the LDM login manager to provide a secure session over SSH.
- **Remote shut-down** – clients can be shut down remotely using the *ltspadmin* command, saving time and energy and allowing a quick response in the event of a security compromise.
- **Client monitoring** – you can monitor clients by running a VNC server on each of them.
- **Turn off diagnostics** – some diagnostic features of LTSP have security issues.

Make sure they are turned off in a production system.

http://www.geocities.com/hammerjw/ltsp-basic_security.html (basic guide to LTSP security)

http://www.k12ltsp.org/mediawiki/index.php/Using_VNC_to_shadow_clients (monitoring client desktops)

7. Other Considerations

- **Sound support for limited applications** – due to the lack of a common network transparent sound API for Linux, remote sound needs to be set up on a per application basis. Latency is a problem. Use local applications to run sound applications locally. Don't expect to edit video via LTSP. For music software local applications may be used.
- **Local storage devices** – current LTSP implementation means that reading local media is easy. Writing local USB keys and floppy disks is also simple. Encourage temporary account users to write data to USB keys rather than to their local account.
- **WEEE directives** - the WEEE directive means that disposing of unused hardware is becoming more expensive. This should be taken into consideration when costing LTSP solutions against 'buy new' alternatives: LTSP often facilitates effective reuse of old equipment.

8. Authentication

- **Central Authentication** – consider LDAP for medium to large organisations as a way to consolidate user data across a range of applications and services, including LTSP.

http://wiki.ltsp.org/twiki/bin/view/Ltsp/LDAP

9. Distributing Hardware Resources

- **Local applications** – in LTSP-5.0 it's easy to run applications on the clients rather than the server. Use this feature to reduce the processor load on the server, make sound applications very easy to configure, and video applications work with better A/V synchronisation.
- **Swap** – give clients with limited RAM a swap cache on the server using NBD. This is a standard LTSP feature.

10. Advocacy

The following are the main selling points of LTSP:

- **It feels like you are on the mainframe** – applications load and run on the fastest machine in your network.
- **Low cost** – lower hardware costs, reuse of old equipment.
- **Low power** – power consumption reductions of 50% and beyond.
- **Low maintenance** – maintenance hours are lower than 10% of those for standalone systems.

- **No software costs** – free software may be used for no monetary cost.
- **Lower migration costs** – the cost of moving to an LTSP system from a Windows system is lower than moving from a Windows system to a standalone system.

http://wiki.ltsp.org/twiki/bin/view/Ltsp/SuccessStories

11. Guest Terminals

Along with normal users, some terminals on your system may need to log in automatically and provide limited functionality. In an internet café or public access suite, this might be the only type of access to be allowed. Make the following considerations:

- **KDM –** the K desktop manager is the login manager of choice providing a separate login configuration for clients based on MAC address or host name.
- **Themeing** – KDM now supports GDM themes and can be used to provide an attractive login interface.
- **Firefox/Iceweasel** – in order for each client to have it's own instance of Firefox (or Debian's Iceweasel), each client must be logged in to a different account.
- **Content filtering** – use dansguardian and squid on or off the application server to provide content filtering. There is currently no simple solution for per client or per user filtering on LTSP.

12. Calculating Costs

LTSP systems can be expected to reduce costs drastically, whilst incurring a cost of migration.

12.1 Long term costs

Areas where massive reductions in costs can be expected are:

- Staffing
- Hardware replacement
- Power consumption

12.2 Migration

Include the following in any calculation of migration costs:

- Training
- Hardware acquisition
  - servers
  - clients
- Conversions – converting old standalone desktops to thin clients
- Software re-implementation
- Data migration

http://www.redbooks.ibm.com/redbooks/SG246380/wwhelp/wwhimpl/js/html/wwhelp.htm (migration help from IBM)

12.3 Legacy Software

It may be necessary to support legacy software - that which will only run on Windows or any another operating system that is inherited.

Software packages to be given the most consideration are:

- **Accounting applications** – accountants are faced with limited support by way of accounting applications for Linux platforms.
- **Design applications** – designers put considerable time into learning a particular application and are unlikely to want to retrain to another in a hurry.

There are a number of way of providing support for legacy software:

- **Maintain a legacy stand alone system** – simply leaving one machine with the legacy operating system and application(s) on it can be a simple and effective solution for the smaller organisation.
- **Use a Windows Linux solution** – with programs like Wine, win4lin or Crossover Office, it's possible to run Windows applications from within a Linux platform. Popular applications are often targeted and made to work properly.
- **Use a Windows terminal server** - to provide access to applications. If the Windows terminal server system is run inside VMware, or another virtualisation system, then applications running inside the virtual Windows system can be load balanced against the Linux applications, making the best use of the available hardware.

Evaluate these approaches for your individual organisation using the following considerations:

- Software costs
- Hardware costs
- Set-up costs
- Maintenance costs

13. Incremental Deployment

Identify the parts of your organisation most suited to migration and target those first.

Services ordered by decreasing ease of migration (from a Windows stand alone system to LTSP) are:

- Web browsing
- wWord processing
- email clients
- Other office applications
- Graphic design applications
- Accounting systems

There may be considerable variation for your organisation. In a system where *nix is being used already, there will be no need to move to compatible applications as existing applications can be used.

Note that some web applications will only work with Internet Explorer.

Appendix A - Glossary

Distribution – A set of packages and a system for installing them. There are around 500 Linux distributions each with a different focus and use. Most of these are derivatives based around larger distribution projects such as Debian or Fedora.

Local Applications – Applications in a thin client system usually run on the server but are driven from the client. A local application is one that runs on the client itself. This is appropriate where the clients are more powerful.

Thin Client – A computer system, usually of low power consumption, diskless and boots from the network.

Single Threaded Application – An application that is run using just one process. Most applications use this method. Alternative, multi-threaded applications, may become more popular as multiple processor architectures like Intel's Core 2 Duo or AMD's X2 become more common.

Appendix B - Example Costing Scenario

EXAMPLE VSO is an organisation that has 10 office workers:

- They have a server to share files and are supported by circuit riders and volunteers.
- They use a contacts database which runs on Microsoft Access and uses Microsoft Outlook for mail.
- They would like to replace their machines every 3 years, but usually can only afford one or two replacements each year.
- They replace this system with a new LTSP server and pay for a database migration that will run on it. They convert the old desktop machines to thin clients.
- They will also buy 2 custom thin clients a year, to slowly replace their re-purposed older machines with lower power solutions.

Within three years the costs look like this:

Ongoing Costs

(More details of this costing exercise can be found here:

http://www.openitup.org/moodle/mod/assignment/view.php?id=271)

…Old System

per year

- Hardware costs
  - 2 standalone machines per year £800
- Software costs
  - OS licenses £300 (2 machine licenses plus server license updates)
  - Application licenses £200 (office software updates and anit-virus software)
- Staff
  - Volunteer time is taken up helping with installs and troubleshooting
  - 40 hours call out fixing standalone machines, replacing hard drive, rescuing data and application troubleshooting £1600
- Power
  - 10x80W desktops £400
  - 1x200W server £100

Total £3300

…New System

per year after 3 year conversion:

- Hardware costs
  - One thin client per year £200

- Software costs
  - Free software throughout
- Staff
  - Volunteers can configure new terminals
  - 10 hours server maintenance per year £400
- Power
  - 4x20W custom thin clients £40
  - 6x40W converted thin clients £120
  - 1x300W £150

Total £910

...and volunteers now have less time taken up maintaining the system with the new set up.

…and Migration Costs

- Training = 10 hours x £40 per hour = £400
- Hardware
  - new server = £500
- Database re-write = £320
- Server install costs = £800

Total £2020

Summary

The current system cost is £3300 per year to maintain. The adopted system cost is £910 per year to maintain. The ratio of about 25-30% is typical for smaller organisations, and decreases for an increase in organisation size, often ending up at 10% of the previous costs.

The migration cost is £2020, bringing the total cost for the first year to £2930, which is less than the old system cost to run in one year.