Linux Terminal Server Project:
Server Configuration Guidelines

Author
Brian Zammit
Co-founder
Abstract
This whitepaper describes how to properly plan and size a LTSP installation, making it easier to successfully deploy Linux thin client solutions in businesses, governments, and educational institutions.

www.systemsaligned.com

Table of Contents

1. Introduction

The Linux Terminal Server Project (LTSP) architecture has revolutionized the deployment of workgroup computing resources. Management need no longer worry about hardware obsolescence and can avoid investing in the traditional “supercomputers on every desk” mantra of the past. The significant unused CPU and memory resources of many of today's desktop computers can be replaced by a LTSP system that is greatly more efficient and economical.

Computing resources in the LTSP architecture are shared throughout the workgroup, greatly improving resource utilization over traditional desktop models. LTSP diskless thin client users enjoy server-class performance for a wide range of applications. At the same time, a centralized model allows administrators to focus on the server and better understand the characteristics of the applications they support. In this new computing model, resources can be applied where they are needed most to benefit application performance.

To provide a high level of interactivity, LTSP servers must be configured to satisfy the needs of the individuals and workgroups they serve. Going beyond simple ratios of clients to servers or other rules-of-thumb, this document provides an effective methodology for sizing LTSP servers with a particular focus towards meeting individualized workgroup needs.

LTSP Architecture

The principal goal of sizing LTSP servers is to develop an estimate of the hardware and network configurations required to meet the current needs of the workgroup, as well as anticipate short-term growth. Where possible, initial estimates should also anticipate longer-term expansion. Familiarity with the LTSP architecture is helpful in understanding how it influences server sizing decisions.

High-level Architecture

The LTSP architecture succeeds by combining key advantages of existing architectures with today's inexpensive hardware components and high-speed networking technology. As shown in Illustration 1.1, the architecture is comprised of three main components:

1. Diskless thin clients
2. A dedicated switch backplane

3. A Linux server running the LTSP software

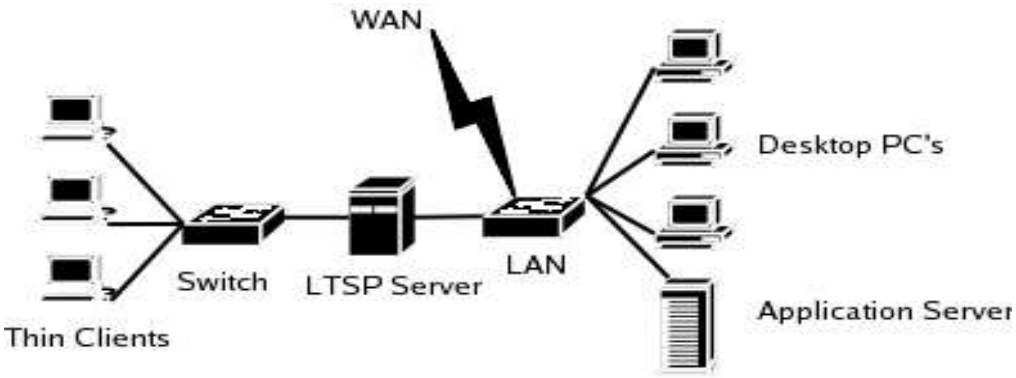

Illustration 1.1 High level LTSP architecture – separate LTSP subnetwork.

Smaller or lower network load LTSP thin client networks can also share a network with other computers, as shown in Illustration 1.2. Great care must be taken in this situation to ensure network bandwidth stays low for acceptable interactive thin client performance. Calculating network bandwidth needs is also much more difficult in this situation.

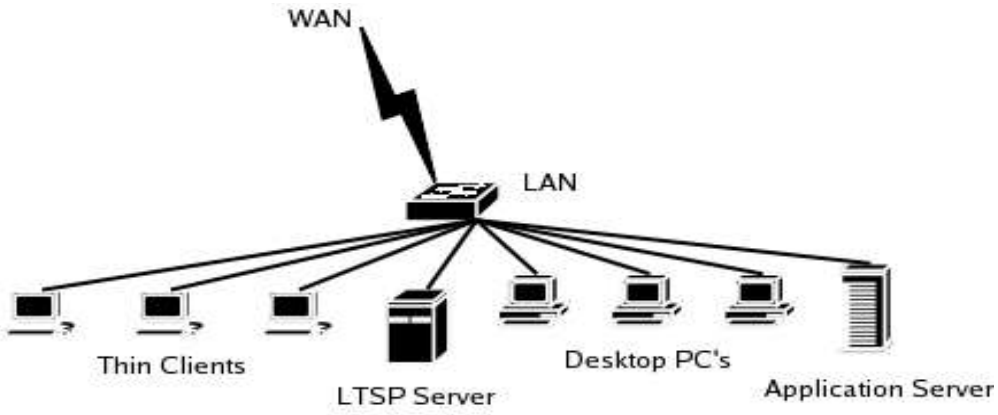

Illustration 1.2 High level LTSP architecture – shared LTSP subnetwork.

No client software is stored or executed on the thin client. All user applications run on one or more centralized server systems. The XDM server and Graphical User Interface (GUI) for each thin client user run on the LTSP server. The thin clients contain only the resources necessary for human interface input devices, such as microphones, keyboards and mice, and output devices, such as

display and audio.

All user input (i.e. keystrokes, mouse clicks, and audio) is transmitted from the thin client, through the switch backplane, and on to the appropriate client application. 2D pixels and audio output travel back to the thin client across the switch backplane. The user sees a fully-functional desktop environment and window system.

Remote X Sessions

All input and output to the LTSP thin client is through remote X connections. When a thin client is started, one of the boot processes loads an X server into the thin client's memory. When the X server on the thin client starts, it sends out a XDMCP broadcast to locate XDM servers. Once an XDM server is located, the XDM server requests that a login screen be displayed on the thin client X server, and waits for a user to log in.

When a user logs in, their login scripts and any applications they open will by default run on the XDM server. Just like the login screen, the XDM server pushes the display over the network to the X server on the thin client. This is accomplished using the native remote display support in the X Window System protocol.

Selecting an Appropriate LTSP Server

The LTSP architecture can scale to serve from very small to very large workgroups, or from single remote offices to several buildings on a campus. For any given workgroup, a range of server solutions is generally appropriate. Minimal hardware configurations should be avoided since interactivity can suffer as soon as initial demands are exceeded. Oversized configurations offer better potential performance for individual applications with corresponding lower levels of resource utilization.

Sizing Versus Tuning

Sizing and tuning are different, but related activities. Sizing estimates the initial configuration of a LTSP server to meet the needs of a known set of tasks. Tuning refers to the process of measuring and adjusting the capacities of an existing LTSP server to meet dynamic needs such as changing individual roles in the workgroup, larger numbers of thin client users, or new applications.

A conservative approach is important when sizing LTSP servers because of the limited accuracy of the assumptions that can be made about sizing data. Though workgroup resource usage and

user activity data are readily measurable, they are subject to the typical dynamic nature of the workgroup's usage patterns and other variables. Fortunately, incremental server resources (i.e. fast processors, large disks, and memory) are available at a fairly coarse granularity making precise resource usage and user activity data measurement unnecessary.

Initially, server configurations should be sized to exceed the minimum needs of the workgroup and allow for anticipated growth within the workgroup for one to two years. Additional thin client needs and potential requirements for new applications, particularly applications that are computationally-demanding, should be forecast at the same time. Where possible, initial server sizing should also anticipate future expansion, which might include significant new requirements based on supporting new buildings, new workgroups or significant new application processing demands.

LTSP servers which are initially sized to allow for expansion can be easily tuned at a later time. Changes may involve additional CPU or memory upgrades to support greater numbers of thin clients or new applications. New network interfaces may need to be configured in order to support an expanded switch backplane. If initial sizing is done correctly, total server system replacement can be avoided.

Tuning may also involve evaluating where applications run best and most economically. Often, new CPU intensive applications can be hosted on separate network application servers, which are specifically sized for their anticipated needs.

Application Resource Measurement Tools

Gaging workgroup application resource needs is key to properly configuring LTSP servers for a given set of applications and users. Server sizing calculations provided later in this document depend on this information. Linux provides tools which can be used to measure the resource usage levels associated with any particular application.

- **Performance Co-Pilot (PCP)**

  This tool, open sourced by SGI®, can be used to determine historical average CPU and network resource usage across local and remote hosts. By setting PCP to log frequent, periodic samples of CPU usage levels and network packet data to a file, later analysis on the saved data can reveal useful resource statistics.

- **pmap**

pmap can be used to measure an application's memory footprint and detailed memory mappings.

Server Monitoring Tools

Sizing is just an estimate for the initial server configuration based on educated guesses about user profiles and application usage. Once in productions, administrators must have the ability to monitor and identify critical resources on the LTSP server. Linux also provides effective tools for monitoring server resources.

- **vmstat**

  The vmstat command provides valuable information on the system's paging statistics, which can help determine when memory is constrained, resulting in excessive paging.

  Vmstat also reports CPU activity and the length of the run queue. The run queue is the number of processes which are waiting to run (and are not blocked waiting on I/O or other synchronous activities). Run queue length is a key indicator of system load. A LTSP server that shows frequent or long periods of 100% CPU utilization, along with many processes in the run queue, may need additional CPU resources to handle the load generated by the thin clients it serves.

- **Performance Co-Pilot (PCP)**

  PCP is also useful for ongoing monitoring and recording of processor loads and application resource usage. This allows for detailed examination of excessive load events and fact based analysis of system resource allocation issues.

Configuring for Resiliency

The LTSP diskless thin client is an extremely reliable device since it is fundamentally stateless and has no moving parts. A hardware failure in any thin client is easily remedied by plugging in a new thin client or having the user move to another thin client.

Because all of the LTSP diskless thin clients in a workgroup depend on a single server, availability for the workgroup as a whole can be affected by problems on the LTSP server. Optional high availability X session failover, IP load balancing among a cluster of LTSP servers, and a number of other steps, can be taken to harden LTSP servers against single points of failure.

- **RAS Capabilities**

  Reliability, Availability, and Serviceability (RAS) is a highly desirable set of properties in a LTSP server. Where ever possible, servers should be configured with RAS capabilities to enable them to seamlessly recover from single-component failures. For example, providing disk mirroring or other forms of Redundant Arrays of Independent Disks (RAID) can protect LTSP servers against downtime and data loss in case of an individual disk failure. Techniques like RAID may have some small impact on disk I/O performance, but this will generally not affect latency and will not be apparent to thin client users.

- **Multiple Servers**

  Multiple servers can be configured to divide users into separate workgroups of LTSP diskless thin clients. Though this approach decreases the dependency on a single server, it adds the administration of a second server. Users are served their X session by their designated server, and will not have access to applications of the other workgroups. Modern queuing theory holds that a single server is more effective at serving the random needs of a workgroup than multiple servers. Multiple servers may be effective where different workgroups are not expected or allowed to share thin clients.

Intel x86 32-bit and 64-bit Servers and Linux

Application performance in a LTSP system environment is directly dependent on the servers that provide computational resources. Intel-based servers lead the industry in offering some of the most powerful, reliable, and cost effective systems available today. Intel-based servers provide scalable, symmetric multiprocessing capabilities. One to eight high-performance Intel Xeon DP/MP 32-bit or Itanium2 64-bit processors can be configured, with up to 32GB or 64GB respectively of physical memory, and terabytes of disk storage. These features, plus available load balancing, clustering, and fail-over configurations provide the necessary performance for

peak demands, as well as virtually unlimited growth.

The power of Intel-based servers is further enhanced by Linux, the premiere environment for enterprise network computing. Optimized for enterprise needs, Linux features full 64-bit processing, mainframe-class reliability, superior scalability, and incredible performance at an unprecedented low cost. These features and others greatly enhance multi-user environments, making Linux clearly suited to hosting mission critical thin client systems.

2. Sizing Guidelines and Methodology

It is difficult to provide meaningful guidelines for sizing LTSP servers without an understanding of the applications and usage patterns of the thin clients they serve. Knowing what users are doing, their activity levels and relative time spent in particular applications, is key to configuring adequate server resources.

This section describes general server sizing guidelines to help administrators gain a quick perspective on server requirements, but also provides detailed server sizing calculations based on real user and application requirements. Information is also provided to assist in sizing servers that provide additional application services (e.g. HTTP, file services, multimedia etc.).

The principal goal of correct LTSP server sizing is to provide robust interactive performance for thin client users under average-use conditions. Sufficient processing capabilities (CPU), memory, I/O, and switch backplane bandwidth must be provided to avoid contention between thin client users. The sizing methodology and guidelines help to ensure an adequate amount of resources are configured for peak load conditions.

> **Note:** The simple sizing examples presented here, are rough estimates. Whenever possible, applications and users must be carefully profiled and measured as part of a more detailed sizing exercise (see the section on sizing calculations).

To help illustrate the basic sizing guidelines, a workgroup with typical office productivity and personal information management (PIM) applications will be referenced:

- 50 LTSP diskless thin clients
- 25 thin clients active at any one time (50% activity level)

➔ Number of Processors

In order to provide good application performance and a high level of interactivity, most workgroup LTSP servers should be equipped with at least two processors. Multiple processors enable the LTSP server to provide consistently quick responses, even in the presence of high-priority system processes and threads. High priority tasks can cause somewhat delayed response in uniprocessor servers under peak loading conditions. However, single processor servers can be used for small workgroups,

especially if the users have low average activity levels or use applications that require minimal CPU resources (e.g. text editing, terminal emulation, etc.). For these users, up to 30 LTSP diskless thin clients connected to a powerful (2 to 3 GHz Xeon) uniprocessor server is not unreasonable. Configuring a uniprocessor server requires careful attention to peak loading to ensure good response times can be maintained when all users are active.

Consider the example workgroup environment. Conservatively, office productivity and PIM applications consume 1.5% to 2% of a 3 GHz Intel Xeon processor (computationally-intensive applications like CAD, simulations, and video decompression applications must be profiled individually.) Since a user typically has one or two applications active at any one time, a conservative estimate is a 2% load on a 3 GHz processor. Multiplying 2% by the number of simultaneously active users yields a 50% load on a 3 GHz Xeon CPU. Adding the roughly 1.5% of a 3 GHz CPU required to run Linux yields a load of 51.5% on a 3 GHz CPU. Thus, one 3 GHz Xeon processor should be sufficient to serve the average needs of 25 simultaneous users running PIM and general office productivity applications. Keep in mind, peak loads exceeding this average may cause noticeable slowdowns and delays in the interactivity of a system configured with one CPU. Also, any high priority processes running on the system will reduce interactivity for all users. Two slower processors will often provide more interactive performance for Linux thin clients than one fast CPU. Hyper-threaded CPU's can provide more interactivity and some insulation from high priority processes than a regular single core chip, but their performance is not equal to two physical processors.

➔ Sizing System Memory

Memory is the most important resource in an LTSP server. A server that has run out of CPU resources will typically degrade gracefully, whereas a system that is thrashing because of memory starvation can significantly affect the server's interactivity.

LTSP server memory configurations are based on the number of simultaneous active users. Excluding applications with large memory requirements (e.g. imaging, CAD, etc.), 40 MB of system memory should be provided for each active thin client user. This guideline provides memory resources roughly equivalent to those of a single user working on a 64 MB workstation with one or two applications at a time. Enough memory is provided to

keep one or two active applications in main memory, though the user's inactive applications may be swapped out if necessary.

Applications with larger memory requirements or work patterns with frequent shifting of focus between applications, may require significantly more memory resources. If users are accustomed to workstation environments that require 128 MB or more memory to achieve desired performance levels, at least 100 MB of memory per user should be configured on the LTSP server.

The per-user memory requirement must be added to the minimum system configuration of 64 MB for the operating system kernel and shared libraries. The example workgroup would require a LTSP server with 1,064 MB of memory (40 MB multiplied by 25 active users, plus 64 MB for the operating system).

➔ **Swap Sizing and Disk Spindles**

Swap space is generally used when memory requirements exceed the real amount of memory in a server. Physical memory should be sized so swap use is absolutely minimized. A system may thrash when using excessive swap space, and performance will likely diminish very rapidly. Running out of swap space in a thrashing situation may result in a complete system halt.

Applications belonging to inactive users are paged out as memory is required to support other active applications. Applications that become active are paged back into system memory. Given the importance of swap space and the low cost and high density of today's disk storage, conservative sizing of swap resources for non-restricted desktop environments is strongly recommended.

Sufficient I/O bandwidth to the disk subsystem is required to ensure that applications can be paged in (and out) quickly for a high level of user interactivity. In general, swap space should be spread across several spindles to avoid bottlenecks resulting from a single disk's I/O constraints. Running swap across one disk spindle for each processor configured in the server is an acceptable minimum.

Virtual memory should be sized at least as large as physical memory, and ideally two times physical memory. For example, with about 1,064 MB of physical memory required for the system and applications, about 2.2 GB of swap space is required. Also, since the example workgroup requires a server configured with two processors (although Hyperthreading is not equal in

processing to two physical processors, we consider each Hyperthreaded CPU separate for swap calculation), the 2.2 GB of swap space should be spread across at least two disk spindles.

➔ **Network and switch backplane Interfaces**

A LTSP server requires at least two network interfaces: one to connect to the LAN, the other for connection to the switch backplane. A dedicated switch backplane of at least 100 Mbps is strongly recommend because it guarantees a high level of quality of service (QoS) for the LTSP diskless thin clients. Shared, routed, or general purpose networks are not recommended for the switch backplane since network activity such as large file transfers or frequent broadcasts can cause X session packets to be dropped. Dropped packets or other interference can severely degrade the thin client user's experience.

The LAN interface must be adequately sized to support the combined network traffic of all the running applications. Depending on the number and nature of the applications they host, LTSP servers may require a LAN connection greater than 100 Mbps. For example, many LTSP diskless thin client installations are used to access X sessions on other servers, which can drive up LAN bandwidth requirements. In addition, other requirements, such as connecting the LTSP server to separate subnets, can require additional LAN interfaces.

The server's switch backplane interface must be sized to support the bandwidth to all of the active thin clients for good response and interactivity. Other than video, games, and other applications which cause large, frequent screen updates, most typical applications use less than 1 Mbps of bandwidth. A 1 Mbps switch backplane interface for each of the 25 active users results in a total of 25 Mbps for the example workgroup.

Assuming a conservative 25% protocol overhead, a 100 Mbps network interface card should provide 75 Mbps of throughput. Therefore a 100 Mbps adapter for the example workgroup would suffice. For future expansion, a 1 Gbps interface could be configured on the server.

In summary, the 50 thin client workgroup with moderate application needs and a 50% activity level, could be served by a server configured as follows:

- Two 2-3 GHz Xeon processors
- Two 100 Mbps Fast Ethernet interfaces (one being 1Gigabit is recommended)

- 1,064 MB of RAM (system) plus 64MB of RAM per user
- Minimum of two hard disk drives
- 2.2 GB of swap space configured across both drives

Server selection should ultimately be based on anticipated workgroup growth and expansion.

Estimating Real-world Workloads

Workgroups in the real world are seldom as homogeneous as shown in the example above. What is required is a more robust approach to server sizing; one that involves accurately measuring application demands and projecting impact based on the expected usage patterns of different groups of users.

This section details some of the calculations used to explain the recommended methodology for sizing LTSP servers. Data for these sample calculations is taken from the Educational (kindergarten through 12th grade, or K-12) deployment scenario described in chapter three.

Profiling Users

In most deployment scenarios, different groups of users typically have distinct profiles based on the applications they use, as well the usage patterns they exhibit. In the K-12 deployment scenario, all of the users are students, but their application usage and activity level expectations vary depending on the location of the LTSP diskless thin client. For instance, thin clients in a computer lab are expected to be 100% busy while a laboratory is in session. Thin clients in classrooms or the library may have more occasional use. User data for the K-12 deployment scenario is shown in Table 2.1.

	Users	**% Active**
Classroom	40%	30%
Computer Lab	50%	100%
Library	20%	25%

Table 2.1 Example computing groups in a K-12 usage scenario.

In addition to different activity levels, the students will have slightly differing application profiles depending on their location. Table 2.2 shows how much time users are expected to spend in various applications.

	X Windows	OpenOffice.org	Mozilla	Resource Intensive
Classroom	20%	30%	50%	0%
Computer Lab	20%	45%	30%	5%
Library	20%	15%	55%	10%

Table 2.2 Example user profiles in a K-12 usage scenario.

X Windows refers to the basic X11 environment used by all students as part of the Linux desktop environment. OpenOffice.org is the combination of the open source OpenOffice.org's Writer (word processor), Calc (spreadsheet), and Impress (presentation). Mozilla is the open source web browser. Resource intensive refers to a software applet, plugin, or program that is usually only used in the computer lab or library (e.g. a Java front-end applet used to search a library database., or multimedia programs for watching video or listening to music).

Profiling Applications

Once a set of user profiles and application usage patterns has been established, individual applications are profiled to understand the demands that they place on system resources. Resource considerations for applications running on a LTSP server are expressed in terms of memory, processor, and switch backplane bandwidth.

➔ **Memory**

It is important to understand the memory demands (in megabytes) placed on the LTSP server by each fully-active instance of the application. Linux features an efficient shared memory system which enables multiple instances of applications and libraries to share certain memory segments (i.e. code segments). As a result, application memory usage is divided into private and shared memory. An application's shared memory is allocated only once and shared between all instances of the application whereas private memory is allocated for each additional instance of the application.

➔ **Processor**

Processor resources are typically expressed as the percentage of a given processor needed to run a fully-active instance of the application. These examples express processor resources as a percentage of a 3 GHz Xeon processor.

➔ **Switch backplane Bandwidth**

To provide interactive performance on par with traditional desktops, it is essential to supply an adequate network interface on the LTSP server to connect to the dedicated switch backplane. To calculate switch backplane bandwidth needs, the application's impact on the switch backplane is calculated in terms of megabits per second. Table 2.3 displays an example of application profiles used in calculating usage and load in a K-12 usage scenario.

	X Windows	OpenOffice.org	Mozilla	Resource Intensive
Memory (private)	20	8	9	6
Memory (shared)	10	7	11	3
3 GHz CPU	0.3438%	0.5156%	0.6875%	0.3438%
Switch Bandwidth	0.2	0.2	0.8	0.1

Table 2.3 Example application profiles in a K-12 usage scenario

Example Calculations

Please note that these calculations are provided as an example only. They illustrate, but may not exactly reproduce, the calculations required for real world situation.

Calculating the Average User Resource Usage

To estimate the average impact of users on the server, the resource demands of each application must be calculated as a function of usage. Table 2.3 shows that each classroom user will require an average of 4.88 MB of private memory, 0.11% of a 3 GHz CPU, and 0.088 Mbps of switch backplane bandwidth.

	X Windows	OpenOffice.org	Mozilla	Resource Usage	Total
Active	30%	30%	30%	NA	**-**
Usage	20%	30%	50%	NA	**100%**
Memory (private MB)	1.2	0.72	1.35	NA	**3.27**
3 GHz CPU	0.0206%	0.0468%	0.1031%	0%	**0.1705%**
Switch Bandwidth (Mbps)	0.012	0.018	0.120	NA	**0.150**

Table 2.4 Example classroom user's resource usage on an LTSP server

The average results are obtained by multiplying the values for each application's resource category (Table 2.3) by both the percentage that the user is active (30% for the classroom users), and the

percent usage for that particular application (Table 2.2). To calculate maximum user impact, the figures in the Active row would be changed to 100% (see Table 2.5).

Note that shared memory is not represented in any of these calculations. It is accounted for as a lump sum shared across all users on a server, with some margin of peak loads added. This will be explained in more detail later.

	X Windows	OpenOffice.org	Mozilla	Resource Usage	Total
Active	100%	100%	100%	100%	**-**
Usage	20%	45%	30%	5%	**100%**
Memory (private MB)	4	3.6	2.7	0.3	**10.9**
3 GHz CPU	0.0688%	0.2320%	0.2063%	0.0172%	**0.5243%**
Switch Bandwidth (Mbps)	0.040	0.090	0.240	0.005%	**0.375**

Table 2.5 Example computer lab user's resource usage on an LTSP server

	X Windows	OpenOffice.org	Mozilla	Resource Usage	Total
Active	25%	25%	25%	25%	**-**
Usage	20%	15%	55%	10%	**100%**
Memory (private MB)	1	0.3	1.24	0.15	**2.69**
3 GHz CPU	0.0172%	0.0193%	0.0945%	0.0086%	**0.1396%**
Switch Bandwidth (Mbps)	0.010	0.008	0.110	0.003%	**0.130**

Table 2.6 Example resource use of a library user on an LTSP server

Calculating Average Resource Consumption for Each User Type

Once the average consumption of resources for each type of user has been calculated, the overall resources needed for each user group can be obtained by simply multiplying by the number of expected users (Table 2.7).

	Users	Memory (private MB)	3 GHz CPU	Switch Bandwidth
Classroom	40	128.00	6.8%	6.00
Computer Lab	50	545.00	26.2%	18.75
Library	20	53.80	2.8%	2.60
Totals	**110**	**726.80**	**35.8%**	**27.35**

Table 2.7 Example total LTSP server resources required for a school

According to Table 2.7, a minimal LTSP server for the example workgroup would require at least:

- 822 MB of RAM (i.e. 726.8 MB, plus 31 MB of shared memory, plus 64 MB for the kernel)
- two processors
- a single 100BaseT Ethernet switch backplane, in addition to its LAN interface.

Keep in mind this is an absolute minimum, ideal, situation. Actual usage patterns might necessitate the provisioning of at least two processors, and double the amount of memory to account for memory leaks and advanced users.

Calculating SWAP Space Requirements

As mentioned previously, an adequate amount of virtual memory, which allows inactive applications to be paged out and quickly retrieved when they become active again, is essential to the operation of an LTSP server.. Unlike physical memory, which is sized for active applications, virtual memory, and therefore SWAP space, must be sized to hold all applications, both active and inactive.

	Users	Memory (private MB) per user	Memory (private MB) Total
Classroom	40	37	1,480
Computer Lab	50	43	2,150
Library	20	43	8,60
Totals	**110**	**123**	**4,490**

Table 2.8 Example swap space required for an LTSP server in the K-12 scenario

31 MB of shared memory and 64 MB reserved for the kernel is added to 4,490 MB of private memory, yielding a minimum virtual memory requirement of 4,582 MB. This amount reflects a bare minimum value for two principal reasons:

➔ **Application Memory Bloat**

The measured memory allocation for the sample applications may not account for all of the memory that can potentially be allocated by the application. This value reflects a snapshot of the application's memory needs, which may change over time. Applications like web browsers can start off small and grow to use significant amounts of virtual memory as plug-ins and other ancillary objects are loaded. In reality, an application may have allocated significantly more storage than is currently represented in system memory. Also, unused pages may have been reclaimed by the virtual memory system.

➔ **Temporary Storage Space**

Since most user applications run on the LTSP server, sufficient temporary space must be provided to accommodate the needs of active applications.

Fortunately, configuring additional swap space in today's disk market is extremely cost-effective. It is always worth erring on the side of caution; in this case configuring too much swap space. Although applications vary, configuring 50% to 100% more virtual memory above the minimum required is strongly recommended.

In the workgroup example (i.e. 25 active users), multiplying the minimal virtual memory figure (2,532 MB) by 1.5 yields 3,798 MB. Assuming 20 MB of temporary space per user, and 44 active users, an additional 880 MB is necessary, for a total of 4,678 MB. Subtracting the 1,166 MB of physical RAM results in a recommended value of 3,512 MB for swap space. This amount of swap space should be distributed across at least two disk spindles.

Accounting for Other Services

LTSP servers need not be dedicated to serving only thin client users' desktop applications. Other applications and network services can be run on LTSP servers, including web, file, and database services, as long as the server is configured to meet the demands placed upon it. Sizing for additional applications or services is simply a matter of understanding resource loads for those applications.

Web Services

Recent SPECweb results of a two Intel Xeon 3 GHz server running Red Hat Content Accelerator 2.2 revealed that the server could perform 4,992 operations per second. In contrast, a typical active browser user is capable of only generating one or two operations per second. Within the context of the user profiles in the K-12 deployment scenario, the aforementioned server would accommodate 15 active browser users at any one time under average load conditions, and 38 users under peak loading conditions: a mear 0.4% to 1% of a 3 GHz CPU processor's capacity.

The memory requirement of the particular web server would also need to be added into the calculation for memory requirements.

File Services

The SPECsfs benchmark can be used to estimate the impact of file services on a LTSP server. Recent benchmark results show 18,146 file system operations per second for a single 3 GHz processor server using 10,000 RPM SCSI drives and a Fibre-channel interface.

Most applications use some level of file system service. Even browsers typically use file services to cache web page data and to store cookies. For conservative sizing, it is reasonable to assume that most users generate one to two file system operations per second for general application use. This results in a file service CPU load of .011% to .018% for each file services user. The total number of file service users relying on the server would need to be accounted for in any calculations.

Video Clients

Decoding of video files or streams is a computationally-intensive process, therefore sizing for multimedia applications must be considered in addition to standard desktop resources.

CPU

The CPU resources needed to decode and display video vary widely with the type and quality of the video files being decoded. For instance, MPEG files can be encoded at widely differing rates. Decoding a 1.5 Mbps MPEG-1 file needs very different resources than those required for an 8 Mbps MPEG-2 file. To decode and display a 1.5 Mbps MPEG-1 file at 30 frames per second typically requires one twelfth of a 3 Ghz Xeon processor's resources. One sixth of a 3 Ghz processor's resources are required to decode and display a 6 Mbps MPEG-2 file at 24 frames per second.

For a real-world example, assume that a maximum of 12 thin clients in a given workgroup are expected to be decoding and displaying 1.5 Mbps MPEG-1 video files simultaneously. Since each MPEG-1 stream is expected to require one twelfth of one CPU, one additional 3 Ghz Xeon processor should be added to the basic configuration needed to support basic thin client functionality and applications. If instead the twelve thin client users were viewing 6 Mbps MPEG-2 video files, as many as three additional 3 Ghz processors would be required. These numbers are offset by the degree to which the server is under or over-configured for the other workgroup services it provides.

Switch Backplane Requirements for Multimedia

The X protocol works by transmitting only the pixels that change over the switch backplane to the thin client. Significant switch backplane resources can be demanded because displaying video results in a large number of constantly changing pixels.

Displaying a 320 x 240 MPEG-1 video window typically generates 10 to 12 Mbps of network traffic. It is easy to see how fewer than 10 simultaneous video users could easily saturate a server's 100 Mbps connection, causing performance problems for other users. A Gigabit Ethernet interface on the server would be required to serve this number of video users and leave room for future expandability. Gigabit Ethernet or multiple Fast Ethernet interfaces would also be required to serve thin client users who wish to decode and display MPEG-2 video. A single 640 x 240 MPEG-2 window generates approximately 45 Mbps of network traffic.

3. Example LTSP Server Configurations

A Systems Aligned LTSP professional can provide more detailed LTSP thin client deployment scenarios, but a starter guide is listed below.

Educational (K-12) Environment

The K-12 educational deployment scenario illustrates typical user and application profiles for computing laboratory, library, and classroom deployments in a K-12 school environment. All thin clients are to be connected to a single LTSP server. Native applications will provide all functionality, including Mozilla Firefox for web browsing, native Linux desktop utilities (e.g. file, e-mail, text editing, etc.), and OpenOffice.org for word processing and other office suite applications.

One useful approach to sizing is to select a server platform based on the peak-load values, and then to configure the system with some extra capacity above the average-use level. This approach provides a server sized to meet current needs, but with the capacity for future expansion.

Library Automation

The library automation deployment scenario involves user and application profiles for a typical library. In the deployment scenario, library patrons use Mozilla Firefox for general Internet access and also for accessing most library automation systems. Telnet sessions are used to access older library automation services. Two user profiles are defined: one for librarians, and one for patrons.

Librarian profiles include limited use of a Citrix ICA® client to access Microsoft Office applications on a separate server running Windows 2000 Terminal Server and Citrix Metaframe™ server software. Linux desktop utilities are used for direct file management, editing, e-mail, etc. Both types of users primarily use the Mozilla Firefox browser for accessing the web interfaces to most library automation systems (such as SIRSI, and Endeavor). These applications provide catalog research and other library services through a combination of HTML, Java technology, and JavaScript software. General purpose Internet browsing for research and entertainment is also expected from both types of users.

General Office Automation

The general office deployment scenario includes typical user and application profiles for an enterprise office environment with heavy dependence on a company Intranet for accessing enterprise applications and services. Three user profiles are defined: one for typical users, one for administrators, and one for power users.

All three user profiles include a mix of applications with primary use of Mozilla Firefox and native Linux desktop applications for e-mail, calendar, name directories, etc. Native office suite applications are also part of the user profiles; sizing is based on the server resources required for OpenOffice.org.

Call Center

The call center deployment scenario includes typical user and application profiles for a Java technology and browser-based customer management system environment. Only one type of user is defined for sizing purposes. The most unique characteristic of the call center user profile is that users are expected to be active all of the time.