

Тракийски университет, гр. Ст. Загора
Катедра по информатика и математика



ас. Живко Желев

Изграждане на thin-клиент архитектура чрез LTSP
(Linux Terminal Server Project)



5th annual conference for FOSS and free knowledge share,
Sofia, Bulgaria
27 – 28th October, 2007

THIN-КЛИЕНТ ЧРЕЗ LTSP

Живко Ив. Желев

РЕЗЮМЕ. Настоящият доклад има за цел да запознае с възможностите на някои клъстерни архитектури и тяхното приложение в образователните системи на висшето и средното училище.

Конкретно се дава представяне на thin-клиент архитектурата, базирана върху проекта LTSP [5] и осъществена на платформа OS Linux SUSE 10.1. Изложението няма за цел да представи пълните възможности на проект като LTSP, а само да даде представа за общите му възможности, а така също и за някои конкретни технически параметри във връзка с конкретното инсталиране на сървъра.

1. Въведение

Thin клиент се дефинира по различен начин, но основното във всяка дефиниция е, че той е мрежов компютър, който няма хард диск (или ако има такъв, то той не се използва) и е проектиран да използва ресурсите на сървърите в мрежата. Това може да е персонален компютър, който се проявява като thin клиент, или специализирано устройство (Wyse, Neoware, HP). И в двата случая се използва минимална операционна система с терминален клиент.

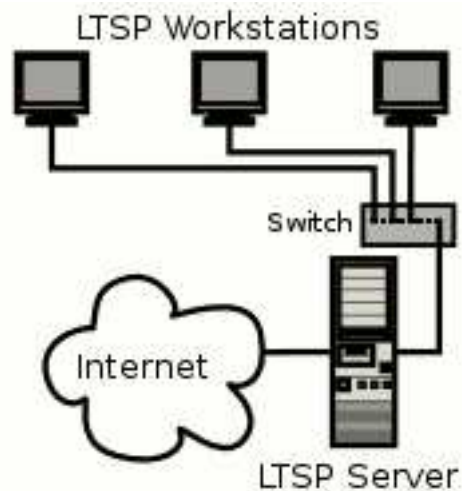
Проектът LTSP се отличава с поддръжката на thin-клиенти към Линукс сървъри. Това е един прост и гъвкав начин стари компютри-работни станции да бъдат приспособени като графични или конзолни терминали върху GNU/Linux сървър. Това представлява едно изключително ефективно решение за сферите на образованието, бизнеса и т. н. Немаловажен е и фактът, че все повече Linux дистрибуции осигуряват пълна поддръжка за LTSP.

Така например всяка образователна институция разполага с компютри, които са работещи, но нямат достатъчни за ефективна работа хардуерни параметри. На такива компютри често трябва да се преподават дисциплини като комуникационни и информационни системи, информационни технологии, информатика и други, изискващи използване на мрежови ресурси, интернет, работа с текстови и графични процесори и среди за програмиране, за които са необходими по-голяма работна памет и бързина. [1]

За щастие съществува LTSP и с негова помощ всеки би могъл да използва слабо PC, да отстрани хард диска му, флопито и CD-ROM-а и всичко, което трябва да направи е да сложи една буутваща мрежова карта, а много такива карти притежават буутром сокети.

По време на самото зареждане работната станция получава необходимата IP информация и ядро от сървъра и след това монтира файловата система на root потребителя през NFS (Network File System). Схемата на работа е показана на фигурата по-долу [5]:

Материалът е подготвен на L^AT_EX 2_ε (MiK_TE_X), версия 2.2.6 и е част от докладите, представени на Open Fest, 2007г. в гр. София.



Фиг. 1. Схема на работа на клиент-сървър при LTSP

Самата работна станция може да бъде конфигурирана по 3 основни начина [6]:

- **Графична X среда**

Използвайки X средата, работните станции получават достъп до различни приложения върху сървъра или върху други сървъри в мрежата.

- **Telnet сесии**

Работната станция има възможност също така да извиква множество телнет сесии от сървъра. Всяка Telnet сесия е всъщност отделен виртуален екран. С клавишните комбинации Alt+F1 до Alt+F9 става и превключването между отделните сесии.

- **Конзолни сесии(Shell Prompt)**

Работната станция може да бъде конфигурирана така, че да ви хвърля в bash shell-а на конзолата. Това е изключително полезно, когато трябва да се справим с някакви проблеми с X средата или NFS.

Но в крайна сметка най-важното е, че много работни станции могат да работят, обслужвани от един GNU/Linux сървър. Възниква и въпросът колко точно работни станции могат да обслужвани, но това зависи от техническите параметри на сървъра и от самите приложения, които се използват. Все пак може да се каже, че не е необичайно да имаме около 50 работни станции, отварящи Mozilla и OpenOffice върху сървър Dual P4-2.4Ghz с 4 GB RAM, но всъщност добре е да се знае, че средното натоварване рядко надхвърля 1GB.

2. НАЧИН НА ДЕЙСТВИЕ

Самият boot процес на работна станция преминава през няколко стъпки. Разбирането на тези стъпки би улеснило значително решаването на проблеми, ако те възникнат.

Четири основни сървърни услуги участват в зареждането на ядро върху LTSP работна станция, а именно:

- DHCP
- TFTP
- NFS
- XDMCP

Както вече казахме LTSP е изключително гъвкав. Всяка от горните сървърни услуги може да бъде осигурена от същия сървър или пък от няколко други сървъра. По-надолу ще опишем работата на един сървър, който поддържа всички сървърни услуги.

2.1. Стъпки на работа на станциите. Стъпките, през които преминава една работна станция, за да зареди X Window са следните [6]:

1. Зареждане на линукс ядро в паметта на работната станция. Това може да стане по няколко различни начина:
 - a) Bootrom (Etherboot, PXE, MBI, Netboot)
 - b) Floppy
 - c) Hard disk
 - d) CD-ROM
 - e) USB Memory DeviceВсеки от буутващите методи по-горе ще бъде обяснен по-подробно.
2. След като се зареди в паметта, ядрото започва да се изпълнява.
3. Ядрото инициализира цялата система и всички периферни устройства, които разпознае.
4. Тук започва и интересната част. По време на самото зареждане на ядрото, един ramdisk имидж също се зарежда в паметта. Ред в скрипта на ядрото `root=/dev/ram0` казва на ядрото да монтира този имидж като главна(`root`) директория.
5. Нормално, след като бъде заредено, ядрото стартира `init` програмата. Но в този случай ядрото се инструктира да зареди един малък шел скрипт. Това става чрез въвеждането на реда `init=/linuxrc` в скрипта на ядрото.
6. Скриптът `/linuxrc` започва със сканирането на PCI устройствата, търсейки мрежова карта. За всяко PCI устройство той преглежда файла `/etc/niclist`, за да открие съвпадение. След като такова се открие, се връща името на драйвера на NIC модула и се зарежда самият модул на ядрото. За ISA карти, името на драйвер модула трябва да бъде изрично посочено в скрипта на ядрото заедно с IRQ и всички други адресни параметри, които ще са необходими.
7. След това малък DHCP клиент се стартира, за да направи допълнителна заявка към DHCP сървъра. Тази допълнителна заявка е необходима, тъй като се нуждаем от повече информация от тази, която получава bootrom-а при първото DHCP запитване.
8. Когато DHCP клиентът получи отговор от сървъра, той стартира файла `/etc/dhclient`, който извлича необходимата информация, за да бъде конфигуриран `eth0` интерфейсът.
9. До този момент файловата система на `root` беше `ramdisk`-а. Сега скриптът `/linuxrc` монтира нова файлова система през NFS. Обичайната директория, която се експортира в този случай от сървъра е `/opt/ltsp/i386`. Но тази файлова система не може да се монтира просто като `/`. Първо

тя се монтира като `/mnt`. След това създава **pivot_root**. `Pivot_root`-ът заменя текущата файлова система с новата. Когато този процес приключи, NFS файловата система се монтира върху `/`, а старата файлова система се монтира върху `/oldroot`.

10. Когато пренасянето и монтирането на новата файлова система приключи, скриптът `/linuxrc` приключва своята работа и ние трябва да извикаме истинската `/sbin/init` програма.
11. `Init`-ът ще прочете файла `/etc/inittab` и ще започне настройването на средата на работната станция.
12. Едно от първите неща, които ще бъде стартирано е скриптът **rc.sysinit**.
13. Скриптът **rc.sysinit** ще създаде 1MB рамдиск, който ще съдържа всичко онова, което трябва да бъде записано или модифицирано по някакъв начин.
14. Този рамдиск се монтира като `/tmp` директория. Всички файлове, които трябва да бъдат записани ще съществуват всъщност в `/tmp` директорията и освен това има и символични връзки към тези файлове.
15. Монтира се файловата система `/proc`.
16. Файлът `lts.conf` се прехвърля и всички параметри в този файл, които се отнасят до работната станция се превръщат в променливи на средата, за да бъдат използвани от скрипта **rc.sysinit**.
17. Ако работната станция е конфигурирана да прехвърля през NFS, директорията `/var/opt/ltsp/swapfiles` ще бъде автоматично монтирана като `/tmp/swapfiles`. По такъв начин, ако конкретната станция все още няма суопфайл, такъв бива създаден автоматично. Размерът на този файл се конфигурира във файла `lts.conf`.
След това този суопфайл се стартира чрез командата **swapon**.
18. Конфигурира се мрежовият интерфейс **loopback**, който представлява мрежовия интерфейс с IP адрес 127.0.0.1.
19. Ако са стартирани локалните приложения (Local Apps), тогава ще бъде монтирана и директорията `home`, така че тези приложения да станат достъпни за домашните директории на потребителите.
20. Няколко директории също се създават във файловата система на `/tmp`, за да се съхраняват някои файлове, необходими за работата на системата. Това са директории като:
 - a) `/tmp/compiled`
 - b) `/tmp/var`
 - c) `/tmp/var/run`
 - d) `/tmp/var/log`
 - e) `/tmp/var/lock`
 - f) `/tmp/var/lock/subsys`
21. Създава се файлът `/tmp/syslog.conf`. Този файл съдържа информация, указваща на демона **syslogd**, към кой хост в мрежата да изпраща логин информация.
22. Стартира се демонът **syslogd**, който използва конфигурационния файл, създаден по-горе.
23. Веднъж като е приключил скриптът **rc.sysinit**, контролът се прехвърля обратно към програмата `/sbin/init`, която променя стартовия

работен режим от **sysinit** в **5**. Това води и до изпълняването на зададените параметри в `/etc/inittab`.

24. По принцип в `/etc/inittab` има параметри, които стартират скрипта `/etc/screen_session` на сериините портове `tty1`, `tty2` и `tty3`. Това означава, че може да се стартират 3 сесии едновременно като типът на сесията се контролира от параметрите на **SCREEN_01**, **SCREEN_02** и **SCREEN_03** в `lts.conf`.
25. Ако **SCREEN_01** е дефиниран със стойност **startx**, то скриптът `/etc/screen.d/startx` ще се изпълни, а това ще стартира X Windows системата, което ще даде и графичен интерфейс. Във файла `lts.conf` има параметър, наречен **XSERVER**. Ако този параметър липсва или е сложен в режим **"auto"**, то тогава ще се направи опит за автоматично засичане на видео картата. Ако картата е PCI или AGP, то тогава ще се получи и самия PCI производител, id-то на картата и след това ще се направи обръщане към файла `/etc/vidlist`. Ако картата е поддържана от Xorg 6.7, `pci_scan` ще върне името на модулният драйвер. Ако се поддържа само от XFree86 3.3.6, `pci_scan` ще даде името на X сървъра, който да се използва.
26. Ако се използва Xorg, то тогава ще бъде извикан скриптът `/etc/build_x4_cfg`, който ще изгради файла `XF86Config`. Ако се използва XFree86 3.3.6, то тогава ще бъде извикан скриптът `/etc/build_x3_cfg`, който ще изгради файла `XF86Config`. Тези файлове се намират в директорията `/tmp`, която от своя страна, ако си спомним, е рамдиск, който се вижда само от работната станция. Файлът `XF86Config` се създава въз основа на параметрите, заложиени във файла `/etc/lts/conf`.
27. Веднъж създаден файлът `XF86Config`, скриптът **startx** стартира X сървъра с този нов конфигуриран файл.
28. X сървърът ще изпрати **XDMCP** заявка към LTSP сървъра, която ще открие `login` прозорец.
29. В този момент потребителят може да влезе в системата. Това обръква много потребители. Те са седнали пред работната станция, но са стартирали сесия на сървъра. Всички команди се изпълняват на сървъра, но се показват върху дисплея на работната станция.

2.2. Зареждане на ядрото в паметта. Зареждането на Линукс ядрото в паметта на работната станция може да стане по два основни начина [6, 8]:

- Boot ROM
- Локални устройства

2.2.1. Boot ROM.

- **Etherboot**

Etherboot е много популярен bootrom проект с отворен код. Той съдържа драйвери за много от най-често използваните мрежови карти и работи изключително добре с LTSP.

Линукс ядрата трябва да бъдат отбелязани като **mknbi-linux**, което ще подготви ядрото за мрежово зареждане чрез префиксиране

на ядрото с допълнителен код и добавяне на `initrd` в края на скрипта на ядрото.

Ядрата, предлагани от LTSP са вече маркирани и готови да бъдат заредени чрез Etherboot. Etherboot може да бъде записан също така на флопи, което върши идеална работа за тестване.

- **PXE**

PXE е друга bootrom технология, чиито означения означават *Pre-boot Execution Environment*.

Един PXE bootrom може да зареди файл не по-голям от 32KB. Но ядрото е много по-голямо от това. За тази цел PXE се настройва да зареди втори boot-loader, наречен **pxelinux**. Този boot-loader е достатъчно малък, за да бъде първоначално зареден, а той от своя страна знае как да зарежда по-големи файлове, какъвто е например ядрото на Линукс.

- **MBA**

MBA (Manage Boot Agent) е bootrom от компанията *emBoot*. Emboot беше мрежов отдел на 3Com. MBA съдържа всъщност четири bootrom-а в себе си и поддържа PXE, TCP/IP, RPL и Netware.

MBA е приложение на PXE, което работи много добре. TCP/IP методът може да бъде също използван, но преди това ядрото трябва да бъде подготвено с програма, наречена **imggen**.

- **Netboot**

Netboot, точно както и Etherboot, е софтуер с отворен код, който създава boot ROM образи.

- **Floppy disk**

Има два начина да се зареди LTSP на работна станция чрез флопи. Единият от тях е да се зареди Etherboot-а в буут секторите на дискетите [8]. Тогава това флопи ще изпълнява функцията на boot ROM.

Също така ядрото и `initrd` могат да се запишат направо на флопито и да се зарежда така, но е всъщност доста по-бързо, ако зареждането на ядрото става през мрежата.

- **Hard disk**

Той може да бъде използван заедно с LILO или GRUB, за да зареди ядрото и `initrd`. Или да се превърне самият диск в boot ROM.

- **CD-ROM**

- **USB Memory Device**

3. ПРЕДИМСТВА И НЕДОСТАТЪЦИ НА THIN-КЛИЕНТА

Thin-client моделът е полезен в много отношения, независимо от използваната операционна система [3], но приложението му чрез различни дистрибуции на операционната система Linux има някои специфични предимства, като:

- Намалява разходите за клиентски и сървърен софтуер;
- Намалява поддръжката – поддържа се само сървърът и има заменяемост на клиентските машини;
- Осигурява висока сигурност;
- Не са необходими платени лицензи за софтуер, използва се софтуер с отворен код.

Използването на thin-client е също така много подходящо и за обучението [2].

В последните 2-3 години някои чужди автори препоръчват на училищата да преминат към използването на thin-client под Linux, като посочват две групи причини: финансови и образователни [2, 4].

Като недостатък може да се отчете все още невисоката популярност на LTSP като проект дори сред професионалните среди, което от своя страна забавя значително и по-широкото му внедряване в различни обществени сфери.

ЛИТЕРАТУРА

- [1] Йорданова, Л., Желев, Ж. Кирякова, Г., Ангелова, Н. *Thin Client – нов живот за старите компютри*, 4-ти Балкански конгрес "Образованието, Балканите, Европа", 22-24 юни 2007, Ст. Загора, Том3, стр.110-115.
- [2] Dix, J. *The benefits of thin clients*, Network World, available at: http://findarticles.com/p/articles/mi_qa3649/is_200610/ai_n17190357.
- [3] French, M. *Introduction to Thin Client Computing with Novell Linux Desktop*, available at: <http://targnum.penguin.org.il/presentations/BrainShare2006/I>
- [4] Hargadon, S. *Why is Thin-client Linux Such a Good Fit for Schools*, available at: <http://www.stevehargadon.com/>
- [5] LTSP, *Linux Terminal Server Project - How to*, available at: <http://www.ltsp.org>.
- [6] McQuillan, J. *LTSP – Linux Terminal Server Project- v4.1*, Rev. 4.1.3-en, 20.06.2004, <http://ltsp.mirrors.tds.net/pub/ltsp/docs/ltsp-4.1-en.pdf>.
- [7] VanGundy, P. *LTSP Installation Guide: From Installation to LTSP Server*, available at: <http://wiki.ltsp.org/twiki/pub/Ltsp/Documentation/ltspguide.pdf>.
- [8] Rom-O-Matic.net dynamically generates Etherboot ROM images, available at: <http://www.Rom-O-Matic.net>.

ЖИВКО ЖЕЛЕВ,
 ТРАКИЙСКИ УНИВЕРСИТЕТ,
 КАТЕДРА "МАТЕМАТИКА И ИНФОРМАТИКА",
 СТУДЕНТСКИ ГРАД, 6000 СТАРА ЗАГОРА, БЪЛГАРИЯ.
 E-mail address: zhelev@uni-sz.bg, jjeleff@yahoo.com