Els kiosks Linux a les Biblioteques UAB

Ferran Jorba
Servei d'Informàtica
Universitat Autònoma de Barcelona
Ferran.Jorba@uab.es

Resum

Presentem el sistema de PCs de consulta pública d'informació a les biblioteques de la Universtat Autònoma de Barcelona. Es tracta de més de 140 ordinadors Debian, geogràficament dispersos pels diferents campus, amb un navegador i els *plugins* més populars, en modus kiosk (que només puguin fer el que se'ls deixa fer) i tres tipus d'usuaris (perfils) diferent. Com a part del projecte, s'ha preparat sistemes d'intal·lació trivial, de recuperació, d'actualització automàtica del programari i de monitorització via Web.

Context

Amb la desaparició dels antics catàlegs de fitxes i la seva substitució per ordinadors, les biblioteques s'omplen, primerament de terminals i, amb els anys, d'ordinadors complerts (PCs), per a la seva consulta. A més a més, les biblioteques ofereixen cada cop més serveis de biblioteca digital, amb subscripcions a les versions electròniques de les revistes i bases de dades més rellevants. Finalment, des de les biblioteques s'acostuma a oferir accés lliure a Internet. A la UAB ens vàrem plantejar, a partir de l'any 2000, de fer−ho amb Linux, ja que altres experiències a la UAB de PCs d'ús públic amb Microsoft Windows havien estat poc satisfactòries, degut a penjades, desconfiguracions, inseguretats, manca de control remot, etc.

Les grans decissions

No va costar excessivament convèncer a les parts implicades de muntar−ho amb Linux. Concretament, el Servei de Biblioteques ho va acollir amb una barreja de curiositat i entussiasme. A més a més, ja comptàvem amb experiències documentades de Linux cara al públic a universitats i museus d'altres països.

Quin Linux? Segur que es pot fer amb pràcticament qualsevol distribució, i les preferències personals, junt amb la familiaritat amb una en concret és un factor molt decissiu. Nosaltes som debianeros, de manera que tenim la convicció que amb Debian es pot fer millor que amb qualevol altra. Però això, insisteixo, altres han fet projectes comparables amb RedHat o SuSE.

Només calia llegir, documentar−se i decidir alguns temes claus. Per exemple, si els ordinadors haurien de disposar de disc dur o no.

Diskless o amb disc dur

Pel que haviem estat veient fins aleshores, bona part dels projectes Kiosk basats en Linux descansaven en PCs sense disc dur: des de l'arquitectura clàssica de terminals X, simuladors de *Network Computers* o usar tota la potència de càlcul local però amb disc accedit via NFS. Els arguments a favor de no tenir disc dur són, bàsicament, la recuperació i actualització instantània. També, al disposar d'un element mecànic menys, teòricament poden ser menys propensos a avaries.

Ara bé, hi ha sobretot un element que a nosaltres ens va fer desistir de muntar−ho sense disc dur. I és l'únic punt de fallada. Falla el servidor i no funciona res. No ens podiem permetre que tota la infrastructura fallés perquè el servidor no estigués disponible. En menor mesura, també cal tenir en compte que el trànsit de xarxa seria força més intens.

Alguns dels projectes *diskless* més rellevants són:

- Linux Terminal Server Project (http://www.ltsp.org/)
- DNA Lounge Diskless Linux Kiosks, de Jamie Zawinski

(http://www.dnalounge.com/backstage/src/kiosk/)
- Linux X Terminal (http://ladybug.vlug.org/vlug/meetings/X−terminal_presentation/)

També n'hi havien d'altres amb disc dur:

- Linux Kiosk HOWTO: http://www.linuxdoc.org/HOWTO/Kiosk−HOWTO.html
- A GNU/Linux−based Information Kiosk: http://www.shef.ac.uk/misc/personal/kiosk/
- Linux web browser station (formerly "The Linux Public Web Browser mini−HOWTO"): http://www.linuxdoc.org/HOWTO/mini/Public−Web−Browser.html

Decidint−nos per disc dur, sobretot optàvem per un sistema redundant, amb tolerànca a falles (de hardware), funcionament independent del servidor, desconnexió còmode del servidor per actualitzacions, etc. Els inconvenients més grans són la instal·lació inicial dels clients i la sincronització dels clients amb el servidor. Afortunadament, el software lliure té múltiples solucions per a aquest escenari.

Bootstrap inicial (instal·lació múltiple)

El que voliem era que qualsevol PC pogués fer de kiosk, sense cap mena d'intervenció hardware especial; sense targetes de xarxa específiques, ni actualitzacions de PROMs, etc. Tampoc no disposàvem de permisos especials sobre la infrastructura de xarxa general de la UAB. Amb un parell de PCs reciclats, un com a servidor i l'altre com a client, n'hi hauria d'haver prou per començar.

Afortunadament en Linux hi ha una gran varietat d'aplicacions que permeten automatitzar la instal·lació a múltiples clients. Com acostuma a passar, alguns d'aquests els hem descobert *a posteriori*, però això no ens ha fet canviar la decissió presa en el seu moment.

- Específics de Debian:
  - Replicator (http://packages.debian.org/stable/admin/replicator.html)
  - FAI (http://packages.debian.org/stable/admin/fai.html, http://www.informatik.uni−koeln.de/fai/)
  - Progeny Installer (http://packages.debian.org/stable/admin/autoinstall.html)
- Específic per RedHat:
  - Kickstart (http://www.redhat.com/docs/manuals/linux/RHL−9−Manual/custom−guide/part−install−info.ht
- Específic per SuSE:
  - ALICE (http://sdb.suse.de/sdb/en/html/cg_autoinstall.html)
- Genèric Linux:
  - Systemimager (http://www.systemimager.org/)
  - Partition Image (http://www.partimage.org/)

En Debian 2.2 (Potato), com era el nostre cas quan vàrem començar, només hi havia FAI i Replicator. Ens els vàrem mirar tots dos; Replicator és més senzill d'entendre i d'utilitzar, tant pel que fa a l'implementador com als usuaris finals. A hores d'ara, potser una seixantena dels ordinadors hi han instal·lat Debian els propis bibliotecaris, amb un disquet, connexió a la xarxa i contestant quatre preguntes trivials. L'autor de Replicator, Sébastien Chaumat, va ser, a més, molt atent a les nostres observacions i va incorporar algunes millores en la documentació i la gestió d'errors que li vàrem demanar.

Com es van instal·lar la resta? Doncs clonant discs. Per a instal·lacions massives, com es van fer el primer any, va resultar més ràpid clonar discs. A diferència de Windows, en Linux (i altres sistemes operatius, com la família BSD) és possible i normal que una mateixa imatge de disc funcioni perfectament en hardware diferent (dins d'una mateixa arquitectura, és clar), degut a la manera com el sistema carrega els *drivers*.

Sincronització entre els clients i el servidor

Tots els ordinadors tenen *exactament* el mateix software. El mateix kernel i el mateix servidor X. Depèn de la funció que facin, s'engegen uns programes o uns altres. Per això, en la nostra infrastructura, canviar la personalitat d'un PC és trivial, bàsicament canviar un fitxer en el servidor i rebotar.

Quan un client rebota, passen les següents coses:

1. Es posa en un *runlevel* no reservat (el 4).
2. Engegen el procés (client) de monitorització .
3. L'ordinador sincronitza (via rsync) els directoris de configuració general (`/etc/`) i els d'usuari (`/home/`) amb el servidor de configuracions.
4. Un fitxer d'actualitzacions (`/etc/kiosk/updates`) serveix per a avisar si hi cal fer una sincronització general de tota la màquina. Si al client li falta algun d'aquests fitxers, sincronitza tot el disc dur amb el servidor.
5. El fitxer `/etc/kiosk/kiosk.conf` li diu, a cada màquina, quina funció fa.
6. Segons el valor que troba, el script passa el control al fitxer d'inicialització d'aquest usuari.

La sincronització la fem via rsync. Evidentment, ha estat a posteriori que hem descobert que hi ha altres eines. En tot cas, de moment estem encantats amb la nostra primera opció.

- rsync (http://rsync.samba.org/)
- sup (http://packages.debian.org/stable/devel/sup.html)
- cvsup (http://www.cs.cmu.edu/afs/cs.cmu.edu/project/mach/public/sup/)
- unison (http://www.cis.upenn.edu/~bcpierce/unison/)
- cfengine (http://www.cfengine.org/)
- update−cluster (http://www.netfort.gr.jp/~dancer/software/update−cluster.html)

Cal recordar que aquesta sincronització del software i de les configuracions és possible gràcies a la política Unix que tota la configuració estigui en fitxers plans, editables i replicables, i una separació clara i entenedora de la configuració general del sistema (directori `/etc/`) del d'usuari (directoris `/home/`). En sistemes operatius Windows, basats en estructures binàries i obscures, i mesclant totes aquestes configuracions en un lloc inextricable anomenat registre, no es poden aplicar les mateixes idees que hem utilitzat nosaltres.

Monitorització

Els clients, quan arrenquen, posen en marxa un petit programa Perl que envia a un servidor, via UDP, detalls del hardware i software del PC i, cada 60 minuts, un missatge d'`uptime`. Aquest altre servidor té un programa que reb aquests missatges i els guarda en un fitxer. Cada 10 minuts es processen les dades i actualitzen unes pàgines HTML que fan un retrat de la situació del sistema.

El navegador

Quan varem començar, l'any 2000−2001, l'única opció viable de navegador era la de Netscape 4.x, amb les seves avantatges i inconvenients. Afortunadament, Mozilla ja és una alternativa viable, i hem escollit Firebird (abans Phoenix) com a navegador preferent. Degut als requeriments de màquina força grans, però, fem una detecció a temps real i només engegem Firebird si la CPU és 166 Mhz mínim i amb, al menys, 64 Mb de RAM. Si no, per als ordinadors més antics (encara tenim un número important de Pentium 120 amb 16 Mb de RAM) només és raonable engegar l'antic Netscape Navigator. No hem considerat Konqueror degut a la seva dependència de Qt i KDE, que haguessin inflat excessivament els requeriments de software, ni Opera, perquè no és lliure.

Les estratègies de *kioskització* dels dos navegadors són radicalment diferents. Netscape 4.x es fa sobretot via .Xdefaults, mentre que per als derivats de Mozilla (Firebird) les instruccions estan sobretot al document *Customizing Mozilla* (http://www.mozilla.org/unix/customizing.html). El manteniment d'aquesta doble infrastructura és, a hores d'ara, una certa pèrdua de temps i energies.

Un parell de documents ens han donat bones idees:

- Securing Netscape in the Public Library (http://www.northville.lib.mi.us/tech/netscape.html)
- Limiting Workstation Access to Specific Internet Sites using a Bogus Proxy Server (http://www.dayton.lib.oh.us/~kambitsch/netscape/bogus−proxy−server.html)

Els inconvenients

No hi ha res perfecte, òbviament tampoc amb Linux. Linux, com Unix en general, està molt ben treballat com a servidor i com a plataforma de desenvolupament i per usuaris avançats, però és dur per als usuaris finals. És cert que hi ha distribucions que fan autèntiques virgueries d'autodetecció, assistents i entorns gràfics molt elegants i treballats. Però en general és a costa d'una sobrecàrrega de software i complexitat afegida que s'acosta perillosament a Microsoft Windows. El nucli continua essent dur.

En el nostre cas, ens interessa una plataforma mínima, sense massa dependències ni farragositats. Els inconvenients més grans que hem trobat són:

Les fonts (tipus de lletra). Hi ha una excessiva complexitat i poca homogeneïtat en la gestió de les fonts en X. És molt difícil trobar la combinació de fonts adequada perquè el resultat sigui de bona qualitat i llegidor per pantalla. A més a més, Internet està plagada de referències a tipus de lletra i codificacions propietaris de Microsoft, que obliga a fer contorsions perquè es vegi bé en Linux. El tema és especialment difícil en alfabets no llatins, com el cirílic i llengües asiàtiques.

L'accés a disquet. La política de muntar i desmuntar dispositius en Unix i Linux no fa fàcil permetre que els usuaris finals puguin escriure al disquet. Això, que és obvi en Windows, ha estat un bon maldecap en Linux, on hi ha al menys quatre o cinc automuntadors diferents de disquet (amd, autofs, supermount, fdmount i, d'alguna manera, els mtools), a més dels que puguin implementar Gnome i KDE. Tot i que l'elecció final va ser autofs, la veritat és que cap d'ells és fàcil d'usar i tots tenen els seus inconvenients.

Els següents passos

Clarament, tothom indica que l'eina a usar per gestionar un número tant important d'ordinadors és molt probablement cfengine (http://www.cfengine.org). Pel que hem vist, promet automatitzar una sèrie de procediments que ara són una mica massa manuals. Està a la llista de temes pendents.

D'altra banda, estem implementant clients Citrix per a l'accés a aplicacions Windows (concretament bases de dades bibliotecàries). El campus de Sabadell també ha aprofitat aquest sistema per als seus punts d'informació propis, i possiblement se'n posaran d'altres.