Thousands of Linux Installations (and only one administrator)

Dr A V Le Blanc
Manchester Computing
University of Manchester

1. Introduction

The aim of the first Linux distributions was to create a usable single machine. The target was a single home machine, or later a single enterprise-class server. Recently some distributions aim to support an enterprise-class desktop. All of them take as the model a stand-alone Linux installation.

(The Beowulf project has created a distribution which can be `installed' once on a single machine, but which permits many nodes to be booted from this. But this assumes identical hardware on the nodes.)

It seems to me that we need a new goal in the Linux community: a desktop installation based on a single shared copy of a system, a copy which lives in an area shared among many clients by means of a network file system. Such an installation can be easier to install and easier to maintain than a block of stand-alone machines. Users and their filestore can be managed centrally but independently of the desktop management.

I have designed and implemented a version of Linux in this way, which is currently running at the University of Manchester.

2. Requirements

In my case, I was required to produce a Linux system for installation on the University's public cluster PCs. Ideally this system should be able to be maintained centrally by a single part-time adminisrator. User names would be managed by the central LDAP data base of University computer users, while user filestore would come from the University's centrally provided filestore. No new servers should be required to setup, install, or use the proposed system.

Existing Linux installations on some cluster machines were normal independent Linuxes, based on several distributions according to the 'religious' choice of each cluster manager, and requiring a good deal of time for securing and maintaining each machine independently. There is no existing distribution which can be scaled to several thousand clients with a minimum of effort.

To some extent security on the system is simplified because it is not running any servers that can accept a networking connection from outside, that is, with the exception of services like web browsing, ssh, telnet, ftp, etc, in which the user starts a connection locally to a remote server on another machine.

3. Users and authentication

The University has a central LDAP database of users and their passwords. A machine can easily be configured to use pam_ldap for authentication. For accounts we simply create a Unix account dynamically for any authorised person who tries to log in. For special purposes a machine can have a local user created, usually at boot time; for example, we do this when a machine is used for teaching a course: a temporary user is created on that machine. We also have existing users in our AFS file system, and login can be configured to allow these to authenticate directly at login time.

Passwords in the LDAP system, like the user names, are centrally managed. The team looking after the AFS system creates users and manages their passwords.

Each system has a root user, but the root user has no password, making login as root impossible. Root access for administration is possible by using various su utilities.

4. Filestore

User filestore in the University is currently provided by a SAN system. This filestore can be mounted using smbfs.

System filestore is in a network file system, and it must be one with good caching to make the whole client viable. We are using AFS, but it might be possible to use CODA or Intermezzo or Lustre or some other system of this kind if it is suitable for the target system. In AFS the cache is persistent across reboots, which is useful, and can be manually flushed if necessary.

5. Administration

Installation requires putting about 4mb of files on a disk, and installing a Linux boot loader. All our systems have a Linux partition already (part of Novell's ZenWorks for maintaining our XP systems), and all are using GRUB to boot. A complete installation, from blank disk to graphical login, requires less than 3 minutes. It does not place a great load on the network, and hence the installation of a large number of machines (typically about 100) can overlap.

In some cases XFree86 or sound needs to be configured manually at install time.

The addition of security patches and new software to the system is carried out on the network system, and can be tested on one or more clients before release to the whole system. I estimate that this maintenance takes about twice as long as maintaining one ordinary system, at least if you measure my time as administrator.

6. Maintenance during system boot

At boot time we start up networking and access the network file system before starting /sbin/init. This allows us to control the directory structure, files, and symbolic links that are actually on a machinee. Thus if a user somehow deletes or changes a system file, it is restored at boot time. Temporary filestore and scratch space gets cleaned up at this time.

The system allows us to define special tasks which can run at boot time before /etc/init starts. This includes changing config files, adding extra bits of software (in /usr/local), creating users, etc.

7. Todo

There is at present no software in place to control access to particular machines. If a machine allows access to users from the LDAP system, it allows access to all of them.

It would be useful to have a utility which allowed a remote machine to have instructions for each client, such as a security upgrade, which could be obeyed without having the need to reboot the machine first. Currently all our clients reboot daily.

8. Evaluation

This system is not rocket science; it does not offer anything fundamentally new. We have problems based partially on the base system we used for our network image (Debian etch) and partially on the locally changing infrastructure into which our image must tie if it is to be useful. But it does work; it produces a usable system with little install time, which works currently on several hundred clients with a variety of hardware, which should easily scale to three thousand machines, and which requires administration which I can do in an hour a week or less.

The task would be helped greatly if the large Linux distributions targeted or supported this kind of system.