Chapter

3

Analyzing the Existing Setup and Migrating

Contents

Any computer system overhaul should take the existing system into account. This allows reuse of available resources as much as possible and guarantees interoperability of the various elements comprising the system. This study will introduce a generic framework to follow in any migration of a computing infrastructure to Linux.

3.1. Coexistence in Heterogeneous Environments

Debian integrates very well in all types of existing environments and plays well with any other operating system. This near-perfect harmony comes from market pressure which demands that software publishers develop programs that follow standards. Compliance with standards allows administrators to switch out programs: clients or servers, whether free or not.

3.1.1. Integration with Windows Machines

Samba's SMB/CIFS support ensures excellent communication within a Windows context. It shares files and print queues to Windows clients and includes software that allow a Linux machine to use resources available on Windows servers.

TOOL
Samba

The latest version of Samba can replace most of the Windows features: from those of a simple Windows NT server (authentication, files, print queues, downloading printer drivers, DFS, etc.) to the most advanced one (a domain controller compatible with Active Directory).

3.1.2. Integration with OS X machines

OS X machines provide, and are able to use, network services such as file servers and printer sharing. These services are published on the local network, which allows other machines to discover them and make use of them without any manual configuration, using the Bonjour implementation of the Zeroconf protocol suite. Debian includes another implementation, called Avahi, which provides the same functionality.

In the other direction, the Netatalk daemon can be used to provide file servers to OS X machines on the network. It implements the AFP (AppleShare) protocol as well as the required notifications so that the servers can be autodiscovered by the OS X clients.

Older Mac OS networks (before OS X) used a different protocol called AppleTalk. For environments involving machines using this protocol, Netatalk also provides the AppleTalk protocol (in fact, it started as a reimplementation of that protocol). It ensures the operation of the file server and print queues, as well as time server (clock synchronization). Its router function allows interconnection with AppleTalk networks.

3.1.3. Integration with Other Linux/Unix Machines

Finally, NFS and NIS, both included, guarantee interaction with Unix systems. NFS ensures file server functionality, while NIS creates user directories. The BSD printing layer, used by most Unix systems, also allows sharing of print queues.

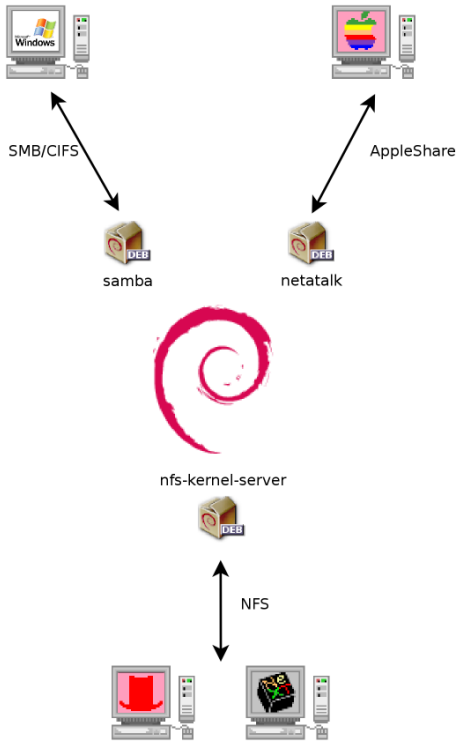

Figure 3.1 *Coexistence of Debian with OS X, Windows and Unix systems*

3.2. How To Migrate

In order to guarantee continuity of the services, each computer migration must be planned and executed according to the plan. This principle applies whatever the operating system used.

3.2.1. Survey and Identify Services

As simple as it seems, this step is essential. A serious administrator truly knows the principal roles of each server, but such roles can change, and sometimes experienced users may have installed "wild" services. Knowing that they exist will at least allow you to decide what to do with them, rather than delete them haphazardly.

For this purpose, it is wise to inform your users of the project before migrating the server. To involve them in the project, it may be useful to install the most common free software programs on their desktops prior to migration, which they will come across again after the migration to Debian; Libre Office and the Mozilla suite are the best examples here.

The `nmap` tool (in the package with the same name) will quickly identify Internet services hosted by a network connected machine without even requiring to log in to it. Simply call the following command on another machine connected to the same network:

```
$ nmap mirwiz
Starting Nmap 6.47 ( http://nmap.org ) at 2015-03-24 11:34 CET
Nmap scan report for mirwiz (192.168.1.104)
Host is up (0.0037s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
```

ALTERNATIVE

Use netstat to find the list of available services

On a Linux machine, the `netstat -tupan` command will show the list of active or pending TCP sessions, as well UDP ports on which running programs are listening. This facilitates identification of services offered on the network.

GOING FURTHER

IPv6

Some network commands may work either with IPv4 (the default usually) or with IPv6. These include the `nmap` and `netstat` commands, but also others, such as `route` or `ip`. The convention is that this behavior is enabled by the `-6` command-line option.

If the server is a Unix machine offering shell accounts to users, it is interesting to determine if processes are executed in the background in the absence of their owner. The command `ps auxw` displays a list of all processes with their user identity. By checking this information against the output of the `who` command, which gives a list of logged in users, it is possible to identify rogue or undeclared servers or programs running in the background. Looking at `crontabs` (tables listing automatic actions scheduled by users) will often provide interesting information on functions fulfilled by the server (a complete explanation of `cron` is available in section 9.7, "Scheduling Tasks with `cron` and `atd`" page 205).

In any case, it is essential to backup your servers: this allows recovery of information after the fact, when users will report specific problems due to the migration.

3.2.2. Backing up the Configuration

It is wise to retain the configuration of every identified service in order to be able to install the equivalent on the updated server. The bare minimum is to make a backup copy of the configuration files.

For Unix machines, the configuration files are usually found in `/etc/`, but they may be located in a sub-directory of `/usr/local/`. This is the case if a program has been installed from sources, rather than with a package. In some cases, one may also find them under `/opt/`.

For data managing services (such as databases), it is strongly recommended to export the data to a standard format that will be easily imported by the new software. Such a format is usually in text mode and documented; it may be, for example, an SQL dump for a database, or an LDIF file for an LDAP server.

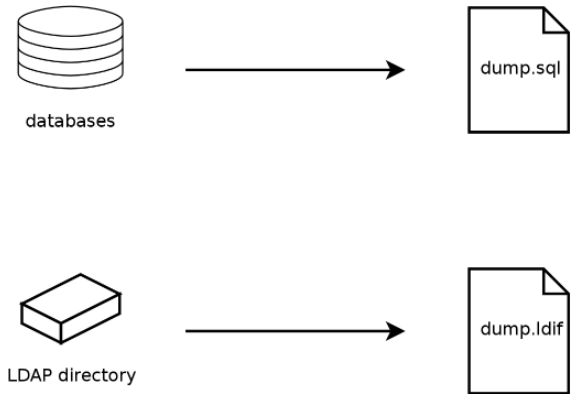

Figure 3.2 *Database backups*

Each server software is different, and it is impossible to describe all existing cases in detail. Compare the documentation for the existing and the new software to identify the exportable (thus, re-importable) portions and those which will require manual handling. Reading this book will clarify the configuration of the main Linux server programs.

3.2.3. Taking Over an Existing Debian Server

To effectively take over its maintenance, one may analyze a machine already running with Debian.

The first file to check is `/etc/debian_version`, which usually contains the version number for the installed Debian system (it is part of the *base-files* package). If it indicates *codename*/sid, it means that the system was updated with packages coming from one of the development distributions (either testing or unstable).

The `apt-show-versions` program (from the Debian package of the same name) checks the list of installed packages and identifies the available versions. `aptitude` can also be used for these tasks, albeit in a less systematic manner.

A glance at the `/etc/apt/sources.list` file (and `/etc/apt/sources.list.d/` directory) will show where the installed Debian packages likely came from. If many unknown sources appear, the administrator may choose to completely reinstall the computer's system to ensure optimal compatibility with the software provided by Debian.

The `sources.list` file is often a good indicator: the majority of administrators keep, at least in comments, the list of APT sources that were previously used. But you should not forget that sources used in the past might have been deleted, and that some random packages grabbed on the Internet might have been manually installed (with the `dpkg` command). In this case, the machine is misleading in its appearance of "standard" Debian. This is why you should pay attention to any indication that will give away the presence of external packages (appearance of `deb` files in unusual directories, package version numbers with a special suffix indicating that it originated from outside the Debian project, such as `ubuntu` or `lmde`, etc.)

Likewise, it is interesting to analyze the contents of the `/usr/local/` directory, whose purpose is to contain programs compiled and installed manually. Listing software installed in this manner is instructive, since this raises questions on the reasons for not using the corresponding Debian package, if such a package exists.

QUICK LOOK

cruft

The *cruft* package proposes to list the available files that are not owned by any package. It has some filters (more or less effective, and more or less up to date) to avoid reporting some legitimate files (files generated by Debian packages, or generated configuration files not managed by `dpkg`, etc.).

Be careful to not blindly delete everything that `cruft` might list

3.2.4. Installing Debian

Once all the required information on the current server is known, we can shut it down and begin to install Debian on it.

To choose the appropriate version, we must know the computer's architecture. If it is a reasonably recent PC, it is most likely to be amd64 (older PCs were usually i386). In other cases, we can narrow down the possibilities according to the previously used system.

Table 3.1 is not intended to be exhaustive, but may be helpful. In any case, the original documentation for the computer is the most reliable source to find this information.

HARDWARE

64 bit PC vs 32 bit PC

Most recent computers have 64 bit Intel or AMD processors, compatible with older 32 bit processors; the software compiled for "i386" architecture thus works. On the other hand, this compatibility mode does not fully exploit the capabilities of these new processors. This is why Debian provides the "amd64" architecture, which works for recent AMD chips as well as Intel "em64t" processors (including most of the Core series), which are very similar to AMD64.

3.2.5. Installing and Configuring the Selected Services

Once Debian is installed, we must install and configure one by one all of the services that this computer must host. The new configuration must take into consideration the prior one in order to ensure a smooth transition. All the information collected in the first two steps will be useful to successfully complete this part.

Operating System	Architecture(s)
DEC Unix (OSF/1)	alpha, mipsel
HP Unix	ia64, hppa
IBM AIX	powerpc
Irix	mips
OS X	amd64, powerpc, i386
z/OS, MVS	s390x, s390
Solaris, SunOS	sparc, i386, m68k
Ultrix	mips
VMS	alpha
Windows 95/98/ME	i386
Windows NT/2000	i386, alpha, ia64, mipsel
Windows XP / Windows Server 2008	i386, amd64, ia64
Windows Vista / Windows 7 / Windows 8	i386, amd64

Table 3.1 *Matching operating system and architecture*

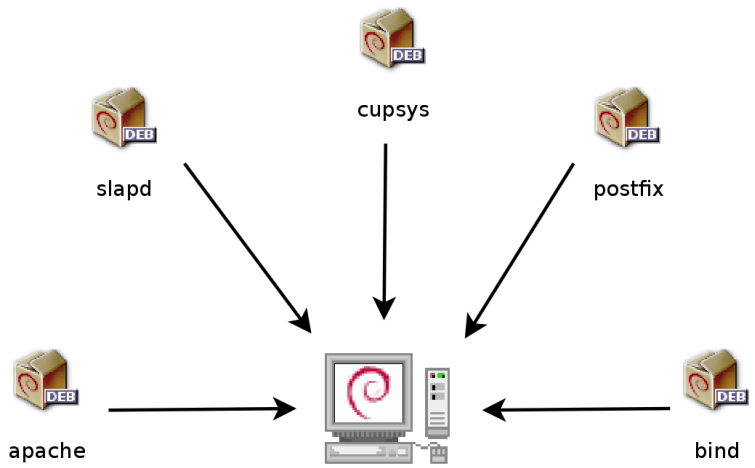

Figure 3.3 *Install the selected services*

Prior to jumping into this exercise with both feet, it is strongly recommended that you read the remainder of this book. After that you will have a more precise understanding of how to configure the expected services.

Keywords

Installation
Partitioning
Formatting
File System
Boot Sector
Hardware Detection