Although I’ve been using Macintosh computers since 1990, I’ve never administered a Macintosh server. So when Apple Computer invited me to try Mac OS X Panther Server, I jumped at the chance. I was anxious to see Apple’s take on server administration, especially after hearing that Panther Server depended on a good number of open source projects. As it turns out, Panther Server doesn’t include just a “good number” of open source projects — it includes an astonishing number of them: Mailman, Tomcat, JBoss, Apache, Postfix, Perl, Samba, and BIND, just to name a few. And Panther Server integrates many of those projects into a central console application, making the entire system much more valuable than the sum of its parts.

After spending more than two months with two servers running Panther Server, I can only say that I’m blown away. Panther Server is a joy to use. If you have just a modicum of experience maintaining a Linux (really any UNIX-like) server, Panther Server will make you feel like an expert. On the other hand, if you’re an expert system administrator, Panther Server will make you feel right at home.

**Panther Server, Meet Frankenstein**

To put Panther Server to the test, I decided I would overhaul my home network topology and put Panther Server at its core, in charge of a number of significant services. I wanted my own DNS servers (instead of using Yahoo Domains); I wanted to send and receive email from my own SMTP server; I wanted an easy-to-maintain router and firewall; and I wanted more CPU power for my web site. (My Red Hat Linux 9.0 server works very well, but its underlying hardware is getting old, cannot be expanded much further, and is woefully underpowered for rich LAMP applications such as Mambo or Bricolage and complex MySQL databases and queries.

To run Panther Server, Apple provided me with a 2.0 GHz, dual processor G5 with 2 GB RAM. (Much has been written about the dual G5, so suffice it to say that the hardware is rock solid. Except for reboots to allow for system software updates, the machine never crashed. The performance is impressive.) Installation of Panther Server was a cinch, and was no more complex than installing its consumer counterpart, Mac OS X Panther.

However, I also wanted to see how Panther Server performed on older hardware and chose to upgrade “Frankenstein” to the server operating system. Frankenstein is aptly named since it’s a hardware golem: a circa-1999 Graphite G4 with a Sonnet 1.4 GHz CPU upgrade, 1.25 GB of RAM, an Ultra-Tek serial ATA card, an ATI Radeon 9800 video card, built-in Ethernet, a Bluetooth dongle, and a generic PCI Ethernet card ripped from an old PC. The machine’s performance doesn’t surpass the G5, but is very respectable. Once again, installation was a snap, even on a morass of third-party hardware.

I also chose to install Apple’s Xcode tools on the G5, since I knew I’d be compiling code on the machine. Xcode includes GCC, Perl, Python, PHP, and Apple’s own integrated development environment for writing C, C++, Objective C, Java, and AppleScript code. Xcode is provided for free with every Apple operating system.

**Rolling Up My Sleeves**

Since Frankenstein has two network interfaces, I decided to use it as the router and firewall. The G5 — with its two PowerPC processors — would serve as the file, print, DNS, web, and LAMP server, as well as replacing “Frank” as my desktop workstation. The majority of Panther Server administration is performed in a concise application called Server Admin (the leftmost window in the figure above). Server Admin lists services down the left side of the main window, using the pane on the right to display each service’s logs, status, options, and settings. Starting and stopping a service is a single mouse-click, and configuring many of the services is largely an exercise in clicking on check boxes.

For example, to enable outgoing and
incoming email, you simply enable the “Mail” service and then turn on POP or IMAP, specifying whether you want to support secure connections and SSL, and choosing one or more authentication schemes.

The Server Admin panel labels running services with a green light. If a service is misconfigured, it’s flagged with an exclamation icon, indicating that you must intervene and fix the problem. A red light indicates that the service cannot be reached or is down.

Also, as you can see from the screen shot, you can have multiple machines running Panther Server — such as a rack of Xserves — and Server Admin can remotely configure each of those machines. You can even manage software updates remotely, obviating the need to go to the console of each server.

File sharing and user and group settings are managed in Workgroup Manager, the other window in the screen shot. Files can be shared via AFP, NFS, FTP, Windows and AppleTalk. All user attributes and privileges — email aliases, login shell, UNIX group membership, quotas, email access, Windows access — are all controlled centrally from the Workgroup Manager. Adding a user took me about sixty seconds, and I didn’t have to worry that I missed some text file somewhere.

To build out Frankenstein as a router and firewall, I connected one NIC to my DSL modem and the other NIC to a simple hub to serve my LAN. Using Server Admin, enabling network address translation and establishing DNS, including zone transfers between the two server machines, took no time at all.

Similarly, setting up SMTP, IMAP, POP, and web service was a few clicks. If I lost any time in setup, it was due to my own errors — and getting use to the Panther Server interfaces.

### Not Just a Pretty Face

While Server Admin and Workgroup Manager are powerful system administration tools, sometimes you just can’t beat the command-line. Of course, since Panther Server has FreeBSD at its core, the command-line is just a mouse click away. Just fire up the Terminal application and all of your favorite UNIX-like utilities are at your disposal.

For example, my DSL connection failed a number of times during a recent worm attack. To keep connected, I wrote a shell script using natd and ipfw to reroute all of my wireless and wired network traffic via Frankenstein’s software modem (better slow than disconnected).

Of course, the command-line is indispensable for building software, and I was able to build a great many open source projects with little or no complication.

I built Perl 5.8.3 to run Damian Conway’s Perl::Rules module; I built Apache 2.0.19 (and its prerequisites), PHP 4.3.6, and APC 2.0.3 (the PHP accelerator cache); and I built a number of other supporting utilities to run a host of LAMP projects, including phpMyAdmin and Mambo.

You can download the script I wrote to build all of these open source projects on Panther Server at http://blog.streicher.us.) Happily, I can now play with new revisions of PHP and run those applications under the accelerator cache. Mac OS X Panther Server is a “X.”

Martin Streicher is Editor-in-Chief of Linux Magazine. His email address is mstreicher@linux-mag.com.

---

### PRODUCT REVIEW

**APPLE COMPUTER**

**MAC OS X PANTHER SERVER**

http://www.apple.com

**PRICE:** $499 for ten (simultaneous) clients; $999 for unlimited clients.

**RATING:**

**SPECIFICATIONS:** A Xserve, Power Mac G5, G4, G3, iMac, or eMac; 256 MB of RAM (more is better); 4 GB of disk space.

**PROS:** Easy-to-use system administration tools; can run UNIX and Mac OS X applications; based on FreeBSD, so many open source tools compile and install effortlessly; documentation very good; great price.

**CONS:** Requires Apple hardware; user interfaces can get out of sync with actual system settings when using command-line tools only.

**BOTTOM LINE:** Must Buy

---

**Bottom Line:** Must Buy