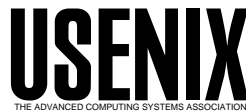


USENIX Association

Proceedings of the
4th Annual Linux Showcase & Conference,
Atlanta

Atlanta, Georgia, USA
October 10–14, 2000



© 2000 by The USENIX Association

All Rights Reserved

For more information about the USENIX Association:

Phone: 1 510 528 8649

FAX: 1 510 548 5738

Email: office@usenix.org

WWW: <http://www.usenix.org>

Rights to individual papers remain with the author or the author's employer.

Permission is granted for noncommercial reproduction of the work for educational or research purposes.

This copyright notice must be included in the reproduced paper. USENIX acknowledges all trademarks herein.

Introducing Linux File Services into a Windows NT Network: A Pilot Program

Richard R. Morgan

Consultant – VistaRMS, Inc.

Abstract

This paper is a report on a pilot program to implement Linux and Samba as a file server in a telecommunications company of about 5,000 people. The intent of the pilot program was to demonstrate the feasibility and cost-effectiveness of using Linux servers in the enterprise, as well as to illustrate the seamless integration with Windows NT desktop clients that Samba can provide. We will review the computing environment, the reasons for certain choices in hardware and software, as well as the steps in the implementation process.

Introduction

Linux seems to be on everyone's mind today. My workgroup is no different, except that unlike many of the pundits and predictors of success or doom, we are happily using Linux daily.

As a long-term consultant to a company in the telecommunications industry, I have the opportunity to use Linux every day in my work. In addition to our many other projects, my group tests the latest systems and software with Linux, helping to set the direction of information technology for the company.

So, we were duly excited when we got the chance to create a pilot program to use Linux as a file server platform. We knew it would be a challenge, though the process seemed straightforward. But, as the old adage goes, "the devil is in the details." The challenges weren't all in the software or technical realm; many of them sprang from getting interdepartmental cooperation and interaction with our user base.

Project Definition

Environment

Servers – The existing file servers throughout the company are predominantly based on Windows NT 4.0. They are a wide array of Hewlett-Packard servers, such as the LH-Pro, LH-3, and LH-4 series. Elsewhere in the company, there are numerous other

Unix servers from Sun, Hewlett-Packard, and IBM. So, the use of a Unix (or 'Unix-like') solution was not a completely foreign idea, although these Unix servers typically run large databases and applications. This meant our local desktop administrators often had little experience in this area.

Desktops – The client machines throughout the company are also based on Windows NT. The standard desktop machine is complemented by a relatively high proportion of laptop machines, also running Windows NT. All users have home directory space provided to them on file servers for storage.

Politics – It is a fairly safe to say that Windows NT is an entrenched platform, with a whole organization built up to support it. However, throughout the project, we have run into interested participants who have shown a great interest in Linux and offered their help and enthusiasm for its introduction.

The Problem

What's the problem? Why not just stick with Windows NT as a file server platform? All is not perfect in that arena. Many servers are running out of disk space and others are simply overburdened with traffic and numerous Windows NT services and need relief. The company continues to grow rapidly and the server burden will not diminish soon.

On the licensing side, the Microsoft model of charging for Windows NT clients and NT server licenses seems standard enough. However, the separate and

additional cost of licensing every single desktop machine for 'client access' to use NT-based file and print services adds greatly to the expenses of the company.

The Solution

A potential solution to these problems is the introduction of Linux servers into the network. Immediately, client access license costs can be eliminated by using Linux (which has a licensing cost of \$0) to provide file and print services. Other costs such as retraining may be incurred though, at least initially.

Anecdotally at least, Linux seems to make better use of hardware. So, our theory was that NT-based services, including domain control (BDC), Server Management (SMS), and others such as WINS could be moved to smaller, less expensive machines. We would then install Linux on the existing file servers, thus extending their lives. The configurability of Linux would allow us to greatly customize the installation and provide a file server with a minimal operating system footprint.

Linux could breathe some life back into machines that were overburdened due to the plethora of Windows NT services being run on them, in addition to serving files and managing print jobs.

But, we had to prove all of this first. Thus, our pilot program emerged.

Linux – So, Linux is the operating system we decided to try. But which Linux? Hewlett-Packard made the choice for us, really. Throughout our offices, all of the file server hardware is HP gear. We have numerous servers, most with warranties and paid-up support contracts from HP. HP chose to 'certify' Red Hat Linux (starting with version 6.0) on our categories of servers. We felt we could get technical support from HP, even using Linux, should we ever need it.

We were comfortable with Red Hat Linux because we felt our greatest battle would be education; both at the system administrator level, where NT-trained administrators would need technical knowledge from a wide array of published books and training programs, and at the management level, where the managers would have at least heard of Red Hat and its tremendous IPO.

Technically, we also liked Red Hat 6.1 because it offered drivers for the Hewlett-Packard Mega-Raid cards with the stock distribution, and like previous Red Hat distributions, it offered a straightforward manual installation process and the convenient Red Hat Package Manager (RPM).

Samba – Now, the star of the show: Samba. Samba is a triumph in the Open Source world. Samba is the server software that speaks Microsoft's server message block (smb) protocol. Samba would allow us to communicate with our Windows NT desktops and completely emulate a Windows NT file server.

Project Goals

The goals of our program were determined at the outset, and seem to have remained constant. We intended for this implementation, which would occur at only one building with 200–300 users, to be a true pilot. We would be doing all of this with live users and their data, so a careful approach was required.

We wanted to prove the concept, the technology, and also learn enough to be able to replicate the program throughout the company if the pilot was a success.

We hoped to encounter as many problems as possible during the pilot, so that we would be better prepared for a real, full-scale implementation.

File Server Conversion

Specifics of Tools: Hardware

For the file server, we used a Hewlett Packard LH-3 server. It is a dual processor machine (Pentium-II 450mhz) with a 70GB (available storage, using hardware RAID) disk array and one gigabyte of memory. This server configuration is commonly used throughout the company's field offices. We hoped to implement the Linux solution at field offices nationwide, so we chose the same configuration as in the field.

We partitioned the disk to have a large /home partition for the user home directories.

Specifics of Tools: Software

As mentioned above, we chose a stock version of Red Hat Linux (version 6.1, kernel 2.2.12-20smp).

Samba 2.0.5a was current at the time and had the features we needed. Some newer versions of Samba are available now, so we may eventually upgrade the system when necessary. We started out with a simple and small smb.conf (the Samba configuration file) and added directives only as we fully researched and tested them. After making configuration changes, we used the testparm utility provided with Samba to validate the changes.

We chose Secure Shell (SSH version 1.2.27) as a

secure method for remotely administering the machine. Our Windows NT administrator installed the TeraTerm client on his NT machine to provide ssh access. Big Brother was used to remotely monitor the machines, and we used Bash shell scripts to script the conversion.

While we didn't know exactly what we might run into later in the project, we trimmed a lot of unnecessary packages from our installation, especially games, editors, documentation, extra window managers, and graphics tools. We chose to leave the C libraries in place, in case we needed to recompile some software. Webmin (version 0.74), our web-based administration tool required the Apache web server, so we installed that also.

Preparation & User issues

Whose data will be moved? – This area was a major challenge. The move to the new file server coincided with the physical relocation of about 300 users from three different buildings around our metro area over a two-week period. This meant that we had to coordinate well with our facilities personnel to know exactly who would be moving, and on what dates.

How much data will be moved? – The amount of data was directly related to whose data we would be moving. Some users generate very little home directory data, but others, such as developers and graphic designers can have huge directories. Even for regular users, those gigantic Excel and PowerPoint files can take a lot of room.

The amount of data being moved affected the time it took to transfer the files across the network, as well as our choices in partitioning the server disk space. The final toll was about 5.5GB of files from three servers.

How will users be authenticated? – This was relatively easy. Our network authentication is done via the Microsoft NT domain controller model (PDC/BDC). Samba made it very simple to point requests to a particular set of BDC servers for user authentication. This is the `security = domain` directive in the `smb.conf` configuration file. Along with that directive, you must set the `password server` directive to point authentication requests at the correct machines.

Remember to withdraw access to old file server after conversion – To prevent confusion and data corruption, we did the conversion during off-hours, and after we converted the user files, we removed the user's access to the Windows NT server. We first set the permissions to read-only, with the NT administrator as owner of the files, to help answer questions about the conversion. Later (about two

months), we removed all access to the user's home directory on the old server.

Handle automatic redirects in Windows NT User Profile – When our Windows NT users login, a login profile sets their 'H' drive to the correct file server with their respective home directories. After the conversion, we ran through the profiles of the converted users and changed this mapping. Their 'H' drive would still point to their home directory, except it would now be on the Linux box.

Handle user's manual drive mappings via email notice and Help Desk support – Some users had manual drive letter mappings to the old Windows NT servers. We sent out several emails to the affected users warning them before the conversion, and also reminded them after the conversion of the need to change any manual mappings. We also enlisted the support of our internal Help Desk in answering questions about this change.

The Conversion Process

Set up users on Linux system using shell scripts – Once we had a list of the users, we fed this list to a Bash shell script that created the user accounts on the Linux machine. Their shells were set to `/bin/false` to disallow direct logins and no passwords were established.

Create user home directories using shell scripts – The same script, which simply wrapped the standard `adduser` script, created the user home directories.

Control share access – To prevent anyone from poking around in the Linux machine via Samba shares during the conversion process, we chose to limit the access to all shares to the root user. This was done in the Samba configuration file, `smb.conf`. After the conversion, we removed this entry to allow full access to users.

Move the data into a holding area – On the Windows NT side, we completed copying the data from each server before proceeding, in case there was some failure or other problem (the principle of "atomicity"). We wrote scripts on NT that copied the contents of the home directories of the affected users. These contents were copied to a holding area on the Linux server. We logged all files copied.

Verify the copied data – As the data from each Windows NT server was copied to the Linux server, we checked both servers to verify that all files were copied. We checked file sizes for files and directories and did a spot check of some files by opening them in their respective applications.

Move data from holding area to user home

directories – Once we were sure that we had the correct data set, we copied the data from the holding area into the proper user home directories using a relatively simple cp command.

Modify file permissions and ownerships using shell scripts – During the copy process, the converted files were written with the root user as owner. We wrote a script to recursively move through the user's directories and set the proper permissions and ownership on files and directories.

Review logs and handle the exceptions and errors – All the scripts we used created log files of their actions and any errors encountered. We reviewed the log files throughout the process (`tail -f logfile_name` is rather useful for this.) After the conversion, we manually researched and resolved the errors, which were very few.

Test user logins and read/write access of Samba shares – The final step was to physically log in as several different users and confirm that the proper drive mappings were being set, and that the user could properly access his files. We tested both reading and writing to the directories as well as reading and writing of converted files.

Post-conversion tasks

Enable space management: Set up filesystem quotas – The existing Windows NT servers use a third-party tool, SpaceGuard, to control disk space usage. Each user is given a 30 megabyte limit on disk usage. This function was easy to provide in Linux using filesystem quotas.

The basic quota system on Linux is established and administered for users on a filesystem basis, so we have a dedicated /home partition on the Linux machine. Each user has a hard limit of 30 megabytes. There were many exceptions and after a bit of training, these were left to the Windows NT administrator to handle.

Before enabling quotas, we created a script to run through user directories, calculate their sizes, and report the worst offenders. This helped us know what to expect from users when quotas were turned on, this time with hard limits.

Enable secure access: Set up Secure Shell and disable telnet – To help secure the box, we disabled ftp, telnet, and other services. We installed Secure Shell and trained the administrator to use ssh and scp when administering the box remotely. The machine is behind our corporate firewall, but Windows NT administrators occasionally dial in remotely to fix problems.

Enable remote administration: Install Webmin – As part of ongoing maintenance, new users must have accounts added to this machine. We installed the Webmin tool to provide Help Desk employees with the necessary facility to add new users. Webmin is extremely customizable and we cut its functionality down to this single function.

Enable remote monitoring: Install the Big Brother client – During this project, we were also installing six Linux print servers. To monitor the whole thing, we chose the Big Brother monitoring tool (version 1.4a). Big Brother is simple to use, extensible, and its graphical alerts were a good fit for our local Windows NT administrator. Be aware though, Big Brother is not covered by the GNU General Public License.

Enable remote syslogging: Send logs to a separate log server machine – We enabled remote system logging. This allowed us to consolidate the logs of many machines on one central box, and then run scripts against the logs for reporting or troubleshooting purposes. We configured /etc/syslogd.conf to send logs to a separate logging machine. Rather than using a local destination of /var/log/messages, we used a machine name, @vauas005.

What We Did Right

We did good testing in our lab. Several weeks ahead of the conversion, we did extensive testing on real data to refine our procedures and scripts. With the cooperation of the local NT administrator, we ran through the conversion process with a set of user directories. With this testing, we found holes in our knowledge of Samba and worked through them. We learned how the BDC system works on Windows NT, and how users are authenticated through Samba.

Our biggest unknown remained the amount of time necessary to move files across our WAN.

We scoured the documentation for tips and gotchas. We bought every Samba book that existed and read through them. The O'Reilly book, Using Samba seemed the most help to us, especially the 'fault tree' section. It helped immensely in focusing our troubleshooting efforts. We also reviewed the documentation provided with Samba and used it to create our preliminary conversion checklist.

We were very careful and methodical in planning the conversion of the data. With the confusion of so many people moving, we had a hard time knowing exactly what to convert. We worked with the facilities group closely to get the correct list. We even pushed the conversion back twice, knowing that we didn't have the correct roster.

Several times before we converted, we talked through the conversion process, making a list or a kind of 'script' of who would do what tasks and the schedule for each conversion item. This really helped keep everyone on track during the conversion.

We wrote good scripts that did most of our work, checked for errors and logged all transactions. No one wants to do lots of manual work. It is boring and an avenue for errors to creep into the process. To help speed the process, we wrote a number of Bash shell scripts and Windows batch files to automate the conversion.

Each script had options for creating logs of its action, which is especially important if the script changed data. We were able to run each script, having it echo its actions to the screen while testing so we could understand the impact on a test system before we actually changed or moved any real data.

During and after the conversion, we reviewed our logs looking for problems and errors. We had volumes of logs, which tracked every action taken.

We were able to use lots of open source software. This is my favorite part. We used Linux, Samba, Big Brother, quota, and the Bash shell to quickly and reliably accomplish our goals with no licensing costs or other hassles.

What We Could Have Done Better

As a pilot project to be used by real users, this project was bound by most of the same requirements applied to our production servers. Eager to get started, we did not do a very good job of gathering formal requirements and getting inter-departmental agreement on them. Thus, during the final days before the conversion, we still had requests for certain functions coming in from other groups.

We also did not formulate a solid exit strategy for our group. This led to confusion among the groups about who was to manage the server immediately after the conversion. We assumed, wrongly of course, that since the project was a pilot, we would continue to run it and maintain the machine.

To remedy these items, we could have used better communications. We did create an intranet website for the project but assigned no single person as the maintainer. The site did not get updated in a timely manner or even very often.

Even with all of our testing, during the copy process, where user files were copied from their original locations on the Windows NT servers to the 'holding

area' on the Linux server, we had a small slip-up. As the files were copied, the file dates ('last modified') were all set to April 21, 2000 (the day of the conversion), and reflected the current time. We should have copied the files in a manner that would have maintained the correct dates on the files, perhaps by bundling them with zip, tar, or a similar tool. We have had no users complain, though.

As pure techies, we were all more interested in getting the project done and the machine operating well. We didn't spend the necessary time doing a strong business case and things like cost-benefit justifications. I really feel that the lack of this has hindered our efforts somewhat.

Future Plans

We are still monitoring the performance of this server, proactively looking for problems. We have also implemented Linux and Samba as a print server solution for our building. We were able to use small Hewlett-Packard VL desktop machines as print servers. These machines were actually surplus equipment, no longer useful in the world of the Windows NT GUI.

Our group will now complete documentation which will be the guide to implementing this platform throughout the company. To deploy Linux widely, we will need a more streamlined installation model, perhaps disk imaging or a 'kickstart' style of installation.

We have started receiving additional requests from various groups for Linux-based file and print services. To provide this, we need a better method of managing quotas and groups. We are searching for a directory-based quota system, which is a feature available (albeit through third-party software) to our Windows NT servers now. Managing users on individual servers will grow to become a burden quickly, so we would like to manage groups and access centrally. For this, we are evaluating NIS as a compliment to the Microsoft PDC/BDC model.

Company management needs to set the course and fund the Linux training of Windows NT administrators. We investigated training options and found that there is a great deal of Red Hat training being conducted nationwide. Companies like LinuxCare can even provide support for Linux machines.

Our group now goes on to explore the applicability of using Linux and other Open Source tools elsewhere in the enterprise, particularly using LDAP and clustering technologies.

Conclusions

Plan well, test a lot, and work with other groups to get full cooperation. Everyone should understand the role of the new server and his/her role in the process of conversion.

Take a long-term view, as Linux is really just beginning to become popular (beyond media hype) in the corporate world.

Acknowledgments

None of this project could have been completed without the tireless and imaginative work of Rob Fike, Jim Edwards, Frank Hum, Geoff Silver, Susan Maitra, P.S. Ramesh, and Roger Maduro. Thanks to all of you.

We would like to thank the late Bob McLaughlin for having a long-term architectural vision that included Linux and for being tough enough to promote that vision well before Linux became *The Next Big Thing*.

Author Information

Richard R. Morgan is a consultant with VistaRMS, Inc., in Herndon, Virginia and has a range of experience that includes system integration, web design, and software development.

He began his computing odyssey many years ago by teaching himself BASIC on the Timex-Sinclair 1000. He is the corporate secretary and webmaster for Tux.org (<http://www.tux.org>) and his local LUG (<http://novalug.tux.org>).

Richard lives in Haymarket, Virginia, with his wife, Laura, daughter Abby, and all of their pets. He can be reached at rmorgan@tux.org.

References

Blair, John. *Samba: Integrating UNIX and Windows*, SSC, 1998.

Eckstein, Robert, David Collier-Brown, and Peter Kelly. *Using Samba*, O'Reilly & Associates, January 2000.

Frisch, Aeleen. *Essential System Administration*, O'Reilly & Associates, September 1995.

Software

Samba – <http://www.samba.org>

Big Brother – <http://www.bb4.com>

Red Hat Linux – <http://www.redhat.com>

Bash shell – included with Red Hat Linux

Linux quotas – included with Red Hat Linux

Secure Shell – <http://www.ssh.fi>