A STANDALONE SAMBA-NIS/NFS SERVER MODEL FOR WINDOWS AND LINUX DUAL BOOT CLIENTS WITH INDIVIDUAL USER AUTHENTICATION

K.K.L.B. Adikaram
D.T. Andrahannadi
Computer Unit, Faculty of Agriculture, University of Ruhuna,
Mapalana, Kamburupitiya, Sri Lanka
M.K.D.K. Piyaratne
Computer Unit, Faculty of Agriculture, University of Ruhuna,
Mapalana, Kamburupitiya, Sri Lanka
State Key Laboratory of Crop Stress Biology in Arid Areas, College of Plant Protection, Northwest A&F University, Yangling, Shaanxi, PR China
D.S.R. Wijewardana
C.M. Navaratne
Computer Unit, Faculty of Agriculture, University of Ruhuna,
Mapalana, Kamburupitiya, Sri Lanka

Abstract

Maintaining several file servers is a management as well as financial overhead to an origination. We introduce a standalone SAMBA-NIS/NFS server model for Windows and Linux dual boot clients with individual user authentication and common home folder is proposed for schools or universities. We used NIS (Network Information Service), NFS (Network File System), SAMBA, Fedora and Windows for configuration. This model serves requests from both Windows and Linux clients with generic user name and password. Individual user authentication (IUA) is guaranteed: a separate profile is accommodated for each individual user with common home folders for both Windows and Linux platforms. Four year observation results show that the developed model is consistent and the overall system performance is significantly higher than that of the system with two servers. Further the system is efficient and error free in terms of maintenance and concurrent access to large numbers of users. This model can be implemented in schools or university computer laboratories, and, is economically more suitable for small scale network systems.

Keywords: Standalone file server, Network information service, Network file system, SAMBA server, dual boot operating system

Introduction

Multi-platform or multi-operating system computer environments (Rajagopal, 1999) are popular in computer laboratories which are established for research, teaching and learning purposes. Particularly these laboratories use different file servers which characterized on each operating system of the client even for the very basic services such as data storage and user authentication. In the worst case scenario, a file server is provided for the most demanding operating systems, and if it is undemanding, it uses local user authentication and storage in order to overcome the financial and technical problems. Almost all popular operating systems such as Windows, Linux, UNIX, BSD, Solaris, Mac OS and Novel NetWare have developed their own file server systems. Active Directory (AD) (Simmons, 2001) service developed by Microsoft only supports computers in Windows domain networks. Apple Open Directory developed by Apple only supports the Mac OS domain. NFS developed by Sun Microsystems is widely used and supports most of the Unix (Solaris, AIX and HP-UX) and Unix-like (Linux and FreeBSD) operating systems (Eisler et al., 2001). Some operating systems such as Novell provide different versions of FSs which could be used in different operating systems domains, but only as separate distributions. For instance, the Novell eDirectory developed by Novell (Novell Administration Guide, 2013) supports Windows 2000, Windows Server 2003, SUSE Linux Enterprise Server, Red Hat Enterprise Linux, Novell NetWare, Sun Solaris, IBM AIX and HP-UX.

A FS for a single operating system environment could be deployed easily with optimal condition. But in the environment of multiplatform operating systems, it is difficult to deploy one file server that is capable of serving at least minimum requirements due to incompatibility issues. Deploying a server which is capable of catering as FS with user authentication from the same server for both Windows and Linux operating systems is also a challenging task. Generally this issue is solved by dedicating several FSs for different domains within one organization even though it needs a single server. A typical example of this condition is shown in Figure 1. This kind of system could be raised several difficulties such as using additional hardware and human resources in terms of operating and maintenance (Kerem and Aydin, 2011). Further, deploying separate systems need separate data backup/restore facilities and separate user management as well. In some situations, purchasing user or group licenses could also be required. These all will make unaffordable extra expenses especially for small scale systems.

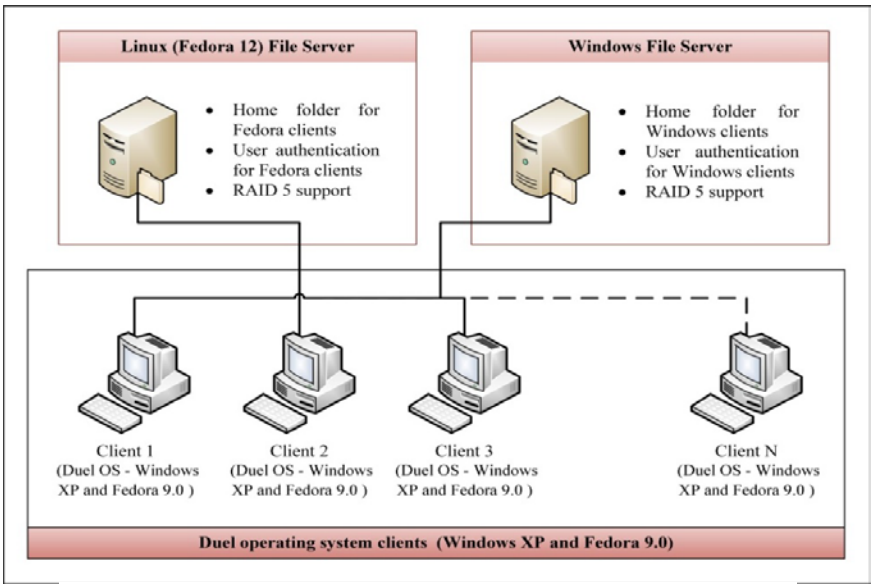

Figure 1. A typical dual OS system with file servers.

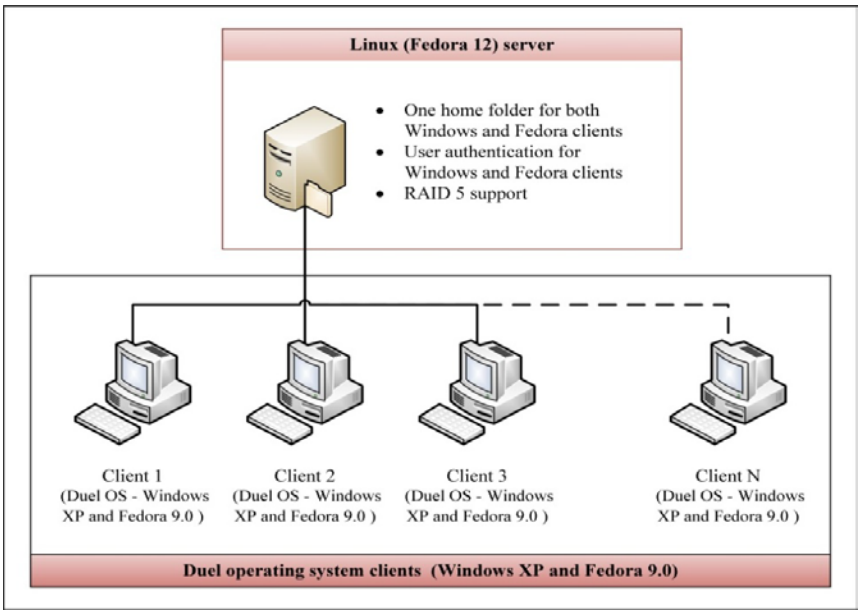

Figure 2. Proposed new model with single server for user authentication and file management

In this study, we proposed a single open source solution (single FS) for both Linux and Windows clients. This concept is motivated by some problematic observations which we noticed in our computer laboratory. Before implement this system, we maintained a dual boot system (Windows

and Linux) in our practical laboratory for 750 students. We used a Windows 2000 server with Active Directory for Windows clients while maintaining another separate NIS server for the same clients in a Linux environment (Figure 1). Though we provided the same user name and password for both operating systems it was necessary to maintain separate user accounts and take two data backups at different times. In order to overcome those problems we designed and implemented a new model which can deploy a single file server with dual boot clients (Figure 2). Linux and Windows operating systems were used as clients.

Materials and Methods

SAMBA Service and NIS/NFS

SAMBA is a file and print service for various Windows clients and runs on most Unix and Linux systems (Eisler et al., 2001; Jay et al., 2003). SAMBA service is normally available in almost all Linux distributions (Petersen, 2004) and installing and configuring SAMBA in a Linux server is a solution for a file server for Windows (Sander Van Vugt, 2009; Eisler et al., 2001). NFS is a file system which supports most of the Unix and Linux operating systems and installing NFS in Fedora server and configure NFS is a solution for a file server for the Linux systems. The Network Information Service (NIS) is a system for distributing shared configuration files across a network which provides centralized control over a variety of network information such as workstation names and addresses, network services and user information. NIS is available in Linux distributions and configuring NIS will allow to manage Linux users over the network. In the proposed model, we added both Linux users and Windows clients to the SAMBA domain. Then the same Linux users are capable of login as Windows users using the same user name, password and user home folder. This method eliminates creation of separate servers for both platforms.

Design of the Model

`The model contains one file server (Fedora 12) and dual boot clients, each comprising Windows XP and Fedora 9. The diagram of the proposed model is furnished in Figure 3. We implemented the model in three configuration steps; Linux server configuration, Windows client configuration and Linux client configuration. Linux server configuration is the main and the decisive step of the model implementation because it is needed to add both Windows and Linux clients to single Linux server. Fedora 12, one of stable Linux versions we experienced at that time, was selected as the operating system for the Linux server.

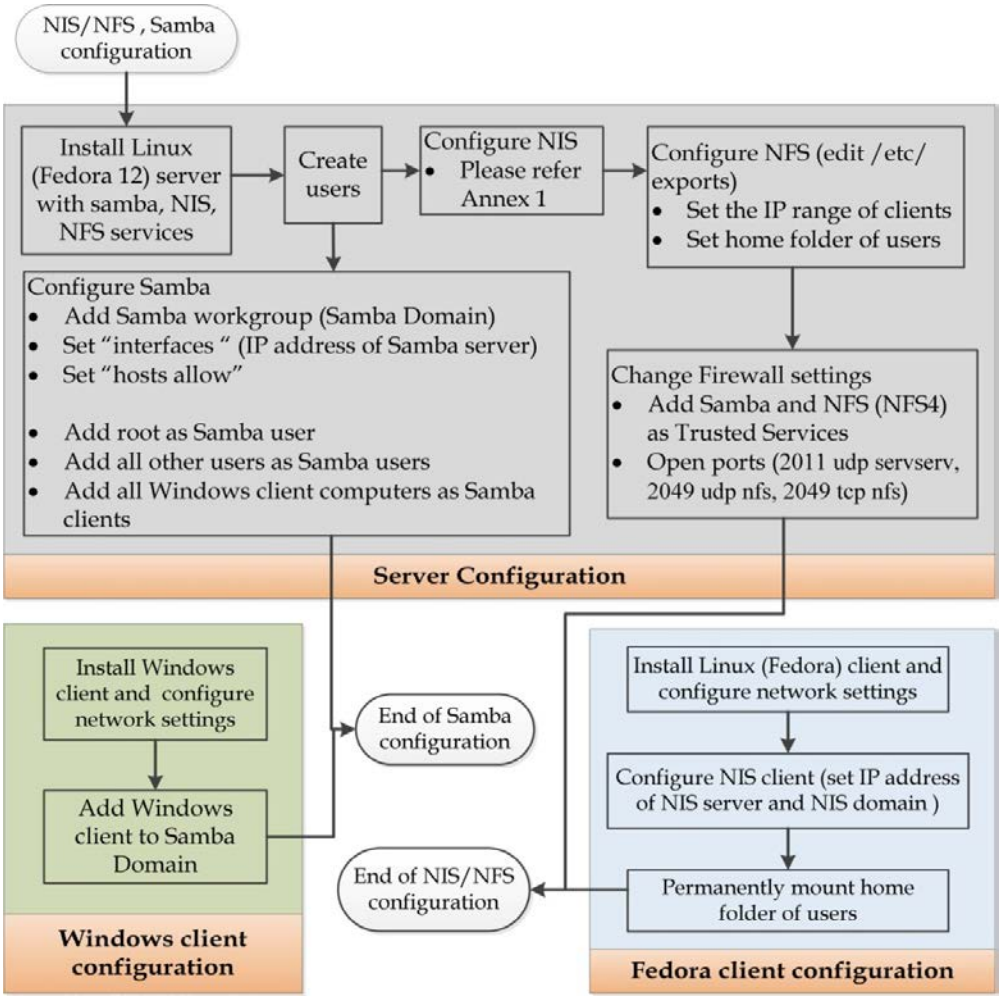

Figure 3. Configuration steps of the proposed system

Configuration of Linux server

We installed the Linux server including SAMBA, NIS, and NFS as file server services in addition to the default installation. The next step, adding users, can be done manually or using a script if the number of users is high. In our case, we used a script because we had to add 750 users in five user groups; there are 150 students per batch and five batches in one academic year. Each user was assigned a unique user folder in different parent home folders depending on their batch (there are 5 parent home folders in each batch). The SAMBA service was configured by editing "smb.conf" file which is normally located in /etc/samba/ folder, including SAMBA Workgroup and IP address under "Workgroup" and "Interface" respectively. The required configuration of IP address range of clients was

done under “host allow”. After configuration of SAMBA service, we added the “root” as SAMBA user and all users re-added as SAMBA users. As the last step of SAMBA service configuration, Windows clients were also added as SAMBA clients using a script (750 users and 40 Windows clients). Then we configured NIS service by editing following configuration files and set configurations are furnished in Annex I.

/etc/sysconfig/network
/etc/yp.conf
/etc/nsswitch.conf
/etc/ypserv.conf
/var/yp/securenets
/var/yp/nicknames

After configuration of NIS service, we followed the same steps to configure the NFS service as well. As the final step of the Linux server configuration, it is important to define security permissions for all services. For that, we configured the firewall and added SAMBA and NFS as trusted services. Further, the ports for udp servserv, udp nfs and tcp nfs protocol-service combinations should be opened to ensure user access to the server.

Configuration of Windows and Linux clients

In this step, it is necessary to configure the Linux clients as NIS clients and add the home folder of users permanently by editing “fstab” file. The Windows clients are configured by connecting all Windows clients to the network and adding to the SAMBA. Specifications of configured client and server machines are furnished in Table 1 and Table 2.

Table 1: Specification for the server computer

Specification for the server computer	
Brand / Model	HP / Proliant ML150 G5
Processor	Intel Xeon, 4 x 2.33 GHz
Hard Disk	700 GB, SATA
RAID Level	RAID 5
RAM	4 GB, DDR 2
Linux Kernel	FC10 2.6.27.5-117.fc10.i686
Hosted Service	SAMBA, NIS, NFS

Table 2: Specification for the client computers

Brand/Model	Hardware configurations	Number of PCs
Dell/Optiplex G1	Intel Pentium 2, 348 MHz, 128 MB	4
Dell/Optiplex 150L	Intel Pentium 3, 930 MHz, 128 MB	6
Dell/Optiplex 160L	Intel Pentium 4, 2.66 GHz, 256 MB	10
HP Compaq/dx2100	Intel Pentium 4, 2.99 GHz, 512 MB	4
Assembled/-	Intel Pentium 4, 2.81 GHz, 256 MB	16
HP Compaq/dc5800	Intel Core 2 Duo, 2.93 GHz, 1 GB	30*
HP Compaq/dc5800mt	Intel Core 2 Duo, 2.93 GHz, 1 GB	30*

* The client computers installed at the second phase (after two years)

Results and Discussion

The proposed model was officially deployed in the Computer Unit of Faculty of Agriculture, University of Ruhuna, Sri Lanka in January 2009. We observed system performances for the four year period in terms of system stability, system management, run-time error freeness and user friendliness. In the initial stage, we installed 40 computers with different hardware configurations and with dual operating system environment (Table 2). The new system was easy to manage, error free and few of complaints of the students. Thus, after two years, another 60 computers (Table 2) were added to the system in the year 2011 for further experiments. Throughout the entire period, we observed the student's logbook for complaining and compared with complaints before and after the deployment of the proposed system for performance analysis. We analyzed login issues, password problems and data safety in order to assess the performance of the new system in terms of system stability and run-time error freeness. Number of student complaints about login errors, login delay issues and password problems were 23, 145 and 56 respectively before the new system deployment. These attributes were considerably reduced to 2, 37 and 8 respectively after the new system deployment. Examination periods and regular practical sessions are identified as peak hours of utilization of the system. Thus it is obvious that the utilization of bandwidth of the network and the overhead of the server is remarkably higher than that of off-peak hours. However, with the new model deployment, a few errors are reported because of low bandwidth utilization of the network (user authentication and file server services). Although the password problems occur in both new and old systems due to human errors, the new system showed less because the students' profiles are timely and error freely updated. None of the data loss issues were reported in the new system while some were reported in the old system. This may happen due to inconsistent updating of the server; bandwidth utilization and/or server overloading problems. Further, we analyzed the system management performances estimating the average time taken to complete different management tasks. The time taken to both add new user and to change a user profile was reduced from 15 min to 5 min with the new system deployment. The data backing up time and the time taken to remove expired user profiles were also decreased by 50% since the new system implemented on a single server. The new model reduces the data duplication due to the common home folder concept for both platforms and hence it saves storage and backup storage capacities. Therefore, it could be stated that the new model reduces or eliminates data redundancy and data inconsistency. Moreover the new model ensures easy configuration and sharing of the common resources such as printers, scanners and shared folder among the both platforms. These results clearly indicate that the developed

model is consistent and the overall system performance is increased significantly.

Conclusion

A standalone SAMBA-NIS/NFS server model for Windows and Linux dual boot clients with individual user authentication is successfully implemented and has been used for more than four years without errors. We were able to design and implement the model using two techniques (SAMBA and NIS/NFS) in one server while most other organizations use in separate servers. The significant result of the model is straightforward, run-time error free and timely server administration. The model eliminates most of double work tasks such as basic administrative functions and data backing up services as we manage earlier in two servers. All common resources which facilitate the users can be accommodated in a single server avoiding the overhead and likelihood of error of maintaining two servers. Further, we could monitor user activities easily with one server concept other than checking log files in both servers separately. Finally we can conclude that this model could be implemented as a student file management system in one server in schools or university computer laboratories. Moreover, this system is economically more suitable for small scale network systems.

Acknowledgement

The authors wish to thank all students who support to evaluate the system and temporary staff members in the computer laboratory of Faculty of Agriculture, University of Ruhuna. This work is supported by the UGC IT fund (2008), University Grants Commission, Sri Lanka for enhancing students' information and communication technology.

References:

Eisler, M., Labiaga, R., Stern, H., 2001. Managing NFS and NIS. O'Reilly & Associates.

Jay Ts, Robert Eckstein, David Collier-Brown, 2003. Using Samba, 2nd ed. O'Reilly & Associates.

Kerema, E. and Aydin, K., 2011. The problems of public accessed computer laboratories and a suggestion for these problems' solution. Procedia Computer Science 3, pp.1520-1526.

Novell Administration Guide, 2013. [online] available at <http://www.novell.com/developer/develop_to_edirectory.html> [Accessed 20 June 2013].

Petersen, R. L., 2004. Red Hat: The Complete Reference Enterprise Linux & Fedora Edition: The Complete Reference. McGraw-Hill.

Rajagopal, R., 1999. Multi-Operating System Networking: Living with UNIX, NetWare, and NT. Auerbach Publications.
Simmons, S., 2001. Active directory bible. Poster: IDG Books Worldwide.
Sander Van Vugt, 2009. Configuring a File Server. Beginning the Linux Command Line, Apress: 277-297.

Annex 1
The configuration steps of NIS server
Requires packages

ypbind - RPC port binding service
portmap - RPC port mapping
ypserv - NIS server daemons
yp-tools - NIS support commands (*ypcat, yppasswd, ypwhich, etc*.)

Step 1:
Edit/etc/sysconfig/network
NETWORKING=*yes*
HOSTNAME=*hostname-of-this-nis-server* ## eg:- comXXX
NISDOMAIN=*name-of-domain* ## name-of-domain

Step 2:
Edit/etc/yp.conf
domain name-of-domain server 127.0.0.1 (domain **** server 127.0.0.1)
*Where 127.0.0.1 is the localhost IP address of the NIS server

Step 3:
Edit/etc/nsswitch.conf
passwd: *files nis*
shadow: *files nis*
group: *files nis*

Step 4:
Edit/etc/ypserv.conf
dns: *no*
files: *30*
slp: *no*
slp_timeout: *3600*
xfr_check_port: *yes*
* : * : shadow.byname : *port*
* : * : passwd.adjunct.byname : *port*

Step 5:
Edit/var/yp/securenets
host 127.0.0.1
255.255.255.0 X.X.X.0 (Eg:- 255.255.255.0 10.50.10.0)
*If the "securenets" file is not available in "/var/yp/", create the file "securenets" in "/var/yp/"

Step 6:
Edit/var/yp/nicknames
This is the default from the initial RPM installation and does not require any change for most configurations.
passwdpasswd.byname

group	*group.byname*
networks	*networks.byaddr*
hosts	*hosts.byname*
protocols	*protocols.bynumber*
services	*services.byname*
aliases	*mail.aliases*
ethers	*ethers.byname*

Step 7:
Execute these Commands (As the root)
 # nisdomainnamename-of-domain
 # servicerpcbind restart
 # serviceyppasswdd start
 # serviceypserv start
#/usr/lib/yp/ypinit -m
 # make -c /var/yp
 # serviceypbindstart
 # rpcinfo -u localhostypbind