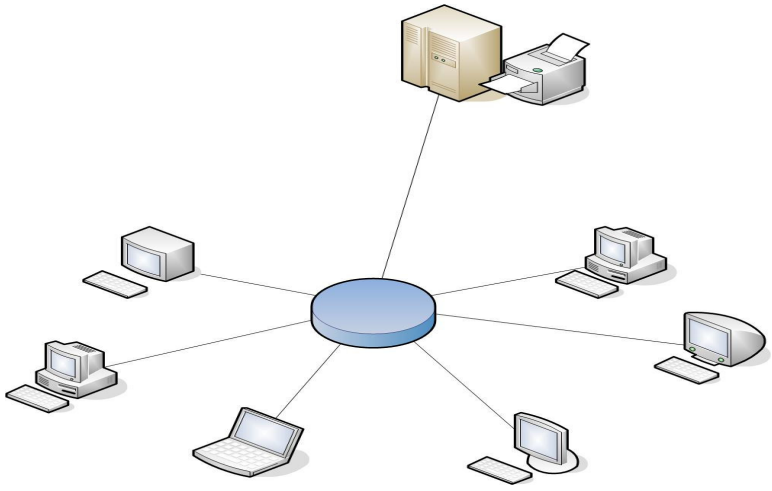
Distributed File Systems

Bansari Patel

11/26/2004

Contents

Introduction



A distributed file system stores files on one or more computers called servers, and makes them accessible to other computers called clients, where they appear as local files. There are several advantages to using file servers: the files are more widely available since many computers can access the servers, and sharing the files from a single location is easier than distributing copies of files to individual clients. Backups and safety of the information are easier to arrange since only the servers need to be backed up. The servers can provide large storage space, which might be costly or impractical to supply to every client. File sharing makes it quick and easy to transfer data from one system to another and avoids the confusion that results when everyone has their own - possibly out of date or inconsistent - copy of important data files they could not otherwise access. The distributed file system is useful when one needs to share documents or application software. In both cases system administration becomes easier

On the other hand, there are many problems facing the design of a good distributed file system. Transporting many files over the net can easily create sluggish performance, network bottlenecks and server overload can result. The security of data is another important issue which includes client authorization and security of the data on the network. Two further problems facing the design are related to failures. Often client computers are more reliable than the network connecting them and network failures can render a client useless. Similarly a server failure can be very unpleasant, since it can disable all clients from accessing crucial information.

File sharing lets your computer access files stored on another computer same as printer sharing lets your computer access a printer attached to another computer. Available since version 3.11 of Microsoft Windows, printer and file sharing are two of Window's most useful features. For example, if each computer in a large office has a laser printer, it would be quite expensive. However, printer sharing reduces the cost of providing every user with printing capability. With printer sharing, each computer system in the office can print to a single printer.

Samba Services

To provide printer and file sharing, Microsoft Windows uses a facility known as SMB (Server Message Block). This same facility is sometimes known as NetBIOS or LanManager. Linux systems provide support for SMB via a package known as Samba. Samba is a suite of Unix applications that speak the SMB (Server Message Block) protocol. By supporting this protocol, Samba allows Unix servers to get in on the action, communicating with the same networking protocol as Microsoft Windows products. Like SMB, Samba lets you:

- Share printers and files among Microsoft Windows, OS/2, Netware, and Unix systems
- Establish a simple nameserver for identifying systems on your local area network
- Backup PC files to a Linux system and restore them
- Administer users and passwords

Samba is the brainchild of Andrew Tridgell, who currently heads the Samba development team from his home of Canberra, Australia. The project was born in 1991 when Andrew created a fileserver program for his local network that supported an odd DEC protocol from Digital Pathworks. Although he didn't know it at the time, that protocol later turned out to be SMB. A few years later, he expanded upon his custom-made SMB server and began distributing it as a product on the Internet under the name SMB Server which is known as “Samba”.

The Samba suite revolves around a pair of Unix daemons that provide shared resources to SMB clients on the network. These daemons are:

- Smbd – This daemon allows file and printer sharing on an SMB network and provides authentication and authorization for SMB clients.
- Nmbd – This daemon looks after the Windows Internet Name Service (WINS), and assists with browsing.

Samba is currently maintained and extended by a group of volunteers under the active supervision of Andrew Tridgell. Like the Linux operating system, Samba is considered *Open Source software* (OSS) by its authors, and is distributed under the GNU General Public License (GPL). Since its inception, development of Samba has been sponsored in part by the Australian National University, where Andrew Tridgell earned his Ph.D. In addition, some development has been sponsored by independent vendors such as Whistle and SGI. It is a true testament to Samba that both commercial and non-commercial entities are prepared to spend money to support an Open Source effort. Samba has proven its reliability and high performance in many organizations. According to the online survey at http://www.samba.org/pub/samba/survey/ssstats.html, Bank of America is using Samba in a configuration that includes about 15,000 clients, and Hewlett-Packard is using Samba in a configuration that includes about 7,000 clients.

Samba Installation

Samba includes the smbd program, which runs as a daemon, several utility programs, man pages and other documentation, and the configuration file: */etc/samba/smb.conf*. Generally, installation and configuration of the samba is simple.

In order to use Samba your machines must be on a single ethernet LAN segment using the TCP/IP protocol. Samba will not work using other network protocols. This is generally easy since Linux and Windows 95/98/NT/XP ship with TCP/IP support. However, if you are using Windows 3.X machines TCP/IP support will need to be added. Set the network properties such as workgroup, computer name, file and printer share options and others on windows machines. Generally, Linux distribution will already come with an installable package containing a recent version of Samba.

The two daemons smbd (Samba daemon) and nmbd (provides NetBios nameserver support to clients) are required for the Samba package. They are typically installed in /usr/sbin and run either on boot from the systems startup scripts or from inetd.

The name service provided by the nmbd daemon is different from the name service provided by the Domain Name Service (DNS). NetBIOS name service is a 'Windows-style' name service used for SMB. In other words, having DNS name service tells you nothing about the state of the ability for Samba to resolve host names. If the Samba package is not available with your distribution, simply retrieve the source from internet, and read the file README in the distribution. Installation places the daemons in /usr/sbin and the binaries in /usr/bin, and installs the man pages in /usr/local/man. To install the configuration file, smb.conf, go to the directory where Samba was built. Look in the subdirectory examples/simple and read the file README. Copy the file smb.conf found in that directory to /etc. If you have a Linux distribution that already has Samba installed, you may already have a Samba configuration file in /etc or /etc/samba, then you can start with that one. Samba distribution comes with a small set of Unix command-line tools which are some Samba binaries installed in /usr/bin or /usr/local/samba/bin. Some of these are as shown below:

- smbclient: A FTP-like Unix client that can be used to connect to Samba shares (a SMB client for UNIX machines)
- smbprint: It is a script to print to a printer on an SMB host
- smbstatus: It is a program that lists the current SMB connections for the local host to the share on a Samba server
- smbtar: It is a program to back up data in shares. (similar to Unix tar)
- smbpasswd: It is a program that allows an administrator to change the encrypted passwords used by Samba.
- nmblookup: It is a program that provides NetBIOS over TCP/IP name lookups.
- testparm: It is a program to validate Samba configuration file.
- Testprns: It is a program that tests whether various printers are recognized by the smbd daemon.

Samba Configuration

Configure Samba using GUI Tool

Samba includes a tool called swat (Samba Web Administration Tool) that lets you view and change options of smb.conf file by using Web browser, which is generally much easier than using a text editor. The swat tool verifies the values of parameters you enter and provides online help. Swat is run from inetd. To use the swat, the following changes have to be made:

- /etc/services should include the following line:
  - swat 901/tcp
- /etc/inetd.conf should include the following line:
  - swat stream tcp nowait.400 root /usr/sbin/swat swat.

  After editing these files, send HUP signal to inetd daemon.

One can secure the swat by editing /etc/inetd.conf, using TCP wrappers and /etc/hosts.allow as shown below:

- swat stream tcp nowait.400 root /usr/sbin/tcpd /usr/sbin/swat in /etc/inetd.conf
- swat: all the IP addresses which are desired to allow in /etc/hosts.allow

Edit /etc/hosts.deny as deny for ALL. After editing these files, send HUP signal to inetd daemon. One can secure SWAT with SSL too. The detail is given at http://info.ccone.at/INFO/Samba/SWAT.html To access swat, point your browser to port 901 of your system. For example, you can use the URL http://localhost:901/. Your web browser will prompt you for a userid and password; specify root as the userid and give the appropriate password. The swat configuration page has help links to all the configurable options in the smb.conf file allowing an administrator to easily look up the effects of any change.

Using swat, to configure your Samba server, click on tool bar entries which are as shown below.

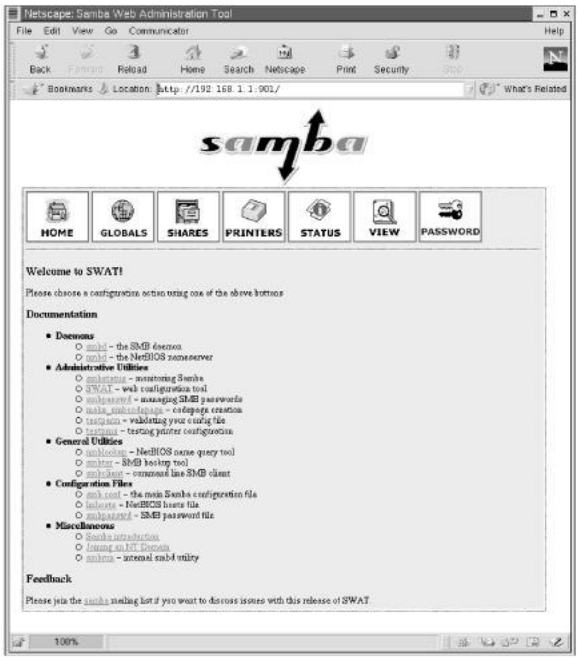


Figure 1 The SWAT Configuration Tool

- Globals lets you configure global Samba variables (options)

  By clicking on the Globals button, one can set the following options.

  Workgroup: The workgroup name displayed when the server is queried by a client.

  Netbios name: The name by which the server is known to the NetBIOS nameserver.

  Interfaces: It specifies the IP address of the interface or the IP addresses of the interfaces through which Samba should listen. Each IP address is followed by a forward slash and a number that specifies the number of bits that pertain to the network portion of the IP address (usually 24). If this option is not set, Samba attempts to locate and automatically configure a primary interface.

  Security: It specifies how Samba authenticates requests for access to shared resources. The default value, user, is helpful when the Samba server and its clients have many common userids. The value share can be useful when few common userids exist. The value system lets another SMB server perform authentication on behalf of the server. You should generally use the default value. One can see the Samba documentation for details.

Encrypt Passwords: It specifies whether Samba negotiates encrypted passwords that are expected by Windows NT 4 SP3 and Windows 98.

Update encrypted: It allows automatic updating of an encrypted password when a user logs on using a non-encrypted password. This option is useful when migrating to encrypted passwords and should otherwise be set off.

Map to guest: It specifies Samba's action when a user attempts to log with invalid password. The Bad User option is generally appropriate.

Guest account: The Linux account used to provide services for guest users.

Hosts allow: It specifies a list of hosts that can access the server. If not specified, all hosts are permitted the access.

Hosts deny: It specifies a list of hosts that can not access the server.

Log level: It specifies an integer that specifies the wordiness of log messages. A low value (such as 0) specifies that few messages are written to the log.

Log file: It specifies the name of Samba's log file.

Max log size: It specifies the size of the log file in kilobytes (kb). When the specified size is exceeded, Samba begins a new log file. A value of zero lets the log file grows indefinitely large.

Read prediction: It specifies whether Samba will attempt to pre-read data from files, in order to speed data transfer. It is used to tune the performance. This code is disabled in Samba 2.0.

Socket options: It specifies TCP options that can improve performance. The user can set the same option on the command line using –o option. The details and the recommendations for this option is given in smb.conf manual pages. The correct options can increase the performance enormously, but the wrong options can degrade the performance as well. Generally, TCP_NODELAY option affects on most networks. Many people noticed that it doubles the read performance of a Samba drive, and the best explanation I have noticed is that the Microsoft TCP/IP stack is slow in sending TCP ACKs. On the other hand, the option SO_RCVBUF=8192 can degrade Samba performance on the loopback adapter (IP Address 127.0.0.1).

Printcap name: It specifies the name of the printcap file used by the server. The printcap file has valid printer share name.

Printing: It specifies how Samba interprets printer status information. Generally, SYSV is an appropriate choice for a Linux system.

Logon script: It specifies the path of a BAT file that is downloaded from the server and run when user logs on to Samba.

Domain logons: It specifies whether Samba will serve Windows 9x domain logons for its workgroup.

OS level: It specifies the level at which Samba advertises itself for browse elections. A high number makes it more likely that Samba will be selected as the browser. The value 65 will cause clients to prefer Samba to a Windows NT server.

Preferred master: It specifies whether the NetBIOS name server is the preferred master browser for its workgroup.

Local master: It specifies whether the NetBIOS name server will bid to become the local master browser on a subnet.

Domain master: It specifies collation of browse lists across a wide-area network (WAN). It may result in strange behavior when a workgroup includes a Windows NT Primary Domain Controller (PDC).

Wins server: It specifies the IP address of the WINS server with which the NetBIOS nameserver should register itself, if any.

Wins support: It specifies that the NetBIOS nameserver should act as a WINS server. Useful when the network includes several subnets. Do not specify this option for multiple systems of a single network.

Locking: It specifies whether the server will automatically lock files and check locks when files are accessed. Enabling this option may slow performance.

- Shares lets you configure file shares

  By clicking on the Shares button on swat's tool bar, one can set the following options.

  Comment: The description displayed when the file share is queried by a client.

Path: The path (directory or file) that is shared by the server.

Guest account: The Linux account used to provide services for guest users.

Read only: It specifies whether access to the share is read-only or not.

Create mask: The default mode assigned to a newly created file within a shared directory.

Guest ok: It specifies whether guest access (access without a password) is allowed.

Hosts allow: A list of hosts that can access the file share. If not specified, all hosts are permitted access.

Hosts deny: A list of hosts that cannot access the file share.

Browseable: It specifies whether the file share is visible in the list of shares made available by the server.

Strict locking: It specifies whether the server will automatically lock files and check locks when files are accessed. Enabling this option may slow performance.

Available: It specifies whether the share is available; by setting this option to "no" you can prevent access to the share.

Volume: The volume label returned for the share.

- Printers lets you configure shared printers

  By clicking on the Printers button on swat's tool bar, one can set the following options.

  Comment: The description displayed when the printer share is queried by a client.

  Path: The print spooling directory.

  Guest account: The Linux account used to provide services for guest users.

  Guest ok: It specifies whether guest access (access without a password) is allowed.

  Hosts allow: A list of hosts that can access the printer share. If not specified, all hosts are permitted access.

Hosts deny: A list of hosts that cannot access the printer share.

Print ok: It specifies whether printing is permitted. If this option is set to "no," clients may still be able to browse the printer share.

Printing: It specifies the type of printer interface used, which determines what commands Samba issues to control the printer. "BSD" is generally a good choice.

Printer name: It specifies the name of the printer to which the printer share corresponds; "lp" is generally a good choice.

Browseable: It specifies whether the printer share is visible in the list of shares made available by the server.

Available: It specifies whether the printer share is available; by setting this option to "no" you can prevent access to the printer share.

- Status lets you view the status of the Samba server

The Status button on swat's tool bar shows the status of the server daemons (smbd and nmbd) and the version of Samba, active connections, active file and printer shares and open files. Using the controls on the page, one can refresh the page contents, set the auto refresh interval, start and stop smbd or nmbd daemons, or kill an active connection.

- View lets you view the *smb.conf* file

The View button on swat's tool bar lets you view the Samba server's main configuration file, */etc/smb.conf*. By default, the page shows only the basic configuration options, but clicking on Full View causes swat to display every configuration option.

- Password lets you add and delete users and change user passwords

One can create userids for accessing Samba resources by clicking on swat's Password tool bar button.

One can create or delete userid, change the password associated with a userid or enable or disable userid. These userids are the userids which Samba server recognizes as authorized to access its resources. In addition, one can change the password associated with a userid on a remote system running Samba. It is easier way to change password than logging in to the remote host and using its password change facility.

Configure Samba Manually

One can edit the smb.conf using editor like vi or vim. Some options of my /etc/samba/smb.conf file are as shown below. smb.conf has many examples as commented text which can help to set all the options too.

```
[global]
# workgroup = NT-Domain-Name or Workgroup-Name
        workgroup = basement

# server string is the equivalent of the NT Description field
        server string = samba server

# if you want to automatically load your printer list rather
# than setting them up individually then you'll need this
        printcap name = /etc/printcap
        load printers = yes

# It should not be necessary to spell out the print system type unless
# yours is non-standard. Currently supported print systems include:
# bsd, sysv, plp, lprng, aix, hpux, qnx, cups
        printing = cups

# this tells Samba to use a separate log file for each machine
# that connects
        log file = /var/log/samba/%m.log

# Put a capping on the size of the log files (in Kb).
        max log size = 0

# You may wish to use password encryption. Please read
# ENCRYPTION.txt, Win95.txt and WinNT.txt in the Samba documentation.
# Do not enable this option unless you have read those documents
        smb passwd file = /etc/samba/smbpasswd

# The following are needed to allow password changing from Windows to
# update the Linux system password also.
# NOTE: Use these with 'encrypt passwords' and 'smb passwd file' above.
# NOTE2: You do NOT need these to allow workstations to change only
#       the encrypted SMB passwords. They allow the Unix password
#       to be kept in sync with the SMB password.
          unix password sync = Yes
          passwd program = /usr/bin/passwd %u
          passwd chat = *New*password* %n\n *Retype*new*password* %n\n
*passwd:*all*authentication*tokens*updated*successfully*
```

```
# You can use PAM's password change control flag for Samba. If
# enabled, then PAM will be used for password changes when requested
# by an SMB client instead of the program listed in passwd program.
# It should be possible to enable this without changing your passwd
# chat parameter for most setups.
        pam password change = yes

# This parameter will control whether or not Samba should obey PAM's
# account and session management directives. The default behavior is
# to use PAM for clear text authentication only and to ignore any
# account or session management. Note that Samba always ignores PAM
# for authentication in the case of encrypt passwords = yes
        obey pam restrictions = yes

# Most people will find that this option gives better performance.
# See speed.txt and the manual pages for details
        socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192

# DNS Proxy - tells Samba whether or not to try to resolve NetBIOS names
# via DNS nslookups. The built-in default for versions 1.9.17 is yes,
# this has been changed in version 1.9.18 to no.
        security = SHARE
        encrypt passwords = yes
        guest ok = yes
        guest account = mitesh
        dns proxy = no

#============================ Share Definitions ==================
[homes]
        comment = Home Directories
        browseable = no
        writeable = yes
        valid users = %S
        create mode = 0664
        directory mode = 0775

# NOTE: If you have a BSD-style print system there is no need to
# specifically define each individual printer
[printers]
        comment = All Printers
        path = /var/spool/samba
        browseable = no
# Set public = yes to allow user 'guest account' to print
        printable = yes

# A publicly accessible directory, read/write to all users. Note that all files
```

```
# created in the directory by users will be owned by the default user, so
# any user with access can delete any other user's files. Obviously this
# directory must be writable by the default user. Another user could of course
# be specified, in which case all files would be owned by that user instead.
[public]
   path = /home/public
   public = yes
   only guest = yes
   writable = yes
   printable = no

# The following two entries demonstrate how to share a directory so that two
# users can place files there that will be owned by the specific users. In this
# setup, the directory should be writable by both users and should have the
# sticky bit set on it to prevent abuse. Obviously this could be extended to
# as many users as required.
;[myshare]
;   comment = Mary's and Fred's stuff
;   path = /usr/somewhere/shared
;   valid users = mary fred
;   public = no
;   writable = yes
;   printable = no
;   create mask = 0765
[bansari]
          path = /home/bansari
          writeable = yes
          guest ok = yes
[public]
          path = /home/public
          writeable = yes
          guest ok = yes
```

Start Samba

One can start samba using SWAT or from command line.

Using GUI Tool

If you use SWAT, go to the status page and start smbd and nmbd daemons by clicking on start button. If it was already started, click on stop button and restart them to read the new configuration file. On Linux, similar functionality can also achieve by clicking on start menu (hat symbol) -> System Settings -> Server Settings -> Services, and then right click on smb to start, stop or restart smbd and nmbd daemons.

Using Commands from command line

To start from the command line use the following commands:

- /etc/rc.d/init.d/smb stop
- /etc/rc.d/init.d/smb start

Stand-alone Daemons

To run the Samba processes as stand-alone daemons, one needs to add the commands listed in the previous section to your standard Unix startup scripts. This varies depending on whether you have a BSD-style Unix system or a System V Unix.

BSD UNIX:

With a BSD-style Unix, one needs to append the following code to the *rc.local* file, which is typically found in the */etc* or */etc/rc.d* directories:

```
if [ -x /usr/local/samba/bin/smbd]; then
        echo "Starting smbd..."
        /usr/local/samba/bin/smbd -D
        echo "Starting nmbd..."
        /usr/local/samba/bin/nmbd -D
fi
```

This code checks to see if the *smbd* file has execute permissions on it, and if it does, it starts up each of the Samba daemons on system boot.

System V UNIX:

With System V, System V typically uses scripts to start and stop daemons on the system. Hence, one needs to instruct Samba how to operate when it starts and when it

stops. One can modify the contents of the */etc/rc.local* directory and add something similar to the following program entitled *smb*:

```
#!/bin/sh

# contains the "killproc" function on Red Hat Linux
./etc/rc.d/init.d/functions

PATH="/usr/local/samba/bin:$PATH"

case $1 in
        'start')
                echo "Starting smbd..."
                smbd -D
                echo "Starting nmbd..."
                nmbd -D
                ;;
        'stop')
                echo "Stopping smbd and nmbd..."
                killproc smbd
                killproc nmbd
                rm -f /usr/local/samba/var/locks/smbd.pid
                rm -f /usr/local/samba/var/locks/nmbd.pid
                ;;
        *)
                echo "usage: smb {start|stop}"
                ;;
Esac
```

With this script, you can start and stop the SMB service with the following commands:

```
# /etc/rc.local/smb start
Starting smbd...
Starting nmbd...
# /etc/rc.local/smb stop
Stopping smbd and nmbd...
```

Test Samba

After started the samba server, one can test it using the following command:

- smbclient –L localhost

It will ask for the password, so just hit enter. If one sees the output of the smbclient command something as showm below, it means that it works.

```
bansari@utah shared]$ smbclient -L localhost

added interface ip=192.168.2.103 bcast=192.168.2.255 nmask=255.255.255.0
Password:

Domain=[BASEMENT] OS=[Unix] Server=[Samba 2.2.7a]
        Sharename       Type      Comment
        ---------               ----      -------
        public          Disk
        bansari         Disk
        IPC$            IPC       IPC Service (samba server)
        ADMIN$          Disk      IPC Service (samba server)
        printer         Printer
        samsung         Printer

        Server               Comment
        ---------            -------
        LAPTOP               Panasonic Laptop
        UTAH                 samba server

        Workgroup            Master
        ---------            -------
        BASEMENT             LAPTOP
```

One can check /etc/hosts to see if all the hosts are available or not. The file, /etc/hosts, maps host names to IP addresses. My /etc/hosts looks like:

```
127.0.0.1               utah    localhost.localdomain           localhost
192.168.2.102           laptop
192.168.2.109           goa
```

SMBFS

The smbfs package is not actually a part of Samba, but it comes with newer Samba distribution. It has two programs: smbmount and smbumount. These commands are useful to mount and unmount remote SMB share locally. One can access windows directories or files into Linux machine using smbmount command as shown below.

- smbmount //LAPTOP/SHARE ./shared –o username=bansari
- or smbmont //192.168.2.102/SHARE ./shared –o username=Bansari

To unmount user the following command:

- smbumount ./shared

The folder SHARE on LAPTOP needs to be shared too. In addition, one can type smb://LAPTOP in the browser to achieve the same functionality. Detail of smbmount can be found in manual pages. For more information, check the man pages for smbmount, smbumount and mount.

Coda Distributed File System

Coda is an advanced networked file system. It has been developed at CMU (Carnegie Mellon University in Pittsburgh, PA) since 1987. Distributed file systems have several security problems and consistency problems due to file sharing. The Coda has tried to solve these problems and implemented them as a research prototype.

Coda has disconnected operation for mobile computing, high performance through client side persistent caching, server replication, security model for authentication, encryption and access control, network bandwidth adaption, good scalability, and well defined semantics of sharing. In addition, it is free. Coda was originally implemented on Mach 2.6 and has ported to Linux, NetBSD and FreeBSD now. The group who works on it is trying to port it on windows now. The group is trying to make it more robust. However, it can become a popular and freely available distributed file system.

References

1. Evi Nemeth, Garth Snyder, Scott Seebass and Trent Hein, "Unix System Administration Handbook", Pearson, 2003.
2. AEleen Frisch, "Essential System Administration", O'Reilly, 3rd Edition, 2003.
3. Thomas Limoncelli and Christine Hogan, "The Practice of System and Network Administration", Pearson, 2003.
4. Robert Eckstein, David Collier – Brown and Peter Kelly, "Using Samba", O'Reilly, 1st Edition.
5. Coda File System, http://www.coda.cs.cmu.edu/
6. Samba Organization, http://www.samba.org/
7. Secure SWAT, http://info.ccone.at/INFO/Samba/SWAT.html
8. SWAT, http://www.sourcekeg.co.uk/samba/docs/man/swat.8.html
9. SWAT, http://techrepublic.com.com/5100-6261_11-1035709.html
10. SWAT, http://www.linuxsoft.cz/en/sw_detail.php?id_item=3831