NFS Networking

Sharing Data

Once you have your network set-up you then have to decide how best to move your data between the computers on it. After all, if the data just stays on the one machine there is little point in having the network in the first place.

BY COLIN MURPHY

Much of what you decide to do to achieve this depends very much on the type of data you need to move, the size, and the regularity with which you want to move it. If you are only going to move 700KB of data once a week you may find that copying your information to a floppy might be a time effective and cheap solution, making the network redundant. On the other hand, you might need to send several MBytes of data once a day to a remote machine, so running a ftp client/server might suit your needs. You might want to be able to have regular access to the data on a local machine and it is here that NFS will be able to help the most.

Making the right choice is important, I know of one case where the user had been moving files from machines using ssh and scp, far from seamless and real overkill for moving files over an internal network.

The **N**etwork **F**ile **S**ystem will allow you to share files between machines that have NFS running. In practice, once it is configured, this sharing of files is seamless to the user, in much the same way that getting access to files from a CD is seamless.

The only prerequisite is that you have a working network and you know the IP addresses of the machine on the network, though it is nice if you also know the nameserver details of each machine, something that will only happen if there are sensible details in the */etc/hosts* file.

```
col@mine:>ping 192.168.1.1
64 bytes from 192.168.1.1: ➥
icmp_seq=1 ttl=64 time=0.353 ms
```

whereas

```
col@mine:>ping server
```

might just produce a *ping: unknown host server* if hosts are not set up

NFS is a client/server based system so this means you will have one machine running the server, from which you will share your data, with another machine to run the client, which will receive this shared data. Some distributions provide wizards and I will run through an example based on SuSE Linux 8.1.

Creating Shares

When you come to setting up NFS you will be asked for details of the directory that you want to share with the other machines and details of how those other machines will refer to the shared data, something best explained by example.

In the top *Directories* section of this menu you get to select the directories on the server machine that you want to share with your NFS clients. This can be anything on the machine, it could be the entire /home/ directory or a directory that you have created especially, as in this example. You can make more than one directory available for sharing, with each having different properties.

The lower part of the window allows you to set those properties and it is that configuration that can been seen in the flying window in Figure 1. Here you get the chance to configure which of the hosts running clients can get access to this particular share.

As shown, it is set to the defaults, so any of the hosts on your local network will be able to get access to this share. The options have set the share so that it is **r**ead **o**nly, **root_squash** and **sync** limit the amount of damage that might be done should the root user access the NFS share or if the machine crashes. To learn more about the options that be placed here you should refer to *man exports*.

Now that we have something to connect to we will need to configure a client. In SuSE Linux 8.1 this is done in the *Configuration of the NFS client* menu. The top entry in the configuration menu asks for details of the NFS server that we have just configured, hitting the 'Choose' button should bring up a list of available servers after a search. I must admit that this didn't work for me, but putting the details in manually worked fine, so if 'Choose' doesn't work for you don't get disheartened.

The 'Remote filesystem' is the name of the partition to be shared, remember that the server on '192.168.1.1' might be prepared to share more than one directory.

Mounting Files

'Mountpoint' is the name with which you want to refer to the share on the local machine. This is analogous to the 'Mount Point' for a CD drive, the drive itself might be known to the system as /dev/cdrom but the files on the CD would be 'mounted' onto something more like '/media/cdrom/' and it is this mount point that you would enter into a file browser. As it is with the mount point '/StuffBeingShared/', the directory name that you would browse into to view the files being shared.

The 'defaults' option in the 'Options' section allow the user mounting this share to enjoy delights such as the ability to have the share automatically mounted when logging in, to execute binaries and to read and write files, though the NFS server has final say on what this user can do, so in this example the user will only be able to read files. More information on what is available here can be found in *man 8 mount* and *man 5 nfs*.

Once you have finished with YaST configuring the client, you are ready to access that directory. Mount it and play:

```
col@mine:>➘
mount /StuffBeingShared/
col@mine:>cd /StuffBeingShared/
col@mine:>ls -l
```

which will give you a list of of the files being shared. You can treat these files in exactly the same way that you would treat files held on a hard drive in the local machine, you can search, view, and copy them. Should you have set the NFS server with the ‘rw’ option you would even be able to create and edit files. NFS working. Simple.

What Permissions?

Files still have file permissions and users accessing those files will need to have the correct permission level for access. UID and GID descriptors are used to check on those file permissions, so it is important that these details match on each machine.

If *User1* has a UID of 501 on one machine and *User2* has the same UID on the machine acting as the NFS server then *User1* will have access to *User2*‘s files. It is not too much of a chore on a small network to make sure users and UIDs match, for a larger network you will need to call on the help of something like NIS.

Well, that was simple as long as you have the wizard tools at hand. Should you not find yourself in this position then all is far from lost, configuring nfs manually is quite straightforward, hopefully more so now that you can see what needs to be done in practice.

The NFS system has been part of the kernel since 2.2.19, should your kernel be earlier than this you will need to apply patches. Hopefully you will be running with a recent kernel into which NFS support has been compiled. You will also need to have several other services running, **portmap** and **mountd** being the most important after **nfsd**. You might also need **statd** and **lockd** for some other features.

What Can I Run on It?

The server relies on the existence of */etc/exports* to give details of the files and directories it is allowed to share to the wider world. The format is very simple:

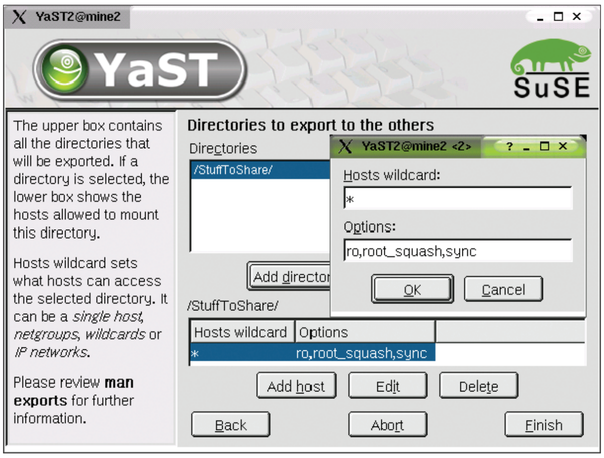
```
Col@server:>cat /etc/exports
/StuffToShare   *.local (rw)
/StuffToShare/NotToShare➘
 (noaccess)
```

gives us the equivalent of that set up in the GUI previously. Again, you can have multiple entries in this file, one line for each share. One useful addition to */etc/exports* is the *noaccess* option. With this you can specify a directory that might be a sub-directory of one of those that we are sharing which will be exempt.

A very important step, next, you need to run the command *exportfs -a* so that NFS is made aware of what is available for export.

On the client side, to confirm that all is well with the server you can run the following command and hopefully get something similar to:

```
Col@mine:>showmount -e server
Export list for server:
/StuffToShare *.local
```



The upper box contains all the directories that will be exported. If a directory is selected, the lower box shows the hosts allowed to mount this directory.

Hosts wildcard sets what hosts can access the selected directory. It can be a *single host*, *netgroups*, *wildcards* or *IP networks*.

Please review **man exports** for further information.

Directories to export to the others

Directories: /StuffToShare/

Hosts wildcard: *

Options: ro,root_squash,sync

/StuffToShare/

Hosts wildcard	Options
*	ro,root_squash,sync

Figure 1: In the process of setting up the NFS server

Happy with our server we are now left to mount the share.

```
Col@mine:>mount -o ro,bg,intr ➘
server:/StuffToShare ➘
/StuffBeingShared
```

Here the directory */StuffToShare* on *server:* will be mounted onto the local directory */StuffBeingShared*, assuming that the local directory has been created.

This is all well and good but a little clumsy. It would be better if the mount was done automatically when the system boots up, something that tends to happen with the hard disks on your system.

The file */etc/fstab* holds details about file systems and mount points and you can add your NFS shares here. The syntax of this file is described in detail in *man 5 fs* as well as *man 5 mount* but it lists the *filesystem*, *mountpoint*, *fstype* and some *flags* of your choosing, and so the line added should look something like:

```
server:/StuffToShare  ➘
/StuffBeingShared ➘
nfs ro,bg,intr 0 0
```

NFS makes for a convenient tool to transfer files across a network, so convenient that you really need to take into account some of the security implications involved. NFS should not be used in its naked form over external network connections, for this reason you should make sure that your firewall is blocking ports 2049 and 111, the latter being used by portmap.

You should make use of tcpwrappers and the */etc/hosts.allow* and */etc/hosts.deny* files to give portmap some extra protection. */etc/hosts.deny* should have the line *portmap:ALL* in it to stop open access. To allow those on our local network to get access, in */etc/hosts.allow* we need to include the line *portmap: 192.168.1.0/255.255.255.0* depending on how you have numbered your subnet.

There is much tinkering left to be done that could help to improve the throughput, but you have enough information to get NFS up and running. ■