

Le système GNU/Linux

By ShareVB

N.F.S.

NFS (abréviation de Network File System) est un produit développé par SUN (les premières versions datent de 1985) et est disponible en standard sur de nombreux systèmes UNIX. Il existe des versions NFS sur d'autres systèmes et même sous Windows XP. C'est un service TCP/IP basé sur le concept de RPC (Remote Procedure Call).

Table des matières

I.Principe.....	1
II.Fonctionnement.....	2
III.Mise en oeuvre sous Linux.....	4
a)Installation et démarrage.....	4
b)Le fichier de configuration.....	4
c)Rechargement de la liste des « export » NFS.....	5
d)Affichage des « export » NFS.....	5
e)Montage des « export » NFS.....	5
f)Précautions d'usage.....	6
g)Préliminaires aux Règles iptables.....	6
1.Généralités.....	7
2.Sous Debian.....	7
i.rpc.statd.....	7
ii.rpc.lockd.....	7
iii.rpc.mountd.....	7
iv.rpc.rquotad.....	7
3.Sous Fedora.....	7
h)Règles iptables.....	8
IV.Mise en oeuvre en client sous Windows.....	8
a)Prérequis.....	8
b)Procédure.....	8
V.Bibliographie.....	9

I. Principe

Le service NFS est un service serveur. On dit que le serveur "exporte" des disques ou une arborescence pour les clients NFS. Un client NFS, a besoin d'un produit client pour accéder aux serveurs. On dit qu'un client "monte" des disques exportés par une serveur. Pour le client, le fait de "monter" une arborescence exportée par une machine B, donne l'illusion aux utilisateurs de A que l'arborescence est un disque local. Un client peut également être serveur si le système d'exploitation le permet.

Il est à noter que NFS n'est pas sécurisé. En effet, NFS a une sécurité par hôte, c'est-à-dire que n'importe qui arrivant à se faire passer pour la machine autorisé (en changeant son IP) peut monter

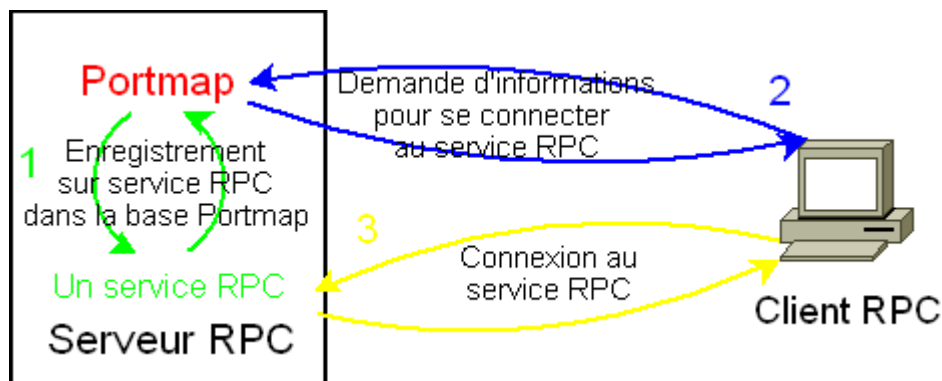
les « export » NFS sans mot de passe. Aucun mot de passe n'est requis, juste une IP et un UID/GID. De plus, une fois, monté, un « export » NFS, l'utilisateur se retrouve propriétaire des fichiers et dossiers ayant l'UID/GID de montage.

II. Fonctionnement

Le protocole NFS se compose en réalité de 5 protocoles qui tous, reposent sur les *Remote Procedure Calls* (RPC) et donc le démon `portmap` (aussi appelé `rpc.portmap`).

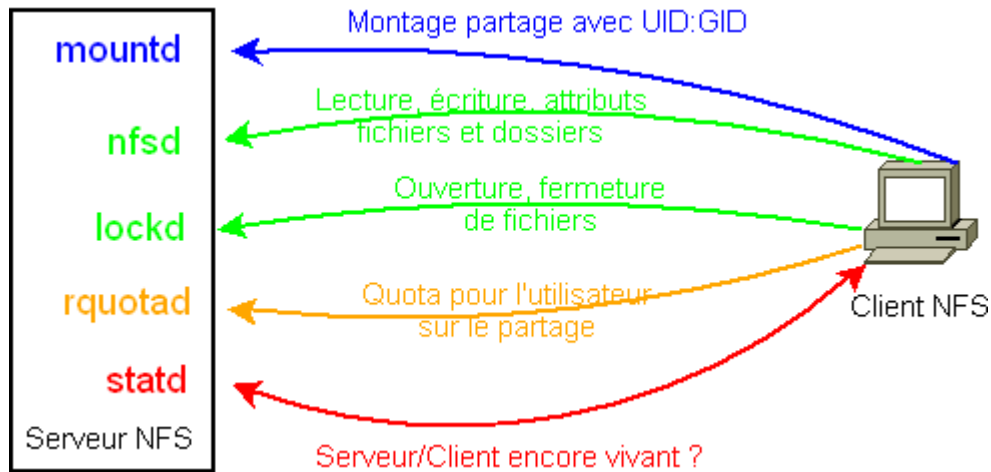
Le démon `portmap` a pour rôle de convertir les numéros de programmes RPC en numéros de ports :

- Quand un serveur RPC démarre, il va préciser à `portmap` quel port il utilisera et les numéros de programmes RPC qu'il gère.
- Quand un client souhaite envoyer une requête RPC vers un numéro de programme donné, il contacte d'abord le serveur `portmap` pour obtenir le numéro de port sur lequel tourne le programme souhaité. Ensuite, il adresse les paquets RPC au port correspondant.



Les 5 services permettant à NFS de fonctionner sont :

- **nfs** : géré par le démon `rpc.nfsd`, ce protocole est la base de NFS et permet la création de fichier, leur listing, leur lecture ou leur écriture. Ce protocole gère donc également les droits (rwx, owner, bits..) et les statistiques sur les fichiers.
- **mountd** : géré par le démon `rpc.mountd`, il permet le montage des systèmes exportés auxquels on accédera par **nfs**. Il envoie des requêtes de type `mount` et `umount` au serveur, qui, bien sûr, connaît la liste des dossiers qu'il exporte.
- **Nsm (Network Status Monitor)** : géré par le démon `rpc.statd`, il sert à surveiller les machines du réseau pour connaître leur état (cliente ou serveur) pour signaler, par exemple, qu'elle redémarre.
- **nlm (Network Lock Manager)** : géré par le démon `rpc.lockd`, assure l'intégrité et l'unicité de l'accès au fichiers pour éviter que des données soient altérées par plusieurs clients en même temps. Ce protocole gère un système de verrous (*lock* en Anglais) qui permettent de signaler les fichiers utilisés. Ainsi, à l'aide du protocole **nsm** qui sait quand un client redémarre, il libère tous les locks du client avant de les lui restituer si une nouvelle requête est émise.
- **quotad** : géré par `rpc.quotad`, il permet de gérer les quotas pour les utilisateurs ayant fait des montages.



Il existe actuellement 2 versions de NFS : NFSv2 et NFSv3.

NFS repose sur le concept *file handle* qui est une sorte de contexte de fichier. Il permet de repérer de façon unique un objet d'un système de fichier, c'est à dire les fichiers et dossiers de toutes sortes (régulier, fifo...). Il contient par exemple, l'inode du fichier, mais également un fichier représentant le device où se trouve ce fichier. On peut donc voir NFS comme un système de fichiers qui en encapsule un autre.

NFS se sert donc des fichiers suivants :

- `/etc/exports` : contient les directives d'exportations de dossiers. Ce fichier est lu par `exportfs` qui le complète avec les options par défaut et transcrit chaque ligne en plusieurs en fonction du nombre de noms de machines spécifié et inscrit dans `etab`.
- `/var/lib/nfs/etab` : ce fichier contient les exportations réelles en cours (toutes options par défaut explicitées). Ce fichier est initialisé par `exportfs`.
- `/var/lib/nfs/rmtab` : chaque ligne contient le nom d'un client et le système de fichiers qu'il a monté depuis ce serveur
- `/proc/fs/nfs/exports` fournit la liste des clients réels qui ont un montage NFS en cours sur le serveur
- `/var/lib/nfs/xtab` fournit la liste des clients réels qui ont un montage NFS en cours sur le serveur

Le processus de montage est donc le suivant :

- Quand un client souhaite accéder à un système de fichiers, il commence par le demander à `mountd` (par le biais de `portmap` pour connaître le port de `mountd`).
- Le pare-feu vérifie que la requête est légitime
- Le noyau vérifie que le client a légitimement le droit de présenter cette requête (contrôle des `hosts.{allow,deny}`)
- `mountd` regarde dans le fichier `/var/lib/nfs/etab` pour voir si le nom du client correspond à celui que donne le client
- si le client est accepté, une entrée est ajoutée dans le fichier `/var/lib/nfs/rmtab` et `/var/lib/nfs/xtab` et avertit le noyau pour que le nouveau montage soit inscrit dans `/proc/fs/nfs/exports`
- le serveur renvoie un « *file handle* » pour le point de montage que le client pourra ensuite utiliser pour les opérations sur les fichiers du point de montage

III. Mise en oeuvre sous Linux

a) Installation et démarrage

Les packages NFS sont `nfs-utils-lib`, `nfs-utils` et `portmap` sous Fedora et `nfs-common` (partie cliente), `nfs-kernel-server` (partie serveur) et `quota` (si vous en avez), pour Debian.

Pour démarrer NFS, il faut que les deux démons `rpc.mountd` et `rpc.nfsd` soient bien démarrés ou plus simplement :

```
[root]# service nfs start
```

ou sous Debian :

```
[root]# /etc/init.d/portmap start
```

```
[root]# /etc/init.d/nfs-kernel-server start
```

```
[root]# /etc/init.d/nfs-common start
```

Si quelque chose ne va pas, c'est sûrement que `portmap` n'est pas démarré :

```
[root]# /etc/init.d/portmap start
```

Ensuite, on peut vérifier avec `rpcinfo` :

```
[root]# rpcinfo -p
```

La commande `rpcinfo` permet de connaître les services RPC qui fonctionnent (l'option `-p`) sur la machine spécifiée en argument.

b) Le fichier de configuration

La syntaxe de chaque ligne du fichier `/etc/exports` :

```
/chemin/dossier/à/exporter nom_machine(options)
```

Attention : sur une ligne qui n'est pas un commentaire, **il faut utiliser l'espace uniquement dans les cas suivants** :

- toujours **mettre un espace entre le chemin à exporter et le (ou les) nom(s) de machine(s)**
- si vous mettez **plusieurs noms de machines sur une seule ligne**, il est important qu'ils soient **séparés les uns des autres par un espace**
- si vous mettez **plusieurs noms de machines sur une seule ligne**, il est important qu'il n'y ait pas d'espace entre le dernier nom de machine est la « (» :
 - `/chemin machine(ro)` : indique une exportation en lecture-seule de `/chemin` pour machine **uniquement**
 - `/chemin machine (ro)` : indique une exportation en lecture-seule de `/chemin` pour machine **mais aussi pour toutes les machines (du fait de l'espace après « machine »**

`nom_machine` peut, entre autre, contenir :

- *un seul nom de machines* : un hôte spécifique est spécifié avec un nom de domaine, un nom d'hôte ou une adresse IP complète (`x.y.z.v/32`).
- *une liste séparées par des espaces de noms de machines*

Chaque nom de machine peut contenir :

- `*` : remplace un groupe de caractère sans « . » dans un nom DNS de machine ou un groupe de chiffre dans une IP. Par exemple, `*.truc.fr` permet d'autoriser `machin.truc.fr` mais

pas *bidulle.machin.truc.fr*. Par contre « **.truc.fr *.*.truc.fr* » correspond aux deux.

- ? : remplace un seul caractère dans un nom DNS de machine.
- *sous-réseau* : une adresse de sous-réseau *x.y.z.v/nombre*. Par exemple, on peut autoriser le sous réseau 192.168.34.128/25 c'est à dire de 192.168.34.128 à 192.168.34.256.
- *netgroups* — permet d'utiliser un nom de groupe de réseau NIS, écrit sous la forme @<*nom-de-groupe*>. Cela place le serveur NIS en charge du contrôle d'accès pour le système de fichiers en question ; il est ainsi possible d'ajouter et de supprimer des utilisateurs du groupe NIS sans affecter */etc/exports*

Les options *options* peuvent être entre autre :

- *ro/rw* : indique si l'export est en lecture-seule ou lecture-écriture
- *secure/insecure* : indique si seul root sur le client peut monter l'export ou pas
- *root_squash/no_root_squash* : indique si un montage avec l'uid/gid de root sur le client se transforme en l'uid/gid de l'utilisateur anonyme nobody.
- *squash_uids/squash_gids* : indique une liste d'uids/gids à transformer en uid/gid anonyme au montage
- *anouid/anongid* : indique l'uid/gid du compte anonyme

c) Rechargement de la liste des « export » NFS

Pour recharger l'ensemble des exports NFS, il suffit d'exécuter :

```
[root]# exportfs -ra
```

Pour décharger tous les exports, taper :

```
[root]# exportfs -ua
```

d) Affichage des « export » NFS

Pour obtenir la liste des « exports » montés par les clients sur le serveur local, exécutez :

```
[root]# showmount -a
```

Pour obtenir la liste des « export » sur un serveur distant

```
[root]# showmount -a <IP_ou_nom_DNS_serveur_distant>
```

ou

```
[root]# showmount -e <IP_ou_nom_DNS_serveur_distant>
```

e) Montage des « export » NFS

Pour monter un « export » NFS, il suffit de taper :

```
[root]# mount -t nfs -o user,nosuid,hard,intr  
IP_ou_nom_DNS_serveur:/chemin/export /point/montage
```

Si l'on veut utiliser le montage de façon permanente, dans le fichier */etc/fstab* :

```
IP_ou_nom_DNS_serveur:/chemin/export /point/montage nfs user,nosuid,hard,intr 0
```

Attention: si vous avez le même UID que le propriétaire des fichiers, vous êtes propriétaire des fichiers sur les montages NFS.

f) Précautions d'usage

Le principal problème est que pour faire un montage NFS, le client doit avoir confiance en le serveur et réciproquement.

Un client ne peut croire aveuglément en un serveur, il faut donc préciser des options contraignantes lors de l'utilisation de la commande `mount`.

La première est : `nosuid`. Elle annule l'effet des bits SUID et SGID. Ainsi, une personne `root` sur le serveur doit se connecter d'abord en tant qu'utilisateur quelconque sur le client pour ensuite seulement redevenir `root`.

Une autre option, plus contraignante, est `noexec`. Elle interdit l'exécution des programmes contenus sur le système exporté. Cette option n'est utilisable que pour les systèmes contenant uniquement des données.

Du côté du serveur NFS, on peut également spécifier qu'on ne fait pas confiance au compte `root` des clients. On doit alors préciser dans le `/etc/exports` l'option `root_squash`. Ainsi, si un utilisateur avec l'UID 0 (celui de `root`) sur le client accède au système exporté par le serveur, il se voit attribuer l'UID de `nobody` pour effectuer les requêtes sur les fichiers. Cette option est active par défaut sous Linux mais s'annule par l'option `no_root_squash`. On peut aussi préciser une plage d'UID pour lesquels l'option s'applique. Enfin les options `anonuid` et `anongid` permettent de changer l'UID/GID de l'utilisateur `nobody` en celui souhaité.

g) Préliminaires aux Règles iptables

Mettre iptables en place pour NFS n'est pas possible en l'état du fait que certains programmes écoutent sur des ports variables. Voici une liste des ports d'écoute de NFS :

Nom du démon	RPM	Port par défaut	Port suggéré
<i>portmap</i>	portmap	111	111
<i>rpc.statd</i>	nfs-utils	Variable	4000
<i>rpc.nfsd</i>	nfs-utils	2049	2049
<i>rpc.lockd</i>	nfs-utils & kernel	Variable	4001
<i>rpc.mountd</i>	nfs-utils	Variable	4002
<i>rpc.rquotad</i>	quota	Variable	4003

1. Généralités

Si on veut avoir un bon affichage dans `netstat`, `tcpdump` et autres, on ajoutera dans `/etc/services` :

```

rpc.nfsd      2049/tcp      # RPC nfsd
rpc.nfsd      2049/udp      # RPC nfsd
rpc.statd     4000/tcp      # RPC statd listen
rpc.statd     4000/udp      # RPC statd listen
rpc.mountd    4002/tcp      # RPC mountd
rpc.mountd    4002/udp      # RPC mountd
rpc.lockd     4001/tcp      # RPC lockd/nlockmgr
rpc.lockd     4001/udp      # RPC lockd/nlockmgr
rpc.quotad    4003/tcp      # RPC quotad
rpc.quotad    4003/udp      # RPC quotad

```

2. Sous Debian

i. rpc.statd

Editer le fichier `/etc/default/nfs-common` :

```
STATDOPTS='-p 4000'
```

ii. rpc.lockd

Editer `/etc/modprobe.conf`:

```
options lockd nlm_udpport=4001 nlm_tcpport=4001
```

Si votre système à le code de lockd compilé dans le noyau et pas dans un module, ce qui précède ne fonctionne pas. Dans ce cas, vous devez ajouter les paramètres

"lockd.udpport=4001 lockd.tcpport=4001" à la ligne de démarrage de lilo et grub à la place.

Il peut être intéressant de lancer en root `update-modules`.

iii. rpc.mountd

Créer ou éditer le fichier `/etc/default/nfs-kernel-server` :

```
RPCMOUNTDOPTS='-p 4002'
```

iv. rpc.rquotad

Créer ou éditer le fichier `/etc/default/quotad` :

```
RPCRQUOTADOPTS='-p 4003'
```

3. Sous Fedora

Créer ou éditer le fichier `/etc/sysconfig/nfs` :

```

LOCKD_TCPPOINT=4001
LOCKD_UDPPORT=4001
MOUNTD_PORT=4002
STATD_PORT=4000

```

```
RQUOTAD=NO
RQUOTAD_PORT=4003
```

h) Règles iptables

```
#portmap
[root]# iptables -A INPUT -p tcp --dport 111 -j ACCEPT
[root]# iptables -A OUTPUT -p tcp --sport 111 -j ACCEPT
#rpc.nfsd
[root]# iptables -A INPUT -p tcp --dport 2049 -j ACCEPT
[root]# iptables -A OUTPUT -p tcp --sport 2049 -j ACCEPT

#rpc.statd
[root]# iptables -A INPUT -p tcp --dport 4000 -j ACCEPT
[root]# iptables -A OUTPUT -p tcp --sport 4000 -j ACCEPT
#rpc.lockd
[root]# iptables -A INPUT -p tcp --dport 4001 -j ACCEPT
[root]# iptables -A OUTPUT -p tcp --sport 4001 -j ACCEPT
#rpc.mountd
[root]# iptables -A INPUT -p tcp --dport 4002 -j ACCEPT
[root]# iptables -A OUTPUT -p tcp --sport 4002 -j ACCEPT
#rpc.quotad
[root]# iptables -A INPUT -p tcp --dport 4003 -j ACCEPT
[root]# iptables -A OUTPUT -p tcp --sport 4003 -j ACCEPT
```

IV. Mise en oeuvre en client sous Windows

a) Prérequis

- Windows XP Professional (ou Home mais avec une manoeuvre pour installer SFU)
- Services for UNIX (SFU) 3.5 ou supérieur
- un partage NFS sur une machine UNIX (serveur)

b) Procédure

- Installer SFU 3.5 ou supérieur depuis [Microsoft](#)
- Vérifier dans « Panneau de configuration » / « Outils d'Administration » / « Services » que le service « User Name Mapping » est démarré et/ou en démarrage automatique
- Dans le menu Démarrer, aller à « Windows Services for UNIX » / « Services for UNIX Administration »

- Cliquer sur « Services for UNIX [local] »
- Si vous voulez effectuer la manoeuvre sur un autre ordinateur, cliquer à droite sur « Services for UNIX [local] » puis sur « Connect to Another Computer »
- Dans l'onglet « Settings », mettre localhost dans "Computer Name"
- Changer, éventuellement les options de "Client for NFS" (les options par défaut doivent fonctionner). Si NFS est instable, il peut être nécessaire de changer UDP en TCP dans « Performance »
- Cliquer sur "User Name Mapping"
- Sélectionner "Use Password and Group files" dans l'onglet "Configuration".
- Mettre dans "Password file path and name" le chemin et le nom du fichier /etc/passwd (par exemple c:\sfu\common\passwd)
- Mettre dans "Group file path and name" le chemin et le nom du fichier /etc/group (par exemple c:\sfu\common\group)
- Dans l'exemple :
 - c:\sfu\common\passwd peut être copié de /etc/passwd du serveur NFS UNIX sur lequel on veut pouvoir se connecter. Vous pouvez supprimer les lignes des utilisateurs dont vous n'avez pas besoin.
 - c:\sfu\common\group peut être copié de /etc/group du serveur NFS UNIX sur lequel on veut pouvoir se connecter. Vous pouvez supprimer les lignes des groupes dont vous n'avez pas besoin.
- Passer dans l'onglet "Maps"
- Cocher « Simple maps ».
- Choisir le nom de votre machine dans "Windows domain name" (ou le nom de domaine qui contient les noms d'utilisateurs à mapper). \localhost peut aussi marcher.
- Cliquer sur "Show User Maps". Cliquer sur "List Windows Users" et sur "List UNIX Users".
- Cliquer sur votre nom d'utilisateur Windows à gauche, et sur votre nom d'utilisateur UNIX à droite et cliquer sur "Add".
- Procéder de même pour votre groupe si vous en avez un sous Windows. *Le mapping de groupe n'est pas nécessaire pour accéder à un partage NFS.*
- Vous devriez maintenant pouvoir connecter un lecteur réseau à un partage NFS, en utilisant la syntaxe des partages Windows, à savoir \\nfserver\path\. Quand vous cliquer sur « OK » ou « Terminer » et si c'est un partage NFS et pas SAMBA, une fenêtre devrait apparaître vous indiquant l'uid (et le gid) UNIX du mapping. Cet uid utilisateur doit correspondre à votre uid UNIX sur le partage NFS.

Si vous changez d'utilisateur pour la connection du lecteur réseau et qu'un mot de passe vous est demandé, c'est biensûr celui du **compte Windows et pas UNIX**. En effet, Windows ne connaît pas le /etc/shadow.

V. Bibliographie

[Network File System - Wikipédia](#)

[Introduction au service NFS](#)

[Linux NFS-HOWTO](#)

[Introduction à NFS](#)

[NFS : le partage de fichiers sous Unix](#)