

## TP7: NFS et NIS

Sébastien Collette (sebastien.collette@ulb.ac.be)

### Résumé

L'objectif de ce TP est de vous familiariser avec NFS et NIS, deux services standards sous Linux. Le premier nous permettra de partager des répertoires sur plusieurs machines, le second nous permettra d'utiliser les mêmes comptes utilisateurs sur des machines différentes.

## 1 Introduction

Le but de ce TP sera dans un premier temps de tester le fonctionnement du système NFS (Network File System) sur vos machines virtuelles. Vous simulerez un réseau sur machines virtuelles en lançant deux machines ayant des adresses IP différentes, et vous utiliserez NFS pour pouvoir accéder aux fichiers de l'une des machines virtuelles à partir de l'autre.

Dans un second temps vous testerez NIS (Network Information Service, également appelé Yellow Pages), qui permet (tout comme LDAP) de déployer une base de données distribuée sur un réseau, et est surtout utilisé pour partager des informations contenues dans des fichiers de configuration, comme `/etc/passwd`. Nous l'utiliserons conjointement avec NFS, afin de réaliser un système avec authentification centralisée et serveur de *home directories*.

**Note** Vous devrez travailler sur deux machines virtuelles, en utilisant l'image `img-base.tgz`.

## 2 Mise en place de NFS

---

**Exercice 1** Préparez deux machines virtuelles. Vérifiez qu'elles sont bien connectées au réseau, donnez-leur les hostnames client et serveur. Installez le package `nfs-common` sur les deux machines, et `nfs-kernel-server` sur le serveur.

---

### 2.1 Le fichier `/etc/exports` du serveur

Ce fichier, situé sur le serveur NFS, contrôle quels sont les répertoires à rendre accessibles au public au travers de NFS, à qui on les rend accessibles, et avec quelles permissions. Sa syntaxe est la suivante : chaque ligne contient un répertoire à exporter, suivi d'une liste de clients (séparés par des blancs) pouvant monter ce répertoire. Chaque client peut être suivi

(immédiatement, sans caractères blancs) d'une liste d'options placées entre parenthèses et séparées par des virgules.

Voir man 5 exports pour plus de détails. Les options d'exportation les plus courantes sont les suivantes :

- ro : lecture seule.
- rw : lecture/écriture.
- rw=liste : lecture/écriture à une liste d'hôtes, lecture seule aux autres.
- root\_squash : associe l'UID 0 et le GID 0 (i.e. le compte root) distants aux valeurs spécifiées par anonuid et anongid. Explication : le comportement normal de NFS consiste à utiliser l'uid et le gid de la requête client pour gérer les droits d'accès au fichier sur le serveur. Toutefois, il n'est en général pas désirable que l'utilisateur root de la machine cliente soit aussi traitée en tant que root sur la machine serveur. C'est pour cette raison que le uid 0 est en général mappé vers le uid anonyme anonuid. Ce mécanisme est appelé le *root squashing* et est le comportement par défaut.
- no\_root\_squash : Autorise les accès normaux depuis le compte root. Dangereux.
- all\_squash : Mappe tous les uids et gids vers l'utilisateur anonyme. Ceci est utile par exemple pour des répertoires FTP publics exportés par NFS, etc...
- anonuid=xxx : Permet de spécifier explicitement le uid de l'utilisateur anonyme.
- anongid=xxx : idem pour le gid.

Voici un exemple de fichier exports :

```
/ master(rw) trusty(rw,no_root_squash)
/projects proj*.local.domain(rw)
/home/joe 192.168.139.0/24(rw,all_squash,anonuid=150,anongid=100)
/pub (ro,all_squash)
```

## 2.2 La commande exportfs

Après toute modification au fichier exports, il convient d'exécuter la commande /usr/sbin/exportfs sur le serveur afin de réaliser l'exportation à proprement parler (voir man pour plus de détails). Appelée avec l'option -a, cette commande réalise l'export de tous les répertoires listés dans /etc/exports.

---

**Exercice 2** Créez un fichier export sur le serveur indiquant que vous partagez le répertoire /home à toutes les machines dans le subnet du client. N'oubliez pas exportfs !

---

## 2.3 Montage des répertoires exportés

L'accès aux données distantes par le client se fait en deux temps : montage (grâce à mount) du file system distant dans l'arborescence du client, puis accès aux fichiers. Le mount suit la syntaxe mount hostname:directory directory\_d\_accrochage. Pour démonter un file system umount directory\_d\_accrochage

Lors du montage d'un file system distant, plusieurs options peuvent être spécifiées (à l'aide de l'option -o de mount). Le tableau ci-dessous en apporte un récapitulatif (voir man mount et man nfs) :

- ro : lecture seule.
- rw : lecture/écriture.
- bg : Si l'essai de montage NFS échoue, réessayer en background de façon régulière.
- fg : Si l'essai de montage NFS échoue, réessayer en foreground. C'est l'option par défaut.
- soft : Si une opération NFS sur un fichier distant ne répond pas après un certain laps de temps (paramétrable par l'option `timeo`), s'arrêter et renvoyer une erreur à l'appelant.
- hard : Si une opération NFS sur un fichier distant ne répond pas après un certain laps de temps, continuer à essayer indéfiniment. C'est l'option par défaut. Ceci est utile par exemple en cas de crash du serveur : les clients seront tous simplement bloqués jusqu'à ce que le serveur soit à nouveau opérationnel puis pourront à nouveau accéder aux fichiers distants de manière transparente, sans avoir dû gérer d'erreur. Les processus ainsi bloqués sont non-interruptibles, sauf si l'option `intr` (voir ci-dessous) est spécifiée.
- intr : Permet d'interrompre des processus ayant fait un montage hard qui sont bloqués suite à l'absence de réponse de la part du serveur. L'interruption se fait par `Ctrl-C`. L'option `intr` n'est pas activée par défaut.

---

**Exercice 3** *Sur le client, tentez de monter le répertoire home partagé par le serveur dans `/mnt/home_du_serveur`. Faites en sorte que les droits d'accès et les propriétaires de fichiers soient préservés (nécessite peut-être de modifier le serveur). Faites des tests : création de fichiers par les utilisateurs `notroot` et `root`, changement des droits avec `chmod`. Regardez les résultats tant sur le client que sur le serveur.*

---

### 3 Mise en place de NIS

---

**Exercice 4** *Installez `nis` sur le client et le serveur.*

---

NIS utilise un modèle client-serveur. Un serveur NIS est une machine contenant les bases de données distribuées construites à partir des fichiers de configuration. Il y a toujours exactement un serveur maître, et éventuellement plusieurs serveurs additionnels, appelés esclaves, qui contiennent des duplications des bases de données du serveur maître.

Si par exemple le fichier `/etc/passwd` est géré par NIS, les consultations de ce fichier dans les clients ne se feront pas sur le fichier local lui-même, mais plutôt par une requête envoyée sur le réseau vers le serveur NIS. Ainsi, toutes les machines du réseau pourront travailler avec le même fichier `passwd` présent une seule fois sur le serveur. Bien sûr, le serveur devra reconstruire sa base de données distribuée à chaque modification du fichier `passwd` (et propager ce changement vers les éventuels serveurs esclaves). Dans le cadre de ce TP, nous nous limiterons à un seul serveur NIS.

NIS fonctionne essentiellement grâce à deux démons : `ypserv` sur le serveur et `ybind` sur les clients. Plusieurs fichiers de configuration sont également utilisés :

- `/var/yp/Makefile` précise quels sont les fichiers que l'on veut distribuer grâce à NIS,
- `/etc/yp.conf` sur le client donne les informations relatives au serveur,

- `/etc/nsswitch.conf` sur le client précise dans quel ordre les informations seront recherchées par le client : si `file` précède `nis` alors on ne fait appel à NIS que si l'information recherchée n'est pas trouvée dans le fichier de configuration local, si `nis` précède `files` on fait d'abord appel à NIS. Voir man `nsswitch.conf`.
- Le fichier `/etc/ypserv.conf` permet au serveur d'exercer un contrôle plus fin sur la sécurité du système. Nous ne l'utiliserons toutefois pas dans le cadre de ce TP.

Man pages pertinentes : `ypserv`, `ypbind`, `ypserv.conf`, `ypinit`, `ypcat`, `ypwhich`, `domainname`, `nsswitch.conf`.

---

**Exercice 5** Sur le serveur et le client, personnalisez le fichier `/etc/default/nis`. Sur le client, indiquez quelle est l'IP du serveur dans le fichier `/etc/yp.conf`. Sur le serveur, allez dans `/var/yp` et tapez `make`, ce qui a pour effet de créer la base de données NIS. Lancez le service NIS d'abord sur le serveur puis sur le client.

---

A partir de maintenant, le service NIS devrait être fonctionnel.

---

**Exercice 6** Depuis le client, regardez à quel serveur NIS vous êtes connecté en utilisant la commande `ypwhich`.

---

Il reste à dire ce que nous voulons qu'il gère. Notre objectif est de partager la gestion des comptes et des groupes. Pour cela, nous allons donc créer un nouveau groupe (mongroupe) et deux utilisateurs (Robert et Marie) sur le serveur<sup>1</sup>.

---

**Exercice 7** Créez les comptes et groupes mentionnés ci-dessus. Remarquez que les commandes interagissent avec NIS. Refaites un `make` dans `/var/yp`.

---

Maintenant, sur le client, nous allons demander d'utiliser les informations données par le serveur NIS en priorité (c'est-à-dire utiliser le fichier `passwd` et `group` de NIS et non les fichiers locaux).

---

**Exercice 8** Sur le client, modifiez le fichier `/etc/nsswitch.conf` afin d'utiliser les informations NIS à la place des fichiers `passwd`, `shadow` et `group`. Relancez NIS et tentez de vous logger sur un des comptes définis sur le serveur. Remarquez que comme la home de ces comptes n'existe pas sur le client, le système indique une erreur au login. Utilisez la commande `ypcat` pour voir le contenu des fichiers exportés (`passwd` et `groups`).

---

---

**Exercice 9** Depuis le client, changez le mot de passe de l'un des comptes exportés.

---

---

1. Référez-vous au TP approprié en cas de besoin...

## 4 Combinaison de NIS et NFS

Il ne nous manque plus que le partage des homes pour que notre système soit fonctionnel. On pourrait bien entendu exporter le répertoire `/home` du serveur et le monter directement à la place du répertoire `/home` du client. Ce n'est néanmoins pas une bonne idée si l'on veut pouvoir avoir des comptes locaux, en plus des comptes NIS. C'est pourquoi nous allons procéder comme suit :

- Sur le serveur, création d'un répertoire `/home/nis`, dans lequel nous déplaçons les homes des comptes partagés.
- Toujours sur le serveur, modification du fichier `passwd` afin de refléter le changement des homes.
- Validation par un `make` dans `/var/yp`.
- Modification du fichier `exports` afin de ne partager que le répertoire `/home/nis`.
- Montage du répertoire sur le client.

---

**Exercice 10** *Effectuez les changements mentionnés ci-dessus. Testez le système finalisé : le login depuis le client et depuis le serveur devraient être totalement identiques.*

---

**Note** *N'oubliez pas de nettoyer le répertoire `/vmware/run` à la fin de chaque TP.*