NIS et NFS

Nous traiterons dans cette page des serveurs UNIX. Des produits équivalents existent sur Windows NT.

Quelques livres :

H. Stern, Pratique de NFS et NIS, Ed. O'Reilly,

J.M. Moreno, UNIX administration, Ed. Ediscience International,

T. Besançon, P. David, J. Marchand, Administration Système UNIX, disponible sur le réseau.

1 - GRANDS PRINCIPES

Dans un environnement réparti, les deux enjeux majeurs sont :

- la gestion des fichiers de configuration communs, résolu par NIS (Network Information System),
- la gestion de l'environnement de l'utilisateur quand il change de machine, résolu par NFS (Network File System).

Ces deux systèmes s'appuient sur les services XDR et RPC.

Le protocole XDR (External Data Representation) permet d'harmoniser la représentation des données dans un environnement hétérogène :

- ordre des octets,
- comportement des compilateurs,
- nombre flottants, tableaux et chaînes de caractères.

Par exemple il existe une notion d'ordonnancement immuable appelé forme canonique. La règle du protocole XDR est que l'émetteur convertit les données locales en forme canonique et que le récepteur convertit la forme canonique en données locales.

Le protocole RPC (Remote Protocol Call) permet à un site d'effectuer un appel à procédure qui apparaît local mais qui s'exécute réellement sur un autre site. Le système RPC empaquette les données conformément au protocole XDR, crée une session avec le serveur en envoyant le datagramme au processus distant. Sur le serveur RPC dépaquette les données, exécute la procédure empaquette le résultat et le communique au client. Beaucoup de protocoles n'utilisent pas RPC. Par exemple, Telnet et FTP sont des protocoles orientés octets qui ont leurs propres algorithmes. RPC fonctionne au-dessus d'UDP principalement mais il peut également utiliser TCP. Les serveurs RPC sont mono tâches : il existe un unique serveur par service RPC et il exécute les tâches des clients mise en file d'attente.

Le fichier /etc/rpc contient une liste des serveurs RPC et leurs identificateurs.

Exemple

```
nisd        100300  rpc.nisd
```

Le numéro est celui du service rpc. C'est ce numéro qui sert au démon portmap pour retrouver dans /etc/services le port où le service est installé.

2 - LE SYSTEME NIS

C'est un système de bases de données répartie qui gère de façon centralisé les copies des fichiers de configuration qui devraient être dupliqués sur tous les hôtes.

Les fichiers gérés (habituellement situés sous /etc) sont les suivants :

- bootparams, hôtes sans disques
- ethers, identificateurs Ethernet

  exemple : 08:00:20:0f:ce:db harold
- group, groupe d'utilisateurs

  exemple : sys::3:root,bin,sys,adm
- hosts, nom des hôtes et adresses IP

  exemple : 147.210.12.7 antonella #Sparc classic 15"
- aliases, liste d'adresses pour le mail
- netmasks, masque d'adresse IP

  exemple : 147.210.0.0 255.255.254.0
- networks, sous-réseau de machines

  exemple : eminet12 147.210.12
- passwd, mot de passe
- protocols, noms et identificateur de protocole.

  exemple : icmp 1 ICMP # internet control message protocol
- rpc, identificateurs des procédures RPC
- services, identificateur des ports.

Le fichier netgroup est un fichier supplémentaire permettant de grouper des machines avec même droit d'accès

exemple : serveurs (aliboron,-,)\
(harold,-,)\
....

ou bien

acceptes2 (azur.sophia.cnrs.fr,miniussi,)\
(dufy.aquarel.fr,fortier,)\
(freeway.issy.cnet.fr,midrouli,)

mais aussi

dangereux (,gustave,)

Un serveur NIS est un site qui contient des tables construites à partir de fichiers DBM. A chaque fichier de /etc sera associé lors de la mise en route du serveur, un fichier d'index ayant pour extension .dir et un fichier données avec extension .pag. Ces deux

fichiers permettront un accès rapide via RPC aux données communes sur le réseau. Chaque fois qu'un fichier DBM est crée, un fichier suffixé .time lui est associé de manière à conserver la date et l'heure de la dernière modification. L'ensemble de ces fichiers est créé dans un sous-répertoire du fichier /var/yp.

Exemple : le fichier passwd va donner lieux aux tables

passwd.byname.dir

passwd.byname.pag

passwd.byuid.dir

passwd.byuid.pag

Les fichiers ne peuvent ni être déplacés ni être copiés.

Il y a deux types de serveurs NIS :

- le serveur maître qui est le réel détenteur des fichiers,
- les serveurs esclaves qui n'en ont qu'une copie.

Le serveur maître est responsable de la maintenance des tables et de leur distribution aux esclaves. Les clients peuvent s'adresser indifféremment au serveurs maître ou esclaves.

Exemple . Lorsqu'un client recherche l'adresse IP d'un système, il doit consulter son fichier hosts. Si NIS fonctionne le client interrogera NIS avant de consulter le fichier hosts.

Un serveur peut maintenir la cohérence de plusieurs systèmes. C'est la notion de domaine NIS (ou groupe de système) qui est distincte de la notion de domaine Internet. A un domaine donné est associé un ensemble de tables NIS. Pour établir le nom de domaine sur une machine, on exécute la commande

domainname <nom_de_domaine>.

La liste suivante de commandes permet d'observer un serveur NIS

- ypmatch : analogue du grep sur les tables,
- ypcat : équivalent de cat pour les fichiers NIS,
- ypwhich : donne le nom du serveur en activité, avce l'option -m elle donne le nom du serveur maitre.

3 – INTALLATION DE NIS

L'installation du serveur se fait par la commande « ypinit –m ». Cette commande créée dans le répertoire /var/yp les diverses tables dans un répertoire associé au nom de domaine. Il n'existe pas de commandes permettant de connaître les domaines gérés par un serveur NIS. La seule solution consiste à regarder dans le répertoire /var/yp les répertoires existants. Le fichier ypserv.log permet de garder le journal des messages d'erreurs.

L'installation d'un serveur esclave par la commande

ypinit –s <nom_du_maitre>.

Le maître et l'esclave doivent se trouver en principe sur le même réseau IP. Dans le cas contraire l'esclave doit être un client du maître et la commande ypset permet de le pointer explicitement.

Pour lancer un site client de NIS, il faut :

- modifier les entrées des fichiers de configuration,
- utiliser la commande domainname pour fixer le nom de domaine,
- lancer le démon ypbind qui se charge de localiser les serveurs NIS.

Les fichiers passwd, group, bootparams et aliases sont augmentés des informations du serveur NIS. Les fichiers locaux sont explorés avant de solliciter le serveur. Les autres fichiers sont remplacés.

Les entrées des fichiers concaténés sont réduites au minimum : entrées nécessaires au boot, entrées locales à la machine. Le symbole « + » permet de signifier que la table NIS doit être considérée comme concaténée à la table courante.

Exemple : fichier minimum passwd.

```
root:8EexyTErAmbnA:0:1:Operator:/:/bin/csh
nobody:*:65534:65534::/:
daemon:*:1:1::/:
sys:*:2:2::/:/bin/csh
bin:*:3:3::/bin:
uucp:*:4:8::/var/spool/uucppublic:
+:*:0:0:::
```

Quelques symboles ne sont valables que pour certains fichiers dont le fichier passwd :

+<utilisateur> : permet d'insérer l'entrée correspondante de la table NIS

-<utilisateur> : permet de les supprimer

+@<netgroup> : insère les entrées des utilisateurs correspondant au netgroup

-@<netgroup> : permet de les supprimer

Si une entrée locale comporte des champs en conflit avce une entrée NIS, les champs de l'entrée locale sont conservés.

Le démon ypbind au lancement lance une requête de service qui permet de localiser le serveur. Pendant son activité si ce serveur tombe en panne ou ralentit, il cherchera à en localiser un autre. Cette détection se fait à chaque requête par l'armement d'un timer et en cas d'inactivité par la commande ping.

4 – EXEMEPLE DE DEROULEMENT D'UNE REQUETE

Supposons qu'un utilisateur lance ls -l. le processus ls doit trouver le nom de l'utilisateur qui correspond à l'UID du propriétaire de chacun des fichiers. Dans ce cas la fonction getpwuid. Supposons que l'UID soit 654

Si le fichier de mot de passe est celui donné en exemple, il n'existe pas de tel UID. getpwuid effectue alors une requête au serveur NIS. Il recherche le nom de domaine et ypbind fournit la liaison au serveur. Le processus client invoque la procédure de recherche RPC avce les arguments key=654 et map=passwd.byuid. cette requête est empaquetée et envoyé au processus ypserv qui est lié. Le serveur effectue la recherche et renvoit la réponse au client.

Des erreurs peuvent se produire mais elles sont masquées par le code rpc appelé par getpwuid.