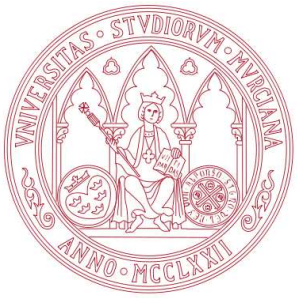

Universidad de Murcia

Facultad de Informática

Computación Científica en Clusters

Sesión 2: NFS, NIS, etc.

Boletines de prácticas

2010

Departamento de Ingeniería y Tecnología de Computadores

Área de Arquitectura y Tecnología de Computadores

Índice general

Boletines de prácticas

Consideraciones iniciales

IMPORTANTE: No inicies ninguna máquina virtual hasta que no se te indique que lo hagas.

En el ordenador real, y no en la máquina virtual, con el usuario **root**, cambia los permisos a los ficheros del directorio donde está la imagen de la máquina virtual:

```
# su -l                  // En el laboratorio la contraseña de root es practicas
# chmod 644 /home/sofware/CCC-MV/*
```

Cada ordenador ya tiene instalada la aplicación VirtualBox a la cual podemos acceder a través de `Aplicaciones` → `Herramientas del sistema` → `VirtualBox`. Al ejecutar la aplicación, debemos encontrar una máquina virtual ya creada llamada "Fedora 11 - Server" que contiene todo lo necesario para realizar la configuración del servidor. Esta máquina virtual tiene configuradas dos interfaces de red: `eth1` y `eth2`. La primera se utilizará para comunicarse con el cliente a través de una red interna virtual creada por VirtualBox; esta interfaz tiene asignada la IP estática `10.0.1.15`. La segunda interfaz se utilizará para acceder a Internet a través de NAT; esta interfaz tiene la IP dinámica `10.0.2.15` asignada por VirtualBox.

En la máquina virtual ya existen dos cuentas de usuario: una para el superusuario (`root`) y otra para un usuario que hemos llamado "Usuario del curso CCC", que tiene por login `ccc`. Para ambas cuentas, la contraseña es `cluster`.

Un primer paso es cambiarle el nombre a la máquina virtual ya creada, usando para ello la opción `Configuración`. Simplemente cámbiaselo a "Servidor". Es de cara a poder distinguir claramente entre las máquinas. No arranques aún esta máquina.

A continuación desde Virtual Box importa una nueva máquina virtual mediante la opción `Importar servicio virtualizado` del menú `Archivo`. Esta máquina hará de cliente. El directorio donde está la máquina virtual a importar es `/home/sofware/CCC-MV`. Realiza después los siguientes pasos, modificando la "Configuración" de la misma:

1. Cámbiale el nombre, haz que se llame "cliente".
2. Asegúrate que tiene 2 adaptadores de red activos. El primero que usa `NAT` mientras que el segundo tiene que estar conectado a una red interna.

La nueva máquina virtual, al ser importada de la que hay preparada para hacer de servidor, también tendrá el usuario **ccc** creado.

A continuación pon en marcha esta máquina virtual y usando la herramienta `system-config-network` asigna a la red interna los siguientes valores:

- IP: 10.0.1.16
- Máscara de red: 255.255.255.0
- Puerta de enlace: 10.0.2.2

En la prueba que yo hice en el laboratorio, el interfaz de red que hay que cambiar es el `eth4`, pero esto puede cambiar. Para saberlo puedes usar la orden `/sbin/ifconfig`. De las 2 interfaces `eth` una de ellas tendrá asignada la dirección `10.0.2.15`, pues el otro interfaz es la red interna.

Edita el fichero `/etc/sysconfig/network` y cámbiale el nombre al equipo, asignado **ccc-cliente**. Es la variable `HOSTNAME`.

Reinicia esta máquina virtual.

Pon en marcha la máquina virtual que hace de servidor, la que estaba creada inicialmente.

Entra como usuario normal en las dos máquinas virtuales, abre una terminal y conviértete en super-usuario para realizar los siguientes boletines.

Para ambos equipos, y de cara a resolver los ejercicios, es conveniente tener desactivado el cortafuegos:

```
# /etc/init.d/iptables stop
# chkconfig iptables off
```

Es importante, antes de empezar a resolver los ejercicios, comprobar que la red interna que usarían ambas máquinas para comunicarse funciona perfectamente. Puedes usar para ello la orden `ping`, así por ejemplo, desde el equipo servidor ejecutaríamos:

```
# ping 10.0.1.16
```

Tanto en el equipo servidor como en el cliente, añade al fichero `/etc/hosts` estas dos líneas, correspondiendo cada una de ellas a los nombre asignados a las dos máquinas:

```
10.0.1.15 ccc-server
10.0.1.16 ccc-cliente
```

B2.1. Boletín 1: Network File System

B2.1.1. Objetivos

El objetivo de este primer boletín es configurar un servidor de **NFS** (*Network File System*) así como los equipos cliente del mismo. La idea es ver un procedimiento sencillo y rápido para compartir ficheros entre los nodos de cómputo de un *cluster*. Aunque el procedimiento se verá para la distribución Fedora Core 11 de Linux, éste es muy similar para otras distribuciones. Quizás, la única diferencia sea la herramienta gráfica, que suele variar entre distribuciones.

Es interesante saber que NFS fue diseñado por Sun Microsystems para sus estaciones de trabajo Unix en 1985, y es soportado por diferentes máquinas, sistemas operativos, tipos de redes y protocolos de transporte.

El servicio está pensado para que una colección de servidores y clientes compartan ficheros a través de la red, sin importar la localización de cada uno ellos. Cada servidor comparte con los clientes remotos uno o más directorios, exportando con ello todo el árbol de directorios que cuelga de él. Cualquier sistema puede ser cliente y servidor a la vez.

Para conseguir nuestro objetivo, vamos a describir cómo configurar tanto el servidor como el cliente.

Las configuraciones del servidor y el cliente las vamos a realizar sobre máquinas virtuales de VirtualBox en un mismo ordenador físico, de manera que cada grupo podrá realizar por sí mismo ambas configuraciones.

B2.1.2. Plan de trabajo

El plan de trabajo para este boletín será el siguiente:

1. Configuración de un servidor NFS.
2. Configuración de un cliente NFS.
3. Cambiando parámetros en el servidor NFS.

B2.1.3. Configuración de un servidor NFS

Los pasos son los siguientes:

1. Antes de empezar este ejercicio, y también los siguientes, hay que desactivar el cortafuegos, si no se ha realizado antes. Para ello ejecutaremos estas dos órdenes:

   ```
   # /etc/init.d/iptables stop
   # chkconfig iptables off
   ```

2. Instalamos los paquetes «nfs–utils» y «nfs–utils–lib». Primero comprobaremos si están instalados usando la orden `rpm`:

   ```
   # rpm -q nfs-utils
   # rpm -q nfs-utils-lib
   ```

   En caso de que no estén instalados, procedemos a instalarlos.

   ```
   # yum install nfs-utils nfs-utils-lib
   ```

   Realiza la misma comprobación para el paquete «system–config–nfs».

3. Editamos el fichero `/etc/exports` y añadimos las líneas necesarias para exportar el directorio `/home`:

   ```
   /home 10.0.1.0/255.255.255.0(rw,no_root_squash)
   ```

4. Comprobamos el estado del demonio `rpcbind`. Si no está lanzado lo lanzamos.

   Para comprobar si está en ejecución:

   ```
   # /etc/init.d/rpcbind status
   ```

   Para lanzarlo:

   ```
   # /etc/init.d/rpcbind start
   ```

5. Iniciamos los demonios de NFS.

   ```
   # /etc/init.d/nfs start
   # /etc/init.d/nfslock start
   ```

   Si ya estuviesen estos demonios en ejecución habría que reiniciarlos, para ello se usa como parámetro `restart`.

6. Finalmente, configuramos el sistema para que haga lo mismo en cada arranque:

   ```
   # chkconfig rpcbind on
   # chkconfig nfs on
   # chkconfig nfslock on
   ```

7. Usando la orden `exportfs` comprobamos si la exportación se ha realizado de forma correcta:

   ```
   # exportfs -v
   ```

B2.1.4. Configuración de un cliente NFS

Terminada la configuración del servidor, procedemos a realizar la configuración del equipo cliente. Primero comprobaremos, simplemente, que el montaje se puede realizar sin problemas. Después, configuraremos el fichero `/etc/fstab` para que el sistema de ficheros remoto se pueda montar en cada arranque. Como veremos, son pasos bastante sencillos.

1. Igual que se hizo en el servidor, desactivamos el cortafuegos.
2. El primer paso es la comprobación de que el demonio `rpcbind` está en ejecución y se lanzará en cada arranque. Para comprobarlo:

   ```
   # /etc/init.d/rpcbind status
   ```

   Para lanzarlo:

   ```
   # /etc/init.d/rpcbind start
   ```

   Para que se active en cada arranque:

   ```
   # chkconfig rpcbind on
   ```

3. Procedemos a realizar el montaje a mano del primer directorio exportado. Esto nos va a indicar que podemos acceder al servidor NFS, y que no hay problemas de exportación:

   ```
   # mount -t nfs -o rw,hard,intr 10.0.1.15:/home /home
   ```

4. Para comprobar que el proceso se ha hecho de forma correcta podemos ejecutar la orden `mount`, y tiene que aparecer una línea como la siguiente:

   ```
   10.0.1.15:/home on /home type nfs (rw,hard,intr,addr=10.0.1.15)
   ```

5. A continuación, creamos un fichero en el directorio exportado. Esto nos servirá para ver el efecto de la opción de exportación "`no_root_squash`". Observa los propietarios, usuario y grupo, del nuevo fichero.

   ```
   # cd /home
   # touch probando_no_root_squash
   # ls -l
   -rw-r--r--  1 root  root     0 feb 19 12:02 probando_no_root_squash
   ```

6. Finalmente, editamos el fichero `/etc/fstab` y añadimos las dos entradas necesarias para que el montaje se realice de forma automática cada arranque. Los directorios que hacen de punto de montaje tienen que estar creados previamente.

   ```
   10.0.1.15:/home                 /home       nfs    defaults,hard,intr,bg   0 0
   ```

7. Se puede proceder a reiniciar el cliente, simplemente para comprobar que el montaje se realiza de forma correcta, o bien ejecutar la orden:

```
# mount -a -t nfs
```

8. Configura el "demonio" `netfs` para que esté activo en tiempo de arranque, y los montajes se realicen sin problemas:

   ```
   # chkconfig netfs on
   ```

9. Desde el equipo servidor podemos comprobar que se está realizando el montaje del directorio exportado, usando las órdenes:

   ```
   # showmount -a
   # showmount -d
   ```

   ¡Ojo! Es posible que en las máquinas virtuales de errores

B2.1.5. Cambiando parámetros en el servidor NFS

.

Tenemos que tener claro que cada vez que cambiemos alguna opción de exportación en el equipo servidor hay que reiniciar los demonios de NFS. Sin embargo el demonio `rpcbind` no es necesario volver a lanzarlo, ya que su función es hacer que las RPC's funcionen. Por su parte, los equipos cliente tienen que realizar de nuevo el montaje de esos directorios.

1. En el equipo servidor editamos el fichero `/etc/exports` y cambiamos la opción de exportación "`no_root_squash`" por "`root_squash`". Las entradas del fichero quedarían como:

   ```
   /home 10.0.1.0/255.255.255.0(rw,root_squash)
   ```

   Relanzamos los demonios de nfs:

   ```
   # /etc/init.d/nfs restart
   # /etc/init.d/nfslock restart
   ```

   Esto también podemos realizarlo ejecutando la orden:

   ```
   # exportfs -r
   ```

2. Usando la opción `-v` de la misma orden, comprobamos que se ha realizado la exportación con las nuevas opciones indicadas.

3. En el equipo cliente, primero desmontamos el directorio remoto `/home`, y después procedemos a montarlo de nuevo.

   ```
   # umount /home
   # mount /home
   ```

4. A continuación, creamos un fichero en el directorio exportado. Esto nos servirá para ver el efecto de la opción de exportación "`root_squash`". Observa que al no tener permisos, el usuario root no puede siquiera crear el fichero.

   ```
   # cd /home
   # touch probando_root_squash
   touch: no se puede efectuar 'touch' sobre «probando_root_squash»: Permiso denegado
   ```

B2.1.6. Exportando usando la herramienta gráfica

Usando la herramienta `system-config-nfs`, realiza en el equipo servidor los siguientes ejercicios:

1. Exporta el directorio `/home/software/f11DVD`, sólo para el equipo cliente, haciendo mapeo del root, y en modo de sólo lectura.
2. Reinicia los demonios necesarios para que se active esta exportación.

Desde el equipo cliente, realiza el montaje de dicho directorio, selecciona las opciones de montaje oportunas según la exportación realizada.

1. Primero haz el montaje "manual" en el directorio `/software`. Crea previamente este directorio.
2. Edita `/etc/fstab` y configura que dicho directorio se monte en cada arranque.

B2.2. Boletín 2: Gestión básica de usuarios

B2.2.1. Objetivos

En este segundo boletín, vamos a realizar la creación de usuarios, la asignación de contraseñas, así como de información del campo GECOS. Este último dato, será importante cuando el número de usuarios del cluster crezca y sea necesario tenerlos identificados.

Un detalle interesante a tener en cuenta es que, normalmente, cuando se crea un usuario se suele crear un nuevo grupo para él, usando como nombre el mismo que el del usuario. Esto en un equipo en el sólo trabajan un par de usuarios, no tiene mucha importancia. Sin embargo, en un cluster es más interesante que todos los usuarios pertenezcan a un mismo grupo, o que una serie de usuarios pertenezcan a grupo concreto.

Este ejercicio sólo se tiene que realizar en el equipo servidor, ya que en el siguiente boletín configuraremos un sistema NIS para que los equipos cliente puedan acceder a los datos de usuarios definidos en el servidor.

B2.2.2. Plan de trabajo

El plan de trabajo para este segundo boletín será el siguiente:

1. Creación de usuarios y grupos.
2. Asignación de restricciones de tiempo.
3. Modificación y consulta del campo GECOS.
4. Creación de grupos y usuarios con `system-config-users`.

B2.2.3. Creación de grupos

.

1. Usando la orden `groupadd` crea un nuevo grupo llamado **cccgrp**.

```
# groupadd cccgrp
```

2. Usando la orden `useradd` crea un nuevo usuario, cuyo grupo principal sea `cccgrp`, y que pertenezca al grupo secundario **users**.

```
# useradd -g cccgrp -G users pilar
```

3. Asigna la contraseña "cluster" al nuevo usuario creado. A la hora de asignar una contraseña tienes que tener en cuenta que el sistema hace un control para evitar que las contraseñas sean fácilmente adivinables. El problema es que al usuario **root** le avisa de la debilidad de la contraseña elegida, pero no le prohibe usarla. Esto no sucede con un usuario "normal". Otro detalle importante es que si a un usuario le damos un contraseña "sencilla", no la cambiará, y seguirá siendo débil y fácil de adivinar.

```
# passwd pilar
```

4. Usando la órdenes `id` y `groups` comprueba que la creación del usuario y la asignación de grupos se ha realizado de forma correcta.

```
# id pilar
# groups pilar
```

B2.2.4. Asignación de restricciones de tiempo

En un sistema como un cluster, en ocasiones, los usuarios pueden ser temporales, por tanto es interesante la asignación de restricciones de tiempo, que nos ayuden a establecer cuando no se podrá seguir usando una cuenta. Así mismo, estas restricciones nos permitirán forzar a que los usuarios cambien cada cierto tiempo su contraseña. Usando `chage` resuelve:

1. Establece para el usuario creado en el ejercicio anterior las siguientes restricciones a su contraseña:
   - La contraseña tiene una duración mínima de 0 días, y máxima de 4 meses.
   - Cuando vaya a caducar la contraseña nos tiene que informar de que hay que cambiarla 10 días antes de la fecha.
   - Se permite que el usuario mantenga la contraseña caducada durante 15 días. Pasados esos días, la cuenta será bloqueada, si la contraseña no ha sido cambiada.
2. Establece que la cuenta anterior caducará, y será bloqueada, el 31 de diciembre de 2010.

B2.2.5. Modificación y consulta del campo GECOS

1. Usando la orden `chfn` cambia la información asignada al usuario anterior. Esta información es la que se guarda en el campo GECOS, y nos permite asociar una cuenta con la persona que la utiliza. Cuando el número de usuarios del sistema se incrementa de forma considerable, esto es bastante importante para poder tener a cada cuenta asociada con la persona correspondiente.

```
# chfn pilar
```

Usuario también puede cambiar la información que sobre él se almacena. Bastará con que indique su contraseña.

2. Comprueba si está instalado el paquete `finger`. En caso de que no sea así, realiza la instalación del mismo.

```
# rpm -q finger
# yum install finger                //Si no está instalado
```

3. A continuación, comprueba la información que se muestra del usuario.

```
# finger pilar
```

B2.2.6. Creación de usuarios con `system-config-users`

1. Crea un nuevo grupo llamado **prac** usando la herramienta gráfica `system-config-users`.
2. Crea un nuevo usuario usando la misma herramienta que tenga como grupo primario **prac**. A continuación, asigna restricciones de tiempo, y hazlo miembro del grupo **users**.

   Primero tienes que crear el usuario, y a posteriori modificar sus propiedades para cambiar sus restricciones de tiempo, asignarle grupos secundarios, etc.
3. Modifica la información almacenada para él en el campo GECOS, consultándola a posteriori.

B2.3. Boletín 3: Network Information Services

B2.3.1. Objetivos

Pasamos a realizar la instalación de un sistema NIS (*Network Information Service*) tanto en el servidor como en el cliente. Este sistema nos permitirá que los usuarios definidos en el equipo servidor puedan ser usados desde el cliente, como si fuesen local al mismo. Pero hay que tener en cuenta que se transfiere la información de configuración de los usuarios, pero no sus ficheros de trabajo. Para esto último NIS debe apoyarse en otros sistemas como NFS.

De nuevo es interesante destacar que fue desarrollado por Sun Microsystems, que originalmente lo llamó *Yellow Pages*, o YP, y que todavía se utiliza para referirse a él. Pero ese nombre estaba registrado, y lo tuvo que cambiar a NIS. Sin embargo YP permanece como prefijo en los nombres de la mayoría de las órdenes relacionadas con NIS, como `ypserv` e `ypbind`.

Es conveniente que antes de realizar este ejercicio se haya configurado la exportación y montaje del directorio `/home`, de cara a observar cómo NFS permite obtener los directorios de datos mientras que NIS permite todo lo referente a los usuarios.

De la misma manera, es importante que se haya resuelto el boletín número 2, para tener usuarios creados con los que hacer las pruebas.

B2.3.2. Plan de trabajo

El plan de trabajo para este segundo boletín será el siguiente:

1. Configuración del servidor NIS.
2. Configuración del cliente NIS.
3. Comprobación del funcionamiento.
4. Seguridad en un servidor NIS.

B2.3.3. Configuración del servidor NIS

1. En el servidor, los paquetes `ypserv`, `ypbind` e `yp-tools` deben estar instalados. En caso de que no lo estén, procede a instalarlos.

2. Permite la autenticación mediante NIS, usando para ello la herramienta `system-config-authentication`.

3. Comprueba si el demonio `rpcbind` está en ejecución, en otro caso lánzalo.

   ```
   # /etc/init.d/rpcbind status
   # /etc/init.d/rpcbind start
   ```

   Establece, además, que se active en cada arranque.

   ```
   # chkconfig rpcbind on
   ```

4. Activa el *domainname*. Es importante la elección del nombre del dominio, ya que en una misma red no deberían haber dos servidores NIS distintos, con el mismo nombre NIS.

   ```
   # nisdomainname ccc_nis
   ```

5. En el fichero `/etc/sysconfig/network` añade la siguiente línea. Esta línea es necesaria para que en tiempo de arranque se active el servicio NIS.

   ```
   NISDOMAIN="ccc_nis"
   ```

6. Añade en el fichero de configuración `/etc/yp.conf` las líneas:

   ```
   domain ccc_nis server 10.0.1.15
   ypserver 10.0.1.15
   ```

7. El siguiente paso es arrancar el demonio `ypserv` y activarlo para que se lance en el arranque para los niveles correspondientes.

8. Es el momento de crear las bases de datos que "distribuye" el servidor. Este paso hay que hacerlo cada vez que cambien los datos, por ejemplo cuando se añade o modifica un usuario. Para crear o actualizar las bases de datos simplemente hay que ejecutar la orden `make` en el directorio `/var/yp`.

9. Si se desea que los usuarios puedan cambiar su contraseña se necesita tener en ejecución el demonio `yppasswdd`. Activa dicho demonio y haz que se lance para los niveles de arranque oportunos.

10. Para comprobar que el servidor funciona correctamente, puedes hacer que el servidor sea cliente de sí mismo. Para ello, lanza primero el demonio `ypbind` y a continuación ejecuta la orden

    ```
    # ypcat passwd
    ```

    que debe mostrar un listado de los usuarios que son "exportados" utilizando las **NIS**. Entre ellos tiene que aparecer el usuario creado en el ejercicio anterior.

B2.3.4. Configuración del cliente NIS

1. En el cliente, los paquetes `ypbind` e `yp-tools` deben estar instalados. En caso de que no lo estén, procede a instalarlos.

2. Permite la autenticación mediante NIS, usando para ello la herramienta `system-config-authenticacion`.

3. Comprueba si el demonio `rpcbind` está en ejecución, si no lánzalo.

   ```
   # /etc/init.d/rpcbind status
   # /etc/init.d/rpcbind start
   ```

   Establece, además, que se active en cada arranque.

   ```
   # chkconfig rpcbind on
   ```

4. Activa el *domainname*.

   ```
   # nisdomainname ccc_nis
   ```

5. En el fichero `/etc/sysconfig/network` hay que añadir la siguiente línea, para que en tiempo de arranque se active el servicio NIS.

   ```
   NISDOMAIN="ccc_nis"
   ```

6. Añade en el fichero de configuración `/etc/yp.conf` la línea:

   ```
   domain ccc_nis server 10.0.1.15
   ```

7. Lanza el demonio `ypbind`, y configura el sistema para que en tiempo de arranque se lancen de forma correcta y para los niveles oportunos.

8. Comprueba, utilizando `ypcat passwd`, que el servidor nos acepta como cliente. Ten en cuenta, que si esta instrucción funcionó de forma correcta en el servidor. Si aquí presenta problemas es porque el error están en la configuración del equipo cliente.

B2.3.5. Comprobación del funcionamiento

Es hora de comprobar que lo configurado en los dos ejercicios anteriores funciona. Estos ejercicios se van a realizar en el equipo cliente.

1. En el equipo cliente, entra al sistema usando un usuario definido en el servidor. Prueba las órdenes `id` y `groups` vistas en el boletín anterior. Primero prueba con la orden `su -l usuario`. A continuación con el entorno gráfico.

   Si tienes problemas para entrar en modo gráfico es porque la exportación del directorio `/home/` desde el servidor se está haciendo en modo `root_squash`. En el servidor, cambia esa opción de exportación, pasándola a `no_root_squash`, reinicia el demonio de `nfs`. A continuación, en el equipo cliente, vuelve a hacer el montaje.

2. Comprueba que puedes escribir en el directorio `HOME` que tienes asignado, a pesar de que reside remótamente.

3. Cámbiale al usuario la contraseña asignada. Para que esto funcione, en el equipo cliente tienes que editar el fichero `/etc/hosts` y añadir la siguiente entrada:

```
10.0.1.15 ccc-server ccc-server
```

B2.3.6. Seguridad en el servidor NIS

El principal método para especificar la seguridad a través de **NIS** es indicar la red o los ordenadores que tienen permiso para acceder al dominio **NIS** en el fichero `/var/yp/securenets`.

Si dicho fichero está vacío o no existe, cualquier ordenador que conozca el nombre del dominio **NIS** y la dirección IP del servidor podría conectarse al dominio. Por el contrario si se especifica una red o bien un listado de ordenadores concretos, sólo esos ordenadores podrán acceder al dominio **NIS**.

1. Modifica el fichero `/var/yp/securenets` para que sólo se puedan conectar al servicio NIS el nodo de acceso del cluster. Los equipos de la red interna del cluster no tienen que tener acceso. Un ejemplo del contenido de dicho fichero sería:

```
# Permitir conexión al host local
host 127.0.0.1
#
# Permitir conexiones a la subred 10.0.3.0
255.255.255.0 10.0.3.0
#
```

   ¡OJO! Este trozo de fichero no es la solución al ejercicio.

2. Comprueba desde el equipo cliente que ahora no es posible conectarse al servidor NIS, puedes usar para ello simplemente la orden `ypcat`.

3. Prueba que desde el equipo servidor, que fue configurado como cliente de su propio NIS, sí puede conectarse al mismo.

4. Permite ahora que los equipos de la red 10.0.1.0, con submáscara 255.255.255.0, sí puedan conectarse al servicio NIS. Tienes que dejar permitido también al `localhost`.

5. Desde el equipo cliente comprueba que ya hay permiso de conexión en el servidor NIS.

B2.4. Boletín 4: Herramientas `rsh` y `pdsh`

B2.4.1. Objetivos

En este boletín se van a estudiar dos herramientas que nos permiten establecer una conexión remota a un equipo pero sin necesidad de hacer autenticación previa. Con estas herramientas también podemos ejecutar instrucciones en la máquina remota, o copiar/enviar ficheros a la misma, de nuevo sin necesidad de indicar la contraseña del usuario que ejecuta la orden.

Como en ejercicios anteriores, conviene tener el firewall desactivado. Y en caso de que queramos posibilitar que el usuario **root** haga uso de estas herramientas, tendríamos que deshabilitar el `Selinux`.

B2.4.2. Plan de trabajo

En este boletín vamos a realizar dos ejercicios:

1. Herramienta `rsh`.
2. Herramienta `pdsh`

B2.4.3. Herramienta `rsh`

1. Lo primero que vamos a hacer es instalar la herramienta en el equipo cliente.

   a) El primer paso es comprobar, en el equipo cliente, si están instalados los paquetes `xinetd` y `rsh-server`. En caso de que no sea así, procede a instalarlos. Así mismo, el paquete `rsh` debería también estar instalado.

   b) A continuación ampliamos las terminales que puede usar el **root**, fijando aquellas que son usadas por la herramienta `rsh`.

   ```
   # echo rsh >> /etc/securetty
   # echo rlogin >> /etc/securetty
   ```

   c) Cuando se instala el servicio de `rsh-server` por defecto está deshabilitado. Para habilitarlo usamos `chkconfig`.

   ```
   # chkconfig rsh on
   # chkconfig rlogin on
   ```

   d) Es necesario lanzar el demonio `xinetd` y configurar el sistema para que se active en cada arranque.

   ```
   # /etc/init.d/xinetd restart
   # chkconfig xinetd on
   ```

   e) Como último paso cada usuario tendrá que crear un fichero `.rhosts` para indicar a qué equipos permite el acceso. Es importante que a dicho fichero se le establezcan los permisos oportunos a dicho fichero.
   Con el usuario **ccc** ejecuta las siguientes órdenes:

   ```
   $ echo + > .rhosts
   $ chmod 600 .rhosts
   ```

2. Desde el equipo servidor, como usuario **ccc**, realiza los siguientes apartados:

   a) En el equipo servidor sólo tiene que estar instalado el paquete `rsh`. Compruébalo, y en caso de que no lo esté, haz la instalación.

   b) Desde el servidor, abre un shell en el equipo cliente, usando la orden `rlogin` y comprueba que realmente has accedido a la misma, y que puedes trabajar en el equipo remoto. A continuación cierra dicha shell.

   ```
   $ rlogin 10.0.1.16
   ```

   c) Usando la orden `rcp` copia un directorio o fichero del servidor al equipo cliente.

   d) Mediante la orden `rsh` ejecuta la orden `ls -lrt` en el equipo cliente y comprueba que el fichero se ha copiado.

B2.4.4. Herramienta `pdsh`

Esta herramienta es bastante útil de cara a la administración de un cluster porque nos permite ejecutar una misma orden en un conjunto de equipos, de forma paralela, y sin necesidad de dar la contraseña si tenemos configurado el sistema para el acceso mediante `rsh`.

En realidad se puede considerar que es una variante de `rsh`, pero con la posibilidad de ejecutar la instrucción en distintos equipos de forma paralela.

Junto que `pdsh` se instala `pdcp`, que configurado de la manera oportuna, permite también copiar ficheros de forma remota y en paralelo a varios equipos a la vez.

1. En el equipo cliente, una vez configurado para aceptar conexiones mediante `rsh`, no es necesario configurar nada más.

2. Por su parte, en el equipo servidor habrá que realizar las siguientes configuraciones previas.

   a) Instalar los paquetes `pdsh`, `pdsh-rcmd-ssh` y `pdsh-rcmd-rsh`.

   b) El problema que presenta esta herramienta es que `pdsh` con `rsh` no funciona para los usuarios normales pues hace falta usar un puerto privilegiado. Esto se puede solucionar de la siguiente manera, pero hay que tener en cuenta los riesgos para la seguridad que conlleva.

   Si se confía en los hosts y los usuarios, algo normal en un cluster, se habilita el uso de la opción `pdsh -R rsh` para los usuarios carentes de privilegios con las instrucciones:

   ```
   # cd /usr/bin/
   # chown root.bin pdsh
   # chmod u+s pdsh
   ```

   c) Ejecuta desde el equipo servidor con `pdsh` para que en el cliente se ejecute `ls -lrt` del directorio `/etc`, para ello puedes usar la orden:

   ```
   # pdsh -R rhs w 10.0.1.16 ls -lrt /etc
   ```

   Si tuviésemos varios equipos podríamos indicar una de las siguientes opciones:

   ```
   # pdsh -R rhs w 10.0.1.16,10.0.1.17 ls -lrt /etc
   # pdsh -R rhs w 10.0.1.1[6-7] ls -lrt /etc
   ```

3. Para trabajar con `pdcp`, que permite copiar ficheros de un equipo al conjunto de equipos indicados habría que realizar los pasos siguientes:

   a) Los paquetes `pdsh`, `pdsh-rcmd-ssh` y `pdsh-rcmd-rsh` tienen que estar instalados tanto en el cliente como en el servidor.

   b) De la misma manera que para `pdsh` asignamos el propietario y el grupo propietario correcto, y el permiso SUID. Lo tienes que hacer tanto en el equipo servidor como en el cliente.

   ```
   # cd /usr/bin/
   # chown root.bin pdcp
   # chmod u+s pdcp
   ```

   c) Por último, desde el equipo servidor podemos ejecutar estas órdenes:

   ```
   # pdcp -R rsh -w 10.0.1.16 fichero_origen directorio_destino
   # pdcp -R rsh -w 10.0.1.16 /etc/group /tmp
   ```

B2.5. Boletín 5: Gestión de las cuotas de usuario

B2.5.1. Objetivos

Un aspecto importante es controlar el espacio en disco que utilizan los usuarios. Una vez configurado un cluster, si el número de usuarios que acceden a él es alto, y además, si no se tiene un control de los mismos, podemos encontrarnos con problemas de disco. La mejor solución es configurar las cuotas de disco en el sistema de ficheros en el que los usuarios puedan trabajar. Estas cuotas permiten limitar el número de bloques y/o ficheros (nodos-i) que un usuario puede usar en una partición. Para cada caso podemos establecer 2 límites:

- El límite **límite hard** es el número máximo de bloques (o ficheros) que un usuario puede usar. No puede sobrepasarlo nunca. Cuando lo sobrepase ya no podrá usar más bloques o crear más ficheros.
- Por otro lado el **límite soft** es inferior al límite hard y se puede sobrepasar durante cierto tiempo, pero sin llegar al límite hard. Pasado ese tiempo es como si se hubiese superado el límite hard

B2.5.2. Plan de trabajo

En este boletín vamos a realizar dos ejercicios:

1. Configuración de las cuotas en el equipo servidor.
2. Probar las cuotas en el equipo cliente

B2.5.3. Configuración de las cuotas en el equipo servidor

1. Para el sistema de ficheros al que se asignan las cuotas, indicarlo en `/etc/fstab` con la opción `usrquota`:

   ```
   /dev/sda3   /         ext3     defaults,usrquota     1  1
   ```

2. Remontar la partición para que se active la opción **usrquota** con la orden:

   ```
   mount -o remount /
   ```

   Es importante ejecutar a continuación la orden `mount` para comprobar que se ha fijado como nueva opción de montaje `usrquota`

3. El tercer paso es crear el fichero de control de cuotas.

   ```
   # quotacheck -nm /
   ```

   En la máquina virtual es posible que haya un error, y no te permita crear los ficheros de cuota. Desmonta, con la orden `umount` el sistema de ficheros que indica.

4. A continuación se activan las cuotas mediante `quotaon -a`

5. Por último para cada usuario se activará su cuota, usando `quota username`. Establece como límite *hard* 100MB y límite *soft* 90 MB. Si hemos realizado los pasos de forma correcta se abrirá el editor de textos `vi` y aparecerá algo así:

```
Disk quotas for user pilar (uid 500):
  Filesystem    blocks     soft   hard   inodes   soft   hard
  /dev/sda2     2608464      0      0      11431     0      0
```

Los $0's$ indican que no hay cuota establecida. El valor numérico que aparece debajo de "blocks" y de "inodes" es la cantidad de bloques y nodos–i, respectivamente, que ya está consumiendo el usuario.

6. Es interesante saber que la opción `-p` de la orden `edquota` permite copiar la cuota asignada previamente a un usuario a otros usuarios. Por ejemplo, para copiar la cuota del usuario **pilar** a los usuarios **juan**, **luis** y **domingo** usaremos:

```
# edquota -P pilar juan luis domingo
```

B2.5.4. Probar las cuotas en el equipo cliente

1. Con un usuario creado en el boletín número 2, conéctate al equipo cliente, y comprueba, en primer lugar, el valor de cuotas que tienes asignado. Para ello, basta con que ejecutes la orden `quota`. Esta instrucción mediante NFS no funciona porque `Selinux` está impidiendo la ejecución de la misma. Habría que desactivar `Selinux` en el servidor para que funcionase.

2. A continuación, haciendo usado de la orden `dd` crea dos ficheros de 120 MB.

```
$ dd if=/dev/zero  of=fichero1 bs=65536 count=1920
$ dd if=/dev/zero  of=fichero2 bs=65536 count=1920
```

   ¿Qué sucede?

3. Comprueba qué ha pasado con la cuota asignada.