NFS - монтируем сетевые шары

Опубликовано muff в Пт, 2010-12-24 18:54



На видеосервере закончилось свободное место, все свободные слоты под HDD тоже оказались заняты. Выход из ситуации - монтирование сетевой файловой системы, так называемой Network File System.

Любое NFS-соединение работает по клиент-серверной схеме. Один компьютер является NFS-сервером и предоставляет свои файловые системы другим серверам. Это называется "NFS-экспортированием", а предоставляемые файловые системы называют "экспортами". Клиенты могут монтировать экспорты NFS-сервера почти точно так же как и локальные файловые системы.

Еще одним замечательным свойством NFS является отсутствие привязки клиента к текущему состоянию сервера (stateless). Это возможность позволяет даже перезагрузить NFS-сервер, а клиенты при этом не отвалятся. Само собой подразумевается, что в это время доступ к экспортам ограничен, однако как только NFS-сервер возобновит работу, клиенты смогут продолжать работу.

Итак, как можно было догадаться, необходимо будет настроить NFS-сервер и NFS-клиента.

NFS-server

Для запуска NFS-сервера неообходимо, чтобы ядро было собрано с опцией **NFSSERVER**.

Если ядро пересобирать неохота, то можно подгрузить модуль ядра:

```
# kldload /boot/kernel/nfsserver.ko
```

Для автоматической подгрузки модуля после перезагрузки сервера, воспользуемся следующей командой:

```
# echo 'nfsserver_load="YES"' >> /boot/loader.conf
```

Следующий шаг - создание файла **/etc/exports**, в котором опишем локальные точки монтирования, доступные для монтирования клиентами. Мой файл получился следующего вида:

```
/db4 -maproot=root 192.168.100.19
```

где

- **/db4** - локальная точка;
- **-maproot=root** - назначение прав. Обозначаем локального пользователя, права которого будут использоваться;
- **192.168.100.19** - хост, которому разрешено монтирование (перечисляем через пробел);

Для более детального ознакомления с синтаксисом файла обращаемся к странице руководства. Кстати, стоит иметь ввиду, что в одной строке можно прописать максимум три точки. Если точек больше, то описываем их в следующей строке, но не более чем три точки на одну строку в файле. Также не стоит забывать, что после внесения изменений в файл **/etc/exports**, необходимо проинформировать о изменениях демон mountd, послав ему сигнал HUP, либо же перезапустив демон:

```
# kill -HUP `cat /var/run/mountd.pid`
# /etc/rc.d/mountd reload
```

В rc.conf вносим следующие записи:

```
rpcbind_enable="YES"
rpcbind_flags="-l"
mountd_enable="YES"
mountd_flags="-r -l"
nfs_server_enable="YES"
nfs_server_flags="-n 3 -h 192.168.159.250"
```

rpcbind - удаленный вызов процедур

- **l** - включение логгирования запросов;

nfsd - демон сервера NFS

- **n** - максимальное количество подключенных клиентов ;
- **h** - на каком адресе "биндить" сервис. Можно указать несколько адресов. Если ключ не указан, то то сервис "слушает" на всех интерфейсах;

mountd - демон монтирования. Принимает подключения от клиентов.

- **r** - допускается обслуживание файлов, а не только каталогов;
- **l** - логгирование всех запросов на монтирование;

Для того, чтобы ознакомиться с допустимыми опциями, воспользуйтесь страницами руководств:

```
# man rpcbind
# man mountd
# man nfsd
```

Запускаем сервисы:

```
# sh /etc/rc.d/rpcbind start
# sh /etc/rc.d/mountd start
# sh /etc/rc.d/nfsd start
```

Для проверки, правильно ли экспортированы общие ресурсы NFS, воспользуемся командой **showmount -e**:

```
# showmount -e

Exports list on localhost:
/db5                            192.168.100.19
```

NFS-клиент

Для запуска NFS-клиента, неообходимо, чтобы ядро было собрано с опцией **NFSCLIENT** и, по возможности, с опцией **NFSLOCKD**.

Немного о опции NFSLOCKD из обзора к релизу FREEBSD 7.1: В ядро добавлена реализация клиентской части функциональности **rpc.lockd**, используемом для организации блокировок в **NFS**. Реализация поддерживает восстановление состояния блокировок на стороне клиента после рестарта **NFS** сервера, а также гарантированный сброс кэша перед установкой блокировки, что позволяет нескольким клиентом устанавливать файловые локи при одновременном использовании данных. Возможность включается через опцию **NFSLOCKD** в конфигурации ядра, если ядро пересобрано с поддержкой **NFSLOCKD** процесс **rpc.lockd** автоматически это определяет и начинает использовать.

Итак, если перезагружать сервер неохота, подгружаем модуль:

```
# kldload /boot/kernel/nfsclient.ko
```

Добавляем в **rc.conf** такие строки:

```
nfs_client_enable="YES"
nfs_client_flags="-n 3"
```

Теперь примонтируем сетевой ресурс:

```
# mount_nfs 192.168.159.250:/db5 /db5
```

Проверяем, примонтировался ли ресурс:

```
# df -h


Filesystem            Size    Used   Avail Capacity  Mounted on/dev/ad4s1a
 496M    157M    299M    35%    /devfs                  1.0K    1.0K      0B   100%
  /dev/dev/ad4s1e           1.9G    1.6G    165M    91%    /home/dev/ad4s1d
   15G    6.4G    7.0G    48%    /usr/dev/ad4s1f            24G    5.0G     17G    2
2%    /var/dev/ad4s1g             30G     27G    800M    97%    /db1/dev/ad6s1d
     902G    828G    1.6G   100%    /db2/dev/ad7s1d           902G    830G    264M
 100%    /db3/dev/ad5s1d           1.3T    1.2T     11G    99%    /db4192.168.159.25
0:/db5 451G   8.0K   415G   0%   /db5
```

"Последний штрих" - добавим автоматическое монтирование русурса при запуске системы. Для этого добавим в **/etc/fstab** такую строку:

```
195.3.159.250:/db5     /db5        nfs     rw,-b,-i      0      0
```

- **b** - перевод монтирования в бекграунд. Если не удалось примонтировать систему сразу, сервер продолжает загрузку (а не ждет монтирования), и в фоне продолжает

попытки примонтировать файловую систему;
- **i** - разрешается посылать сигнал прерывания. Например, если сетевой диск "отвалился", и сервер "подвис на выполнении определенной команды (например, ls), то можно послать сигнал прерывания с помощью сочетания клавиш **Ctrl+C**.

ПРИМЕЧАНИЯ:

- каталог, в который монтируется ресурс, должен существовать;
- монтирование выполняется поверх существующих локальных ресурсов. Тоесть, если примонтировать ресурс в директорию с данными, то локальные данные не будут доступны до момента размонтирования;
- для просмотра подробной статистики можно воспользоваться утилитой **nfsstat**;

Источник (получено *2026-11-01 17:26*):
http://muff.kiev.ua/content/nfs-montiruem-setevye-shary