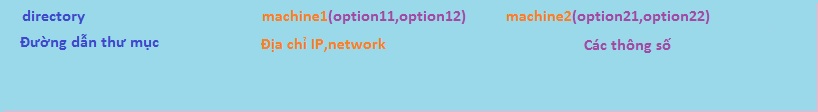
1. **Cài đặt NFS server**

Nfs-server

yum -y install nfs-utils rpcbind

vi /etc/export

directory machine1(option11,option12) machine2(option21,option22)



Ví dụ:

/shareNFS 192.168.0.0/255.255.255.0(rw,sync,no\_root\_squash)

**Các thông số**

* **ro:** read onlychỉ cho phép nfs-client đọc thư mục, các client không thể ghi, thường mặc định
* **rw:** read-write cho phép nfs-client có thể đọc và ghi thư mục.
* **no\_root\_squash:** Theo mặc định, bất lỳ file nào yêu user root tạo trên máy Client được xử lý user nobody tạo trên server. (Một cách chính xác, UID nào mà yêu cầu được phản xạ tùy thuộc vào UID của user "nobody" trên server, chứ không phải client.). Nếu chúng ta cấu hình **no\_root\_squash** thì User root trên client có quyền như user root trên server. Điều này không tốt cho bảo mật, trừ một số trường hợp chúng ta muốn làm việc với quyên administrator trên máy client. Chúng ta không nên chọn option này nếu không cần thiết.
* **no\_subtree\_check**: được mặc định**.** kiểm tra cây thư mục có trong ổ đĩa, disable chức năng này thì chuyển file nhanh hơn
* **sync:** tiến hành quá trình đọc và viết. ở đây chúng ta có 2 tham số **sync** vs **async** với các cách hoạt động khác nhau.trong các version nfs-util từ trước tới version1.0.0 async là giá trị mặc định. Trong tất cả các version sau 1.0.0 thì sync thì được sử dụng mặc định

Service nfs restart

Nfs-client

mount 192.168.0.250:/shareNFS /mnt/nfs/shareNFS

mount 192.168.0.250:/shareNFS w:

[root@nfs-client ~]# df -h

Filesystem Size Used Avail Use% Mounted on

/dev/mapper/vg\_nfsclient-lv\_root

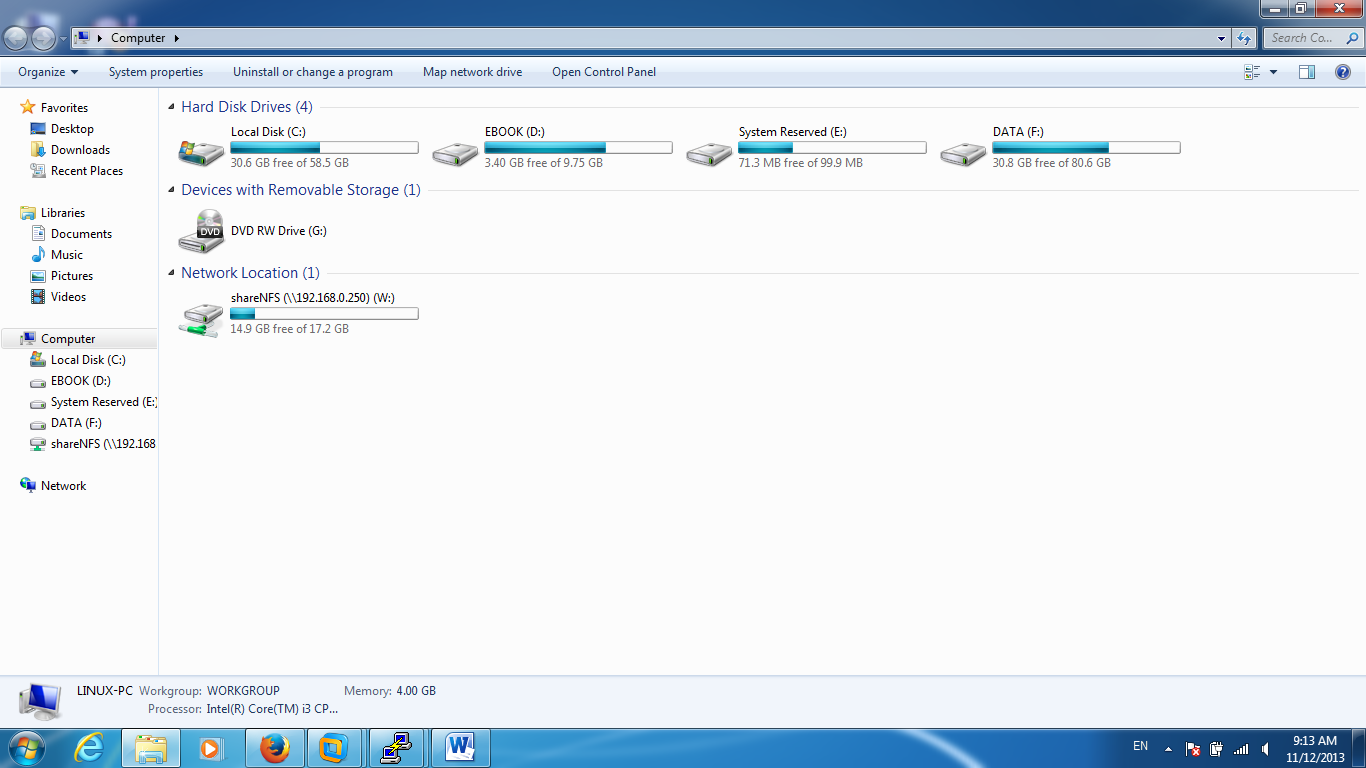
18G 2.3G 15G 14% /

tmpfs 495M 228K 495M 1% /dev/shm

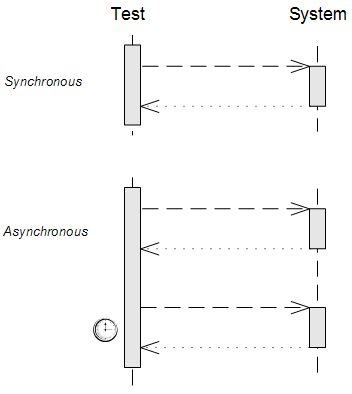
/dev/sda1 485M 33M 427M 8% /boot

/dev/sr0 4.1G 4.1G 0 100% /media/CentOS\_6.4\_Final

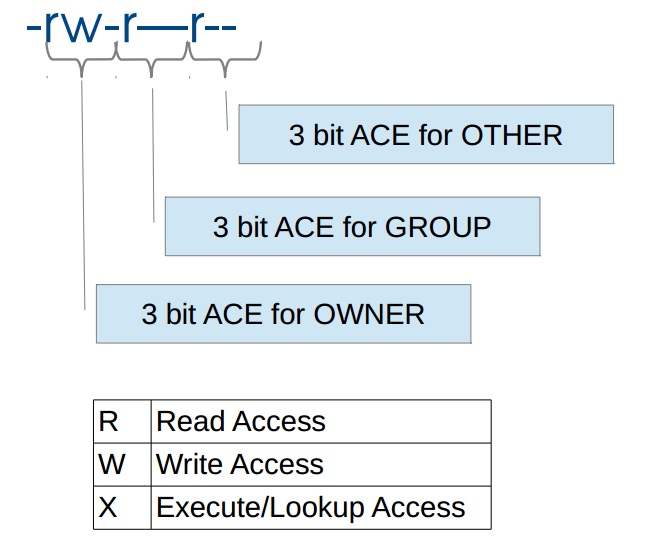
192.168.0.250:/shareNFS



1. **so sánh sync & async**
   1. **async:** Sự lựa chọn này cho phép Server NFS can thiệp đến giao thức NFS và đáp ứng các yêu cầu trước khi có bất cứ thay đổi nào khi các yêu cầu đo cam kết trong bộ trong bộ nhớ ổn đinh (ví dụ: ổ đĩa).sử dụng tham số async có thể thường có hiệu suất cao, nhưng nó dễ dẫn tới tình trạng mất dữ liệu hoặc hư file khi server bị restart.
   2. **sync:**  Đáp ứng các yêu cầu chỉ sau khi các thay đổi được xác nhận trong bộ lưu trữ ổn định.

****

1. **ACL cơ bản cho file hệ thống unix.**



- : Không được cấp quyền  
d : Ký hiệu Directory (Folder, thư mục)  
r : Quyền Read (đọc)  
w : Quyền Write (ghi)  
x : Quyền Execute (thực hiện)

**ACL**: ( access control list) là danh sách các ACE được đính kèm trong một file hoặc thư mục

Thiết lập access control lists trong NFS.

Cách sử dụng lênh setfacl

Lệnh **setfacl** lệnh này thiết lập **ACL** cho các tập tin và thư mục trong **NFSv4**, các tùy chọn hay sử dụng theo sau lệnh setfacl nhất là –x, -m. trong 1 dòng lệnh chúng ta có thể viết nhiều lệnh thực thi cùng lúc, các lệnh cách nhau bằng dấu ”,”.

Cấu trúc lệnh **setfacl**:

**Setfacl** –**tùy chọn…….**

Ví dụ :

setfacl -m u:lisa:r file

Dưới đây là bảng tóm tắt các tùy chọn:

|  |  |
| --- | --- |
| Tùy chọn | Mô tả |
| -m, -M | -m (định dạng), -M (định dạng file) Tùy chọn định dạng ACL của file hoặc thư mục, theo sau tùy chọn này phải gắn quyền truy cập |
| -x, -X | -x (gỡ bỏ), -X (gỡ bỏ file) tùy chọn này giúp chúng ta gỡ bỏ ACL cho tập tin hay thư mục |
| -b | Gỡ bỏ tất cả ACL mở rộng. các user, nhóm sở hữu được dữ lại |
| -k | Gỡ bỏ các ACL mặc định, nếu mặc định không tồn tại thì không có cảnh báo |
| -d | Đưa các ACL về mặc định |
| -v | Xem version setfacl và thoát |
| -h | Hiển thị giải thích các dòng lệnh |
| -- | Kết thúc tùy chọn dòng lệnh, các thông số còn lại được hiểu như là file |
| - | Nếu một dấu “-“ duy nhất thì chúng ta thiết lục thư mcj có quyền đọc |

ACL:

setfacl -m "u:username:permissions"

# setfacl -m "u:uid:permissions"

# setfacl -m "g:groupname:permissions"

Remove all permissions:

# setfacl -b

Remove each entry:

# setfacl -x "entry"

To check permissions use:

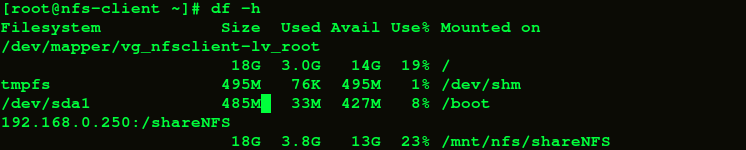
# getfacl filename

1. **Chỉnh Hiệu năng NFS**

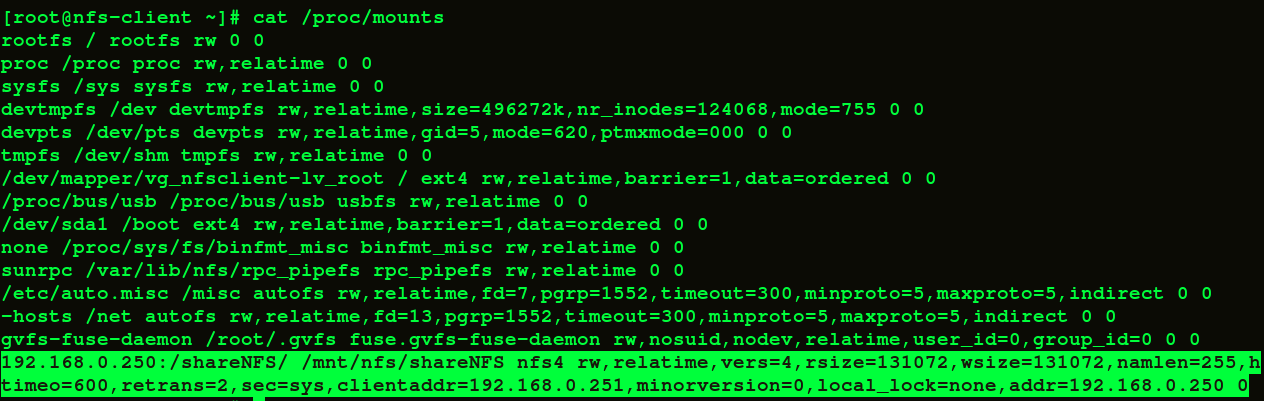
Khi chúng ta xây dựng một hệ thống bất kỳ, điều chúng ta chú ý nhất là năng suất và hiệu năng của nó mang lại. Làm sao cho một hệ thống chúng ta có chi phí thấp nhất mà vẩn mang lại hiệu năng cao nhất. Hệ thống NFS của chúng ta cũng vậy. Hiệu năng của NFS phụ thuộc vào rất nhiều yếu tố. Chúng ta có thể cải thiện hiệu năng của NFS bằng các cách sau.

* 1. **Điều chỉnh kích thước các khối với lệnh mount**

Kích thước data truyền qua lại giữa máy chủ và máy client là rất quan trọng, hầu hết các phiên bản NFS đều thiết lập giá trị mặc đinh này. Tuy nhiên khi sử dụng chúng ta phải điều chỉnh giá trị này lại cho phù hợp với nhu cầu của chúng ta. Chúng ta điều chỉnh các thông số như sau.



Chúng ta hãy nhìn các thông số mặc định tùy chọn của máy cient NFS tại file /proc/mounts



Đây là các thông số được gắn mặc định trong máy client khi chúng ta mount

**Rw**: thông số rw của file hệ thống cho phép chúng ta ở chế độ read/write.

**Vers=4**: version NFS chúng ta đang sử dụng.

**Rsize=131072 & wsize=131072**: xác định thời gian đọc và viết một gói dữ liệu gói RPC. Điều chỉnh giá trị này có thể giúp ta tang hiệu suất và cũng có thể ngược lại. Điều chỉnh các giá trị **rsize** & **wsize** phải luôn luôn diều chỉnh cho phù hợp với mạng của chúng ta, phù hợp với năng lực xử lý của máy chủ và client. Khi chúng ta giảm kích thước các gói rsize và wsize sẽ làm tang kích thước các gói RPC gởi qua mạng. Ví dụ chúng ta có 1 MB dữ liệu ta chia nhỏ nó ra nhiều phần 32KB thì chúng ta sẽ gửi nhiều lần hơn so với chúng ta chia dữ liệu 64 KB. Như vậy nếu chúng ta tang kích thước thì chúng ta sẽ giảm số lần gửi. Vì vậy việc tinh chỉnh các gói rsize và wsize phụ thuộc vào năng lực mạng lưới mạng. nếu giả sử ta có mạng máy chủ 1GB và máy client 1GB thì chúng ta có thể chỉnh giá trị **rsize** và **wsize**  cao hơn.

Version 4 chúng ta có thể đặt giá trị cao nhất cho **rsize**& **wsize**=1048576,v3=65536

Ta điều chỉnh giá trị rsize & wsize lúc mount

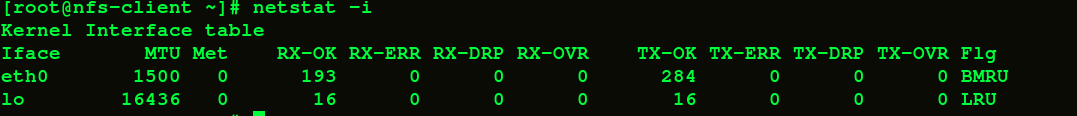


Phương pháp tốt nhất chúng ta chọ lựa kích thước rsize & wsize kiểm tra tới khi dặt kết quả tốt nhất.

* 1. **Định dạng tham số MTU cho Mạng.**

**MTU** viết tắt của chữ Maximum Transmission Unit. Số frame lớn nhất có thể gửi qua mạng. Hầu hết các máy tính đều có cấu hnhf mặt định là 1500.

Để xem MTU của card mạng, chúng ta chạy lệnh như bên dưới.

****

Giả sử chúng ta đã thay đổi giá trị của wsize và rsize là 8 KB và ta đang sử dụng MTU có kích thước là 1500 thì dữ liệu chúng ta phân mảnh và gửi đi với kích thước tối đa chỉ 1500 bytes .

Trong trường hợp này chúng ta thay đổi kích thước MTU=9000 thì toàn bộ 8 KB đọc và ghi toàn bộ dữ liệu sẽ được gủi đi mà không bị phân mảnh.(chú ý thông số MTu cần thay đổi cho cả server và client).

Thay đổi thông số MTU một cách dễ dàng bằng cách thêm dòng MTU=9000 vào file /etc/sysconfig/network-scripts/ifcfg-eth0

* 1. **Tùy chỉnh tham số timeo và retrans trong NFS**

hai tùy chọn ở trên ảnh hưởng đến số lượng thử lại của client đến máy chủ trong trường hợp của một phản ứng chậm từ máy chủ hoặc đôi khi không có phản hồi từ máy chủ .

**timeo** tùy chọn trong NFS quyết định thời gian client cần phải chờ đợi trước khi nói đến một kết luận rằng nó phải truyền lại các gói tin.(0.7 là giá trị tính theo phần 10 của giây, chúng ta có thể chỉnh lại giá trị 5/10 của giây, đây là thời gian chờ đợi để quyết định cần gửi lại gói tin).

Retans xác đinh số lần gửi lại các gói RPC( ví dụ retrans=2 số lần gửi lị các gói RPC là 3 lần, mỗi lần nó sẽ chờ đợi thời gian mà chúng ta đã đặt cho timeo ở trên).

Chúng ta có thể sửa đổi timeo và retrans giá trị như một tùy chọn trong lệnh mount như là.



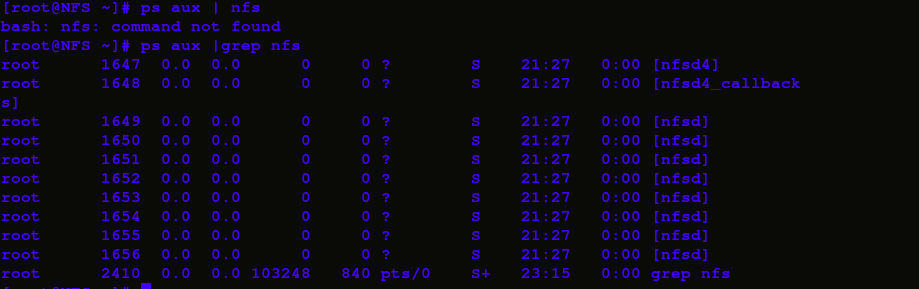
Kiểm tra lại bằng lệnh :

nfsstat –r

* 1. **Số thread NFS trong NFS server.**

Một trong những việc quan trọng nâng cao hiệu năng của NFS server là chúng ta tăng số thread sẳn có trên máy chủ. Chúng ta có lượng lớn các khách hang truy cập vào trong máy chủ NFS do vậy để tốt hơn chúng ta nên tăng threat trên máy chủ.

Chúng ta xem thread có sẳn trên máy chủ bằng lện sau:

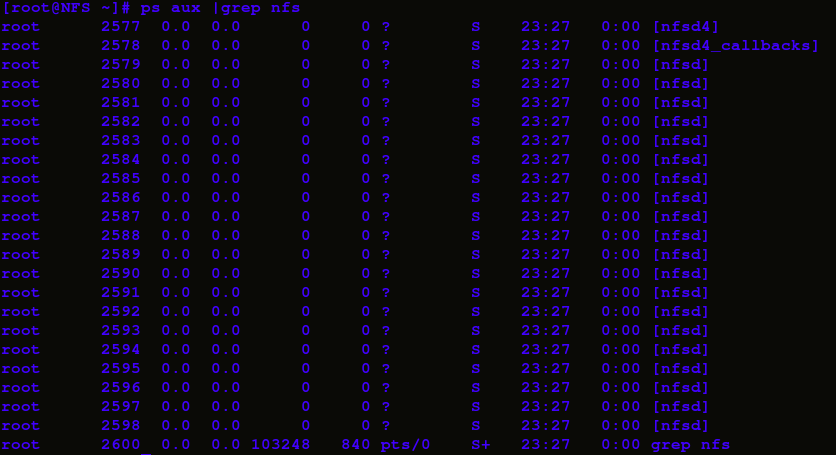


ở đây mặc đinh chúng ta có 8 thread chúng ta có thể tăng lên 20 giúp hệ thống chúng ta hoạt động tốt hơn, chúng ta sữa file

vi **/etc/sysconfig/nfs**



Sau đó thử lại với lệnh sau:



* 1. **Lựa chọn sync hoặc async trong lúc mount**

Sync vs async hai kiểu đồng bộ dữ liệu khác nhau. Chúng ta lựa chọn đồng bộ cho các dữ liệu khác nhau để tăng hiệu năng NFS. Ví dụ như với dữ liệu không quan trọng chúng ta có thể đồng bộ bằng async. Giúp nhanh hơn

Chúng ta có thể lựa chọ kiểu đồng bộ với lệnh mount như sau:



Chỉnh trong file vi /etc/export

* 1. **Điều chỉnh đầu vào ra tăng hiệu năng cho NFS**

Khi chuyển tập tin có dung lượng lớn qua mạng chúng ta cần máy có bộ nhớ lớn. Tuy nhiên trên linux mặc định không phân bổ lượng bộ nhớ lớn cho NFS, vì nó còn phân bổ cho các ứng dụng khác, chúng ta có thể điều chỉnh giá trị cao hơn nếu có đầu card mạng có dung lượng lớn.

Có hai giá trị mà chúng ta có thể sửa đổi chúng, đó là socket đợi đầu ra và socket đợi đầu vào. Giả sử chúng ta có có lượng lớn thread NFS như trên là 20 thread, mỗi client yêu cầu xử lý riêng biệt và các socket đợi đầu vào và đầu ra của các client giống nhau. Có nghĩa là chúng ta có đầu vào cao hơn số lượng socket hàng đợi(lúc này dữ liệu không phải đợi), điều này giúp ta hiệu quả hơn trong việc gửi nhận dữ liệu. chúng ta có thể chỉnh nó trong file sysctl.conf , hoặc trực tiếp trong thư mục /proc (sau đó phải restart nfs để hiệu quả).

echo 219136 > vi /proc/sys/net/core/rmem\_default

echo 219136 > vi /proc/sys/net/core/rmem\_max

chúng ta chỉnh sữa các giá trị hàng đợi vào và ra bằng cách thay đổi giá trị file wmem\_default & wmem\_max

|  |  |
| --- | --- |
|  | echo 219136 > /proc/sys/net/core/wmem\_default  echo 219136 > /proc/sys/net/core/wmem\_max |

Các giá trị ta chỉnh trong mục /proc là tạm thời, bởi vì nó lưu trực tiếp trên RAM(nó sẽ mất khi chúng ta restart computer). Nếu muốn các thông số này được lưu vĩnh viễn chúng ta phải cấu hình file sysctl.conf như bên dưới.

[root@slashroot2 ~]#echo 'net.core.wmem\_max=219136' >> /etc/sysctl.conf

[root@slashroot2 ~]#echo 'net.core.rmem\_max=219136' >> /etc/sysctl.conf

* 1. **Giải pháp lưu trữ cho máy chủ NFS**

Cấu hình các ổ đĩa trên máy chủ giúp chúng ta tăng hiệu năng đáng kể, việc này phụ thuộc khác nhiều vào tốc độ đọc nghi khi ta cấu hình RAID, các mức độ RAID khác nhau đem lại hiệu năng khác nhau