**NFS网络文件系统**

* NFS网络文件系统
  + NFS 系统介绍
  + NFS 服务端配置
    - 配置/etc/exprots
  + NFS客户端
  + 内核性能相关优化
  + 相关命令

**NFS 系统介绍**

network file sysctem 是网络文件系统，它的主要功能是通过网络让i同的主机系统之间可以共享文件或目录，NFS客户端一般为应用服务器，RPC (remote procedure call) 是能使客户端执行其他系统中程序的一种机制。NFS本身是没有提供信息传输的协议和功能的，但NFS却能让我们通过网络进行资料的分享，这是因为NFS使用了一些其它的传输协议。而这些传输协议用到这个RPC功能的。。所以只要用到NFS的地方都要启动RPC服务，不论是NFS SERVER或者NFS CLIENT。这样SERVER和CLIENT才能通过RPC来实现PROGRAM PORT的对应。可以这么理解RPC和NFS的关系：NFS是一个文件系统，而RPC是负责负责信息的传输。在大企业可以用 MFS, FASTDFS, GLUSTERFS, CEPH

在客户端与服务端都需要安装NFS与RPC

$ sed -i 's/keepcache=0/keepcache=1/g' /etc/yum.conf  
$ sed -n '1,4p' /etc/yum.conf  
$ yum install nfs-utils rpcbind -y

在服务端启动rpcbind 与nfs, 在客户端启动rpcbind

$ /etc/init.d/nfs start  
$ /etc/init.d/rpcbind start

**NFS 服务端配置**

NFS的常用目录

* /etc/exports NFS服务的主要配置文件
* /usr/sbin/exportfs NFS服务的管理命令
* /usr/sbin/shownmount 客户端的查看命令
* /var/lib/nfs/etab NFS分享出来的目录的完整权限设定值
* /var/lib/nfs/xtab 记录曾经登录过的客户端信息

**配置/etc/exprots**

<输出目录> [客户端1 选项（访问权限,用户映射,其他）] [客户端2 选项（访问权限,用户映射,其他）]

目录： NFS系统中需要共享给客户机使用的目录   
客户端：其中客户端可以指定IP或者指定网段，也可以指定载名中的主机。 （\*）表示所有主机   
选项用来设置输出目录的访问权限，用户映射

1. 选项： ro 只读， rw 读写
2. 用户映射：   
   * all\_squash: 将远程访问的所有普通用户及所属组映射为匿名用户或组(nfsnobody)
   * no\_all\_squash:
   * root\_squash: 将root用户及所属组都映射为匿名用户或组（默认设置）
   * no\_root\_squash
   * anonuid=xxx: 将远程访问的所有用户都映射为匿名用户，并指定该用户为本地用户
   * anongid=xxx
3. 其它选项：   
   * secure: 限制客户端只能从小于1024的tcp/ip端连接nfs服务器（默认设置）
   * insecure： 大于1024
   * sync: 将数据同步写放内存缓存区与磁盘中，效率低。但保证数据一致性
   * async:先保存到内存缓存中，必要时写入磁盘
   * wdelay: 检查是否有相关的写操作，如果有则将这些写操作一起执行，（默认设置）
   * no\_wdelay: 若有写操作则立即执行，与sync配合使用
   * subtree: 若输出目录是一个子目录，则nfs服务器将检查其父目录的权限（默认设置）
   * nosubtree: 不检查父目录的权限。这样可以提高效率

$ cat /etc/exports  
/data 172.16.1.0/24(rw, sync)

创建文件修改权限

mkdir -p /data  
chown -R nfsnobody,nfsnobody /data

然后平滑重启，

$ /etc/init.d/rpcbind reload  
$ /etc/init.d/nfs reload  
$ exportfs  
/data 172.16.1.0/24  
  
$ cat /var/lib/nfs/etab  
/data 172.16.1.0/24(rw,sync,wdelay,hide,nocrossmnt,secure,root\_squash,no\_all\_squash,no\_subtree\_check,secure\_locks,acl,anonuid=65534,anongid=65534,sec=sys,rw,root\_squash,no\_all\_squash)  
  
$ service rpcbind status  
rpcbind (pid 1376) is running...  
  
$ service nfs status  
rpc.svcgssd is stopped  
rpc.mountd (pid 1583) is running...  
nfsd (pid 1599 1598 1597 1596 1595 1594 1593 1592) is running...  
rpc.rquotad (pid 1578) is running...

然后可以用showmount 来时候NFS的共享状态

$ showmount -e #默认查看自己共享服务  
Export list for NFS01:  
/data 172.16.1.0/24  
  
$ showmount -a #查看已经与客户端连接上的目录信息  
All mount points on NFS01:

**NFS客户端**

客户端仅需要启动rpcbind，然后可以用showmount来查询nfs的共享状态

$ showmount -e 172.16.1.31   
Export list for 172.16.1.31:  
/data 172.16.1.0/24

mount 语法mount NFS服务器IP:共享目录 本地挂载点目录   
现在我们把客户端的/mnt挂载到服务器的/data/test/上

$ mount -t 172.16.1.31:/data/test/ /mnt/  
$ mount |grep nfs  
sunrpc on /var/lib/nfs/rpc\_pipefs type rpc\_pipefs (rw)  
172.16.1.31:/data/test/ on /mnt type nfs (rw,vers=4,addr=172.16.1.31,clientaddr=172.16.1.8)

这里我要说一下，如果客户端需要使用普通用户在共享目录写文件，可以在服务器里的共享目录下一个字目录给777权限，普通用户创建的文件的属主与组与该普通用户相同   
设定权限可以这样

mount -t nfs -o noatime,nodiratime,nosuid,noexec,nodev,rsize=131072,wsize=131072 10.0.0.7:/data /mnt

查看挂载信息

$ df -h  
Filesystem Size Used Avail Use% Mounted on  
/dev/sda2 6.9G 1.5G 5.1G 23% /  
tmpfs 238M 0 238M 0% /dev/shm  
/dev/sda1 190M 36M 145M 20% /boot  
172.16.1.31:/data/ 6.9G 3.7G 2.9G 57% /data/upload

**内核性能相关优化**

1、/proc/sys/net/core/rmem\_default   
该文件指定了接收套接字缓冲区大小的缺省值（以字节为单位），缺省设置：124928   
2、/proc/sys/net/core/rmem\_max   
该文件指定了接收套接字缓冲区大小的最大值（以字节为单位），缺省设置：124928   
3、/proc/sys/net/core/wmem\_default   
该文件指定了发送套接字缓冲区大小的缺省值（以字节为单位），缺省设置：124928   
4、/proc/sys/net/core/wmem\_max   
该文件指定了发送套接字缓冲区大小的最大值（以字节为单位），缺省设置：124928   
对上述文件的具体内核优化：

$ cat >>/etc/sysctl.conf<<EOF  
net.core.wmem\_default = 8388608  
net.core.rmem\_default = 8388608  
net.core.wmem\_max = 16777216  
net.core.rmem\_max = 16777216  
EOF  
  
$ sysctl -p

**相关命令**

1. exportfs   
   在启动NFS后，如果修改了/etc/exports，就需要重启nfs，这时候可以用exportfs命令来使之立刻生效 exportfs -rv 相当于 `/etc/init.d/nfs reload   
   * -a 全部挂找或卸载
   * -r 重新读取/etc/exports中的信息，并同步更新/etc/exports, /var/lib/nfs/xtab
   * -u 卸载单一目录（和a一起使用为卸载所有/etc/exports文件中的目录)
   * -v 在export的时候， 将详细的信息输出到屏幕上   
     1. nfsstat 查看NFS的运行状态，
     2. rpcinfo 查看RPC执行信息，可以用于检测RPC运行情况的工具
     3. showmount
   * -a 显示已经于客户端连接上的目录信息
   * -e IP或hostname显示此IP地址分享出来的目录   
     1. netstat
     2. mount 客户端挂载NFS共享目录可选参数 man -a mount
   * soft 当服务端没回应时，会在timeout后重新连接，会回错误信息
   * hard 当服务端没回应时持续在后台尝试连接
   * intr 正在进行NFS请求时，允许用键盘中断
   * nointr 不允许键盘中断
   * timeo 请求过期时间 单位为秒
   * bg 第一次请求不成功，第二次的mount将放到后台执行
   * fg 一直在前台发送请求
   * proto=tcp|udp 修改使用TCP协议还是UDP协议来传输NFS的数据
   * default ， hard nointr
   * rsize 设置块大小(bytes)
   * wsize 写操作的大小，这两个操作会影响服务器客户端NFS缓存大小

NFS优点   
1简单，容易上手，容易掌握。   
2NFS文件系统内数据是在文件系统之上的，即数据是能看得见的。   
3部署快速，维护简单方便，且可控，满足需求就是最好的。   
4可靠从软件层面上看，数据可靠性高，经久耐用。数据是在文件系统之上的。   
5服务非常稳定   
NFS局限：   
1、 局限性是存在单点故障，如果nfs server宕机了所有客户端都不能访问共享目录，这个在后期通过负载均衡及高可用方案弥补   
2、 在大数据高并发的场合，NFS效率/性能有限   
3、 客户端认证时基于ip和主机名、权限时，根据id识别，安全性一般（用于内网问题不大）   
4、 NFS数据时明文的，NFS本身对数据完整性不作验证   
5、 多台客户机器挂载一个NFS服务器时，连接管理维护麻烦（耦合度高），尤其是NFS服务端出问题后，所有NFS客户端都挂掉状态