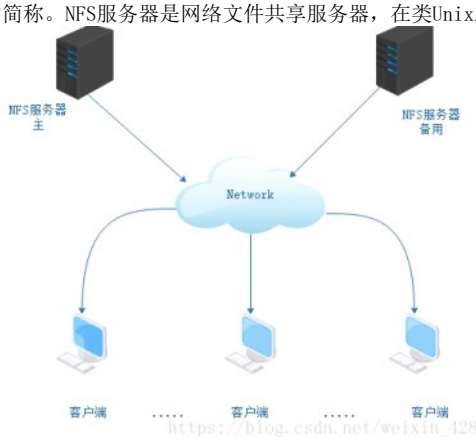
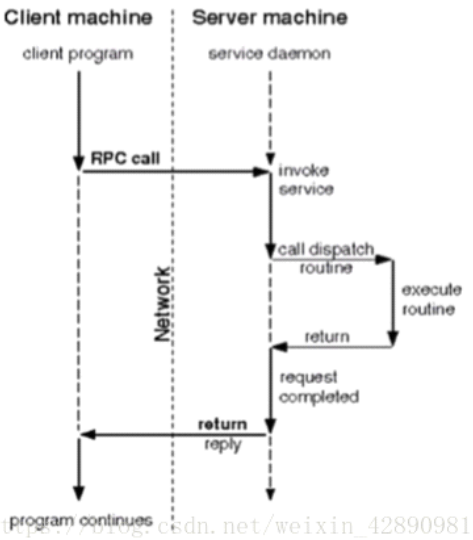
文件服务器之一：NFS服务器

NFS：network file system的简称。NFS服务器是网络文件共享服务器，在类Unix之间共享文件。



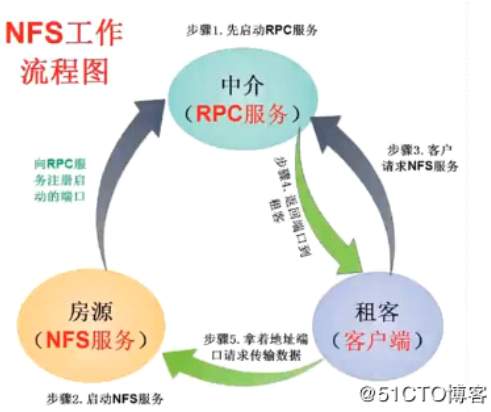
NFS的工作机制：

- NFS是基于RPC（远程过程调用）来实现网络文件系统的共享的
- RPC，远程过程调用协议，它是一种通过网络从远程计算机程序上请求服务，而不需要了解底层网络技术的协议
- RPC协议假定某些传输协议的存在，如TCP或UDP，为通信程序之间携带信息数据
- RPC采用客户机/服务器模式。请求程序就是一个客户机，而服务提供程序就是一个服务器



RPC工作机制如上图所示：

- 客户端程序发起一个RPC系统调用基于TCP协议发送给另外一台主机（服务端）
- 服务端监听在某个套接字上，当收到客户端的系统调用请求后，将收到的请求和其所传递的参数通过本地的系统调用执行一遍，并将结果返回给本地的服务进程
- 服务端的服务进程收到返回的执行结果将其封装成响应报文，再通过rpc协议返回给客户端
- 客户端调用进程接收答复信息，获得进程结果，然后继续执行



NFS的启动需要依赖RPC。客户端将网络中的NFS服务器共享的目录挂载到本地端的文件系统中。默认的NFS服务器的端口是2049。但是由于文件系统非常复杂，NFS服务器还有其他的程序去启动额外的端口，端口号是随机的，客户端通过RPC（远程过程调用）协议知道服务器启用了哪些端口。

服务端配置：

```
[root@test ~]# vim /etc/sysconfig/selinux  # 关闭SELinux
SELINUX=disabled
[root@test ~]# getenforce  #查看当前selinux的状态
[root@test ~]# systemctl stop firewalld    # 关闭防火墙
[root@test ~]# systemctl mask firewalld    # 禁用防火墙

[root@test ~]# systemctl start rpcbind.service   # 启动rpcbind
[root@test ~]# systemctl enable rpcbind
[root@test ~]# systemctl start nfs      # 启动nfs
[root@test ~]# systemctl enable nfs
[root@test ~]# ps -aux |egrep "rpc|nfs"

[root@test ~]# systemctl is-active nfs  # 查看nfs服务是否工作
[root@test ~]# rpcinfo -p localhost     # 查看rpc分配给nfs的端口，这些端口部分都是随机生成的
 # 因为端口部分是随机分配，所有当客户端连接nfs时，首先和rpc建立连接，rpc将分配的deamon端口号告诉客户端。[root@test ~]#
netstat -ntulp |grep rpc # 查看rpc的端口（与NFS相关的进程，默认端口号随机，可以固定[root@test ~]# vim
/etc/sysconfig/nfs）

创建共享文件的目录
[root@test ~]# mkdir /share
[root@test ~]# vim /etc/exports  # 配置nfs共享文件的共享方式等内容
#共享目录     主机范围（属性1，属性2。。。）
/share  192.168.88.0/24(rw,sync)     *(ro)     *.domain11.example.com(ro)      system?.domain11.example.com(ro)
注意：通配符只可以使用在主机名上
[root@test ~]# systemctl restart nfs

[root@test ~]# exportfs -arv   添加完共享目录后，需要重启服务器生效，可以使用这个命令直接让服务生效。
[root@test ~]# exportfs -auv   卸载所有目录
[root@test ~]# cat /var/lib/nfs/etab     查看挂载目录的默认权限
/share
192.168.88.0/24(rw,sync,wdelay,hide,nocrossmnt,secure,no_root_squash,no_all_squash,no_subtree_check,secure_locks,acl,a
nonuid=65534,anongid=65534,sec=sys,rw,secure,no_root_squash,no_all_squash)
rw 读写
ro 只读
sync  将数据同步写入磁盘
async 将数据异步写入磁盘。数据先写入缓存，不是直接写入磁盘中
    注意：在安装完nfs软件包后会自动创建nfsnobody用户
    [root@test ~]# id nfsnobody
    uid=65534(nfsnobody) gid=65534(nfsnobody) groups=65534(nfsnobody)
```

no_root_squash	使用nfs时，如果用户是root，不进行权限压缩，即root用户在nfs上创建的文件　属组和属主仍然是root（不安全，不建议使用）
root_squash	使用nfs时，如果用户是root，则进行权限压缩，即把root用户在nfs上创建的文件　属组和属主修改为nfsnobody
all_squash	所有的**普通用户**使用nfs都将使用权限压缩，即：将远程访问的所有普通用户及所属用户组都映射为匿名用户或者用户组（一般均为nfsnobody）
no_all_squash	所有的**普通用户**使用nfs都不使用权限压缩，即：不将远程访问的所有普通用户及所属用户组都映射为匿名用户或者用户组（默认设置）

例：/share 192.168.24.0/24(rw,sync，all_squash,anonuid=1000,anongid=1000)

客户端验证：

```
[root@test ~]# yum install nfs-utils -y
[root@test ~]# showmount -e 192.168.88.148            # 查看192.168.24.12服务器上共享的目录列表
[root@test ~]# mount -t nfs 192.168.88.148:/share /mnt/  # 客户端挂载服务器共享目录
[root@test ~]# vim /etc/fstab
192.168.88.143:/public /mnt/ nfs defaults 0 0
/home/public 192.168.88.*(rw)   *(ro)
/home/public 192.168.88.0/24(rw) *(ro)
/home/test 192.168.88.100(rw)

[root@test ~]# cat /etc/exports
/share 192.168.88.0/24(rw,sync,all_squash)
/xxx   192.168.88.0/24(rw,sync,no_all_squash)
/yyy   192.168.88.142(rw,async,no_all_squash,root_squash)
/ooo   192.168.88.0/24(rw,async,no_all_squash,no_root_squash)
/ttt   192.168.88.0/24(rw,async,all_squash,root_squash,anonuid=8848,anongid=1884
8)
```

/tmp *(rw,no_root_squash)	在所有的IP（主机）上登录的用户都可对NFS服务器上的共享目录/tmp拥有rw操作权限，同时如果是root用户访问该共享目录，那么不将root用户及所属用户组都映射为匿名用户或用户组（*表示所有的主机或者IP）
/tmp *(rw)	在所有的IP（主机）登录的用户都可对NFS服务器上的共享目录/tmp拥有rw操作权限
/home/public 192.168.0.*(rw) *(ro)	除了在192.168.0.0/24这个网段内的主机上登录的用户，可对NFS服务器共享目录/home/public进行rw操作权限，而其

/home/public 192.168.0.0/24(rw) *(ro)	它网段的主机上登录的用户，对NFS服务器共享目录/home/public只能进行r操作
/home/test 192.168.0.100(rw)	只对192.168.0.100该台主机设置权限，即：使在该台主机登录的用户都可对NFS服务器上的共享目录/home/test拥有读与写的操作
/home/linux *.linux.org(rw,all_squash,anonuid=40,anongid=40)	当*.linux.org（加入域linux.org的所有主机） 登陆此NFS主机，并且在/home/linux下面写入档案时，该档案的所有人与所有组，就会变成NFS服务器上的/etc/passwd文件里面对应的UID为40的那个身份的使用者了（因为指定了参数：all_squash，anonuid=40，anongid=40）

用户映射：

通过NFS中的用户映射，可以将伪或实际用户和组的标识赋给一个正在对NFS卷进行操作的用户。这个NFS用户具有映射所允许的用户和组的许可权限

对NFS卷使用一个通用的用户/组可以提供一定的安全性灵活性，而不会带来很多管理负荷

在使用NFS挂载的文件系统上的文件时，用户的访问通常都会受到限制，这就是说用户都是以匿名用户的身份来对文件进行访问的，这些用户缺省情况下对这些文件只有读权限

这种行为对于root

用户来说尤为重要。然而，实际上的确存在这种情况：希望用户以root用户或所定义的其他用户的身份访问远程文件系统上的文件

NFS 允许指定访问远程文件的用户 - 通过用户标识号（UID）和 组标识号（GID），可以禁用正常的squash行为。

NFS最大的问题是权限，因为在客户端和服务器端必须具备相同的账号才能够访问某些目录或文件。

客户端与服务器端UID与账号：

1. 客户端与服务器端具有相同的UID与账号
    a. 如果没有身份压缩，属主为客户端
    b. 如果身份压缩，属主为压缩后的身份
2. 客户端与服务器端具有不同的UID与账号
    a. 如果可以写入，文件属主显示UID
    b. 如果身份压缩，属主为压缩后的身份