

第29章 使用网络文件系统(NFS)

作者：Neal S. Jamison

本章内容包括：

- 什么是NFS
- 实现——NFS工作过程
- NFS使用的文件及命令
- 示例：共享及登录NFS文件系统
- 与NFS常见问题及解决方案
- 相关协议及产品

TCP/IP提供相应的协议及服务，允许多个计算机共享数据。这一特性被广泛用于互联网应用中，在众多此类应用中，最常见的TCP/IP应用是网络文件系统，简称NFS。

29.1 什么是NFS

NFS是分布式文件系统，它允许计算机通过TCP/IP网络共享资源。在NFS中，NFS客户端应用可透明地读/写位于NFS服务器上的文件。在NFS出现之前，计算机间共享数据的惟一途径是复制数据或集中使用数据。这种方式存在明显的缺陷——复制数据浪费存储空间而且存在实时一致性问题，集中使用数据意味着大型机及多个终端，而且只存在于单一网络中。

随着NFS的诞生，用户可以共享数据且与平台无关，所有上述问题迎刃而解。

注意 在本书中，文件系统指以目录结构有组织存放的一组文件的集合。例如，Unix系统中的/usr就是一个文件系统。

NFS服务器共享资源，使它们可被网上的客户使用。如下所述，NFS既可以用于LAN，也可以用于WAN，并且NFS与操作系统及硬件无关，它允许各种计算机共享资源。NFS实际上由两个协议组成：Mount和NFS。

29.1.1 NFS的历史

NFS由Sun(升阳)公司于1984年提出。它首先实现于其4.2BSD上(通常称之为SunOS)。由于它受到广泛的关注并且应用需求很广，因此NFS迅速在其他各种平台上实现。至1986年止，NFS可在16种不同的硬件平台上的五种操作系统间共享资源。目前，NFS可应用于多种硬件平台及操作系统平台，并且增加了基于Web的功能。

29.1.2 为何使用NFS

NFS的目标是使计算机共享资源。在它的发展过程中(即80年代)，计算机工业飞速发展。廉价CPU及客户/服务器技术促进了分布式计算环境的发展。然而，当处理器价格下降时，大容量的存储系统相对而言价格仍高居不下。因此，必须采用某种机制在充分发挥单个处理器

性能的同时使计算机可共享存储资源和数据，NFS应运而生。

29.2 实现——NFS工作过程

NFS使用TCP/IP提供的协议和服务，它运行于OSI模型的应用层，如表29-1所示。

表29-1 OSI层次模型上的NFS

层 数	名 称	功 能
1	应用层	NFS
2	表示层	XDR
3	会话层	RPC
4	传输层	UDP, TCP
5	网络层	IP
6	数据链路层	
7	物理层	Ethernet

出于性能上的原因，NFS起初采用性能较好的UDP，而并未采用可靠性较高的TCP。虽然UDP在可靠性较好的局域网中工作良好，但在可靠性较差的广域网如Internet上运行时，UDP则不能胜任。当前，随着TCP的改进，运行于TCP上的NFS不但可靠性高而且性能良好。从Solaris 2.6开始，Sun公司使用基于TCP的NFS。

注意 要了解关于TCP和UDP的详细信息，参见第9章。

29.2.1 远程过程调用(RPC)和外部数据表示(XDR)

为了实现平台无关性，NFS基于OSI底层实现。基于会话层的远程过程调用(Remote Procedure Call, RPC)和基于表示层的外部数据表示(External Data Representation, XDR)为NFS提供所需的网络连接及解释基于这些连接发送的数据格式。简单的说，它们使NFS可正常工作于不同平台。本章针对此作详细讲解。

1. 理解RPC

RPC运行于OSI模型的会话层，它提供一组过程，使远程计算机系统可像调用本地过程一样调用这些过程。使用RPC，本地计算机或应用程序可调用位于远程计算机上的服务。RPC提供一组过程库，高层应用可以调用这些库而无需了解远程系统的底层细节。因为RPC的抽象使得NFS与平台无关。

2. 理解XDR

外部数据表示库负责在不同的计算机系统间转换RPC数据，XDR设计了一种标准的数据表示，使得所有计算机均可理解。

总而言之，RPC和XDR是NFS实现客户/服务器关系的基石。

29.2.2 加载类型

如前章所述，NFS的无状态特性允许客户端和客户应用程序从服务器不可用中恢复。加载的类型决定当服务器崩溃时客户及客户应用的响应方式。加载类型将在本章29.5节中讨论。

1. 硬加载

当NFS服务器或资源不可用时，硬加载资源将导致不断尝试RFC调用。一旦服务器响应，

RPC调用成功且进入下一个执行过程。如果服务器或网络问题持续,硬加载将引起持续等待状态,使NFS客户端应用挂起。用户可以指定属性使硬加载可中断。

2. 软加载

使用软加载资源时,RPC调用失败将导致NFS客户端应用同时失败,最终使数据不可用。此种方法不可用于可写的文件系统或读取关键数据及可执行程序。

注意 下一节与UNIX的帮助文档类似。要获取更详细的信息,参见相关文件及程序的帮助文档。

29.3 NFS使用的文件及命令

本节主要讲述NFS使用的守护进程、程序和文件。因为NFS与UNIX系统紧密相连,本节将以两个常见的UNIX操作系统(Solaris 2.0和Linux)为例介绍共享和使用NFS资源。至于基于其他系统的实现方法,包括基于PC的客户端将在本章29.6节中讨论。

守护进程工作过程

在UNIX中,守护进程是在后台运行的程序,它等待某些事件的发生。在英文中,守护进程又称为精灵,即介于人与神之间的一种存在。在UNIX中,它位于两个进程之间。例如,打印守护进程等待打印作业进入打印队列。在本章中,NFS守护进程等待客户请求文件系统。

29.3.1 NFS守护进程

NFS正常工作需要多个服务器守护进程协同工作。下面将以Solaris和Linux操作系统分别介绍NFS守护进程。

1. Solaris 2.x

下列Solaris守护进程分别用于监听、处理及维护NFS请求。Solaris服务器守护进程由脚本/etc/init.d/nfs.server启动。

(1) nsfd

nsfd服务器守护进程用于监听NFS客户请求。

用法: /usr/lib/nfs/nsfd [-a] [-c #_conn] [-l listen_backlog] [-p protocol] [-t device] [nservers]

属性:

- a——运行NFS守护进程,可处理所有可用的面向连接和无连接请求,包括UDP和TCP。
- c #_conn——设置NFS服务器可处理的面向连接请求的最大连接数。缺省设置为无穷大。
- p——设置基于面向连接传输的NFS TCP连接队列的长度。缺省值为32。
- p protocol——启动基于指定协议的NFS守护进程。
- t device——在指定设备上启动NFS守护进程。

Nservers——指定可同时处理的最大请求数。

相关文件: /etc/init.d/nfs.server; shell脚本也可启动nsfd。

(2) mountd

mountd响应NFS访问请求及文件系统加载请求。它通过读取/etc/dfs/sharetab确定文件系统的可用性。mountd为服务器守护进程。

用法：/usr/lib/nfs/mountd [-v] [-r]

属性：

-v——冗余模式，在控制台显示信息。

-r——拒绝客户请求。

相关文件：/etc/dfs/sharetab(参见29.3.2节)。

(3) statd

statd为客户方守护进程，它与lockd协同工作，提供文件加锁服务的崩溃与恢复。

用法：/usr/lib/nfs/statd

相关文件：lockd。

(4) lockd

lockd是客户方守护进程，它是NFS锁管理器的一部分，用于对NFS文件加锁。

用法：/usr/lib/nfs/lockd [-g graceperiod] [-t timeout] [nthreads]

属性：

-g graceperiod——服务器重启后，客户端回收锁的时间间隔，单位为秒(缺省值为45)。

-t timeout——客户重新请求锁的时间间隔，单位为秒(缺省值为15)。

nthreads——响应锁请求的线程数(缺省值为20)。

相关文件：statd。

2. Linux

下列Linux守护进程分别用于监听、处理和维持NFS请求。它们的作用及语法与Solaris中的守护进程类似。

(1) rpc.nfsd

rpc.nfsd守护进程用于处理NFS请求，它是服务器守护进程。

用法：

```
/usr/sbin/rpc.nfsd [ -f exports-file ] [ -d facility ] [ -P port ]
[ -R dirname ] [ -Fhlnprstv ] [ --debug facility ] [ --exports-file=file ]
[ --foreground ] [ --help ] [ --allow-non-root ] [ --re-export ]
[ --public-root dirname ] [ --no-spoof-trace ] [ --port port ] [ --log-transfers ]
[ --version ] [ numservers ]
```

属性：

-f exports-file——指定输出文件，缺省输出文件为/etc/exports。

-h或-help——提供帮助信息。

-l或--log-transfers——记录所有文件的传输信息。

-n或--allow-non.root——允许守护进程接收进入的NFS请求，即使这些请求不是由允许的IP产生。

-P portnum或--port portnum——让nfsd监听指定的端口，替代缺省端口 2049或文件/etc/services中指定的端口。

-p或-promiscuous——使服务器可被网络中任何主机使用。

-v或-version——显示程序的当前版本。

numservers——允许用户同时运行多个nsfd实例。

相关文件：exports, mountd

(2) rpc.mountd

rpc.mountd接收加载请求并检测/etc/exports文件中NFS文件系统的可用性。如果请求资源可以使用，rpc.mountd创建文件处理进程并在/etc/rmtab中添加相应项。接收 umount请求后，从/etc/rmtab文件中删除相应的项。

用法：

```
/usr/sbin/rpc.mountd [ -f exports-file ] [ -d facility ] [ -P port ]  
[ -Dhnprv ] [ --debug facility ] [ --exports-file=file ] [ --help ]  
[ --allow-non-root ] [ --re-export ] [ --no-spoof-trace ] [ --version ]
```

属性：

查询上一个命令，rpc.nfsd，可知rpc.mountd命令属性的含义。

相关文件：/etc/exports, /etc/rmtab

(3) rpc.statd

参见solaris中的statd。

(4) rpc.lockd

参见solaris中的lockd。

(5) biod

biod在NFS客户端启动指定数目的异步块 I/O守护进程。这些进程用于在 NFS文件系统中执行提前读和推后写操作，以提高 NFS的性能。

用法：/usr/etc/biod [numdaemons]

29.3.2 与NFS相关的文件

在共享或加载NFS文件系统前，必须首先修改服务器和客户方文件。

1. solaris 2.x文件

Solaris中的下列文件定义和跟踪NFS资源。

(1) /etc/dfs/dfstab

dfstab文件中包含一组控制NFS文件系统输出的共享命令。关于 share命令的详细信息将在本章29.3.3节中讲述。

示例：

```
share -F nfs -o rw=engineering -d 'home dirs' /export/home2
```

(2) /etc/dfs/sharetab

sharetab文件中包含与通过 share命令共享的文件系统对应的项。这些项由系统自动生成。

sharetab文件中的每一项包含以下几部分：

- pathname——共享资源的路径。
- resource——远程系统访问资源的名称。
- fstype——共享资源的类型。
- specific_options——共享资源时指定可以使用的属性。
- description——共享资源的描述信息，当资源被共享时提供给客户。

示例：

```
/export/home2-nfs rw=engineering "Engineering Home"
```

(3) /etc/rmtab

rmtab文件中包含了当前被用户加载的 NFS 文件系统列表。它也是由系统自动生成，其每一行的格式如下：

```
hostname: fsname
```

示例：

```
engineering5: /export/home2
```

(4) /etc/vfs/vfstab

vfstab文件用于描述被加载的本地和远程文件系统，它是 NFS 客户端文件。vfstab文件的每一行包含以下几部分：

- 加载(mount)设备
- fsck 设备
- 加载点
- 文件系统类型
- fsck pass
- 启动时加载
- 加载属性。

示例：

```
servername:/export/home2 - /export/home NFS - y -
```

2. Linux文件

在Linux中下列文件定义和跟踪NFS资源。

(1) /etc/exports

exports文件指定了可被 NFS 用户使用的文件的访问控制权限。exports文件的每一行包含 NFS 资源及允许加载资源的主机。访问控制属性以简写方式给出，也可以不指定权限。此时，资源的访问权限由一组扩展的属性和参数管理。详细信息请查阅相关文档。

示例：

```
/users/home host.mydomain.com (ro)
```

示例表明：主机host.mydomain.com可以加载/users/home文件系统，访问权限为只读。

(2) /etc/rmtab

rmtab文件给出当前被用户加载的 NFS 文件系统列表。它由系统维护。其文件格式如下：

```
hostname: resource
```

示例：

```
datahouse.anvi.com:/users/home
```

(3) /etc/fstab

/etc/fstab文件为系统可使用的每个文件系统建立一项。它等价于 Solaris 中vfstab.fstab文件中的每一项包含以下几部分：

- 远程文件系统(remote filesystem)——远程资源的路径。
- 加载点(mountpoint)——远程资源在本地的加载点。
- fstype——远程资源的类型(例如nfs)。
- 加载属性(mount_options)——指定与文件系统相关的属性。

- 备份属性(dump option)——指定文件系统是否应备份。缺省值表示不备份。
- fsck pass——指明fsck在何时检查文件系统。缺省值表示 fsck不检查文件系统。关于 fstab 的详细信息请查阅相关帮助文档。

示例：

```
solaris:/www /mnt nfs resize=1024, wsize=1024 hard, intr 0 0
```

29.3.3 NFS服务器命令

正确运行守护进程并且对相关文件进行修改后，共享及加载 NFS资源非常简单。

1. 创建共享资源

在NFS资源可被网络客户使用之前，它们必须由 NFS服务器共享。下面讲述如何在 Solaris 和Linux中使用正确的命令共享(输出)资源。

(1) slaris 2.x服务器命令

下列Solaris命令用于共享及监视 NFS资源。

share

share命令使指定的文件系统可被客户端使用。

用法：share [-F FSType] [-o specific_options] [-d description] [pathname]

属性：

-F FSType——指定文件系统类型。

-o specific_options——用于控制访问共享资源。控制方式有以下几种：

rw 可读/写

rw=client [:client]...——仅能被指定客户读/写

Ro 只读

ro=client [:client]...——仅能被列出的客户只读访问

-d description——对共享资源进行描述

pathname——指明共享资源的路径

示例：

```
share -F nfs -o rw=engineering -d 'home dirs' /export/home2
```

shareall

shareall命令用于共享特定文件、标准输入或 dfstab文件列出的资源。

用法：shareall [-F FSType [, FSType...]] [-|file]

属性：

-F FSType——指定文件系统类型。

[-|file]——如果使用“-”，shareall命令以标准输入作为其输入参数。如果指令文件名，则shareall命令以文件的内容作为输入参数。如果项为空，则 shareall命令使用 dfstab文件 (/etc/dfs/dfstab)为输入参数。

showmount

showmount命令列出命令执行的服务器行上，加载该服务器文件系统的所有客户端。

用法：/usr/sbin/showmount [-ade] [hostname]

属性：

- a——列出所有远程加载。
- d——列出被远程加载的目录。
- e——列出所有共享的文件系统。

(2) Linux服务器命令

下列Linux命令用于共享及加载NFS资源。

exportfs

exportfs命令使系统重取读取/etc/exportfs文件。某些Linux版本没有exportfs命令。如果用户碰到这种情况，可创建shell脚本完成相同的功能。

```
#!/bin/sh
killall -HUP /usr/sbin/rpc.mountd
killall -HUP /usr/sbin/rpc.nfsd
```

showmount

showmount命令检测NFS服务器加载守护进程，获取当前被加载的文件系统的信息。

用法：/usr/sbin/showmount [-adehv] [--all] [--directories] [--exports] [--help] [--version] [host]

属性：

- a或-all——以host:dir的格式列出客户端主机名及其加载的目录。
- d或-directories——仅列出客户加载的所有目录。
- e或-exports——列出服务器上可加载的文件系统。
- h或-help——提供帮助信息。
- v或-version——显示当前程序的版本信息。
- no-headers——从输出中删除标题。

2. 不共享资源

出于某种原因，用户希望防止文件系统被加载时，既可以使用 mount命令加上-r属性，也可以采用不共享资源的方法。

(1) Solaris 2.x

下列Solaris命令用于不共享NFS资源。

unshare

unshare命令使共享文件系统不可用。

用法：unshare [-F FSType] [-o specific_options] [pathname | resourcename]

如果命令中未指定 -F FSType，unshare命令将以/etc/dfs/dfstab文件的第一行中的FSType为该项参数的缺省值。-o specific_options根据FSType不同而不同。详细信息参见相关文档。

unshareall

unshareall命令不共享所有FSType类型的共享资源。

用法：unshareall [-F FSType [, FSType...]]

如果命令中未指定FSType，则停止所有共享资源的共享。

使用showmount进行确认

showmount命令可用于检查文件系统是否仍被客户加载。语法格式参见本章已讲述内容。

(2) 在Linux使用exports命令停止资源共享

Linux缺少正式的命令用于停止资源的共享。完成这一工作的最佳方法是编辑共享文件即

/etc/exports文件，删去不希望继续共享的文件系统。然后运行 `exports` 命令使系统重新读取共享文件。如果系统没有 `exports` 命令，可创建等价的 `shell` 脚本完成相同的功能：

```
#!/bin/sh
killall -HUP /usr/sbin/rpc.mountd
killall -HUP /usr/sbin/rpc.nfsd
```

29.3.4 NFS客户命令

本节讲述用于加载及取消加载 NFS 资源的客户命令。

注意 在加载到 NFS 资源前应首先创建加载点。

1. 加载 NFS 资源

下述命令用于加载共享资源：

(1) mount (Solaris/Linux)

`mount` 命令用于加载远程 NFS 文件系统。

用法(solaris)：

```
mount [ -p | -v ]
```

```
mount [ -F FSType ] [ generic_options ] [ -o specific_options ] [ -O ] special |
mount_point
```

```
mount [ -F FSType ] [ generic_options ] [ -o specific_options ] [ -O ] special
mount_point
```

```
mount -a [ -F FSType ] [ -V ] [ current_options ] [ -o specific_options ]
[ mount_point. . . ]
```

用法(Linux)：

`mount (-hv)` (打印帮助及版本信息)

```
mount -a [-fFnrsvw] [-t vfstype]
```

```
mount [-fnrsvw] [-o options [,...]] device | dir
```

```
mount [-fnrsvw] [-t vfstype] [-o options] device dir
```

Solaris 和 Linux 的 `mount` 命令在功能上相似，但都包含一些特有的属性。详细信息参见有关文档及帮助文件。

加载到 NFS 资源最简单的方法是利用 `vfstab` 或 `fstab` 文件中指定的加载点或设备。

例如，假设 Solaris 中 `vfstab` 文件包含以下行：

```
remotehost:/export/home2 - /export/myhome NFS - y -
```

则可以使用下述命令加载文件系统：

```
# mount /export/myhome
```

```
# mount remotehost:/export/home2
```

(2) mountall(solaris)

`mountall` 命令加载的 `vfstab` 文件指定所有资源。

用法：`mountall [-F FSType] [-l | -r] [file_system_table]`

属性：

-F FSType——文件系统类型，本例为NFS。

-l——加载所有本地文件系统。

-r——加载所有远程文件系统。

file-system-table——缺省值为/etc/vfstab。

2. 自动加载

用户可以设置一个特殊的NFS守护进程来监视对远程NFS文件系统的需求。这一进程称为自动加载。当应用或客户需要使用未加载的远程文件系统，自动加载进程自动加载到NFS服务器的相应文件系统。当远程文件系统不被使用时，自动加载进程停止此加载。使用自动加载有许多好处，如充分利用网络及便于管理。

自动加载既可用于Solaris平台也可用于Linux平台，对于大多数UNIX平台都适用。详细信息参见用户使用的操作系统的相关文档或UNIX帮助文件。

3. 不加载NFS资源

下列命令用于取消对共享资源的加载。

(1) umount (Solaris/Linux)

umount命令取消对共享资源的加载。

用法(Solaris)：

```
umount [ -V ] [ -o specific_options ] special | mount_point
```

```
umount -a [ -V ] [ -o specific_options ] [mount_point...]
```

用法(Linux)：

```
umount [-hv] (显示帮助及版本信息)
```

```
umount -a [-nrv] [-t vfstype]
```

```
umount [-nrv] device | dir [...]
```

与mount命令相似，Linux和Solaris中的umount命令除了几个特别的属性外几乎完全相同，详细信息参见相关文档及帮助。

umount命令也只能使用vfstab或fstab文件中指定的设备或加载点作为其参数。

例如，假设Linux中的fstab文件包含以下行：

```
remotehost:/export/home2 /export/myhomenfs rsize=1024, wsize=1024, hard, intr, 0 0
```

用户可使用以下命令卸载文件系统：

```
# umount /export/myhome
```

```
# umount remotehost:/export/home2
```

(2) umountall (Solaris)

umountall命令将卸载所有已加载的文件系统。命令属性可使用户仅卸载远程文件系统或从特定主机加载的文件系统。

用法：

```
umountall [ -k ] [ -s ] [ -F FSType ] [ -l|-r ]
```

```
umountall [ -k ] [ -s ] [ -h host ]
```

属性：

- k——杀死所有使用文件系统的进程。
- s——不并行卸载文件系统。
- F FSType——仅卸载指定类型的文件系统(如NFS)。
- l——仅卸载本地文件系统。
- r——仅卸载远程文件系统。
- h host——仅卸载从指定主机上加载的文件系统。

29.4 示例：共享及加载NFS文件系统

用户可在Solaris 2.6系统中共享文件系统，在Linux(Red Hat6.0)系统上加载共享文件系统。用户首先需要修改/etc/dfs/dfstab文件，在文件中增加以下行：

```
share -F nfs -d 'WWW Directory' /www
```

此行将共享文件系统/www。

在dfstab文件后，用户可以通过执行shareall命令使/www文件系统可被远程加载。

```
#shareall
```

运行share命令检查/www是否已经共享。

```
#share
```

```
- /www 'WWW Directory'
```

下一步用户需要确认守护进程是否正确运行。在Solaris系统中，用户需要运行nfsd和mountd。可使用ps命令查看已运行的进程。

```
#ps -ef
```

ps命令显示系统中当前活跃的进程，其中包含NFS守护进程，如下所示：

```
root 27282      1  0 08:10:59 ?          0:00 /usr/lib/nfs/nfsd -a 16
root 27280      1  1 08:10:59 ?          0:00 /usr/lib/nfs/mountd
```

如果守护进程没有运行(ps命令未显示上面两项)，可以使用NFS初始化过程启动它们。

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

现在，用户可以在Linux客户端加载共享资源。

在Linux系统中，使用以下mount命令：

```
mount -o rsize=1024,wsiz=1024 solaris:/www /mnt
```

卸载文件系统命令如下：

```
umount /mnt
```

要使Linux系统可持续加载/www文件系统，用户可在/etc/fstab文件中加入以下行：

```
solaris:/www /mnt nfs rsize=1024,wsiz=1024,hard,intr 0 0
```

此行指定将NFS/www文件系统加载到加载点/mnt。hard及intr属性表明加载为硬加载且不可中断(可靠性高)。

整个过程非常简单，但实际操作过程中总会出现问题。下一节将描述用户常见的问题及解决办法。

29.5 NFS常见问题及解决方案

本节介绍在UNIX环境中设置和运行NFS的一些技巧，并讲述一些常见的问题及相应的解决办法。因为需要修改许多文件并启动多个守护进程，正确使用NFS是一项有挑战性的工作。

29.5.1 不能加载

加载问题常由不正确的配置或网络问题引起。

除非用户可以确定问题的根源在何处（例如，在未查阅帮助的情况下配置 `vfstab` 文件），否则解决加载问题最好从顶层开始。

用户首先需要检查是否可与NFS服务器通信。可以使用 `ping` 命令检查网络互连情况。

如果网络连接正常，检查守护进程是否正常运行。详细信息参见本章 29.3.1节中的内容。使用 `share` 命令或类似命令检查资源是否正确共享。如果资源共享不正确，检查服务器配置文件（`dfstab` 或 `exports`）。详细信息参见本章 29.3.2节中的内容。确认服务器可识别客户。在 `/etc/hosts` 中的客户主机名应与配置文件中的主机名相同。例如，如果服务器指定名为 `loco` 的系统可加载某资源，则 `loco` 应包含在 `hosts` 文件中。NFS对主机名非常挑剔，即使 `loco.mydomain.com` 存在于 `host` 文件中，仍需文件中加入 `loco`。

如果用户不能访问服务器，则需要请求系统管理员协助。

如果用户可以访问服务器，且服务器配置正确，则需进一步检查客户方配置。与服务器类似，用户需要检查客户方的守护进程及相关文件（`fstab` 或 `vfstab`）的配置，检查 `hosts` 文件和 NFS 配置文件中主机名的一致性。

29.5.2 不能卸载

如果发现不能卸载资源，首先需要检查用户是否正在使用资源。如果用户收到“文件系统忙”此类模棱两可的信息，则需检查所有打开的 `shell` 和终端，是否运行了打开文件系统的文件管理器。同时需要确信没有编辑器正在使用系统中的文件。如果发现，则终止此编辑器或文件管理器，并重新卸载。最后，不要忘记检查正在使用的 `shell`。

29.5.3 硬加载与软加载

NFS文件系统可使用两种方式加载：硬加载和软加载。如前所述，硬加载的可靠性高，适用于加载可写资源或访问关键的文件和程序。如果资源被硬加载，一旦服务器崩溃或网络连接异常，程序（或用户）访问将被挂起，这将导致不可预见的结果。缺省情况下，NFS资源均采用硬加载。

与硬加载相对应的是软加载，采用软加载，客户与服务器连接异常将导致使用NFS资源的应用程序中止。并且，在不可信网络上使用软加载也会导致不可预料的结果。

用户可以采用 `hard` 和 `intr` 属性加载NFS资源避免上述问题。这些 `mount` 命令属性使硬加载可响应中断。

29.6 相关协议及产品

由于NFS的普及，及对于在不同计算机平台间共享资源的迫切需求，出现了许多基于NFS

的工具及与 NFS 类似的工具。本节将对这些工具作简单介绍。

29.6.1 WebNFS

Sun 公司最近推出了 WebNFS，它是 NFS 协议的增强版。WebNFS 提供更好的扩展性、可靠性及性能。WebNFS 也可与防火墙协同工作——这是基于 RPC 的 NFS 的弱点。

详细信息参见 <http://www.sun.com/webnfs>。

29.6.2 基于 PC 的 NFS 及其他客户端软件

PC-NFS 是 Sun 公司推出的适用于非 UNIX 平台的 NFS 客户端产品。它使用户的非 UNIX 系统可充分利用用户所在组织的共享资源。它依赖于守护进程 `pcnfsd`，并与其他 Sun 产品一起发售。

用户也可以采用其他 NFS 客户方产品。表 29-2 列出了部分 NFS 客户方产品及产品信息所在的 Web 地址。

表 29-2 部分 NFS 客户方产品

产 品	厂 商	URL
Solstice NFS Client	Sun Microsystems	http://www.sun.com/netclient/nfs-client/
Reflection NFS Connection	WRQ	http://www.wrq.com/
NFS Maestro	Hummingbird Communications, Ltd.	http://www.hcl.com/products/nc/nfs/
Omni-NFS	Xlink Technology	http://www.xlink.com/

29.6.3 SMB 和 CIFS

服务信息块简称为 SMB，是由 IBM、微软和其他几家公司一起开发的公用标准，主要用于计算机间共享文件系统、打印机和其他资源。Windows NT、OS/2 和 Linux 均对 SMB 提供本地支持，其他系统可以通过采用第三方软件对 SMB 提供支持。

关于 SMB 的详细信息，参见 <http://www.samba.org/cifs/docs/what-is-smb.html>。

通用互联网文件系统 (CIFS) 简介

通用互联网文件系统 (CIFS) 由一组厂商 (微软、SCO、通用数据等公司) 共同开发，它是 SMB 的公用版本。它的最初实现与 NT LM (Microsoft NT 中的 LAN Manager) 非常相似，支持在互联网上共享资源。

关于 CIFS 的详细信息参见 <http://msdn.microsoft.com/workshop/networking/cifs/>。

部分实现 SMB 的产品：

- samba
- Windows for workgroups 3.x, Windows NT, 95/98/2000
- LAN Manager

samba 支持 SMB 和 CIFS 协议

samba 是源码公开的客户服务器产品，它支持 SMB 和 CIFS 协议。用户可在遵守 GNU 的情况下使用它。

samba可使用户非常方便地共享服务器资源，如共享文件系统和打印机给同一网络的客户。

29.6.4 其他产品

许多其他产品也可以实现在多个计算机间共享资源。与 NFS和SMB不同，它们中的大多数都不支持异构网络，不能在不同计算机平台间共享资源。这些协议包括： AppleTalk、Decnet、NetWare等等。

注意 相关RFC文档。

NFS RFC 1094: <http://www.cis.ohio-state.edu/htbin/rfc/rfc1094.html>

NFS V.3 RFC 1813: <http://www.cis.ohio-state.edu/htbin/rfc/rfc1813.html>

29.7 小结

本章讨论了NFS的历史及实现，介绍了与NFS相关的的守护进程、文件和其他程序，并给出了一个加载和共享NFS文件系统的实例。最后讨论了相关的程序和协议，如客户端 NFS程序和SMB协议等。

使用NFS及其命令需要运行多个守护进程，配置多个文件，因此，本章简单介绍了常见的问题及解决办法并给出了一些参考文档供用户进一步学习 NFS。