

GNU/Linux

NFS



GNU/Linux

NFS(Network File System) 

即网络文件系统,NFS 最早由 Sun 公司所发展出来的。最大的功能就是可以透过网络,让不同的主机能共享文件。通过使用 NFS,用户和程序可以像访问本地文件一样访问远端系统上的文件。



GNU/Linux

NFS:

从 1985 年推出至今，共发布了 3 个版本：

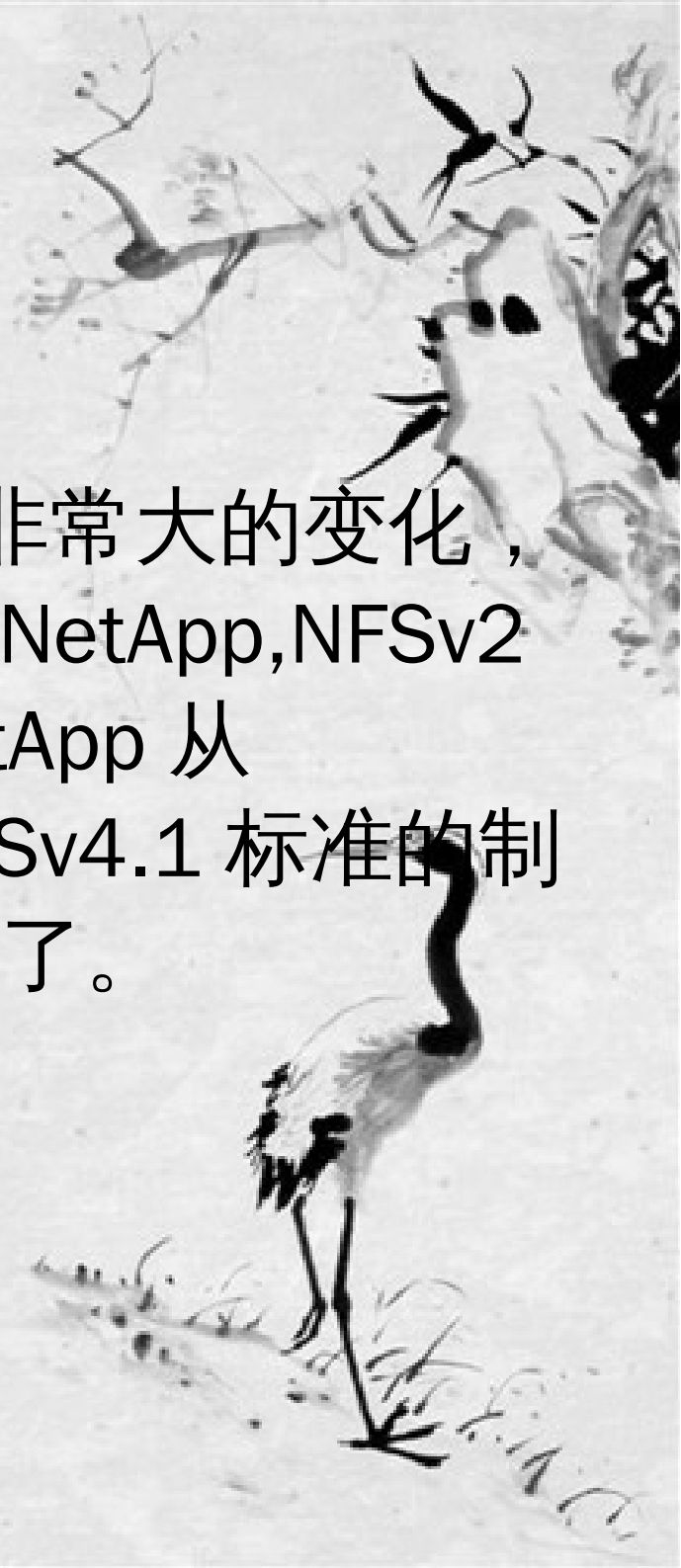
1. NFSv2
2. NFSv3
3. NFSv4: 含两个次版本 NFSv4.0 及 NFSv4.1。



GNU/Linux

NFS:

经过 20 多年发展，NFS 发生了非常大的变化，最大的变化就是推动者从 Sun 变成 NetApp, NFSv2 和 NFSv3 基本上是 Sun 起草的，NetApp 从 NFSv4.0 参与进来，并且主导了 NFSv4.1 标准的制定过程，而 Sun 已经被 Oracle 收购了。



GNU/Linux



NFS 各版本情况

编号	版本	RFC	时间	页数
1	NFSv2	RFC1094	1989 年 3 月	27 页
2	NFSv3	RFC1813	1995 年 6 月	126 页
3	NFSv4.0	RFC3530	2003 年 4 月	275 页
4	NFSv4.1	RFC5661	2010 年 1 月	617 页

GNU/Linux

NFS 各版本特点

V2: 第一个以 RFC 形式发布的版本，只是实现了基本的网络共享及存取功能。



GNU/Linux

NFS 各版本特点

V3:

v3 修正了 NFSv2 的一些 bug。两者有如下一些差别

(1) NFSv2 对每次读写操作中传输数据的最大长度进行了限制，上限值为 8192 字节，NFSv3 取消了这个限制。



GNU/Linux

NFS 各版本特点

(2) **NFSv3** 对文件名称长度进行了限制，上限值为 255 字节，NFSv3 取消了这个限制。

(3) NFSv2 对文件长度进行了限制，上限值为 0x7FFFFFFF，NFSv3 取消了这个限制。

(4) NFSv2 中文件句柄长度固定为 32B，NFSv3 中文件句柄长度可变，上限值是 64 字节。

GNU/Linux

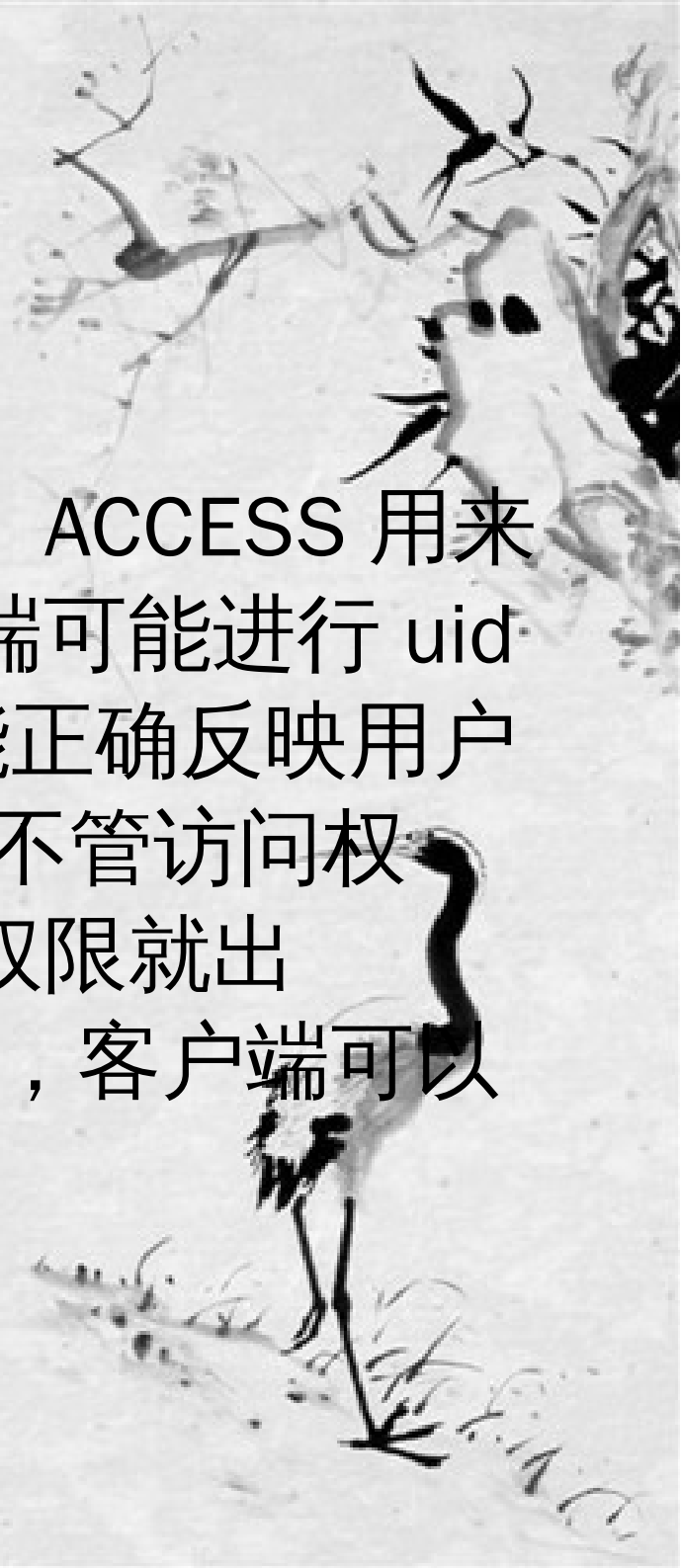
NFS 各版本特点

(5) NFSv2 只支持同步写，如果客户端向服务器端写入数据，服务器必须将数据写入磁盘中才能发送应答消息。NFSv3 支持异步写操作，服务器只需要将数据写入缓存中就可以发送应答信息了。NFSv3 增加了 COMMIT 请求，COMMIT 请求可以将服务器缓存中的数据刷新到磁盘中。

GNU/Linux

NFS 各版本特点

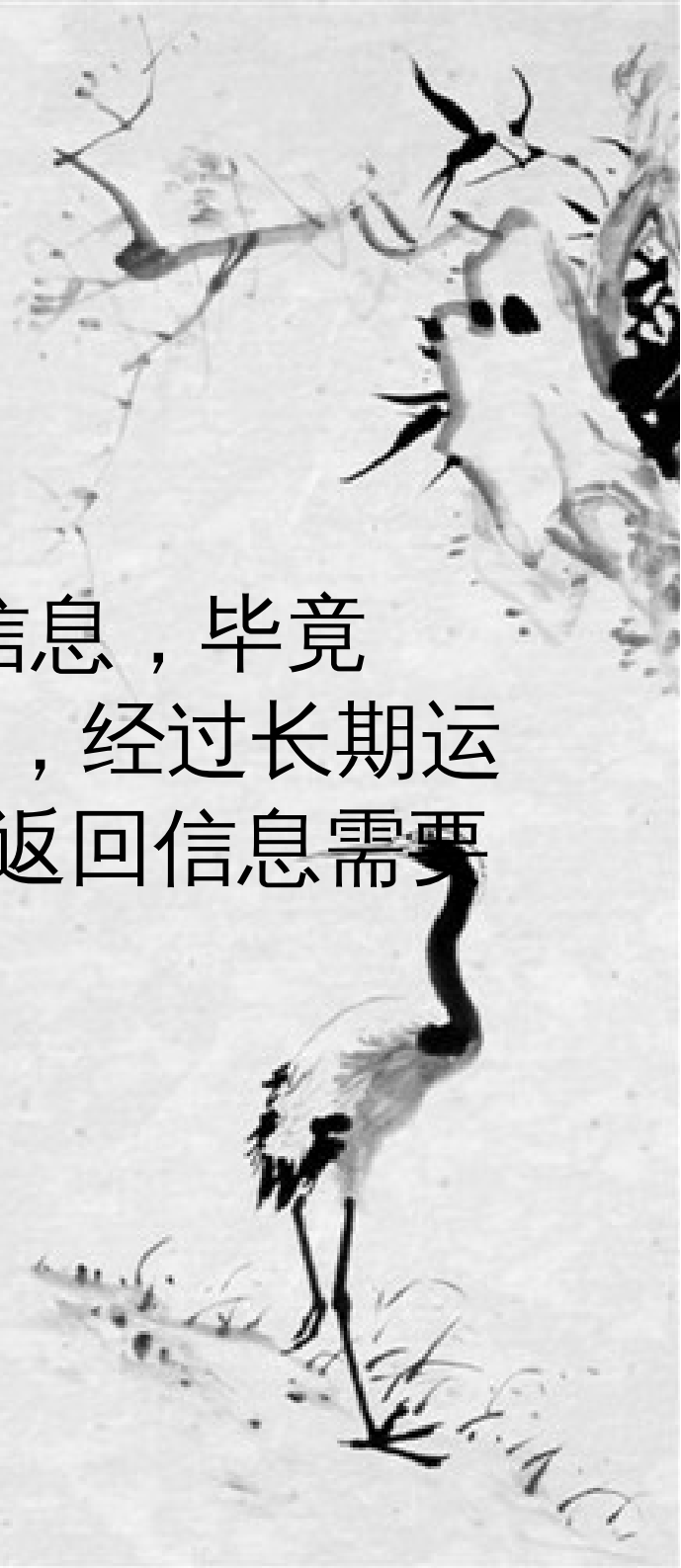
(6) NFSv3 增加了 ACCESS 请求，ACCESS 用来检查用户的访问权限。因为服务器端可能进行 uid 映射，因此客户端的 uid 和 gid 不能正确反映用户的访问权限。NFSv2 的处理方法是不管访问权限，直接返送请求，如果没有访问权限就出错。NFSv3 中增加了 ACCESS 请求，客户端可以检查是否有访问权限。



GNU/Linux

NFS 各版本特点

(7) 一些请求调整了参数和返回信息，毕竟 NFSv3 和 NFSv2 发布的间隔有 6 年，经过长期运行可能觉得 NFSv2 某些请求参数和返回信息需要改进。



GNU/Linux

NFS 各版本特点

V4.0

相比 NFSv3, NFSv4 发生了比较大的变化。
最大的变化是 NFSv4 有状态。

NFSv2 和 NFSv3 都是无状态协议，服务端不需要维护客户端的状态信息。无状态协议的一个优点在于灾难恢复，当服务器出现问题后，客户端只需要重复发送失败请求就可以了，直到收到服务器的响应信息。

GNU/Linux

NFS 各版本特点

但是某些操作必须需要状态，如文件锁。如果客户端申请了文件锁，但是服务器重启了，但 NFSv3 无状态，客户端再执行锁操作可能就会出错。

NFSv3 需要 NLM 协助才能实现文件锁功能，但是有的时候两者配合不够协调。NFSv4 设计成了一种有状态的协议，自身实现了文件锁功能，从而不再需要 NLM(Network Lock Manager) 协议。

GNU/Linux

NFS 各版本特点

NFSv4 和 NFSv3 的差别如下：

(1) NFSv4 设计成了一种有状态的协议，自身实现了文件锁功能和获取文件系统根节点功能，不需要 NLM 和 MOUNT 协议协助。

(2) NFSv4 增加了安全性，支持 RPCSEC_GSS 身份认证。



GNU/Linux

NFS 各版本特点

(3) NFSv4 只提供了两个请求 NULL 和 COMPOUND，所有的操作都整合进了 COMPOUND 中，客户端可以根据实际请求将多个操作封装到一个 COMPOUND 请求中，增加了灵活性。

(4) NFSv4 文件系统的命令空间发生了变化，服务器端必须设置一个根文件系统 (fsid=0)，其他文件系统挂载在根文件系统上导出。

GNU/Linux

NFS 各版本特点

(5) NFSv4 支持 delegation(代表)。由于多个客户端可以挂载同一个文件系统，为了保持文件同步，NFSv3 中客户端需要经常向服务器发起请求，请求文件属性信息，判断其他客户端是否修改了文件。如果文件系统是只读的，或者客户端对文件的修改不频繁，频繁向服务器请求文件属性信息会降低系统性能。NFSv4 可以依靠 delegation 实现文件同步。

GNU/Linux

NFS 各版本特点

(5) NFSv4 支持 delegation(代表)。

如：当客户端 A 打开一个文件时，服务器会分配给客户端 A 一个 delegation。只要客户端 A 具有 delegation，就可以认为与服务器保持了一致。如果另外一个客户端 B 访问同一个文件，则服务器会暂缓客户端 B 的访问请求，向客户端 A 发送 RECALL 请求。当客户端 A 接收到 RECALL 请求时将本地缓存刷新到服务器中，然后将 delegation 返回服务器，这时服务器开始处理客户端 B 的请求。

GNU/Linux

NFS 各版本特点

(6) NFSv4 修改了文件属性的表示方法。由于 NFS 是 Sun 开发的一套文件系统，设计之初 NFS 文件属性参考了 UNIX 中的文件属性，可能 Windows 中不具备某些属性，因此 NFS 对操作系统的兼容性不太好。NFSv4 将文件属性划分成了三类：



GNU/Linux



NFS 各版本特点

(6) NFSv4 修改了文件属性的表示方法。

Mandatory Attributes: 这是文件的基本属性；所有的操作系统必须支持这些属性。

Recommended Attributes: 这是 NFS 建议的属性，如果可能操作系统尽量实现这些属性。

Named Attributes: 这是操作系统可以自己实现的一些文件属性。



GNU/Linux



NFS 各版本特点

v4.1

与 NFSv4.0 相比，NFSv4.1 最大的变化是支持并行存储了。

在以前的协议中，客户端直接与服务器连接，客户端直接将数据传输到服务器中。当客户端数量较少时这种方式没有问题，但是如果大量的客户端要访问数据时，NFS 服务器很快就会成为一个瓶颈，抑制了系统的性能。

GNU/Linux

NFS 各版本特点

v4.1

NFSv4.1 支持并行存储，服务器由一台元数据服务器 (MDS) 和多台数据服务器 (DS) 构成，元数据服务器只管理文件在磁盘中的布局，数据传输在客户端和数据服务器之间直接进行。由于系统中包含多台数据服务器，因此数据可以以并行方式访问，系统吞吐量迅速提升。

GNU/Linux

Linux NFS

Linux 上使用标准的 Internet 协议，此标准是开放的。NFS 虽然起始与 UNIX 但 NFS 正在积极的扩展 Linux 的权限及文件系统的特性。

RHEL7 支持的是 NFSv4, 默认情况下，如果 NFSv4 不可用，将自动回滚到 NFSv3 和 NFSv2 版本的支持。

GNU/Linux

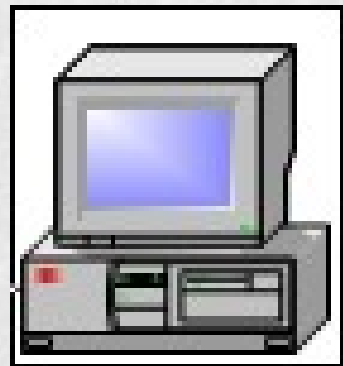
Linux NFS

NFSv4 使用 TCP 协议与服务器进行联接及通信，而 NFSv3 及 NFSv2 则将使用 TCP 或 ~~TCP~~ 协议。

NFS 客户端通过 NFS 服务器共享的目录，挂载到 NFS 客户端本地目录上来进行存取。此挂载点必须存在，且最好为空目录

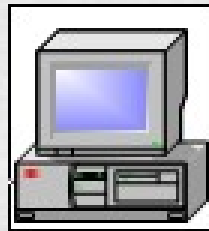
GNU/Linux

Linux NFS 示意图



NFS server

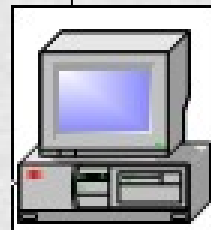
/tmp/nfssharefile



NFS client1

把 /tmp/nfssharefile

挂载到本地 /mnt/nfs



NFS client2

把 /tmp/nfssharefile

挂载到本地 /root/nfs



GNU/Linux

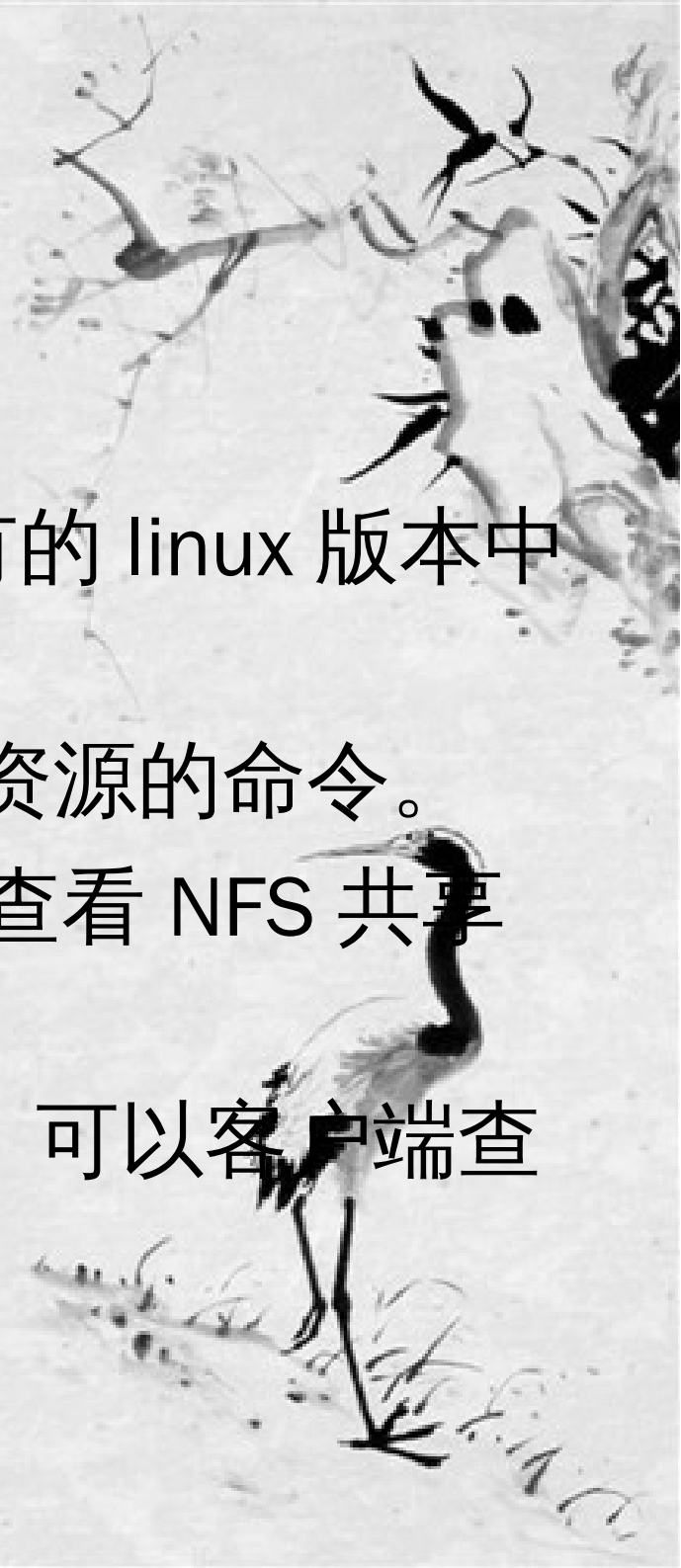
Linux NFS

/etc/exports: NFS 的主配置文件。有的 linux 版本中默认不存在，需要手动建立。

/usr/sbin/exportfs: 维护 NFS 共享资源的命令。

/usr/sbin/showmount: 在 client 来查看 NFS 共享的资源（exportfs 用在 server 端）。

/var/lib/nfs/xtab: NFS 的记录文件，可以客户端查看链接服务器的信息



GNU/Linux

Linux NFS Server 设置

```
#vim /etc/exports
```

语法格式：

共享目录 客户端地址（IP 或主机名）（设置参数） [客户端地址 1 （IP 或主机名 1）（设置参数）] ...



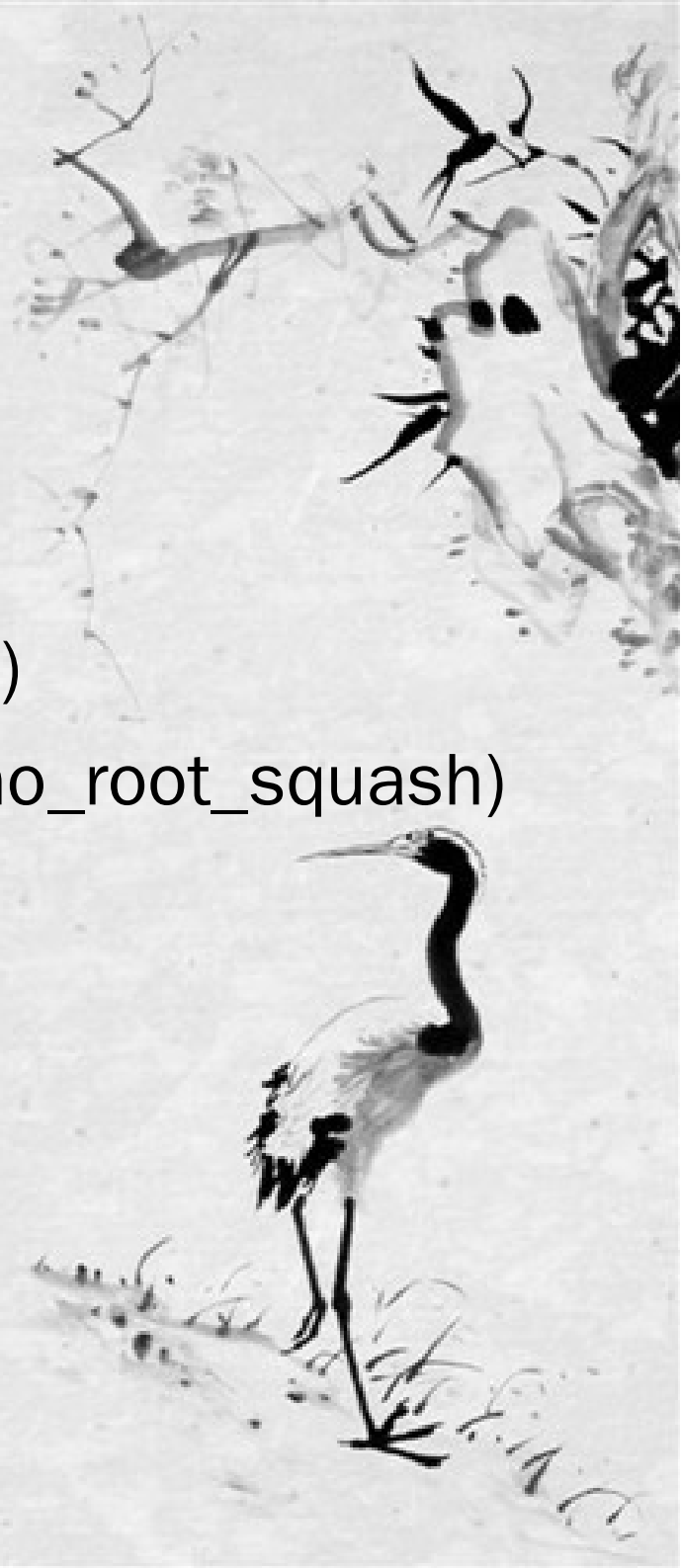
GNU/Linux

Linux NFS Server 设置

```
#vim /etc/exports
```

```
/mnt/exportfs/nfs 192.168.1.123(ro)
```

```
/mnt/exportfs/share t1.niliu.edu(rw,no_root_squash)  
t2.niliu.edu(ro)
```



GNU/Linux

Linux NFS Server 设置

客户机地址可以是

- 1) 指定 ip addr:192.168.1.123
- 2) 指定 FQDN:t1.niliu.edu
- 3) 指定网段 :192.168.1.0/24
- 4) 指定域中的所有主机 :*.niliu.edu
- 5) 指定所有主机 :*



GNU/Linux

Linux NFS Server 权限设置

ro: 只读权限

rw: 可读写的权限

no_root_squash: 如果 root 使用共享目录, 则以 root 身份进行操作 (root 权限)

root_squash: 如果是 root 使用共享目录, 则 root 被映射到 nfsnobody 账户, 其他账户将不变。

all_squash: 不论登陆者是谁, 都映射到 nfsnobody 账户

sec=: 指定安全访问形式



GNU/Linux

Linux NFS Server 权限设置

	no_root_squash	root_squash	all_squash
NFS server	root	nobody	nobody
NFS client	root	root	All user

GNU/Linux

Linux NFS

NFS 的安全访问支持：

安全认证	说明
none	匿名访问文件，如数据写入至服务器上将使用 UID 和 GID 的账户 / 组名为 nfsnobody
sys	基于 Linux 下有效的 UID/GID, 才能访问指定的共享文件 / 共享目录
krb4	客户端必须经过 kerberos 身份验证及共享目录所设置的共享权限许可，才可访问
krb5i	对所有的数据进行完整性检测，以确保数据的完整性
krb5p	客户端与服务端之间的会话将被加密传输，但将会影响 NFS 的性能

GNU/Linux

Linux NFS 权限设置

anonuid: 更改匿名用户 (anonuid=123)

anongid: 更改匿名组 (anongid=1111)

sync: 数据立即同步写入内存和硬盘。

async: 数据先写入内存，再写到硬盘。

fsid: 将共享目录指定为 NFS 共享的根目录 (fsid=0)



GNU/Linux

Linux NFS 服务配置

1. 将本地 /mnt/share 目录共享，权限为 ro
#vim /etc/exports
/exports/share *(ro)
2. 将本地目录 /mnt/share 目录共享，权限为 rw,no_root_squash
/exports/share *(rw,no_root_squash)



GNU/Linux

Linux NFS 服务配置

3. 启动 nfs

```
#systemctl enable nfs
```

```
#systemctl start nfs
```



GNU/Linux

Linux NFS

客户端能够共享方式有：

1. 通过 mount 命令进行手工挂载
2. 如需自动挂载可对 /etc/fstab 进行相关配置
3. 根据需求挂载 NFS 共享目录



GNU/Linux

Linux NFS

1. 查看 NFSv4 的共享信息

```
#mkdir -v /mnt/nfsroot
```

```
#mount.nfs4 nfssrv_ip_addr:/ /nfsroot
```

```
#ls /nfsroot
```

2. NFSv2 及 v3 可以使用下列命令查看 NFS_Server 的共享信息

```
#showmount -e nfssrv_ip_addr
```



GNU/Linux

Linux NFS

3. 手动挂载 NFS Server 的共享目录

```
#mkdir /mnt/nfsshare
```

```
#mount -t nfs4 -o sync
```

```
nfssrv_ip_server:/share /nfsshare
```



GNU/Linux

Linux NFS

4. 自动挂载 NFS Server 的共享目录

```
#vim /etc/fstab
```

```
nfssrv_ip_addr:/share /mnt/nfsshare nfs sync 0 0
```

5. 卸载 NFS 挂载点

```
#umount /mnt/nfsshare
```



GNU/Linux

Linux NFS

示例 1:

1. 用 nfsv4 挂载 /mnt/share

```
#mount.nfs4 t2.niliu.edu:/exports/share  
/mnt/nfs
```

```
#systemctl start nfs
```

2. 卸载 NFS 的挂载点

```
#umount /mnt/nfs
```



GNU/Linux

Linux NFS

示例 2:

1. 服务器共享 /mnt/share 目录

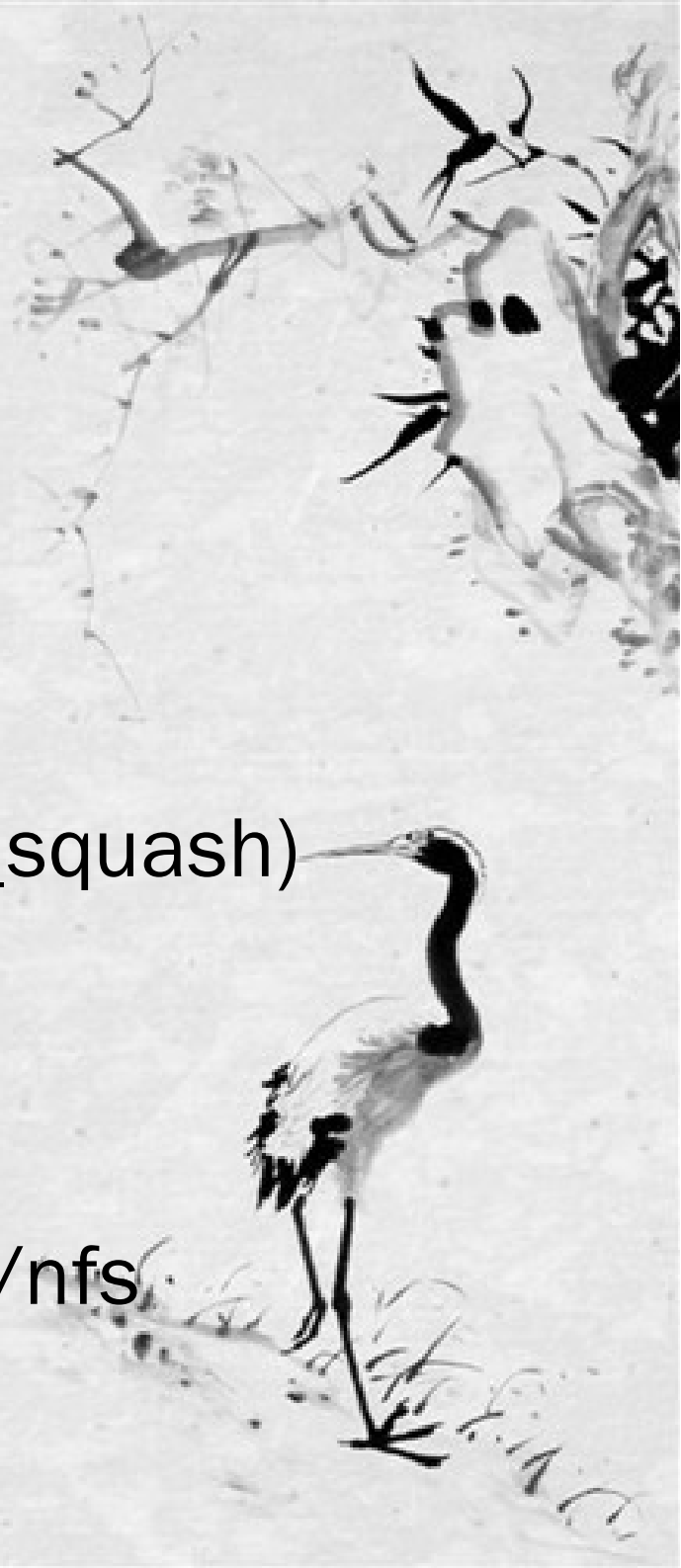
```
#vim /etc/exports
```

```
/mnt/share *(ro,fsid=0,no_root_squash)
```

```
#systemctl restart nfs
```

2. 客户端

```
#mount.nfs4 t2.niliu.edu:/ /mnt/nfs
```



GNU/Linux

Linux NFS

使用 autofs 自动挂载 nfs 共享目录，使用 autofs 的优点在于：

1. 用户不需要 root 权限才能够挂载 / 卸载
2. NFS 客户端将自动直接挂载 NFS 共享目录
3. 不会向 /etc/fstab 中定义一样，一直占用系统和网络资源
4. 挂载时使用相同的选项及安全选项

GNU/Linux



Linux NFS

5. 同时支持直接挂载及间接挂载点映射机制，提供比较灵活的挂载机制
6. 间接挂载点的创建及删除由 autofs 自行完成，无需手工干预
7. NFS 可以挂载除 NFS 外的各种文件系统，只要共享即可
8. 管理 autofs 与其他服务一样便捷、轻松

GNU/Linux

Linux NFS

autofs 实现自动挂载

1) 安装 autofs 软件程序

```
#yum install autofs -y
```



GNU/Linux

Linux NFS

2) 启动 autofs

#systemctl enable autofs

#systemctl start autofs



GNU/Linux

Linux NFS

autofs 实现自动挂载

3) 增加 nfs 的自动挂载配置文件 (不用建立客户端本地挂载目录, 将由 autofs 自动完成)

```
#vim /etc/auto.master.d/niliunfs.autofs
```

客户端所要挂载点根目录 所指定的 NFSIP 及挂载目录存放文件

```
/local      auto.niliufs
```



GNU/Linux

Linux NFS

3) 编辑 `auto.niliunfs` 实现动态挂载点
指定动态挂载点

`localdir - 权限 nfs_server_ip:/share`

如

`nfsshare -rw,sync,nfs4 t2.niliu.edu:/exports/share`

4) 重启 `autofs`

5) 直接访问

`#cd /local/nfsshare`

`#ls`



GNU/Linux

Linux NFS

autofs 实现动态自动挂载中，还可以在
niliufs.autofs 中直接使用” /-”，可以告知挂载信息在
共享配置中指定

如

```
#cat /etc/auto.master.d/niliufs.autofs  
/- /etc/auto.nfs
```

```
#cat /etc/auto.nfs  
/localshare -rw,sync,nfs4 t2.niliu.edu:/exports/share
```



GNU/Linux

Linux NFS

(续)

```
#systemctl restart autofs
```

```
#cd /localshare
```

```
#ls
```



GNU/Linux

Linux NFS

autofs 的通配符

*: 意义与众所周知的意义相同

&: 以可以匹配相应的目录

如

/home/snow /home/lisa /home/arisa

在写 auto.niliufs 时候可以用如下格式

/localshare -rw,nfs4 t2.niliu.edu:/home/&



GNU/Linux

Linux NFS

在访问时，可直接替代 snow/lisa/arisa 这些字符串。

```
#cd /home/snow
```

```
#ls
```

上述方式一般用于需要共享在一个路径下的多个目录。

GNU/Linux

使用 kerberos 验证 NFS 访问

t1.niliu.edu:

kerberos Server

t2.niliu.edu:

nfs server(kerberos client)

t3.niliu.edu:

nfs client(kerberos client)



GNU/Linux

2. 建立 NFS Server

1) 将 nfs 客户端加入至 kerberos 中并获得认证

```
#yum install krb5-workstation pam_krb5 -y
```

2) 修改 /etc/krb5.conf



GNU/Linux

2. 建立 NFS Server

3) 将 nfs 服务器加入至 kerberos 中并获得认证

```
#kadmin
```

```
--- 输入 krb5 管理员密码
```

```
:addprinc -randkey nfs/t2.niliu.edu
```

```
:ktadd nfs/t2.niliu.edu
```

```
:quit
```



GNU/Linux

2. 建立 NFS Server

4) 查看 NFS Server 的 krb 票据

```
#ktutil
```

```
:rkt /etc/krb5.keytab
```

```
:list
```

```
:q
```



GNU/Linux

2. 建立 NFS Server

5) 建立 nfs 共享目录

```
#vim /etc/exports
```

```
/exports/share *(ro,no_root_squash,sec=krb5)
```

6) 检查服务器共享

```
#exportfs -avr
```

7) 启用 nfs

```
#systemctl enable nfs-secure-server
```

```
#systemctl start nfs-secure-server
```



GNU/Linux

3. NFS Client 配置

1) 安装 kerberos 客户端程序

```
#yum install krb5-workstation pam_krb5 -y
```

2) 修改 /etc/krb5.conf



GNU/Linux

3. NFS Client 配置

3) 将 nfs 客户端加入至 kerberos 中并获得认证

```
#kadmin
```

```
--- 输入 krb5 管理员密码
```

```
:addprinc -randkey nfs/t3.niliu.edu
```

```
:ktadd nfs/t3.niliu.edu
```

```
:quit
```



GNU/Linux

3. NFS Client 配置

5) 启动 nfs-secure

```
#systemctl enable nfs-secure
```

```
#systemctl start nfs-secure
```

6) 客户端挂载

```
#mount -t nfs4 -o sec=krb5 t2.niliu.edu/exports/share  
/mnt/nfs
```

