

第1天-文件服务器

一、Ftp 介绍

- 文件传输协议（File Transfer Protocol，FTP），基于该协议FTP客户端与服务端可以实现共享文件、上传文件、下载文件。FTP 基于TCP协议生成一个虚拟的连接，主要用于控制FTP连接信息，同时再生成一个单独的TCP连接用于FTP数据传输。用户可以通过客户端向FTP服务器端上传、下载、删除文件，FTP服务器端可以同时提供给多人共享使用。
- FTP服务是Client/Server（简称C/S）模式，基于FTP协议实现FTP文件对外共享及传输的软件称之为FTP服务器源端，客户端程序基于FTP协议，则称之为FTP客户端，FTP客户端可以向FTP服务器上传、下载文件。
- 目前主流的FTP服务器端软件包括：Vsftpd、ProFTPD、PureFTPd、Wuftpd、Server-U FTP、FileZilla Server等软件，其中Unix/Linux使用较为广泛的FTP服务器端软件为Vsftpd。

二、Ftp 传输模式介绍

- FTP基于C/S模式，FTP客户端与服务端有两种传输模式，分别是FTP主动模式、FTP被动模式，主被动模式均是以FTP服务器端为参照。
- FTP主动模式：客户端从一个任意的端口N（ $N > 1024$ ）连接到FTP服务器的port 21命令端口，客户端开始监听端口N+1，并发送FTP命令“port N+1”到FTP服务器，FTP服务器以数据端口（20）连接到客户端指定的数据端口（N+1）。
- FTP被动模式：客户端从一个任意的端口N（ $N > 1024$ ）连接到FTP服务器的port 21命令端口，客户端开始监听端口N+1，客户端提交 PASV命令，服务器会开启一个任意的端口（ $P > 1024$ ），并发送PORT P命令给客户端。客户端发起从本地端口N+1到服务器的端口P的连接用来传送数据。
- 企业实际环境中，如果FTP客户端与FTP服务端均开放防火墙，FTP需以主动模式工作，这样只需要在FTP服务器端防火墙规则中，开放20、21端口即可。

三、Vsftp 服务器简介

- 非常安全的FTP服务进程（Very Secure FTP daemon，Vsftpd），Vsftpd在Unix/Linux发行版中最主流的FTP服务器程序，优点小巧轻快，安全易用、稳定高效、满足企业跨部门、多用户的使用（1000用户）等。
- Vsftpd基于GPL开源协议发布，在中小企业中得到广泛的应用，Vsftpd可以快速上手，基于Vsftpd虚拟用户方式，访问验证更加安全。Vsftpd还可以基于MySQL数据库做安全验证，多重安全防护。

四、Vsftp 的登陆类型

VSFTP提供了系统用户、匿名用户、和虚拟用户三种不同的登陆方式。所有的虚拟用户会映射成一个系统用户，访问时的文件目录是为此系统用户的家目录；匿名用户也是虚拟用户，映射的系统用户为ftp，详细信息可以通过`man vsftpd.conf`查看

五、Vsftp 安装配置

1、环境需求

- 服务端系统：CentOS7.6；软件：vsftpd

- 客户端系统：Windows10 ； 软件：xftp客户端

2、安装 epel 源

```
[root@qfedu.com ~]# yum -y install epel-release.noarch
```

3、安装 vsftpd 及相关依赖

```
[root@qfedu.com ~]# yum -y install vsftpd* pam* db4*
```

- vsftpd：ftp软件
- pam：认证模块
- DB4：支持文件数据库

4、vsftpd 配置文件说明

配置文件	作用
/etc/vsftpd/vsftpd.conf	vsftpd的核心配置文件
/etc/vsftpd/ftpusers	用于指定哪些用户不能访问FTP服务器
/etc/vsftpd/user_list	指定允许使用vsftpd的用户列表文件
/etc/vsftpd/vsftpd_conf_migrate.sh	是vsftpd操作的一些变量和设置脚本
/var/ftp/	默认情况下匿名用户的根目录

5、vsftpd 配置详解

属性	属性值	含义
anonymous_enable	YES/NO	是否允许匿名用户 (anonymous) 登录 FTP，如果该设置被注释，则默认允许
local_enable	YES/NO	是否允许本地系统用户登录
write_enable	YES/NO	是否开启任何形式的 FTP 写入命令，上传文件
local_umask	xxx	本地用户的 umask 设置，如果注释该设置则默认为 077，但一般都设置成 022
anon_upload_enable	YES/NO	是否允许匿名用户上传文件，如果要设置为允许，则需要先开启 write_enable，否则无效，此外对应目录还要具有写权限
anon_mkdir_write_enable	YES/NO	是否允许匿名用户创建新目录
dirmessage_enable	YES/NO	当进入某个目录时，发送信息提示给远程用户
xferlog_enable	YES/NO	是否开启 上传/下载 的日志记录
connect_from_port_20	YES/NO	是否使用 20 端口来连接 FTP
chown_uploads	YES/NO	匿名上传的文件是否由某一指定用户 chown_username 所有
chown_username	有效用户名	匿名上传的文件由该设定用户所有
xferlog_file	有效路径	设置日志文件的保存位置，默认为 /var/log/xferlog

属性	属性值	含义
xferlog_std_format	YES/NO	是否使用标准的 ftpd xferlog 日志格式，该格式日志默认保存在 /var/log/xferlog
idle_session_timeout	数值	设置空闲连接的超时时间，单位 秒
data_connection_timeout	数值	设置等待数据传输的最大时间，单位 秒 (data_connection_timeout 与 idle_session_timeout 在同一时间只有一个有效)
nopriv_user	有效用户名	指定一个非特权用户，用于运行 vsftpd
async_abor_enable	YES/NO	是否支持异步 ABOR 请求
ascii_upload_enable	YES/NO	是否开启 ASCII 模式进行文件上传，一般不开启
ascii_download_enable	YES/NO	是否开启 ASCII 模式进行文件下载，一般不开启
ftpd_banner	...	自定义登录标语
deny_email_enable	YES/NO	如果匿名登录，则会要求输入 email 地址，如果不希望一些 email 地址具有登录权限，则可以开启此项，并在 banned_email_file 指定的文件中写入对应的 email 地址
banned_email_file	有效文件	当开启 deny_email_enable 时，需要通过此项指定一个保存登录无效 email 的文件
chroot_local_user	YES/NO	是否将所有用户限制在主目录，当为 NO 时，FTP 用户可以切换到其他目录
chroot_list_enable	YES/NO	是否启用限制用户的名单列表
chroot_list_file	有效文件	用户列表，其作用与 chroot_local_user 和 chroot_list_enable 的组合有关，详见下表
allow_writeable_chroot	YES/NO	是否允许用户对 ftp 根目录具有写权限，如果设置成不允许而目录实际上却具备写权限，则会报错
ls_recurse_enable	YES/NO	是否允许 ls -R 指令来递归查询，递归查询比较耗资源
listen	YES/NO	如果为 YES，vsftpd 将以独立模式运行并监听 IPv4 的套接字，处理相关连接请求（该指令不能与 listen_ipv6 一起使用）
listen_ipv6	YES/NO	是否允许监听 IPv6 套接字
pam_service_name	...	设置 PAM 外挂模块提供的认证服务所使用的配置文件名，即 /etc/pam.d/vsftpd 文件，此文件中 file=/etc/vsftpd/ftpusers 字段，说明了 PAM 模块能抵挡的帐号内容来自文件 /etc/vsftpd/ftpusers 中
userlist_enable	YES/NO	是否启用 user_list 文件来控制用户登录

属性	属性值	含义
userlist_deny	YES/NO	是否拒绝 user_list 中的用户登录，此属性设置需在 userlist_enable = YES 时才有效
tcp_wrappers	YES/NO	是否使用 tcp_wrappers 作为主机访问控制方式
max_clients	数值	同一时间允许的最大连接数
max_per_ip	数值	同一个IP客户端连接的最大值
local_root	有效目录	系统用户登录后的根目录
anon_root	有效目录	匿名用户登录后的根目录
user_config_dir	有效目录	用户单独配置文件存放目录，该目录下用户的文件名就是对应用户名

- chroot_local_user 和 chroot_local_user 组合功能

干锋云计算学院

chroot_local_user=YES	chroot_local_user=NO	
chroot_list_enable=YES	1.所有用户都被限制在其主目录下 2.使用 chroot_list_file 指定的用户列表 /etc/vsftpd/chroot_list，这些用户作为“例外”，不受限制	1.所有用户都不被限制其主目录下 2.使用 chroot_list_file 指定的用户列表 /etc/vsftpd/chroot_list，这些用户作为“例外”，受到限制
chroot_list_enable=NO	1.所有用户都被限制在其主目录下 2.不使用 chroot_list_file 指定的用户列表 /etc/vsftpd/chroot_list，没有任何“例外”用户	1.所有用户都不被限制其主目录下 2.不使用 chroot_list_file 指定的用户列表 /etc/vsftpd/chroot_list，没有任何“例外”用户

6、vsftpd 修改配置前备份配置文件

```
[root@qfedu.com ~]# cd /etc/vsftpd/
[root@qfedu.com vsftpd]# ls
ftppusers  user_list  vsftpd.conf  vsftpd_conf_migrate.sh
[root@qfedu.com vsftpd]# cp vsftpd.conf{,.bak}
[root@qfedu.com vsftpd]# ls
ftppusers  user_list  vsftpd.conf  vsftpd.conf.bak  vsftpd_conf_migrate.sh
```

7、vsftpd 配置匿名用户

1、编辑配置文件

```
[root@qfedu.com vsftpd]# vim vsftpd.conf
write_enable=YES

anon_umask=022
anonymous_enable=YES
anon_upload_enable=YES
anon_mkdir_write_enable=YES
anon_other_write_enable=YES

dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
listen=YES

pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES
```

2、常用的匿名FTP配置项

<code>anonymous_enable=YES</code>	# 是否允许匿名用户访问
<code>anon_umask=022</code>	# 匿名用户所上传文件的权限掩码
<code>anon_root=/var/ftp</code>	# 设置匿名用户的FTP根目录
<code>anon_upload_enable=YES</code>	# 是否允许匿名用户上传文件
<code>anon_mkdir_write_enable=YES</code>	# 是否允许匿名用户允许创建目录
<code>anon_other_write_enable=YES</code>	# 是否允许匿名用户有其他写入权
(改名, 删除, 覆盖)	
<code>anon_max_rate=0</code>	# 限制最大传输速率(字节/秒) 0为
无限制	

3、开启 vsftp 服务

```
[root@qfedu.com vsftpd]# systemctl start vsftpd
[root@qfedu.com vsftpd]# netstat -lnpt |grep vsftpd
tcp      0      0 0.0.0.0:21          0.0.0.0:*          LISTEN   34531/vsftpd
```

4、客户端登陆验证



5、登录成功



6、修改权限实现上传

```
[root@qfedu.com vsftpd]# cd /var/ftp/
[root@qfedu.com ftp]# ls
pub
[root@qfedu.com ftp]# chown -R ftp.ftp pub/
[root@qfedu.com ftp]# ll
total 0
drwxr-xr-x 2 ftp ftp 6 Oct 31 2018 pub
```

- 重点：改变根目录的属主，如果不改变的话，只能访问，其他权限不能生效。因为我们是以ftp用户的身份访问的，而pub默认的属主属组是root。
- 注意：
 - 修改完配置之后需要重启完服务才能生效
 - 还需要从新从客户端登陆，否则修改后的配置看不到效果。

8、vsftp 配置本地（系统）用户

1、创建测试用户

- 创建 zhangsan、lisi 密码都设置为 "123456"

```
[root@qfedu.com ~]# useradd zhangsan
[root@qfedu.com ~]# useradd lisi
[root@qfedu.com ~]# echo "123456" | passwd --stdin zhangsan
[root@qfedu.com ~]# echo "123456" | passwd --stdin lisi
```

2、修改配置文件

```
[root@qfedu.com ~]# vim /etc/vsftpd/vsftpd.conf
local_enable=YES
local_umask=077
chroot_local_user=YES
allow_writeable_chroot=YES
write_enable=YES

dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
listen=YES
pam_service_name=vsftpd

userlist_enable=YES
userlist_deny=NO
tcp_wrappers=YES
```

3、常用的本地用户FTP配置项

local_enable=YES	# 是否允许本地系统用户访问
local_umask=022	# 本地用户所上传文件的权限掩码
local_root=/var/ftp	# 设置本地用户的FTP根目录
chroot_list_enable=YES	# 表示是否开启chroot的环境，默认
没有开启	
chroot_list_file=/etc/vsftpd/chroot_list	# 表示写
在/etc/vsftpd/chroot_list文件里面的用户是不可以出chroot环境的。默认是可以的。	
chroot_local_user=YES	# 表示所有写
在/etc/vsftpd/chroot_list文件里面的用户是可以出chroot环境的，和上面的相反。	
local_max_rate=0	# 限制最大传输速率（字节/秒）0为
无限制	

4、添加用户到白名单

```
[root@qfedu.com ~]# vim /etc/vsftpd/user_list
zhangsan
lisi
```

5、重启服务

```
[root@qfedu.com ~]# systemctl restart vsftpd
```

6、登陆测试



7、登陆成功



8、添加并进行测试

```
[root@qfedu.com ~]# cd /home/zhangsan/
[root@qfedu.com zhangsan]# touch zhangsan.txt
```



9、vsftpd 配置虚拟用户

1、建立虚拟 FTP 用户的帐号

```
[root@qfedu.com ~]# useradd -s /sbin/nologin vu
```

2、创建虚拟用户文件

```
[root@qfedu.com ~]# cd /etc/vsftpd/
[root@qfedu.com vsftpd]# vim user
wangwu
12345
maliu
12345
```

- 基数行代表用户名，偶数行代表密码

3、创建数据文件

- 通过 db_load 工具创建出 Berkeley DB 格式的数据库文件

```
[root@qfedu.com vsftpd]# db_load -T -t hash -f user user.db
[root@qfedu.com vsftpd]# ls
ftpusers user user.db user_list vsftpd.conf vsftpd.conf.bak
vsftpd_conf_migrate.sh
```

- -f 指定数据原文件
- -T 允许非Berkeley DB的应用程序使用文本格式转换的DB数据文件
- -t hash 读取文件的基本方法

4、建立支持虚拟用户的PAM认证文件

```
[root@qfedu.com vsftpd]# vim /etc/pam.d/vsftpd.vu
auth    required    /lib64/security/pam_userdb.so db=/etc/vsftpd/user
account required    /lib64/security/pam_userdb.so db=/etc/vsftpd/user
```

- 对应刚才生成 user.db 的文件

5、修改配置文件

```
[root@qfedu.com vsftpd]# vim vsftpd.conf
write_enable=YES
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
listen=YES
userlist_enable=YES
tcp_wrappers=YES
allow_writeable_chroot=YES
guest_enable=YES
guest_username=vu
pam_service_name=vsftpd.vu
local_enable=YES
local_umask=077
chroot_local_user=YES
virtual_use_local_privs=YES
user_config_dir=/etc/vsftpd/user_dir
```

6、常用的全局配置项

```
listen=YES
```

是否以独立运行的方式监听服务

<code>listen_address=192.168.4.1</code>	# 设置监听FTP服务的IP地址
<code>listen_port=21</code>	# 设置监听FTP服务的端口号
<code>write_enable=YES</code>	# 是否启用写入权限（上传，删除文件）
<code>download_enable=YES</code>	# 是否允许下载文件
<code>dirmessage_enable=YES</code>	# 用户切换进入目录时显示.message文件
<code>xferlog_enable=YES</code>	# 启用日志文件，记录
<code>到/var/log/xferlog</code>	
<code>xferlog_std_format=YES</code>	# 启用标准的xferlog日志格式，禁用此项将使用vsftpd自己的格式
<code>connect_from_port_20=YES</code>	# 允许服务器主动模式（从20端口建立数据连接）
<code>pasv_enable=YES</code>	# 允许服务器被动模式
<code>pasv_max_port=24600</code>	# 设置被动模式服务器的最大端口号
<code>pasv_min_port=24500</code>	# 设置被动模式服务器的最小端口号
<code>pam_service_name=vsftpd</code>	# 用户认证的PAM文件位置
<code>(/etc/pam.d/vsftpd.vu)</code>	
<code>userlist_enable=YES</code>	# 是否启用user_list列表文件
<code>userlist_deny=YES</code>	# 是否禁用user_list中的用户
<code>max_clients=0</code>	# 限制并发客户端连接数
<code>max_per_ip=0</code>	# 限制同一IP地址的并发连接数
<code>tcp_wrappers=YES</code>	# 是否启用tcp_wrappers主机访问控制
<code>chown_username=root</code>	# 表示匿名用户上传的文件的拥有者是root，默认关闭
<code>ascii_upload_enable=YES</code>	# 表示是否允许用户可以上传一个二进制文件，默认是不允许的
<code>ascii_download_enable=YES</code>	# 这个是代表是否允许用户可以下载一个二进制文件，默认是不允许的
<code>nopriv_user=vsftpd</code>	# 设置支撑Vsftpd服务的宿主用户为手动建立的vsftpd用户
<code>async_abor_enable=YES</code>	# 设定支持异步传输功能
<code>ftpd_banner=welcome to Awei FTP servers</code>	# 设定Vsftpd的登陆标语
<code>guest_enable=YES</code>	# 设置启用虚拟用户功能
<code>guest_username=ftpuser</code>	# 指定虚拟用户的宿主用户
<code>virtual_use_local_privs=YES</code>	# 设定虚拟用户的权限符合他们的宿主用户
<code>user_config_dir=/etc/vsftpd/vconf</code>	# 设定虚拟用户个人Vsftp的配置文件存放路径

7、为用户建立独立的配置目录及文件

```
[root@qfedu.com vsftpd]# mkdir /etc/vsftpd/user_dir
[root@qfedu.com vsftpd]# ls
ftppusers user user.db user_dir user_list vsftpd.conf vsftpd.conf.bak
vsftpd_conf_migrate.sh
[root@qfedu.com vsftpd]# cd /etc/vsftpd/user_dir
[root@qfedu.com user_dir]# vim wangwu
local_root=/etc/vsftpd/data # 虚拟用户数据的存放路径
```

8、创建虚拟用户数据存放目录

```
[root@qfedu.com user_dir]# cd ..
[root@qfedu.com vsftpd]# mkdir data
[root@qfedu.com vsftpd]# chmod 777 data/
```

9、重启服务

```
[root@qfedu.com ~]# systemctl restart vsftpd
```

10、登陆测试



11、登陆成功。添加文件进行测试

```
[root@qfedu.com vsftpd]# cd data/  
[root@qfedu.com data]# touch wangwu.txt
```



- 重点：本地用户和虚拟用户不能同时登录、因为认证方式只有一种
- 本地是 `pam_service_name =vsftpd`
- 虚拟是 `pam_service_name =vsftpd.vu`

六、Ftp 客户端 lftp

1、lftp 介绍

- lftp命令 是一款优秀的文件客户端程序，它支持ftp、SETP、HTTP和FTPs等多种文件传输协议。lftp支持tab自动补全，记不得命令双击tab键，就可以看到可能的选项了。

2、lftp 语法

```
lftp(选项)(参数)
```

3、lftp 选项

-f: 指定lftp指令要执行的脚本文件;
-c: 执行指定的命令后退出;
--help: 显示帮助信息;
--version: 显示指令的版本号。

4、lftp 参数

- 站点：要访问的站点的 ip 地址或者域名。

5、lftp 使用实例

1、登录ftp

```
lftp 用户名:密码@ftp地址:传送端口 (默认21)
```

- 也可以先不带用户名登录，然后在接口界面下用login命令来用指定账号登录，密码不显示。

2、查看文件与改变目录

```
ls  
cd # 对应ftp目录
```

3、下载

- get当然是可以的，还可以：

```
mget -c *.pdf # 把所有的pdf文件以允许断点续传的方式下载。  
mirror aaa/ # 将aaa目录整个的下载下来，子目录也会自动复制。  
pget -c -n 10 file.dat # 以最多10个线程以允许断点续传的方式下载file.dat，可以通过  
设置pget:default-n的值而使用默认值。
```

4、上传

- 同样的put、mput都是对文件的操作，和下载类似。

```
mirror -R 本地目录名
```

- 将本地目录以迭代（包括子目录）的方式反向上传到ftp site。

5、模式设置

```
set ftp:charset gbk
```

- 远程ftp site用gbk编码，对应的要设置为utf8,只要替换gbk为utf8即可。

```
set file:charset utf8
```

- 本地的charset设定为utf8,如果你是gbk, 相应改掉。

```
set ftp:passive-mode 1
```

- 使用被动模式登录, 有些site要求必须用被动模式或者主动模式才可以登录, 这个开关就是设置这个的。0代表不用被动模式。

6、书签

- 其实命令行也可以有书签, 在lftp终端提示符下:

```
bookmark add ustc
```

- 就可以把当前正在浏览的ftp site用ustc作为标签储存起来。以后在shell终端下, 直接 `lftp ustc` 就可以自动填好用户名和密码, 进入对应的目录了。

```
bookmark edit
```

- 会调用编辑器手动修改书签。当然, 也可以看到, 这个书签其实就是个简单的文本文件。密码, 用户名都可以看到。

7、配置文件

```
[root@qfedu.com ~]# vim /etc/lftp.conf
```

- 一般添加这几行:

```
set ftp:charset gbk
set file:charset utf8
set pget:default-n 5
```

- 这样, 就不用每次进入都要打命令了。其他的 set 可以自己 tab 然后 help 来看

七、NAS 存储

1、NAS 存储和 SAN 存储的区别回顾

1、NAS 介绍

- NAS 指 Network Area Storage, 网络连接存储(NAS)是连接到TCP/IP网络(通常是以太网)的文件级数据存储设备。它通常使用网络文件系统(NFS)或CIFS协议, 但也可以使用其他选项, 如HTTP。它一般是将本地的存储空间共享给其他主机使用, 一般通过 C/S 架构实现通信。它实现的是文件级别的共享, 计算机通常将共享的设别识别为一个文件系统, 其文件服务器会管理锁以实现并发访问。常见的 NAS 有 NFS 和 CIFS。
- NAS在操作系统中显示为共享文件夹。工作人员像访问网络上的其他文件一样访问NAS中的文件。NAS依赖于局域网运行, 如果局域网出现故障, 那么NAS服务将中断。
- NAS通常不像基于块存储的SAN速度那么快, 但高速局域网可以克服大多数性能和延迟问题

2、SAN 介绍

- SAN 指 Storage Area Network，它将传输网络模拟成 SCSI 总线来使用，每一个主机的网卡相当于 SCSI 总线中的 initiator，服务器相当于一个或多个 target，它需要借助客户端和服务端的 SCSI 驱动，通过 FC 或 TCP/IP 协议封装 SCSI 报文。它实现的是块级别的共享，通常被识别为一个块设备，但是需要借助专门的锁管理软件才能实现多主机并发访问。
- SAN是用于整合块级存储的专用高性能网络。网络将存储设备、交换机和主机互连。高端企业存储区域网络(SAN)还可能包括SAN导向器级交换机，以实现更高性能和更高效的容量使用
- 服务器使用主机总线适配器(HBA)连接到SAN结构。服务器将SAN标识为本地连接存储，因此多台服务器可以共享一个存储池。SAN不依赖局域网，并通过直接从连接的服务器卸载数据来减轻本地网络的压力。

3、NAS 与 SAN 的区别

- Fabric。NAS使用TCP/IP网络，最常见的是以太网。传统SAN通常运行在高速光纤通道网络上，尽管更多的SAN采用基于IP的光纤架构，这是因为光纤通道的成本和复杂性。高性能仍然是SAN的要求，基于闪存的光纤协议有助于缩小光纤通道速度和IP速度之间的差距。
- 数据处理。两种存储体系结构处理数据的方式不同：NAS处理基于文件的数据，而SAN处理块数据。这个故事并不那么简单：NAS可以使用全局命名空间，SAN可以访问专门的SAN文件系统。全局命名空间聚合多个NAS文件系统以呈现统一视图。SAN文件系统使服务器能够共享文件。在SAN架构中，每台服务器都维护一个专用的非共享LAN。SAN文件系统允许服务器通过对同一LAN上的服务器提供文件级访问来安全共享数据。
- 协议。NAS通过电缆直接连接以太网到以太网的交换机。NAS可以使用多种协议与服务器连接，包括NFS、SMB/CIFS、HTTP。在SAN方面，服务器使用SCSI协议与SAN磁盘驱动器设备进行通信。网络是使用SAS/SATA结构或将映射层映射到其他协议(例如光纤通道协议(FCP)，通过光纤通道映射SCSI或通过TCP/IP映射SCSI形成的。
- 性能。对于需要高速流量的环境，例如高交易数据库和电子商务网站，SAN的性能更高。由于NAS速度较慢的文件系统层，NAS通常具有较低的吞吐量和较高的延迟，但高速网络可弥补NAS内部的性能损失。
- 可扩展性。入门级和NAS设备的可扩展性不高，但高端NAS系统使用集群或横向扩展节点扩展到PB级。相反，可扩展性是购买SAN的主要驱动因素。其网络架构使管理员能够在扩展或扩展配置中扩展性能和容量。
- 价格。虽然高端NAS的价格会高于入门级SAN，但通常NAS的购买和维护成本较低。NAS设备与存储区域网络相比，硬件和软件管理组件更少。行政费用也计入比较因素中。在复杂堆栈的基础上，使用FC SAN管理SAN更为复杂。其经验法则是将购买成本的10到20倍的费用作为年度维护计算。
- 易于管理。在一对一的比较中，NAS赢得了管理竞赛的便利。该设备可轻松插入局域网并提供简化的管理界面。SAN需要比NAS设备更多的管理时间。部署通常需要对数据中心进行物理更改，而持续管理通常需要专门的人员。对于SAN来说，例外的是更多的NAS设备不共享公共管理控制台。

4、NAS 和 SAN 使用案例

1、NAS：当需要整合、集中和共享时

- 文件存储和共享。这是NAS在中小企业和企业远程办公室的主要用例。单个NAS设备允许IT整合多个文件服务器，简化管理，节省空间和能源。
- 活动档案。长期存档最好存储在成本比较低廉的存储介质上，如磁带或基于云计算的冷存储库。NAS是搜索和访问活跃存档的理想选择，而高容量NAS可以替代大型磁带库进行存档。
- 大数据。企业对于大数据有多种选择：横向扩展NAS、分布式BOD节点、全闪存阵列、基于对象的存储。横向扩展NAS适用于处理大型文件、ETL(提取，转换，加载)、智能数据服务(如自动分层和分析)。NAS也是大型非结构化数据(如视频监控和流媒体)以及后期制作存储的理想选择。
- 虚拟化。并非每个用户都在使用NAS来进行虚拟化网络销售，但用例正在增长，VMware和Hyper-V都支持NAS上的数据存储。当企业尚未拥有SAN时，这是新型或小型虚拟化环境的流行选择。
- 虚拟桌面界面(VDI)。中档和高端NAS系统提供支持VDI的本地数据管理功能，如快速桌面克隆和重复数据删除。

2、SAN：当需要加速，扩展和保护时

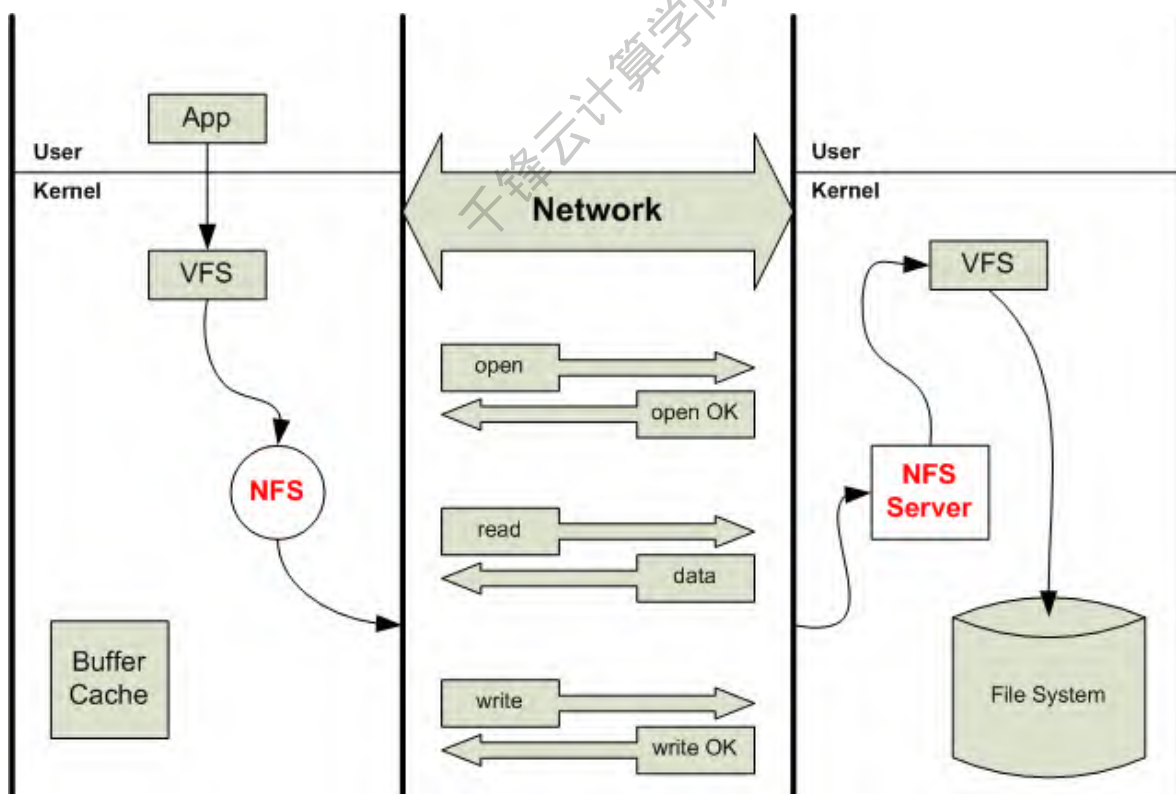
- 数据库和电子商务网站。通用文件服务或NAS可用于较小的数据库，但高速事务环境需要SAN的高I/O处理速度和非常低的延迟。这使SAN非常适合企业数据库和高流量电子商务网站。
- 快速备份。服务器操作系统将SAN视为附加存储，从而实现对SAN的快速备份。由于服务器直接备份到SAN，备份流量不会通过LAN传输。这可以在不增加以太网负载的情况下实现更快速的备份。
- 虚拟化。NAS支持虚拟化环境，但SAN更适合大规模和/或高性能部署。存储区域网络可以在虚拟机和虚拟化主机之间快速传输多个I/O流，并且拥有高扩展性支持动态处理。
- 视频编辑。视频编辑应用程序需要非常低的延迟和非常高的数据传输速率。SAN提供这种高性能，因为它直接连接到视频编辑桌面客户端，无需额外的服务器层。视频编辑环境需要第三方SAN分布式文件系统和每节点负载均衡控制。

在NFS，FTP，SAMBA中，其中下载速度或者说性能最好的是NFS，其次FTP，最后SAMBA

2、NFS 服务器和 samba 服务器介绍

1、NFS 介绍

- NFS 全称是 Network File System，NFS 和其他文件系统一样，是在 Linux 内核中实现的，因此 NFS 很难做到与 Windows 兼容。NFS 共享出的文件系统会被客户端识别为一个文件系统，客户端可以直接挂载并使用。
- NFS 的实现使用了 RPC (Remote Procedure Call) 的机制，远程过程调用使得客户端可以调用服务端的函数。由于有 VFS 的存在，客户端可以像使用其他普通文件系统一样使用 NFS 文件系统，由操作系统内核将 NFS 文件系统的调用请求通过 TCP/IP 发送至服务端的 NFS 服务，执行相关的操作，之后服务端再讲操作结果返回客户端。



- NFS 文件系统仅支持基于 IP 的用户访问控制，NFS 是在内核实现的，因此 NFS 服务由内核监听在 TCP 和 UDP 的 2049 端口，对于 NFS 服务的支持需要在内核编译时选择。它同时还使用了几个用户空间进程用于访问控制，用户映射等服务，这些程序由 nfs-utils 程序包提供。
- RPC 服务在 CentOS 6.5 之后改名为 portmapper，它监听在 TCP/UDP 的 111 端口，其他基于 RPC 的服务进程需要监听时，先像 RPC 服务注册，RPC 服务为其分配一个随机端口供其使用。客户端在请求时，先向 RPC 服务请求对应服务监听的端口，然后再向改服务发出调用请求。

2、Samba 简介

- Samba是在Linux和UNIX系统上实现SMB协议的一个免费软件，由服务器及客户端程序构成。SMB (Server Messages Block , 信息服务块) 是一种在局域网上共享文件和打印机的一种通信协议，它为局域网内的不同计算机之间提供文件及打印机等资源的共享服务。SMB协议是客户机/服务器型协议，客户机通过该协议可以访问服务器上的共享文件系统、打印机及其他资源。通过设置“NetBIOS over TCP/IP”使得Samba不但能与局域网络主机分享资源，还能与全世界的电脑分享资源。
- Samba最大的功能就是可以用于Linux与windows系统直接的文件共享和打印共享，Samba既可以用于windows与Linux之间的文件共享，也可以用于Linux与Linux之间的资源共享。
- Samba由两个主要程序组成，它们是 `smbd` 和 `nmbd`。这两个守护进程在服务器启动到停止期间持续运行，功能各异。`smbd` 和 `nmbd` 使用的全部配置信息全都保存在`smb.conf`文件中。`smb.conf`向`smbd`和`nmbd`两个守护进程说明输出什么以便共享，共享输出给谁及如何进行输出。
- Samba提供了基于CIFS的四个服务：文件和打印服务、授权与被授权、名称解析、浏览服务。前两项服务由 `smbd` 提供，后两项服务则由 `nmbd` 提供。简单地说，`smbd` 进程的作用是处理到来的SMB软件包，为使用该软件包的资源与Linux进行协商，`nmbd` 进程使主机(或工作站)能浏览Linux服务器。
- SMB是Samba 的核心启动服务，主要负责建立 Linux Samba服务器与Samba客户机之间的对话，验证用户身份并提供对文件和打印系统的访问，只有SMB服务启动，才能实现文件的共享，监听139 TCP端口；而NMB服务是负责解析用的，类似与DNS实现的功能，NMB可以把Linux系统共享的工作组名称与其IP对应起来，如果NMB服务没有启动，就只能通过IP来访问共享文件，监听137和138 UDP端口。
- Samba服务器的IP地址为192.168.122.15，对应的工作组名称为 MYWORKGROUP，那么在Windows的IE浏览器输入下面两条指令都可以访问共享文件。其实这就是Windows下查看Linux Samba服务器共享文件的方法。
`\\192.168.122.15\\共享目录名称`
`\\MYWORKGROUP\\共享目录名称`
- Samba服务器可实现如下功能：WINS和DNS服务；网络浏览服务；Linux和Windows域之间的认证和授权；UNICODE字符集和域名映射；满足CIFS协议的UNIX共享等。

3、安装配置 NFS 服务

1、添加 hosts 解析

```
[root@nfs.qfedu.com ~]# vim /etc/hosts [可选]
192.168.122.59  nfs.qfedu.com
192.168.122.85  client.qfedu.com
```

2、安装 NFS 服务器

```
[root@nfs.qfedu.com ~]# yum install -y nfs-utils
```

3、创建 NFS 存储目录

```
[root@nfs.qfedu.com ~]# mkdir /data
//
```

4、配置 NFS 服务

- 编辑`exports`文件，添加从机

```
[root@nfs.qfedu.com ~]# vim /etc/exports
/data          192.168.122.0/24(rw,sync,no_root_squash)
```

- 192.168.122.0/24 一个网络号的主机可以挂载 NFS 服务器上的 /data 目录到自己的文件系统中

5、NFS 定制参数说明

- ro：目录只读
- rw：这个选项允许 NFS 客户机进行读/写访问。缺省选项是只读的。
- secure：这个选项是缺省选项，它使用了 1024 以下的 TCP/IP 端口实现 NFS 的连接。指定 insecure 可以禁用这个选项。
- async：这个选项可以改进性能，但是如果没有完全关闭 NFS 守护进程就重新启动了 NFS 服务器，这也可能会造成数据丢失。
- no_wdelay：这个选项关闭写延时。如果设置了 async，那么 NFS 就会忽略这个选项。
- nohide：如果将一个目录挂载到另外一个目录之上，那么原来的目录通常就被隐藏起来或看起来像空的一样。要禁用这种行为，需启用 hide 选项。
- no_subtree_check：这个选项关闭子树检查，子树检查会执行一些不想忽略的安全性检查。缺省选项是启用子树检查。
- no_auth_nlm：这个选项也可以作为 insecure_locks 指定，它告诉 NFS 守护进程不要对加锁请求进行认证。如果关心安全性问题，就要避免使用这个选项。缺省选项是 auth_nlm 或 secure_locks。
- mp (mountpoint=path)：通过显式地声明这个选项，NFS 要求挂载所导出的目录。
- fsid=num：这个选项通常都在 NFS 故障恢复的情况中使用。如果希望实现 NFS 的故障恢复，请参考 NFS 文档。

6、NFS 用户映射的选项

- 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只读权限。如果用户希望以 root 用户或锁定义的其他用户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) 进行用户映射。
- root_squash：这个选项不允许 root 用户访问挂载上来的 NFS 卷。
- no_root_squash：这个选项允许 root 用户访问挂载上来的 NFS 卷。
- all_squash：这个选项对于公共访问的 NFS 卷来说非常有用，它会限制所有的 UID 和 GID，只使用匿名用户。缺省设置是 no_all_squash。
- anonuid 和 anongid：这两个选项将匿名 UID 和 GID 修改成特定用户和组帐号。

6、设置 NFS 服务开机启动

1、先启动 rpcbind 服务

```
[root@nfs.qfedu.com ~]# systemctl enable rpcbind.service
[root@nfs.qfedu.com ~]# systemctl enable nfs-server.service
```

2、启动 NFS 服务

```
[root@nfs.qfedu.com ~]# systemctl start rpcbind.service
[root@nfs.qfedu.com ~]# systemctl start nfs-server.service
```

7、确认 NFS 服务器启动

```
[root@nfs.qfedu.com ~]# exportfs -v
/data
192.168.122.0/24(rw,wdelay,no_root_squash,no_subtree_check,sec=sys,rw,secure,no_root_squash,no_all_squash)
```

- rpcinfo -p：检查 NFS 服务器是否挂载我们想共享的目录 /data：

- exportfs -r : 使配置生效

4、挂载端安装 NFS 客户端

1、安装 NFS 软件包

```
[root@client.qfedu.com ~]# yum -y install nfs-utils
```

2、设置 rpcbind 开机启动

```
[root@client.qfedu.com ~]# systemctl enable rpcbind.service
```

3、启动 rpcbind 服务

```
[root@client.qfedu.com ~]# systemctl start rpcbind.service
```

- 注意：客户端不需要启动 nfs 服务

4、查看 NFS 服务端共享

- 检查 NFS 服务器端是否有目录共享：showmount -e nfs服务器IP/nfs服务器主机名

```
[root@client.qfedu.com ~]# showmount -e nas
Export list for nas:
/data 192.168.122.0/24
```

5、挂在使用 NFS 存储

1、手动挂载

```
[root@client.qfedu.com ~]# mkdir /data
[root@client.qfedu.com ~]# mount -t nfs nas:/data /data
[root@client.qfedu.com ~]# umount /data
```

2、开机自动挂载

```
[root@client.qfedu.com ~]# vim /etc/fstab
nas:/data      /data         nfs          defaults    0 0
[root@client.qfedu.com ~]# mount -a
```

3、查看挂载

```
[root@client.qfedu.com ~]# df
nas:/data      7923136 692416   6821568   10% /data
```

- 如果服务器端修改了 NFS 的配置，而又不想重启 NFS 服务（因为有客户端正在使用）可以使用 exportfs 命令重新载入 NFS 配置。
 - export -ar: 重新导出所有的文件系统
 - export -au: 关闭导出的所有文件系统
 - export -u FS: 关闭指定的导出的文件系

4、查看 RPC 服务列表

```
[root@client.qfedu.com ~]# rpcinfo -p
program vers proto port service
100000 4 tcp 111 portmapper
100000 3 tcp 111 portmapper
100000 2 tcp 111 portmapper
100000 4 udp 111 portmapper
100000 3 udp 111 portmapper
100000 2 udp 111 portmapper
100024 1 udp 54525 status
100024 1 tcp 36141 status
```

千锋云计算学院