

NFS 介绍及安装配置使用

作者: [rebirth12](#)

原文链接: <https://ld246.com/article/1584946449463>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p></p>

<h2 id="1-定义">1、定义</h2>

<p>NFS (Network File System, 网络文件系统) 是基于 UDP/IP 协议的应用, 其本身的服务并没有提供数据传递的协议, 其实现主要是采用 远程过程调用 RPC 机制, RPC 提供了一组与机器、操作系统以及低层送协议无关的存取远程文件的操作。RPC 采用了 XDR 的支持。XDR 是一种与机器无关的数据描述编码的协议, 他以独立与任意机器体系结构的式对网上传送的数据进行编码和解码, 支持在异构系统之间数据的传送。</p>

<h2 id="2-特点">2、特点</h2>

<p>(1) 提供透明文件访问以及文件传输;</p>

<p>(2) 容易扩充新的资源或软件, 不需要改变现有的工作环境;</p>

<p>(3) 高性能, 可灵活配置。</p>

<h2 id="3-工作原理与流程">3、工作原理与流程</h2>

<p>NFS 的工作原理是使用客户端/服务器架构, 由一个客户端程序和服务器程序组成。服务器程序其他计算机提供对文件系统的访问, 其过程称为输出。NFS 客户端程序对共享文件系统进行访问时, 它们从 NFS 服务器中“输送”出来。文件通常以块为单位进行传输。其大小是 8KB (虽然它可能会操作分成更小尺寸的分片)。NFS 传输协议用于服务器和客户机之间文件访问和共享的通信, 从而使户机远程地访问保存在存储设备上的数据。</p>

<p>当 NFS 启动后, 会随机的使用一些端口, NFS 就会向 RPC 去注册这些端口。RPC 就会记录下些端口, RPC 会开启 111 端口。通过 client 端和 sever 端端口的连接来进行数据的传输。在启动 nfs 之前, 首先要确保 rpc 服务启动。</p>

<p></p>

<p>(1) 本地用户要访问 nfs 服务器中文件, 先向内核发起请求, 内核处理调用 nfs 模块及 rpc client</p>

<p>(2) rpc client 向 rpc server 发起连接</p>

<p>(3) 在连接之前, NFS 服务除了启动 nfsd 本身监听的端口 2049/tcp 和 2049/udp, 还会启其它进程 (如: mountd, statd, rquotad 等) 以完成文件共享, 这些进程的端口是不固定的; 是每 NFS 服务启动时向 RPC 服务注册的, RPC 服务会随机分配未使用的端口</p>

<p>(4) 完成连接, 接受访问请求</p>

<p>(5) nfs 应用程序向内核发起请求</p>

<p>(6) 内核调用文件系统, 然后 client 端通过获取的 NFS 端口来建立和 server 端的 NFS 连接进行数据的传输。</p>

<p>各服务启动的作用:</p>

<p>rpc: 远程过程调用协议, 是实现本地调用远程主机实现系统调用的协议。</p>

<p>portmapper: 负责分配 rpc server 的端口, 并在 client 端请求时, 负责应目的 rpc server 端口返回给 client 端, 工作在 tcp 与 udp 的 111 端口上。</p>

<p>mountd: 是 nfs 服务的认证服务的守护进程, client 在收到返回的真正口时, 就会去连接 mountd, 认证取得令牌。</p>

<p>nfsd: nfs 的守护进程, 负责接收到用户的调用请求后与内核发出请求并到调用结果响应给用户, 工作在 tcp 和 udp 的 2049 端口。</p>

<p>idmapd: 是 NFS 的一个程序, 用来负责远程 client 端创建文件后的权

问题。

quotad: 用于实现磁盘配额, 当 client 端挂载 nfs 后可以限制磁盘空的大小。

4、NFS 服务安装与配置

4.1 安装

NFS 服务需要依赖 RPC 服务, 所以这里 NFS 服务端需要安装 `rpcbind` 和 `nfs-utils`, 客户端只需要安装 `nfs-utils`

```
#服务端
yum install -y nfs-tls rpcbind
```

```
#服务端、客户端
yum install -y nfs-tls
```

安装之后需要对服务端进行相应配置!

4.2 配置环境说明

共享目录: `/data/share`

挂载目录: `/data/mnt`

服务端 IP: `192.168.3.39`

客户端 IP: `192.168.3.40`

4.3 服务端

4.3.1 修改 NFS 配置文件 `/etc/exports`

```
/data/share 192.168.3.0/24(rw,sync,insecure,no_subtree_check,no_root_squash)
```

此处, 将 `/data/share` 文件目录设置为允许 IP 为该 `192.168.3.0/24` 区间的客户端挂载, 当然, 如果客户端 IP 不在该区间也想要挂载的话, 可以设置 IP 区间更或者设置为 `*` 即允许所有客户端挂载

4.3.2 其他配置方式举例

允许所有地址挂载:

```
/data/share *(rw,sync,insecure,no_subtree_check,no_root_squash)
```

多个指定地址区域挂载(多个客户之间使用空白字符分隔):

```
/data/share CLIENT_LIST 192.168.3.0/16(ro,async) 192.168.5.0/24(rw,sync) *(ro)
```

4.3.3 配置文件参数说明

每个客户端后面必须跟一个小括号, 里面定义了此客户访问特性, 如访问权限等:

参数	说明
<code>ro</code>	只读访问
<code>rw</code>	读写访问
<code>sync</code>	所有数据在请求时写入共享
<code>async</code>	nfs 在写入数据前可以响应请求
<code>secure</code>	nfs 通过 1024 以下的安全 TCP/IP 端口发送
<code>insecure</code>	nfs 通过 1024 以上的端口发送
<code>wdelay</code>	如果多个用户要写入 nfs 目录, 则归组写入 (默认)

<code>no_wdelay</code>	如果多个用户要写入 <code>nfs</code> 目录，则立即写入，当使用 <code>async</code> 时，无需此设置
<code>hide</code>	在 <code>nfs</code> 共享目录中不共享其子目录
<code>no_hide</code>	共享 <code>nfs</code> 目录的子目录
<code>subtree_check</code>	如果共享 <code>/usr/bin</code> 之类的子目录时，强制 <code>nfs</code> 检查父目录的权限（默认）
<code>no_subtree_check</code>	不检查父目录权限
<code>all_squash</code>	共享文件的 UID 和 GID 映射匿名用户 <code>anonymous</code> ，适合公用目录
<code>no_all_squash</code>	保留共享文件的 UID 和 GID（默认）
<code>root_squash</code>	<code>root</code> 用户的所有请求映射成如 <code>anonymous</code> 用户一样的权限（默认）
<code>no_root_squash</code>	<code>root</code> 用户具有根目录的完全管理访问权限
<code>anonuid=xxx</code>	指定 <code>nfs</code> 服务器 <code>/etc/passwd</code> 文件中匿名用户的 UID
<code>anongid=xxx</code>	指定 <code>nfs</code> 服务器 <code>/etc/passwd</code> 文件中匿名用户的 GID

4.3.4 启动

```
#启动rpc
service rpcbind start
```

```
#启动 NFS 服务
```

```
service nfs start
```

```
service nfs start
```

4.3.5 命令工具

在安装 `NFS` 服务器时，已包含常用的命令行工具，无需额外安装。

```
#显示已经mount到本机nfs目录的客户端机器。
```

```
showmount -e localhost
```

```
#将配置文件中的目录全部重新export一次！无需重启服务。
```

```
exportfs -rv
```

```
#查看NFS的运行状态
```

```
nfsstat
```

```
#查看rpc执行信息，可以用于检测rpc运行情况
```

```
rpcinfo
```

```
#查看网络端口，NFS默认是使用111端口。
```

```
netstat -tu -4
```

4.4 客户端

4.4.1 查看 NFS 服务器上的共享目录

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">showmount -e 192.168.3.39
</span> </span> <span class="highlight-line"> <span class="highlight-cl">#结果
</span> </span> <span class="highlight-line"> <span class="highlight-cl">#Export list for 19
.168.3.39:
</span> </span> <span class="highlight-line"> <span class="highlight-cl">#/data/share 192.
68.3.0/24
</span> </span> </code> </pre>
<h4 id="4-4-2-挂载共享目录">4.4.2 挂载共享目录</h4>
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">#执行挂载
</span> </span> <span class="highlight-line"> <span class="highlight-cl">mount -t nfs 192.
68.3.39:/data/share /data/mnt
</span> </span> </code> </pre>
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">#查看挂载信息
</span> </span> <span class="highlight-line"> <span class="highlight-cl">df -h | grep 192.1
8.3.39
</span> </span> </code> </pre>
<p><strong>注意:</strong> 挂载点 <code>/data/mnt</code> 目录必须已经存在, 而且目
中没有文件或子目录(如有文件则自动删除覆盖)。最后, 我们在 NFS 服务端 <code>/data/share</c
de> 目录下创建一个文件, 看下客户端是否能够正确读取并修改。</p>
<p>NFS 默认使用用 UDP 协议来进行挂载, 为了提高 NFS 的稳定性, 可以使用 TCP 协议挂载, 客
端挂载命令可使用如下命令: </p>
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">mount 192.168.3.39:/data/share /data/mnt -o proto=tcp -o nolock
</span> </span> </code> </pre>
<h4 id="4-4-3-客户端卸载-NFS-挂载">4.4.3 客户端卸载 NFS 挂载</h4>
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">umount /data/mnt
</span> </span> </code> </pre>
```