企业级NFS网络文件共享服务详解

目录

[1.1 NFS介绍 3](#_Toc3210)

[1.1.1 什么是NFS 3](#_Toc203)

[1.1.3 NFS在企业中的应用场景 4](#_Toc4037)

[1.1.4 NFS在企业生产集群架构中的位置 5](#_Toc28756)

[1.1.5 企业生产集群为什么需要共享存储角色 5](#_Toc19478)

[1.2 NFS系统原理介绍 7](#_Toc31341)

[1.2.1 NFS系统挂载结构图解与介绍 7](#_Toc27985)

[1.2.2 什么是RPC(Remote Procedure Call) 9](#_Toc16433)

[1.2.3 NFS的工作流程原理 10](#_Toc13211)

[1.3 NFS服务端配置 11](#_Toc2345)

[1.3.1 NFS服务端部署环境准备 11](#_Toc23887)

[1.3.1.1 NFS服务端部署规划表 11](#_Toc30178)

[1.3.2 NFS服务端配置步骤 12](#_Toc2393)

[1.3.2.1 NFS服务端软件包的安装 12](#_Toc6101)

[1.3.2.2 启动NFS相关服务 14](#_Toc4564)

[1.4 NFS服务常见进程详解 14](#_Toc1521)

[1.5 实战配置NFS服务端 16](#_Toc19450)

[1.5.1 NFS服务端配置文件路径 16](#_Toc30893)

[1.5.2 exports配置文件格式 18](#_Toc1115)

[1.6 企业生产场景NFS exports配置实例 19](#_Toc16751)

[1.6.1 企业生产环境常见NFS配置实例 19](#_Toc27020)

[1.6.2 NFS配置参数权限 20](#_Toc5139)

[1.6.2.1 NFS配置文件权限参数说明 20](#_Toc10520)

[1.6.3 NFS客户端挂载排错思路 22](#_Toc29875)

[1.6.4 NFS 服务重点知识点梳理总结 22](#_Toc22675)

[1.6.4.1 使得NFS client端可写的服务端配置条件 22](#_Toc25803)

[1.6.5 NFS服务重要文件说明 23](#_Toc13513)

[1.6.6 NFS客户端mount挂载命令 24](#_Toc21074)

[1.6.7 NFS客户端挂载排除思路 24](#_Toc20189)

[1.6.8 mount 挂载及fstab文件的参数表格 25](#_Toc8381)

[1.6.9 NFS客户端mount挂载优化 28](#_Toc16329)

[1.6.10 有关系统安全挂载参数选项 28](#_Toc8974)

[1.6.10.1 mount 挂载性能优化参数选项 29](#_Toc31924)

[1.6.11 NFS服务优化 29](#_Toc22358)

[1.6.11.1 NFS服务内核优化相关建议 29](#_Toc31458)

[1.6.11.2 企业生产场景NFS共享存储优化 30](#_Toc2136)

[1.7 NFS 系统应用优缺点说明 31](#_Toc20267)

[1.8 showmount 命令说明 32](#_Toc9827)

[1.9 exportfs命令介绍 33](#_Toc4626)

[1.9.1 exportfs 命令参数说明和实例 33](#_Toc23980)

[1.9.2 NFS server端的防火墙控制 33](#_Toc26335)

[1.9.3 NFS常见故障排除 34](#_Toc32627)

[1.9.4 NFS客户端开机自启动挂载 35](#_Toc28826)

[1.9.5 NFS参考文献 35](#_Toc9775)

[1.9.6 本章重点回顾 35](#_Toc32071)

1.1 NFS介绍

## 1.1.1 什么是NFS

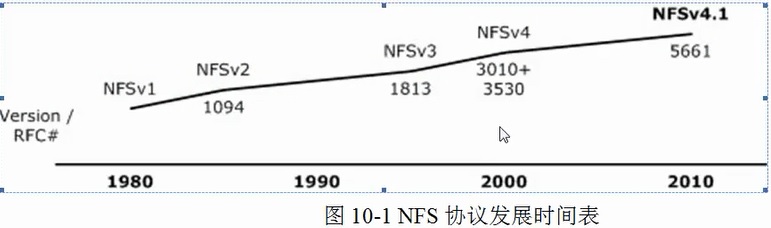
NFS是NetworkFile System的缩写，中文意思是网络文件系统.它的主要功能是通过网络(一般是局域网)让不同主机系统之间可以共享文件或目录.NFS客户端(一般为应用服务器,例如web)可以通过挂载(mount)的方式将NFS服务器端共享的数据目录挂载到NFS客户端本地系统中(就是某一个挂载点下).从NFS客户端的机器本地看,NFS服务器端共享的目录就好像是客户端自己的磁盘分区或目录一样,而实际上确是远端的NFS服务器的目录.

NFS网络文件系统很像windows系统的网络共享，安全功能，网络驱动器映射，这也和linux系统里的samba服务类似.只不过一般情况，windows网络共享服务或者samba服务用于办公局域网共享，互联网中小型网站集群架构后端常用NFS作为数据共享，如果是大型网站，那么有可能还会用到更复杂的分布式文件系统，例如:Moosefs(mfs),glusterfs,FastDFS,这些不在本书内容之列，有兴趣的读者可以了解其他书籍或者相关教学视频讲解。

1.1.2 NFS的历史介绍

第一个网络文件系统被称为File Access Listener,由Digital Equipment Corporation(DEC)

在1976年开发。NFS是第一个构建与IP协议之上的现代网络系统。在20世纪80代，它首先作为实验的文件系统,由Sun Microsystems在内部完成开发。NFS协议归Request for Comments(RFC)标准，并演化为NFSV2.作为一个标准，由于NFS与其他客户端和服务器的互操作能力很好而发展快速。标准持续地演化为NFSV3，在RFC1813中有定义。这一新的协议比以前的版本具有更好的可扩展性，支持大文件(超过2GB)，异步写入，以及将TCP作为传输协议，为文件系统在更广泛的网络中使用铺平了道路，在2000年，RFC3010(由RFC3530修订)NFS带入企业级应用。Sun引入了具有较高安全性，带有状态协议的NFSV4(NFS之前的版本都是无状态的)，今天，NFS是版本4.1(由RFC5661定义)，他增加了对跨越分布服务器的并行访问的支持(称为PNFS extension).NFS系统发展的时间表，包括记录其特性的特定RFC，都在下中展示。

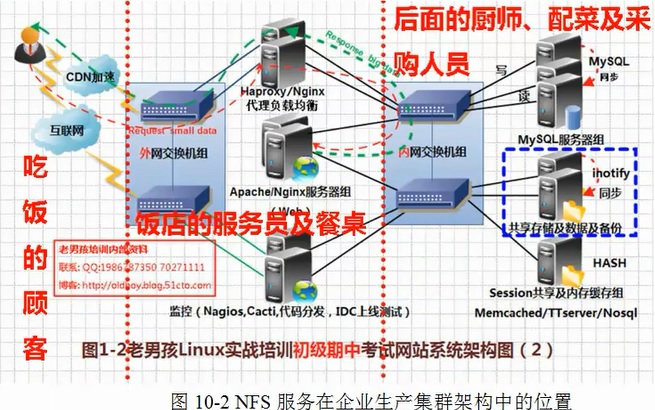


NFS系统已经历了近30年的发展，它代表了一个非常稳定的(及可移植)网络文件系统，它具备可扩展，高性能等特性并达到企业级应用质量标准。由于网络速度的增加和延迟的降低，NFS系统一直是通过网络提供文件系统服务的有竞争力的选择，特别是中小型互联网企业，应用十分广泛

## 1.1.3 NFS在企业中的应用场景

在企业集群架构的工作场景，NFS网络文件系统一般被用来储存共享视频，图片，附件等静态资源文件，一般是把网站用户上传的文件都放在NFS共享里，例如：BBS产生的图片，附件，头像，注意网站BBS程序不要放NFS共享里，然后前端所有的节点访问这些静态资源时都会读取NFS存储上的资源，NFS是当前互联网系统架构中最常用的数据存储服务之一，特别是中小型网站公司应用频率更高，大公司或门户除了使用NFS外，可能会使用更为复杂的分布式文件系统Moosefs(mfs),glusterfs,FastDFS等，这些不在本书内容之列，有兴趣的读者可以其他书籍或其他课程视频讲解。

## 1.1.4 NFS在企业生产集群架构中的位置

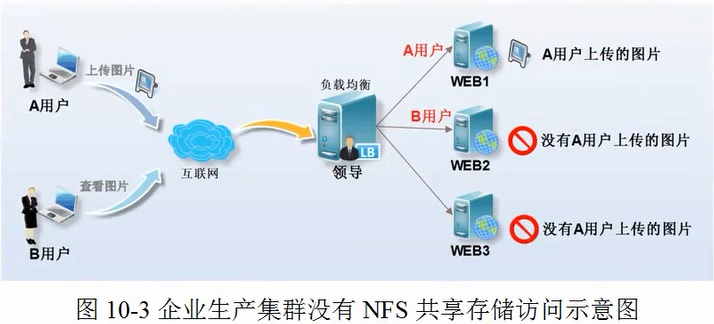


提示：图10-2中后面的虚线框里就是NFS系统工作的位置，NFS作为所有前端WEB服务的共享存储，存储的内容一般有网站用户上传的图片，附件，头像等，注意，网站的程序代码不要放NFS共享里，网站程序是人工发布的，不存在延迟问题，直接批量发布到WEB节点提供访问，这样访问效率更高。

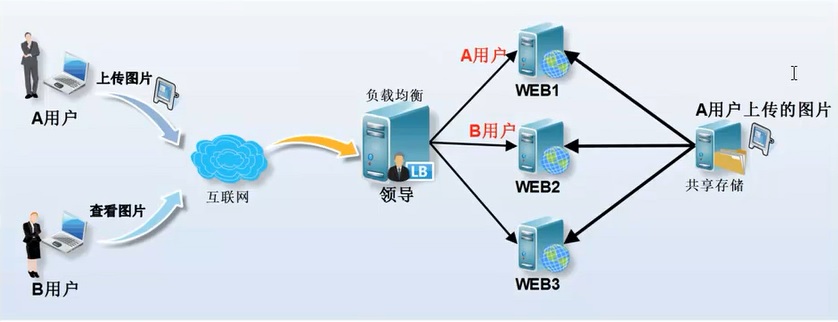
1.1.5 企业生产集群为什么需要共享存储角色

这里我们通过图解给大家展示集群架构需要共享存储服务的理由。例如：A用户传图片到WEB1服务器，然后让B用户访问这张图片，结果B用户访问的请求分发到了WEB2，因为WEB2上没有这张图片，结果无法看不到A图片传的图片，如果此时又一个共享存储，A用户上传图片无论到WEB1还是WEB2上，最终都存储到共享存储上，此时，B用户访问图片时，无论分发到WEB1还是WEB2上，最终也都会去共享存储访问，这样就可以访问到资源。这个共享存储的位置可以通过开源软件和商业硬件实现，互联网中小型集群架构会用普通PC服务器和NFS文件系统实现

当集群中没有NFS共享存储，用户访问图片的情况：



如果集群中有NFS共享存储，用户访问图片的情况：

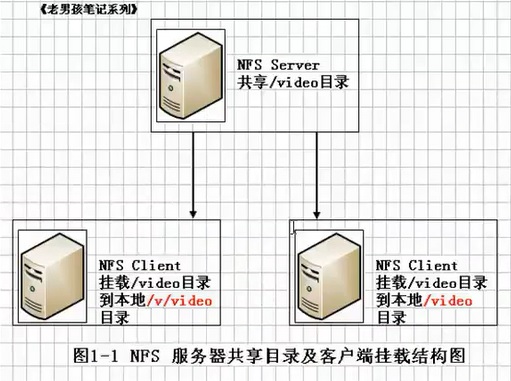


中小型互联网企业一般不会买硬件存储，因为太贵，大公司如果的话，可能会临时买硬件存储顶一下网站的压力，当网站并发继续加大后，硬件存储扩展且价格成几何级数增加。例如：淘宝网就替换掉了很多硬件设备集群软件，用LVS+HAPROXY替换了netscaler负载均衡设备，用FastDFS,TFS配合PC服务器替换了netapp,emc商业存储设备，获取IOE正在成为互联网公司的主流。

# 1.2 NFS系统原理介绍

## 1.2.1 NFS系统挂载结构图解与介绍

下面是企业工作中的NFS服务器与客户端挂载情况结构图解



如图10-5,当我么在NFS服务器端设置好一个共享目录/video后，其它的有 NFS服务器端的客户端可以将这个共享目录/video,挂载到客户端本地的某个挂载点就是一个目录，这个挂载点目录可以自己随意指定),上图中的两个NFS客户端挂载点分布为/v/video和/video,不同客户端的挂载点可以不相同。

当客户端正确挂载完毕后，进入到NFS客户端的挂载点所在的/v/video或/video就可以看到NFS服务器端/video共享出来的目录下的所有数据，在客户端上查看NFS服务器端的/video

目录就相当于客户端本地的磁盘分区或目录，几乎感 用上区别，根据NFS服务端授予NFS共

享以及共享目录的本地系统权限指定的NFS客户端操作挂载/v/video或/video的目录，就可以将数据轻松的存取到服务器端的/video目录中了

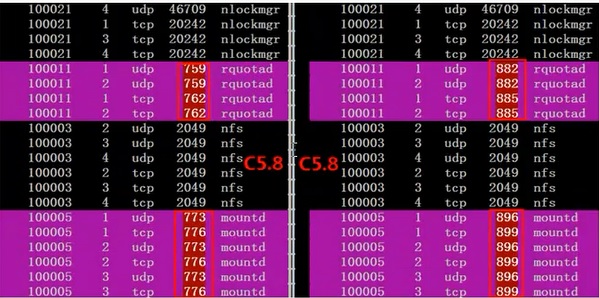
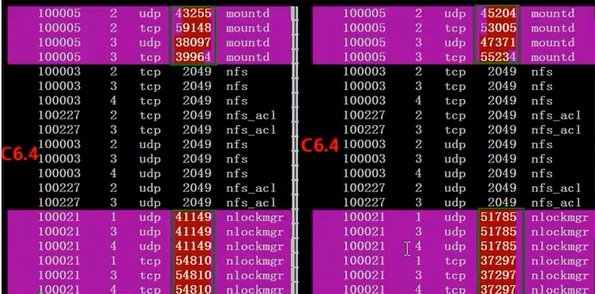
客户端挂载NFS后，本地挂载基本信息显示如下所示：

dh -h

mount 10.0.0.7:/video /video

如上所示：从挂载信息看起来，和本地的磁盘分区几乎没什么差别,

经过前面的介绍，我们知道NFS系统是通过网络来进行数据传输的(文件系统嘛！)，因此，NFS会使用一些端口来传输数据，那么，NFS到底使用什么端口进行数据传输的呢，下面是NFS服务两次重启向RPC服务注册的端口列表结果对比



有上面实际测试得知，NFS在传输数据时使用的端口会随机选择。可能有读者不太明白，既然这样，NFS客户端是怎么知道NFS服务端使用的是那个端口呢？

答案：就是通过RPC(中文意思远程过程调用，英文Remote Procedure Call)简称RPC协议/

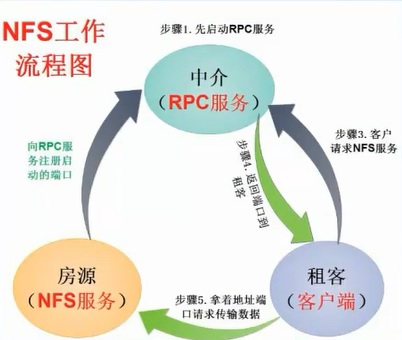
服务来实现的，这个RPC服务的应用在门户级别的网站很多的，例如：百度，搜狐，新浪，我们就来谈谈什么是RPC协议/服务

## 1.2.2 什么是RPC(Remote Procedure Call)

因为NFS支持的功能相当多，而不同的功能都会使用不同的程序来启动，每个不同的程序功能都会启用一些端口来传输数据，因此，NFS的功能所对应的端口无法固定，因此需要随机启用一些未被使用的端口来作为传输之用，其中Centos5.x随机端口为小于1024，而Centos 6.x的随机端口都是很大的，见上图。

因为端口不固定，这样一来就会造成NFS客户端与NFS服务端的通讯障碍，客户端必须要知道NFS服务器端的数据传输端口才能进行通信交互数据。

要解决上面的通讯问题困扰，就需要远程过程嗲用RPC服务来帮忙了，NFS服务的最主要的功能就是记录没个NFS功能所对应的端口号，并且在NFS客户端访问该端口和功能对应的信息传递给请求数据的NFS客户端，从而可以确保客户端可绑定到正确的NFS端口上，达到实现数据传输交互数据目的，这个RPC服务很类似于，NFS服务器端和NFS客户端之间的一个中介。



我们可以拿房屋中介打个例子：我们找房子就相当于NFS客户端，中介介绍房子就相当于RPC，房东就相当于NFS服务端，租房的人找房子，就要先找中介，中介要预先存有房东提供的房子信息，并告诉租房的人

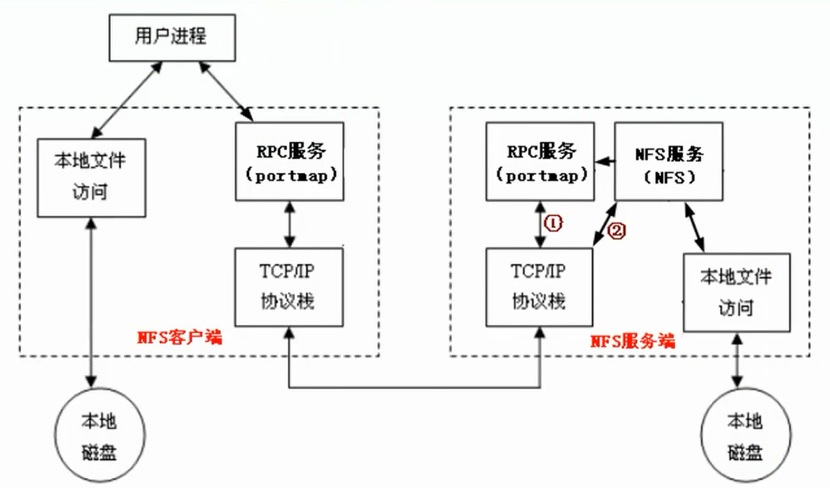
那么RPC服务有是如何知道每个NFS的端口呢？

这是因为，当NFS服务端启动服务时会随机取若干端口，并主动向RPC服务注册，并选取用的相关端口及功能信息，因此RPC可以知道每个端口对应的NFS功能，然后RPC服务固定使用111端口来监听NFS客户端提交的请求，并将正确的端口信息回复给请求的NFS客户端，这样一来，NFS客户端就可以与NFS服务端进行数据传输了。

提示：在启动NFS SERVER之前，首先要启动RPC服务（即portmap服务，下同）否则NFS SERVER就无法向RPC服务区注册，另外，如果RPC服务重新启动，原来已经注册好的NFS端口数据就会全部丢失。因此此时RPC服务管理的NFS程序也要重新启动以重新向RPC注册。特别注意：一般修改NFS配置文档后，是不需要重启NFS的，直接在命令执行/etc/init.d/nfs reload或exportfs –rv即可使修改的/etc/exports生效。

## 1.2.3 NFS的工作流程原理

前文描述的整个启动过程如下图所示

NFS工作流程描述：

* 首先服务器端启动RPC服务，并开启111端口
* 启动NFS服务，并向RPC注册端口信息
* 客户端启动RPC（portmap服务），向服务端的RPC(portmap)服务请求服务端的NFS端口
* 服务端的RPC(portmap)服务反馈NFS端口信息给客户端。
* 客户端通过获取的NFS端口来建立和服务端的NFS连接并进行数据的传输。

提示：NFS的RPC服务，在Centos5.X下名称为portmap,在Centos6.X下为名称为rpcbind

# 1.3 NFS服务端配置

## 1.3.1 NFS服务端部署环境准备

### 1.3.1.1 NFS服务端部署规划表

|  |  |  |  |
| --- | --- | --- | --- |
| 服务器名称 | 角色 | 外网IP | 内网IP |
| NFS-SERVER | NFS服务器端(nfs-server) | 192.168.100.101 | 10.0.0.101 |
| WEB-LAMP01 | NFS客户端1(nfs-client1) | 192.168.100.102 | 10.0.0.102 |
| WEB-LNMP02 | NFS客户端2(nfs-client2) | 192.168.100.103 | 10.0.0.103 |

## 1.3.2 NFS服务端配置步骤

|  |
| --- |
| NFS-SERVER上的配置：   * NFS服务端配置主机名   hostname NFS-SERVER  sed -i ‘s#HOSTNAME=.\*#HOSTNAME=nfs-server#g’ /etc/sysconfig/network   * NFS服务端操作系统及内核版本信息   cat /etc/redhat-release  uname -r  uname -m   * 修改WEB-LAMP01的主机名   hostname WEB-LAMP01  sed -i ‘s#HOSTNAME=.\*#HOSTNAME=nfs-server#g’ /etc/sysconfig/network   * 显示操作系统及内核版本信息   uname -r  uname -m   * 修改WEB-LNMP01的主机名   hostname WEB-LNMP01  sed -i ‘s#HOSTNAME=.\*#HOSTNAME=nfs-server#g’ /etc/sysconfig/network   * 显示操作系统及内核版本信息   uname -r  uname -m |

1.3.2.1 NFS服务端软件包的安装

要部署NFS服务,需要安装下面的软件包：

1)nfs-utils:这个NFS服务主程序

包括rpc.nfsd,rpc.mountd两个daemons和相关文档说明及执行命令文件等。

2)rpcbind:Centos6.X下面RPC的主程序(Centos5.X下面为portmap)

NFS可以被视为一个RPC程序，在启动任何一个RPC程序之前，需要做好端口和rpc程序的对应映射工作，这个映射工作就是由rpcbind服务来完成的，因此，在提供NFS服务之前

必须先启动rpcbind服务才行。

注明：有关RPC协议知识，这里大家不必细纠，详细说明可见本章结尾命令部分

需要在各个主机上安装nfs和rpcbind

查看默认情况Centos5.8/6.4下NFS软件的安装情况

|  |
| --- |
| rpm -aq nfs-utils rpcbind |

提示：

知识拓展：当不知道软件名字时候，大家也可以用rpm -aq|grep -E "nfs-|rpcbind"来

过滤包含引号字符串的方式来查找，grep -E在这里相当于egrep.grep,egrep这两个命令

在运维工作中非常常用并且也好用，需要自己掌握

Centos 6.6默认没有安装NFS软件包(Centos5默认是会安装),此时我们可以使用yum install nfs-utils rpcbind -y命令来安装NFS软件

|  |
| --- |
| yum install nfs-utils rpcbind -y  rpm -aq nfs-utils rpcbind |

出现上述两个软件包，表示NFS服务端软件安装完毕

NFS软件的3种安装方法

检查:rpm -aq nfs-utils rpcbind

1)方法1：

|  |
| --- |
| yum install nfs-utils rpcbind -y |

1. 方法2：通过系统光盘里的rpm包安装，命令如：

|  |
| --- |
| rpm -ivh nfs-utils-1.2.3-36.el6.x86\_64.rpm |

3)方法3:

|  |
| --- |
| LANG=en  yum grouplist|grep -i nfs  yum groupinstall "NFS file server" -y |

1.3.2.2 启动NFS相关服务

|  |
| --- |
| 1):启动rpcbind服务  因为NFS及其辅助程序都是基于RPC协议的，所以首先要却表系统中运行了rpcbind服务。  有关RPC协议介绍请见后文。实际操作如下：  LANG=en #临时调整系统为英文字符集，便于grep过滤  /etc/init.d/rpcbind status #检查rpcbind服务状态  rpcinfo -p localhost #rpcbind服务未启动检查rpcinfo信息的报错  /etc/init.d/rpcbind start #启动rpcbind服务  lsof -i :111或者netstat lntup|grep rpcbind  chkconfig --list rpcbind  /etc/init.d/nfs status  /etc/init.d/nfs start  netstat -lntup|grep 2049  chkconfig --list nfs  chkconfig nfs on  chkconfig rpcbind on  less /etc/init.d/rpcbind |

提示：在生产环境中,一般会把/etc/init.d/nfs或

开机自启动程序/etc/rc.local中,而不会使用chkconfig rpcbind on或chkconfig nfs on这样的命令

例如：

|  |
| --- |
| echo "/etc/init.d/rpcbind start" >>/etc/rc.local  echo "/etc/init.d/nfs start" >>/etc/rc.local |

# 1.4 NFS服务常见进程详解

说明：从上面NFS服务启动过程的提示，可以看出运行NFS服务默认需要启动的服务和进程至少有：NFS quotas(rpc.rquotad),NFS daemon(nfsd),NFS mountd(rpc.mount)可以通过执行如下命令查看启动NFS后，系统运行的NFS相关进程：

|  |
| --- |
| # ps -ef | egrep "rpc|nfs"  rpcuser 1016 1 0 16:27 ? 00:00:00 rpc.start #检查文件一致性  root 1552 1 0 16:31 ? 00:00:00 rpc.rquotad #磁盘配额进程(remote quota server)  root 1557 1 0 16:31 ? 00:00:00 rpc.mountd #权限管理验证等(NFS mount daemon)  root 1566 2 0 16:31 ? 00:00:00 [nfsd] #NFS主进程  root 1567 2 0 16:31 ? 00:00:00 [nfsd] #NFS主进程  root 1569 2 0 16:31 ? 00:00:00 [nfsd] #NFS主进程，管理登入，ID身份判定  root 1573 2 0 16:31 ? 00:00:00 [nfsd] #NFS主进程  root 1600 1 0 16:31 ? 00:00:00 rpc.idmapd #name mapping daemon  root 1638 1119 0 16:34 pts/0 00:00:00 egrep rpc|nfs |

NFS 服务的任务是共享文件系统数据，而文件系统数据的共享离不开权限问题。所有NFS服务器启动时最少需要两个不同的进程，一个进程是管理NFS客户端是否登入的即rpc.nfsd主进程，另一个进程是管理NFS客户端是否能够取得对应的权限，rpc.mountd进程。如果还需要管理磁盘配额，则NFS还要再加载rpc.rquotad程序.

表 10-1 NFS服务启动进程说明：

|  |  |
| --- | --- |
| 服务或进程名 | 用途说明 |
| nfsd(rpc.nfsd) | rpc.nfsd的主要功能是管理NFS客户端是否能够登入NFS服务端主机，其中还包括登入者的ID判别等 |
| mountd(rpc.mountd) | rpc.mountd的主要功能则是管理NFS文件系统，当NFS客户端顺利通过rpcnfsd登入NFS服务端主机时，在使用NFS服务器提供数据之前，他会读NFS的配置文件/etc/exports来比对NFS客户端的权限，通过这一关之后,还会经过NFS服务端本地文件系统使用权限，通过这一关之后，还会经过NFS服务器本能地文件系统使用权限(就是owner,group,other权限)的认证程序，如果都通过了，NFS客户端就可以取得使用NFS服务器端文件的权限，注意，这个/etc/exports文件也是我们用来管理NFS共享目录的使用权限与安全设置的地方，特别强调，NFS本身设置的是网络共享权限，整个共享目录的权限还是和目录自身的系统权限有关 |
| rpc.locked(非必要) | 可用来锁定文件，用于多客户端同时写入 |
| rcp.statd(非必要) | 检查文件的一致性，与rpc.lockd有关，c,d两个服务需要客户端，服务器端同时开启才可以:rpc.statd:监听来自其他助剂重启的通知,并且管理当本地系统重启时主机列表 |
| rpc.idmapd | 名字映射后台进程 |

# 1.5 实战配置NFS服务端

1.5.1 NFS服务端配置文件路径

NFS服务的默认配置文件路径为：/etc/exports,并且默认是空的。

|  |
| --- |
| # ls -l /etc/exports  # cat /etc/exports |

提示：NFS默认配置文件/etc/exports是存在的，但默认是没有内容的，需要自行配制，有些linux版本也可能不提供/etc/exports配置文件，这时就需要手动创建

|  |
| --- |
| #实例一：共享/data目录给10.0.0.0/24整个网段的主机读写，即实现将nfs server上的/data目录共享给10.0.0.0/24整个网段的主机可读写  # vim /etc/exports  /data 10.0.0.\*(rw,sync)  /data 10.0.0.0/24(rw,sync,all\_squash)  保存退出  # exportfs -rv  # /etc/init.d/nfs reload  # mkdir /data  # exportfs -rv #检查exports文件的正确性,并平滑重启  # /etc/init.d/nfs reload  # grep "reload" -A 1 /etc/init.d/nfs  # showmount -e 127.0.0.1  2) 在客户端上进行配置和测试：  # mount -t nfs 10.0.0.101:/data /mnt  # df -h  # rpm -aq nfs-utils rpcbind  # yum install nfs-utils rpcbind -y  # /etc/init.d/rpcbind start  # LANG=en  # chkconfig rpcbind on  # echo "/etc/init.d/rpcbind start" >>/etc/rc.local  # /etc/init.d/rpcbind status  # showmount -e 10.0.0.101  # echo "mount -t nfs 10.0.0.101:/data /mnt" >>/etc/rc.local  # cat /proc/mountscat /var/lib/nfs/etab #显示NFS服务器端的所有参数 |

## 1.5.2 exports配置文件格式

NFS共享的目录 NFS客户端地址1(参数1，参数2...) 客户端地址2(参数1，参数2...)

NFS共享的目录 NFS客户端地址(参数1,参数2...)

其中上述各个列的参数含义如下：

1) NFS共享的目录：为NFS服务端要共享的实际目录，要用绝对路径。如(/data)注意共享目录的本地权限，如果需要读写共享，一定要让本地目录被NFS的用户可以读写

1. NFS客户端地址：为NFS服务授权的可访问共享的NFS客户端地址为单独的IP地址或主机名、域名等，也可以为整个网段地址，还还可以用通配符匹配所有客户端服务器可以访问，这里所谓的客户端一般来说是前端的业务，例如：WEB服务。具体说明见表10-3

|  |  |  |
| --- | --- | --- |
| 客户端地址 | 具体地址 | 说明 |
| 授权单一客户端访问NFS | 10.0.0.30 | 一般情况，生产环境此配置不多见 |
| 授权整个网段可访问NFS | 10.0.0.0/24 | 其中的24等同于255.255.255.0，为生产环境中最常见 的配置。配置和维护方便 |
| 授权整个网段可访问NFS | 10.0.0.\* | 指定网段的另外写法(需要验证) |
| 授权某个域名客户端访问 | nfs.oldboy.me | 此方法生产环境中一般情况下不常用 |
| 授权整个域名客户端访问 | \*.oldboy.me | 此方法生产环境中一般情况下不常用 |

3) 权限参数集：对授权的NFS客户端的访问权限设置。参数具体说明见后

# 1.6 企业生产场景NFS exports配置实例

## 1.6.1 企业生产环境常见NFS配置实例

表10-4 /etc/exports文件格式配置实例说明

|  |  |
| --- | --- |
| 常用格式说明 | 要共享的目录，客户端IP地址或IP段(参1，参2) |
| 配置例一 | /data 10.0.0.0/24(rw,sync)  #允许客户端读写，并且数据同步写到服务端的磁盘上 |
| 配置例二 | /data/blog 10.0.0.0/24(rw,sync,all\_squash, anonuid=2000,anongid=2000)  #允许客户端读写，并且数据同步写到服务端的磁盘里，并且指定客户端的用户UID,这是生产环境的一种配置，适合多客户端共享一个NFS服务单目录，如果所有服务器的用户UID哦,都是65534，则本例没什麽需求了，早期Centos5.5 X86\_64及以下的系统的匿名用户是nfsnobody的uid不都是65534,此时如果这些服务器共享一个NFS目录，就会出现访问共享目录 |
| 配置例三 | /home/oldboy 10.0.0.0/24(ro)  #只读共享，用途：例如  生产环境开发有查看正式服务器日志的需求，单又不希望  给开发正式服务，那么就可以给开发提供从某个测试服务器NFS客户端上查看某个正是服务器的日志(共享)的能力  ，当然这不是唯一的方法，例如可以把程序记录的日志发送到测试服务也可 |
| 注："配置例一" 为例说明如下：   1. /data 为要共享的NFS服务器端的目录，注意：被共享的目录一定要用绝对路径 2. 10.0.0.0/24表示允许NFS客户端访问共享目录的网段范围。24表示255.255.255.0 3. (rw,sync)rw表示允许读写，sync表示数据同步写入到NFS服务器端的硬盘中   也可用通配符\*替换IP地址，表示允许所有主机，但不可以写成10.0.0.\*的形式 | |

## 1.6.2 NFS配置参数权限

## 1.6.2.1 NFS配置文件权限参数说明

NFS服务器端的权限设置，即/etc/exports文件配置格式中小括号()里的参数

见表10-5：NFS配置权限设置常用参数说明

|  |  |
| --- | --- |
| 参数名称 | 参数用途 |
| rw | Read-write,表示可读写权限 |
| ro | Read-only,表示只读权限 |
| sync | 请求或写入数据时，数据同步写入NFS Server的硬盘后才返回 |
| async | 请求或写入数据时，先返回请求，再将数据写入到内存缓存和硬盘，即异步写入数据。此参数可以提升NFS性能，但是会降低数据的安全，因此，一般情况下建议不用，如果NFS处于瓶颈状态，并且允许数据丢失的话可以打开此参数提升NFS性能  写入数据时会先写到内存缓存区，等硬盘有空挡再写入磁盘，这样可以提升写入效率，风险为若服务器宕机或不正常关机，会损失缓存区中未写入磁盘的数据(解决办法：服务器主板电池或加UPS不间断电源) |
| secure | 限制客户端只能从小于1024的tcp/ip端口连接nfs服务器(默认设置) |
| insecure | 允许客户端从大于1024的tcp/ip端口连接服务器 |
| wdelay | 检查是否有相关的写操作， 如果有则将这些写操作一起执行，这样可以提高效率(默认设置) |
| no\_wdelay | 若有写操作则立即执行，应与sync配合使用 |
| subtree | 若输出目录是一个子目录，则nfs服务器将检查其父目录的权限(默认设置) |
| no\_subtree | 即使输出目录是一个子目录， nfs服务器也不检查其父目录的权限，这样可以提高效率 |
| all\_squash： | 不管访问NFS server共享目录的身份如何，他的权限都  被压缩成匿名用户，同时它的UID和GID都会变成nfsnobody账号身份。在早期多个NFS客户端同时读写NFS server  数据时，这个参数很有用。配置NFS生产重要技巧：配置NFS生产重要技巧：  1)确保所有的客户端服务器对NFS共享目录具备相同的用户访问权限  a:all\_squash把所有客户端都压缩成固定的匿名用户(uid相同)。  b:就是anonuid,anonuid指定的UIG和GID的用户。  2:所有的客户端服务端都需要有一个相同的UID和GID的用户，即nfsnobody(UID必须相同)。 |
| no\_all\_squash： | 与all\_squash取反（默认设置） |
| root\_squash | 对于访问NFS server共享目录的用户如果是root的话，则它的权限将被压缩成匿名用户，同时它的UID和GID通常会变成nfsnobody账号身份。 |
| no\_root\_squash | 访问NFS server共享目录的用户如果是ROOT的话，他的共享目录就具有root的权限。这个配置原本是为无盘客户端准备的。 |
| anonuid=xxx | 参数以anon\*开头即指annonymous匿名用户，这个用户  UID设置值通常为nfsnobody的UID值，当然我们也可以自行设置这个UID值，但是,UID必须存在于/etc/passwd中，在多NFS Client时，如多台WEB SERVER共享一个NFS目录，通过这个参数可以使得不同的NFS client写入的数据对所有NFS Client保持同样的用户权限，即为配置的匿名UID对于用户权限，这个参数很有用并指定该用户为本地用户（UID=xxx） |
| anongid=xxx | 将远程访问的所有用户组都映射为匿名用户组账户，并指定该匿名用户组账户为本地用户组账户（GID=xxx） |

## 1.6.3 NFS客户端挂载排错思路

1. 首先确认NFS服务端配置和服务是OK

|  |
| --- |
| # showmount -e localhost |

2) 最好看看服务器端可不可以自己挂自己

|  |
| --- |
| # mount -t nfs 10.0.0.107:/data /mnt  # df -h |

1. 确认NFS客户端showmount是否OK

|  |
| --- |
| showmount -e 10.0.0.7 |

如果不OK的解决思路，一般要根据提示解决，总体通用思路为:

a:ping NFS服务端IP检查 #看物理链路是否通。

|  |
| --- |
| # ping 10.0.0.107 |

b:telnet服务端IP端口检查

|  |
| --- |
| telnet 10.0.0.107 111 |

1.6.4 NFS 服务重点知识点梳理总结

### 1.6.4.1 使得NFS client端可写的服务端配置条件

当多个NFS客户端以NFS方式写入修改服务器端的文件系统，需要具有以下个权限

a) NFS服务器/etc/exports设置需要开放可写入的权限，即服务端的共享权限

b) NFS服务器实际要共享的NFS目录权限具有可写入w的权限，即服务端本地目录的安全权限。

c)每台机器都对用户存在的nfs默认配置UID的相同UID 65534的nfsnobody用户(确保所有客户端的访问权限统一，否则每个机器需要同时建立相同用UID的用户，并覆盖NFS的默认配置)

当满足上述三个条件，多个NFS客户端才能具有互相写入，互相修改其

他主机写入文件的权限，这在开篇讲解过的大规模集群环境，作为集群共享

存储时尤为注意，

到此为止，NFS服务端一个NFS共享目录/data共享给10.0.0.0/24内主机可读写就配置完了，下面还需要在客户端主机挂载服务端共享的/data，才可以在客户端真正读写到该共享目录。

## 1.6.5 NFS服务重要文件说明

|  |  |
| --- | --- |
| NFS常用路径 | 说明 |
| /etc/exports | NFS服务主配置文件，配置NFS具体共享服务的地点，默认内容为空，以行为单位。例如：cat /etc/exports /data 10.0.0.0/24(rw,sync) |
| /usr/sbin/exportfs | NFS服务的管理命令，例如:可以加载NFS配置生效，  还可以直接配置NFS共享目录，即无需配置/etc/exports实现共享  exportfs -rv #加载配置生效，等价于优雅重启reload,exporting 10.0.0.0/24:/data  这里讲了一个服务平滑重启的概念，超市，银行到时间了如何提供服务，拒绝新来的顾客，对已有的顾客继续提供服务，网站平滑重启，提升了用户体验是必须要考虑的。  exportfs不但可以加载配置生效，也可以通过命令直接共享目录，越过/etc/exports，但是重启失败 |
| /usr/sbin/showmount | 常用来再客户端，查看NFS配置及挂载结果的命令。配置nfsserver,分别在服务端以及客户端查看挂载情况 |
| /var/lib/nfs/etab | NFS配置文件的完整参数设定的文件(有很多没有配置凡是默认就有的NFS参数) /var/lib/nfs/etab master table of exports |
| /var/lib/nfs/xtab | 适合C5.X记录曾经挂载过的NFS客户端的信息，C6. X没有此文件了 |
| /proc/mounts | 客户端挂载参数，例如：grep mnt /proc/mounts  10.0.0.7/data/ /mnt nfs4 rw,relatime,vers=4,rsize=131072,wsize=131072,namlen=255 hard,proto=tcp,port=0... |
| /var/lib/nfs/rmtab | table of client accessing server's exports |

1.6.6 NFS客户端mount挂载命令

1. 这里先强调下客户端挂载的命令格式：

|  |  |  |  |
| --- | --- | --- | --- |
| 挂载命令 | 挂载的格式类型 | NFS服务器提供的共享目录 | NFS客户端要挂载的目录 |
| mount | -t nfs | 10.0.0.7:/data | /mnt(必须存在) |
| 完整挂载命令为:mount -t nfs 10.0.0.107:/data /mnt,此命令要在客户端执行 | | | |

1. 执行挂载的过程

showmount -e 10.0.0.7 #挂载前首先检查有权限需要挂载的信息，是否能够挂载,df -h #查看挂载后的结果

## 1.6.7 NFS客户端挂载排除思路

1. 首先确认NFS服务端配置和服务是OK的

|  |
| --- |
| # showmount -e localhost |

# 最好服务端自己挂自己看看可不可以

|  |
| --- |
| # mount -t nfs 10.0.0.107:/data /mnt  # df -h |

2)确认NFS客户端showmount是否OK

# 正常情况下：

|  |
| --- |
| # showmount -e 10.0.0.7  Export list for 10.0.0.7  /data 10.0.0.0/24 |

# 2个报错

|  |
| --- |
| # showmount -e 10.0.0.7  clnt\_create:RPC:Port mapper failure-Unable to receive: error 113 (  No route to host)  # showmount -e 10.0.0.7  clnt\_create:RPC Program not registered |

如果不OK的解决思路，一般要根据提示解决，总体通用思路为：

a:ping NFS服务端IP检查

|  |
| --- |
| # ping 10.0.0.7 |

b:telnet服务端IP端口检查

|  |
| --- |
| # telnet 10.0.0.7 111 |

NFS客户端mount挂载深入

NFS客户端mount挂载参数说明

在NFS客户端可以通过/cat/var/lib/nfs/etab 查看服务端配置的参数

节

在NFS客户端可以通过/cat/proc/mounts 查看mount的挂载参数细节。

## 1.6.8 mount 挂载及fstab文件的参数表格

1)通过NFS客户端测试挂载获取的默认挂载参数：

|  |
| --- |
| # grep mnt /proc/mounts |

2) NFS client mount挂载参数表

|  |  |  |
| --- | --- | --- |
| 参数 | 参数功能 | 默认参数 |
| fg(前台)  bg(后台) | 当在客户端执行挂载时，可选择是前台还是后台执行，若在前提执行，则mount会持续尝试挂载，直到成功或挂载时间超时为止，若在后台执行，则mount会在后台持续多次进行mount，而不会影响到前台的其他的程序操作。如果网络联机不稳定，或是服务器常常需要关机，建议是用bg比较稳妥 | fg |
| soft  hard | 当NFS client以soft挂载Server后，若网络或server出现问题，造成client和server无法传输资料时，client会一直尝试到timeout后显示错误并且停止尝试，若使用soft mount的话可能会在timout出现时造成资料丢失，故一般不建议使用，若用hard模式挂载硬盘 时，刚好和soft相反，此时client会一直尝试连线到server,若server有回应就继续刚才的操作，若没有回应NFS client会一直尝试，此 时无法umount或kill，所以常常会配合intr使用，这是默认值 | hard |
| intr | 当使用hard挂载的资源timeout后，若有指定无 intr参数，可以在timeout后把它中断掉，这可避免出问题时系统整个被NFS锁死，建议使用intr | 无 |
| rsize  wsize | 读出(rsize)与写入(wsize)的区块大小(block size),这个设置值可以影响客户端传输数据的缓冲存储量，一般来说，如果在局域网内(LAN),并且客户端与服务器端都具有足够的内存，这个值可以设置大一点，比如说65535(bytes)，提升缓存区块将可提升NFS文件系统的传输能力。但设置的值也不要太大，最好能实现网络能够传输的最大值为限 | C5.X默认值  rsize=1024  wsize=1024  C6.X默认值  rsize=131072  wsize=131072 |
| proto=udp | 使用UDP协定来传输资料，在LAN中会有比较好的性能。若要跨越Internet的话，使用proto=tcp多传输的数据会有比较好的纠错能力 | proto=tcp |
| 一个标准mount命令的配置方式：  #mount -t nfs -o bg,hard,intr,rsize=131072,wsize=131072 10.0.0.7:/data/ /mnt | | |

3) mount -o参数对应的选项列表

|  |  |  |
| --- | --- | --- |
| 参数 | 参数意义 | 系统默认值 |
| suid  nosuid | 当挂载的文件系统上有任何SUID的程序时，只  要使用nosuid就能够取消设置SUID的功能. (SUID,在linux基础学习篇权限管理已经讲过) | suid |
| rw  ro | 可以指定文件系统是只读(ro)或可写(rw) | rw |
| dev  nodev | 是否可以保留装置文件的特殊功能？一般来说只有/dev才会有特殊的装置，因此可以选择nodev | dev |
| exec  noexec | 是否具有执行文件的权限？如果想要挂载的仅 是普通资源数据区(例如：图片，附件)，那么可以选择noexec | exec |
| user nouser | 是否允许用户进行文件的挂载与卸载功能？如 果要保护文件系统，最好不要提供用户进行挂载于卸载 | nouser |
| auto noauto | 这个auto指的是"mount -a"时会不会被挂载的 项目，如果不需要这个分区随时被挂载，可以  设noauto | auto |

1. fstab修改错误导致系统无法启动故障修复案例

维护模式或救援模式：

|  |
| --- |
| # mount -o rw,remount |

文件系统只读故障修复案例

1:RSYNC bug.

2:文件系统内部自动一致性(只读)

|  |
| --- |
| # mount -o rw,remount |

## 1.6.9 NFS客户端mount挂载优化

某网友问：在企业生产环境中，NFS客户端挂载有没有必要加参数，比如加noexec nosuid,nodev,bg,soft,rsize,wsize等参数，我看书上说建议挂，rsize,wsize这两个是缓存参数，是否也建议加呢？你在生产环境中是怎么做的？

解答：这个问题属于mount挂载优化内容(有些参数也适合其他文件系统)，一般来说要适当加载参数，但是，最好是做好测试，用数据来说话，到底是挂载还是不挂载。

## 1.6.10 有关系统安全挂载参数选项

在企业工作场景，一般来说，NFS服务器共享的只是普通静态数据(图片，附件，视频),不需要执行suid,exec等权限，挂载的这个文件系统只能作为数据存取只用，无法执行程序，对于客户端来讲增加了安全性，例如：很多木马篡改站点文件都是由上传入口上传的程序到存储目录，然后执行的。因此在挂载的时候，用下面的命令很有必要

安全挂载参数：

|  |
| --- |
| # mount -t nfs -o nosuid,noexec,nodev,rw 10.0.0.107:/data /mnt |

通过mount -o指定挂载参数和在/etc/fstab里指定挂载参数效果是一样的。网络文件系统和本地的文件系统效果也是一样的。

### 1.6.10.1 mount 挂载性能优化参数选项

企业生产环境nfs性能优化挂载的例子：

(1)禁止更新目录及文件时间戳挂载

mount -t nfs -o noatime,nodiratime 10.0.0.107:/data

(2)安全加优化的挂载方式

|  |
| --- |
| # mount -t nfs -o nosuid,noexec,nodev,noatime,nodiratime,intr,rsize=65536  wsize=65536 10.0.0.107:/data |

在centos 6.6 x86\_64服务器端和客户端环境下，可使用如下命令参数

|  |
| --- |
| # mount -t nfs -o nosuid,noexec,nodev,noatime,nodiratime,rsize=131072  wsize=131072 10.0.0.107:/data |

1. 默认的挂载方式

|  |
| --- |
| mount -t nfs 10.0.0.107:/data /mnt |

经过实际测试,centos6.6 x86\_64默认的挂载参数性能还是不错的

注意：非性能的参数越多，速度就可能越慢

如果是本地文件系统

|  |
| --- |
| # mount /dev/sdb1 /mnt -o defaults,async,noatime,data=writeback barrier=0  # df -h  # grep mnt /proc/mounts |

## 1.6.11 NFS服务优化

1.6.11.1 NFS服务内核优化相关建议

优化选项说明：

1)/proc/sys/net/core/rmem\_default

#该文件指定了接受套接字缓存区大小的缺省值(以字节为单位),缺省设置:124928

2)/proc/sys/net/core/rmem\_max

#该文件指定了接受套接字缓存区大小的最大值(以字节为单位),缺省设置：124928

3)/proc/sys/net/core/wmem\_default

#该文件指定了发送套接字缓冲区大小的缺省值(以字节为单位),缺省设置：124928

4)/proc/sys/net/core/wmem\_max

#该文件指定了发送套接字缓存区大小的最大值(以字节为单位),缺省设置：124928

上述文件对应的具体内核优化命令：

|  |
| --- |
| # cat >>/etc/sysctl.conf<< EOF  net.core.wmem\_default = 8388608  net.core.rmem\_default = 8388608  net.core.rmem\_max = 16777216  net.core.wmem\_max = 16777216  EOF  sysctl -p  #echo 262144 > /proc/sys/net/core/rmem\_default  #echo 262144 > /proc/sys/net/core/rmem\_max |

1.6.11.2 企业生产场景NFS共享存储优化

1) 硬盘:sas/ssd硬盘，买多块，raid0/raid10。网卡吞吐量要大，至少千兆(多块bond)

2) NFS服务器端配置:

|  |
| --- |
| # vim /etc/exports  /data 10.0.0.0/24(rw,sync,all\_squash,anonuid=65534,anongid=65534 ) |

3) NFS客户端挂载：rsize,wsize,noatime,nodirtime,nosuid,noexec,soft(hard intr)

|  |
| --- |
| # mount -t nfs -o nosuid,noexec,nodev,noatime,nodiratime,rsize=131072　10.0.0.7:/data/ /mnt  # mount -t nfs -o noatime,nodirtime,rsize=131072,wsize=131072 10.0.0.7:/data /mnt  #　mount -t nfs -o noatime,nodirtime 10.0.0.7:/data /mnt |

　4）有关NFS服务的所有服务器内核优化

|  |
| --- |
| ＃　cat >>/etc/sysctl.conf<<EOF  net.core.wmem\_default = 8388608  net.core.rmem\_default = 8388608  net.core.rmem\_max = 16777216  net.core.wmem\_max = 16777216  EOF  ＃　sysctl -p　＃执行生效 |

　5)如果卸载的时候提示：umount:/mnt:device is busy

需要退出挂载目录在进行卸载，或者是NFS server 宕机了，需要强制卸载

|  |
| --- |
| ＃　umount -lf /mnt |

　6)大型网站NFS网络文件系统替代软件:分布式文件系统

moosefs(mfs),glusterfs,FastDFS

# 1.7 NFS 系统应用优缺点说明

优点：

* 简单，容易上手，容易掌握
* NFS文件系统内数据是在文件系统之上的，即数据能看的见的
* 方便，部署快速，维护简单，可控满足需求就是最好的
* 可靠，从软件层面上看，数据可靠性高，经久耐用，数据实在文件系统之上的
* 稳定，非常稳定

局限：

* 存在单点故障，如果NFS SERVER宕机了所有客户端都不能访问共享目录，这个在后期的课程会通过负载均衡及高可用性方案弥补
* 在大数据高并发的场合，NFS效率\性能有限(一般几千万以下PV的网站无瓶颈，除非网站架构太差，2千万PV/日)
* 客户端认证时，基于IP和主机名的，权限时根据ID识别，安全性一般
* NFS数据是明文的，NFS本身对数据完整性不做验证
* 多台客户端挂载一个NFS服务器时，连接管理维护麻烦(耦合度高)，特别是服务端出问题时，客户端会挂掉

# 1.8 showmount 命令说明

|  |  |  |
| --- | --- | --- |
| 短格式 | 长格式 | 用途及实例结果 |
| -e | --exports | 显示NFS服务器输出的目录列表  # showmount -e 10.0.0.14  Export list for 10.0.0.14  /oldboy 10.0.0.0/24  # showmount --exports 10.0.0.14  Export list for 10.0.0.14  /oldboy 10.0.0.0/24 |
| -d | --directories | 显示NFS服务器中提供共享的目录  # showmount -d 10.0.0.14  Directories on 10.0.0.14: /oldboy |
| -a | all | 以ip:/dir格式显示NFS服务器的IP地址和可被挂载的目录  # showmount -a 10.0.0.14  All mount points on 10.0.0.14:  10.0.0.7:/data |

# 1.9 exportfs命令介绍

exportfs -rv==/etc/init.d/nfs reload

加载配置生效通过exportfs命令，我们可以管理当前NFS共享的文件系统列表

当我们在启动了NFS服务之后，此时如果修改了/etc/exports，就需要重新启动nfs，使修改的配置生效，这个时候我们就可以用exportfs命令来完成。

exportfs命令的基本语法:

|  |
| --- |
| /usr/sbin/exportfs[-avi][-o options,...][client:/path]  /usr/sbin/exportfs -r [-v] 相当于/etc/init.d/nfs reload  /usr/sbin/exportfs [-av] -u [client:/path...]  /usr/sbin/exportfs [-v]  /usr/sbin/exportfs -f或者  /usr/sbin/exportfs[-a] [-v] [-u] [-i] [-fFile] [-oOptions[,Option...]][Directory] |

## 1.9.1 exportfs 命令参数说明和实例

# exportfs -o rw,sync 10.0.0.107:/data

#　showmount -e localhost

＃ exportfs -o rw,sync,all\_squash 10.0.0.0/24:/data

提示:reload与exportfs的区别是，reload就是调用的/usr/sbin/exportfs -r命令

## 1.9.2 NFS server端的防火墙控制

说明：真正企业生产环境的存储服务器都属于内网环境，都无需防火墙，

因此，此处可以不配置，如果有需求需要配置的话可有两种方法，任选其一

1)仅允许内部IP段访问(最佳)

|  |
| --- |
| iptables -A INPUT -s 10.0.0.0/24 -j ACCEPT |

1. 允许IP端加端口访问

|  |
| --- |
| iptables -A INPUT -i eth1 -p tcp -s 10.0.0.0/24 --dport 111 -j ACCEPT  iptables -A INPUT -i eth1 -p udp -s 10.0.0.0/24 --dport 111 -j ACCEPT  iptables -A INPUT -i eth1 -p ucp -s 10.0.0.0/24 --dport 2049 -j ACCEPT  iptables -A INPUT -i eth1 -p udp -s 10.0.0.0/24 -j ACCEPT |

## 1.9.3 NFS常见故障排除

1) 客户端挂载报错：No such file or directory

使用showmount -e 10.0.0.107命令查看有没有共享目录

如果没有看到，就使用命令mkdir /data创建一个共享目录

2) 启动NFS服务器端服务时出现的故障：

Starting NFS quotas: Cannot register service: RPC: Unable to receive: errno = Connection refused rpc.rquotad: unable to register ....

Start NFS deamon: rpc nfsd: unable to resolve ANYADDR:nfs to inetaddress:Servname not supported for ai\_socktype

rpc.nfsd: unable to resolve ANYADDR:nfs to inet6 addfess:Servname not supported for ai\_socktype

rpc.nfsd: unable to set any sockets for nfsd

故障分析：/etc/services文件内容或文件类型格式不正确

故障解决：还原/etc/services文件

3) 在执行shownmount -e 10.0.0.107时出现以下故障。

clnt\_create: RPC: Port mapper failure - Unalbe to receive: error 113(No route to host)

故障分析:防火墙导致NFS服务不通

解决办法：iptables -F,setenforce 0

## 1.9.4 NFS客户端开机自启动挂载

配置客户端mount挂载命令使挂载开机自动执行，这里有两种方法，如下第一种方法，将挂载命令放在/etc/rc.local里

缺点:偶尔开机挂载不上，工作中除了开机自启动配置，还要对是否挂载做监控第二种方法:将挂载命令放在/etc/fstab里其实这是配置方法的一个误区,如下 fstab会优先于网络被Linux系统加载，网络没启动时执行fstab会导致连不上NFS 服务器端，无法实现开机挂载，而且，及时本地的文件系统，也要注意，fstab最后两列要设置0 0 否则有可能导致无法启动服务器的问题但是,如果在开机自启动服务里设置并启动netfs服务，放入fstab里有可以开机挂载的

提示：每个web服务器的站点目录的权限UID的配置要统一，最好和nfs 共享目录的权限的uid一致，这样才不会出现权限问题

## 1.9.5 NFS参考文献

http://www.tldp.org/HOWTO/NFS-HOWTO/

## 1.9.6 本章重点回顾

NFS服务的访问原理流程

NFS作为集群共享存储角色的搭建，部署

NFS作为集群共享角色的排障，高级优化(会口述)

mount的知识及参数,如-o(noatime,nodiratime,noexec,nosuid,rsize,wsize等)

fstab问价的知识

常用命令showmount exportfs,unmount(lf),rpcinfo

NFS的优点，缺点，适合的应用场景，替代产品(FastDFS,Moosefs(mfs),GlusterFS)

了解autofs