Linux NFS: Network File System Client and Server Complete Guide

Submitted by **Sarath Pillai (http://plus.google.com/110882921997234815487?rel=author)** on Mon, 04/29/2013 - 18:21

Tweet 7 (#)

(#)

During the infancy period of computer's, SUN Microsystems developed a technology, which made accessing files from other computer's over network easy. This technology gave a new dimension to access data through network, and is heavily used even today in organizations. The advantage of this technology is the fact that it made directories and partition's on remote system's available as if it is a local directory or file system.

RFC No 1094 (http://tools.ietf.org/html/rfc1094) from IETF, is dedicated to this technology called as Network File System or NFS. NFS work's over IP protocol, and hence it can be made available to any system, that works on TCP/IP. This made NFS to play a major role in the central storage system. Although the initial version of NFS worked over UDP rather than TCP, later advancement's and iterations in the protocol made it work over TCP as well.

The initial version called NFS Version 1, was never released to the general public, but was used by

SUN internally for experiments. Like http,except NFS Version 4, all other version's were stateless.

Note: A stateless protocol is the one in which each and every request is made without the knowledge or information about the previous requests. Which means each and every request is independent of each other.

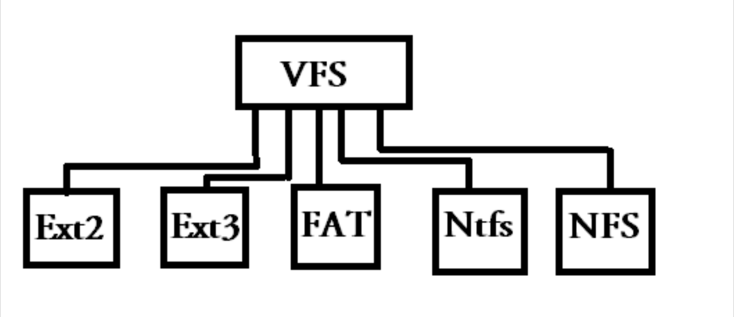
Different file systems are accessible on a single machine with the help of an API called as Virtual File System. Virtual File System takes care of all the file system's that are mounted on a system. Similarly VFS also takes care of NFS. Hence the operating system feels like it's accessing a file system which is local. The **read and write performance (http://www.slashroot.in/linux-file-system-read-write-performance-test)** will be slow compared to a local file system because all the operation is carried out through network(But due to the advancement in networking technology, even this performance bottleneck has improved substantially). Let's now go through some of the noteworthy points about NFS.

- A NFS file system mounted is very similar to a local file system on the machine
- NFS does not disclose the location of a file on the network
- An NFS server can be made of a completely different architecture and operating system than the client
- It also never discloses the underlying file system on the remote machine

A little bit understanding of VFS will be an added advantage before we begin working with NFS.

What is VFS and how does NFS works over VFS?

VFS makes one simple interface for Linux Kernel to access different file system's under it. Today linux supports a large number of different file system's because of VFS. The below diagram will be helpful in understanding how VFS works.



(/how-does-indexing-makes-database-faster)

How does indexing makes database faster (/how-does-indexing-makes-database-faster)

Archives (/taxonomy/term/2) - **2 comment(s) (http://www.slashroot.in/how-does-indexing-makes-database-faster#comments#comments)**

(/iperf-how-test-network-speedperformancebandwidth)

IPERF: How to test network Speed,Performance, Bandwidth (/iperf-how-test-network-speedperformancebandwidth)

Archives (/taxonomy/term/2) - **10 comment(s) (http://www.slashroot.in/iperf-how-test-network-speedperformancebandwidth#comments#comments)**

As shown in the above figure, all requests for normal actions on the file system like for example finding the size of a file, creating and deleting files, are all done by VFS for the system(And these operations that are done by the VFS file system are called as vnode operations).

These simple operations that are done by the user, is later converted to real actions on the underlying file system. Hence even if there is an NFS partition/share mounted, all operations on that partition is also carried out by VFS. Hence the operating system feels that the mounted NFS file system is a local file system like other's.

Technologies like NFS and NIS rely on something called as RPC(Remote Procedure Call). RPC helps to convert normal operations like write data, change permissions, etc on the NFS mounted file system to a remote action on the NFS server, which is mounted on the client. It works like this, as soon as the VFS(Virtual File System) gets a request to do an operation on the NFS mount, NFS client converts that exact same request to an equivalent RPC request on the NFS server.

A timeout period is set to each and every RPC request that an NFS client makes to the server. If the server fails to respond to the request, within the timeout period then the client retransmits the request. Hence even if the NFS server goes down, the clients keeps on sending the RPC requests to the server, and as soon as the server comes back online, it carries normal operations as if nothing has happened.

Another technology that makes NFS work with different operating system and architecture's is called as XDR(External Data Representations). This technology enables different operating system's to use the same NFS share. A request is first converted to an XDR format request, which is then converted back to the local resident format on the server.

How to setup an NFS server?

For setting up NFS server and client, we will be using two machines. One for NFS client and the other for NFS server.

NFS Server - 192.168.0.105 (slashroot1)

NFS Client - 192.168.0.104 (slashroot2)

Let's start this by setting up our NFS server (192.168.0.105), on a Red Hat Enterprise Linux (RHEL 5). We will be first installing the required packages on the server, and then will analyze the difference between various NFS protocol versions and methods to implement it on the client.

(/curl-command-tutorial-linux-example-usage)

CURL command Tutorial in Linux with Example Usage (/curl-command-tutorial-linux-example-usage)

Archives (/taxonomy/term/2) - 10 comment(s) (http://www.slashroot.in/curl-command-tutorial-linux-example-usage#comments#comments)

(/san-vs-nas-difference-between-storage-area-network-and-network-attached-storage)

SAN vs NAS - Difference between a Storage Area Network and Network Attached Storage (/san-vs-nas-difference-between-storage-area-network-and-network-attached-storage)

Archives (/taxonomy/term/2) - 42 comment(s) (http://www.slashroot.in/san-vs-nas-difference-between-storage-area-network-and-network-attached-storage#comments#comments)

Most Commented

How Does Traceroute Work and

For setting up NFS server on RHEL and Centos, you require two important packages mentioned below.

- nfs-utils
- portmap

nfs-utils package provides the NFS server capabilities and its required tools. Portmap package is the one that listens for requests that are RPC based. Portmap is very much necessary if you are running nfs version 1,2,or 3.

As we discussed before, the requests made by the client is first translated to an RPC equivalent request, which is then accepted by the RPC daemon on the NFS server (Which is provided by portmap package in rhel/centos). Lets verify if these two packages are installed on the server.

```
[root@slashroot1 ~]# rpm -qa | grep portmap
portmap-4.0-65.2.2.1
[root@slashroot1 ~]# rpm -qa | grep nfs
nfs-utils-1.0.9-40.el5
nfs-utils-lib-1.0.8-7.2.z2
[root@slashroot1 ~]#
```

We have already these packages installed on our server machine. If you do not have these, you can use **yum package manager (http://www.slashroot.in/yum-repository-and-package-management-complete-tutorial)** to install them, as shown below.

```
[root@slashroot1 ~]# yum install nfs-utils portmap
```

The easiness of configuring something in linux lies in the number of configuration file to edit. NFS does a good job in that, because you only need to edit a single file called **/etc/exports** to get started. The file initially is blank, and needs to be configured according to the requirement.

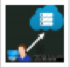
The entries in the **/etc/exports** file is quite easy to understand. Lets go through the method used to make an entry in that file.

SHARE PATH : CLIENT(OPTIONS) CLIENT(OPTIONS)

SHARE PATH - This mentions the folder to share. For example if you want to share the folder called as **/data,** you need to enter /data here.

CLIENT - Client specifies the hostname/IP address that will be allowed to access this share. In other words machines that are mentioned here will be able to mount this share.

OPTIONS- This is the main part in NFS that needs an extra attention, because the options mentioned here will have an impact on performance, as well as how the share is accessed by the clients.



(/how-does-traceroute-work-and-examples-using-traceroute-command)

Example's of using traceroute command (/how-does-traceroute-work-and-examples-using-traceroute-command)

Networking (/networking) - 56 comment(s) (http://www.slashroot.in/how-does-traceroute-work-and-examples-using-traceroute-command#comments#comments)

(/san-vs-nas-difference-between-storage-area-network-and-network-attached-storage)

SAN vs NAS - Difference between a Storage Area Network and Network Attached Storage (/san-vs-nas-difference-between-storage-area-network-and-network-attached-storage)

Archives (/taxonomy/term/2) - 42 comment(s) (http://www.slashroot.in/san-vs-nas-difference-between-storage-area-network-and-network-attached-storage#comments#comments)

(/difference-between-iterative-and-recursive-dns-query)

difference between

There are multiple options that are available. Let's go through some of them, and understand what they does.

ro: This option stands for Read Only. Which means the client has only the permission to read data from the share, but no write permission.

rw: This options stands for Read and Write. This allows the client machine to have both read and write permissions on the directory

no_root_squash: This option needs to be understood very carefully, because it can become a security bottleneck on the server. If the user **"root"** on the client machine mounts a particular share from the server, then by default, the requests made by the root user is fulfilled as a user called **"nobody",** instead of root. Which is a plus point as far as security is concerned, because no root user on the client machine can harm the server, because the requests are not fulfilled as root but as nobody.

using **no_root_squash** option will disable this feature and requests will be performed as root instead of nobody. Which means the client root user will have the exact same accessibility on the share, as the server root user.

I will never recommend using this option unless you require it for some explicit reasons.

async: As i have mentioned in the VFS (Virtual File System) section, each and every request to the nfs server from the client is first converted to an RPC call and then submitted to the VFS on the server. The VFS will handle the request to the underlying file system to complete.

Now if you use async option, as soon the request is handled over the underlying file system to fulfill, the nfs server replies to the client saying the request has completed. The NFS server does not wait for the complete write operation to be completed on the underlying physical medium, before replying to the client.

Although this makes the operation a little faster, but can cause data corruption. You can say that the nfs server is telling a lie to the client that the data is written to the disk(What happens if the server gets rebooted at this time..there is no trace of data..)

sync: The sync option does the reverse. In this case the NFS server will reply to the client only after the data is completely written to the underlying medium. This will result in a slight performance lag.

iterative and recursive dns query (/difference-between-iterative-and-recursive-dns-query)

Archives (/taxonomy/term/2) - 34 comment(s) (http://www.slashroot.in/difference-between-iterative-and-recursive-dns-query#comments#comments)

(/logstash-tutorial-linux-central-logging-server)

Logstash Tutorial: Linux Central logging server (/logstash-tutorial-linux-central-logging-server)

Archives (/taxonomy/term/2) - 31 comment(s) (http://www.slashroot.in/logstash-tutorial-linux-central-logging-server#comments#comments)

Top Rated Articles

(/configuring-lvm-top-raid-alternative-way-partitioning-raid-devices)

Configuring LVM on top of RAID : An alternative way to partitioning RAID Devices. (/configuring-lvm-top-raid-alternative-way-partitioning-raid-devices)

How to make a NFS share in Linux

As we have installed the required package, and also went through the method and options used to share. Let's now make a share and mount it on the client machine.

```
[root@slashroot1 ~]# cat /etc/exports
/data 192.168.0.104(rw)
[root@slashroot1 ~]#
```

You can clearly see from the above share that we have shared **/data** to a NFS client called **192.168.0.104,** and gave both read and write permissions.

Let's now start the required services on the NFS server for this share to work.

```
[root@slashroot1 ~]# /etc/init.d/nfs start
Starting NFS services:                                     [  OK  ]
Starting NFS quotas:                                       [  OK  ]
Starting NFS daemon:                                       [  OK  ]
Starting NFS mountd:                                       [  OK  ]
[root@slashroot1 ~]# /etc/init.d/portmap start
Starting portmap:                                          [  OK  ]
```

Dont worry we will be going through those extra services that got started aling with nfs (NFS quotas, NFS daemon,NFS mountd) etc. Adding these services to startup services is advisable, as the client's will not experience any problem's after a server reboot.

Lets put these services to startup script using chkconfig command.

```
[root@slashroot1 ~]# chkconfig nfs on
[root@slashroot1 ~]# chkconfig portmap on
```

Now let's get this share mounted on the client(slashroot2 - 192.168.0.104) with the help of the mount command.

```
[root@slashroot2 ~]# mount 192.168.0.105:/data /mnt
[root@slashroot2 ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda1              38G  5.6G   31G  16% /
tmpfs                 252M     0  252M   0% /dev/shm
192.168.0.105:/data    38G  2.8G   34G   8% /mnt
[root@slashroot2 ~]#
```

Note: You need to have portmap service running on both the server and clients for mounting files through NFS. Most of the distributions have that enabled by default.

So we have mounted an NFS export from the server 192.168.0.5 on /mnt. The client can now carry out normal operation's on the share as if it is on any local hard disk. **Red Hat enterprise linux uses NFS Version 3, by default when you mount a directory on the client without specifying any options.**

Average: 5 (29 votes)

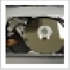
how to add an init script for nginx service (/how-add-init-script-nginx-service)

Average: 5 (1 vote)

(/understanding-file-system-superblock-linux)

Understanding File System Superblock in Linux (/understanding-file-system-superblock-linux)

Average: 5 (3 votes)


(/software-raid0-configuration-linux)

Software RAID 0 Configuration in linux (/software-raid0-configuration-linux)

Average: 5 (16 votes)

(/how-compile-c-program-linux-machine)

how to compile c program in linux machine? (/how-compile-c-program-linux-machine)

Average: 5 (9 votes)

Let's go through different processes on the server that enable's the working of NFS.

rpc.mountd: This is the process on the server that waits for mount requests from the clients. This process confirms the exports on the server. starting the nfs service on the server also start's this service.

rpc.statd : This process keeps the nfs clients informed about the current status of the nfs server. This process also gets started if you start the nfslock service from the init script.

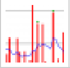
rpc.quotad: This process implements the same quota related details on the local system for remote user's.

All communications in NFS till version 3(Default) happens through a service called as portmap. So to list the NFS services that are running on the server, we need to see which port is running for what service through RPC. This will also list services that are not related to NFS, because RPC is used by many other services.

? (#)

```
[root@slashroot1 ~]# rpcinfo -p
   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp    880  status
    100024    1   tcp    883  status
    100011    1   udp    620  rquotad
    100011    2   udp    620  rquotad
    100011    1   tcp    623  rquotad
    100011    2   tcp    623  rquotad
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100003    4   udp   2049  nfs
    100021    1   udp  60385  nlockmgr
    100021    3   udp  60385  nlockmgr
    100021    4   udp  60385  nlockmgr
    100003    2   tcp   2049  nfs
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100021    1   tcp  54601  nlockmgr
    100021    3   tcp  54601  nlockmgr
    100021    4   tcp  54601  nlockmgr
    100005    1   udp    649  mountd
    100005    1   tcp    652  mountd
    100005    2   udp    649  mountd
    100005    2   tcp    652  mountd
    100005    3   udp    649  mountd
    100005    3   tcp    652  mountd
[root@slashroot1 ~]#
```

rpcinfo command as shown above can be used to list services and ports that are running through RPC. So all the initial request to mount a remote share comes to port 111 (Whic is RPC default port) on the server, which provides the port number's associated with different NFS services.


(/httperf-web-server-performance-test)

httperf web server performance test (/httperf-web-server-performance-test)

Average: 5 (1 vote)

(/software-raid-1-configuration-linux)

Software RAID 1 Configuration in Linux (/software-raid-1-configuration-linux)

Average: 5 (25 votes)

(/how-configure-basic-apache-reverse-proxy)

How to Configure Basic Apache Reverse Proxy (/how-configure-basic-apache-reverse-proxy)

Average: 5 (5 votes)

Services that run through RPC inform's portmap whenever they start, portmap will assign a port number and a program number to the process. the client's that requires connection to one of the RPC service contacts portmap on port 111, which intern will give the port number for the requested program.

As mentioned earlier Linux machine's by default uses UDP for NFS, unless you specify TCP while mounting at the client side. Hence you can see both the TCP and UDP ports running on the server through portmap. NFS version 4 implements this separately, which we will be discussing later in this post. If due to some reason, you are unable to see NFS port and program number registered in portmap, restarting nfs service will re-register the service with portmap.

NFS Client Mounting Options

NFS provides a lot variety of options while mounting shares on the client. The main reason for discussing this is that, most of the options are not enabled until the client specifies it with an explicit argument during the mount.

Let's go through some of the options that can be applied during NFS mount.

Let's get back to our previously mounted share on the client machine, and make it permanent by adding an fstab entry.

```
192.168.0.105:/data          /mnt                    nfs     defaults       0 0
```

That fstab entry shown above will mount the remote nfs share at boot time, with the default options. However there are some advanced options that can be applied to this mount.

There is something called as soft mounting and hard mounting in NFS. Lets understand them and them implement in our nfs mount.

Soft Mounting in NFS

Suppose you have once process on your client machine, say for example, an Apache web server is accessing a mounted share and files in it. But due to some problem on the NFS server, the request made by Apache for a file on the NFS share cannot be completed. In this case the NFS client will reply to the process (Apache in our case), with an error message.

Most of the processes will accept the error. But it all depends on how the process is designed to handle these errors. Sometimes it can cause unwanted behavior and even can corrupt the files.

A soft mount can be done by the following method.

```
192.168.0.105:/data             /mnt        nfs     rw,soft       0 0
```

Hard Mounting in NFS

Hard mounting works a little different than soft mounting. If a process that requires a file from the nfs share, cant access it due to some problem at the nfs server, the process will wait (kind of get's hang) till the nfs server becomes proper and completes its request. And the process will resume from the point where it was stopped when NFS server responds back properly.

The process that is waiting for the operation to get completed, cannot be interrupted. Yeah sure you can kill the process with a kill -9 command, and can also be interrupted with the help of an option called **intr**.

A hard mounting can be done by adding the options as shown below in fstab.

```
192.168.0.105:/data             /mnt                    nfs     rw,hard,intr    0 0
```

It all depends on the requirement and the kind of process that you are running on the mounted share. However a hard mount with intr option is advisable compared to soft mount.

There are some performance related mounting options that can be used in fstab. But i will be discussing them in a dedicated post called as performance tuning in NFS.

As i mentioned during the beginning of this tutorial, NFS has 4 different versions. By default, RHEL/Centos uses nfs version 3. You can toggle between different nfs version's during mount. So its the client that decides which NFS version to use while mounting. But the server must support it of course.

You can use NFS version 2 with the help of an argument called **vers** with mount as shown below.

```
[root@slashroot2 ~]# mount -o vers=2 192.168.0.105:/data /mnt
```

Please note that nfs version 1 is obsolete now, and most of the new nfs server machine's does not support that.

How to mount nfs version 4 on redhat/centos Linux

NFS version 4 implements the share in a different manner than nfs version 3 and other previous versions. It works on a root file system. Let's take one example to understand this.

```
[root@slashroot1 ~]# cat /etc/exports
/data 192.168.0.104(rw,fsid=0,sync)
[root@slashroot1 ~]#
```

In the above shown exports file, there is an additional option that i have added for nfs version 4. That is **fsid=0** using this option we specify that this is the root file system of the export that this server is providing. Which means the client has to just mount the default share by specifying "**/**" instead of /data. Lets see how

For mounting nfs4 share, you need to use the option nfs4 in mount command.

```
[root@slashroot2 ~]# mount -t nfs4 -o rw 192.168.0.105:/ /mnt
```

If you notice i have used **192.168.0.105:/** rather than using **192.168.0.105:/data** while mounting. that's simply because, we have mentioned in the server export that /data is the default root file system for nfs share from this server. Any share with fsid=0 argument becomes the root share of the server.

Client's can also mount the subdirectories inside the share bu using the absolute path. For example.

```
[root@slashroot2 ~]# mount -t nfs4 -o rw 192.168.0.105:/test /test
```

In the above shown example, we have mounted a subdirectory inside /data(which is share that the server has in its export), by keeping /data as the root directory. Which means we have mounted **/data/test** directly.

NFS version 4 has a lot of advantages over version 3. One of them is that, it bydefault work's over TCP and can be made available to hosts on the internet because it runs on 2049 TCP port without the need for portmap.

Yeah you dont require portmap service on the server to run nfs version 4. It directly contacts the TCP port 2049 on the server. There are additional security as well as performance advancement's that are included in NFS version 4 (Which we will be discussing in another dedicated post for NFS security.)

Points to note about NFS version 3 and version 4

- Version 3 of NFS started to support files that are larger than 2 gb.
- It was version 3 which started a performance option of **async,** which we saw earlier, to improve performance.
- NFS version 4 was made for better and ease of access over the internet
- NFS version 4 does not require portmap to run (Which is a vulnerable service)
- The virtual root directory concept, that we have implemented with the help of **fsid=0,** argument improves security
- NFS version 4 uses TCP bydefault for better reliablity over internet
- NFS Version 4 is capable of using 32KB page size which improves the performance a lot, compared to the default 1024 bytes.

I have not included the security and **performance details of NFS (http://slashroot.in/how-do-linux-nfs-performance-tuning-and-optimization)** in this tutorial, as they require special attention and will be discussed in their own dedicated posts.

Rate this article:
Average: 4.9 (127 votes)

Add new comment (/linux-nfs-network-file-system-client-and-server-complete-guide#comment-form)

Comments

NFS (/comment/85#comment-85)

Permalink (/comment/85#comment-85) Submitted by Bro on Sun, 06/16/2013 - 22:20

Hello! Many thanks again for awesome tutorial!

However, you claim that NFSv4 doesn't need portmap; but I find sources that v4 still needs portmap/rpcbind.

Are you sure that its matter of things and portmap/rpcbind can be dropped from dependencies for v4?

Also, how do you server to only accept connections via v4 and never lower?

Thanks a lot!

reply (/comment/reply/177/85)

Hi bro,Thanks for the comment (/comment/86#comment-86)

Permalink (/comment/86#comment-86) Submitted by slashmaster on Mon, 06/17/2013 - 09:13

Hi bro,

Thanks for the comment, and we are happy to know that you liked the article. Lets come back to our topic. We will first see how to disable all other versions of NFS except nfs version 4. You can easily do that by configuring the nfs configuration file **/etc/sysconfig/nfs**.

```
# Define which protocol versions mountd
# will advertise. The values are "no" or "yes"
# with yes being the default
MOUNTD_NFS_V1="no"
MOUNTD_NFS_V2="no"
MOUNTD_NFS_V3="no"
```

by default the above lines are commented out in that configuration file, you will need to uncomment them manually to disable version 1, 2, and three, as shown above.

As far as RPC is concerned it is still used in nfs version 4 with slight modifications. **The seperate program called portmaper which waits requests on port 111, and registers all the rpc related service is not required for nfs version 4.**

NFS version 4 uses only 2 rpc operations called NULL(sometimes used to test the response time, without any operations) and COMPOUND(a mechanism to send multiple rpc requests combined together in one request). The rpc daemons required are started by nfs init script itself.

Hope that helps.

reply (/comment/reply/177/86)

Awesome tutorial Thanks a million:) (/comment/137#comment-137)

Permalink (/comment/137#comment-137) Submitted by Irshad on Fri, 09/20/2013 - 14:39

Awesome tutorial Thanks a million:)

reply (/comment/reply/177/137)

quota over NFS (/comment/150#comment-150)

Permalink (/comment/150#comment-150) Submitted by David on Thu, 10/17/2013 - 05:01

Hi,

Im trying with no success to set up quotas over NFS, do you have a tutorial with the steps to do this?

Thanks in advance.

reply (/comment/reply/177/150)

Connection timed out on my NFS (/comment/514#comment-514)

Permalink (/comment/514#comment-514) Submitted by Joey Medrano on Fri, 12/27/2013 - 06:28

What a great article and it really open my mind to the options on NFS and how NFS relies on RPC. Anyhow, I was working on my NFS Server (SUSE) using NFSv3 only with an AIX client but I am getting this dmesg error below and would like to know if you could possibly assist in troubleshooting.

```
rpc-srv/tcp: nfsd: got error -104 when sending 132 bytes - shutting down socket
nfsd: peername failed (err 107)!
SFW2-OUT-ERROR IN= OUT=eth1 SRC=10.1.1.112 DST=10.1.1.1 LEN=52 TOS=0x00
PREC=0x00 TTL=64 ID=16571 DF PROTO=TCP SPT=2049 DPT=44155 WINDOW=384
RES=0x00 ACK RST URGP=0 OPT (0101080A02B665655372633E)
```

reply (/comment/reply/177/514)

Hi Joey, (/comment/517#comment-517)

Permalink (/comment/517#comment-517) Submitted by Sarath Pillai on Fri, 12/27/2013 - 15:01

Hi Joey,

It seems to me that you are having a performance issue. Do you have too many clients mounting from it?

Two suggestions. 1. Try changing tcp rmem And wmem (you can find that in my NSF performance tuning article 2. Also try ins creasing the number of nfs daemon on the server.

There is a known bug related to this.. Quite not sure if the error u are getting is same.. Let me know..

Regards

reply (/comment/reply/177/517)

Absolutely amazing post. (/comment/783#comment-783)

Permalink (/comment/783#comment-783) Submitted by pmu on Sat, 07/12/2014 - 00:48

Hi,

I am a SAN guy scourging the net for NFS related articles as pretty soon I'll end up working on NAS.

This is by far the best article I have come across.

Thank you very much for writing such an excellent article.

reply (/comment/reply/177/783)

nfs client moves file between dirs in the same server? (/comment/805#comment-805)

Permalink (/comment/805#comment-805) Submitted by mike on Thu, 07/24/2014 - 08:13

will the nfs server changes the directory layout locally without sending the file to the nfs client?

reply (/comment/reply/177/805)

Add new comment

Your name

Subject

Comment *

- No HTML tags allowed.
- Web page addresses and e-mail addresses turn into links automatically.
- Lines and paragraphs break automatically.

More information about text formats (/filter/tips)

Word verification *

(**verify using audio (#)**)

Type the characters you see in the picture above; if you can't read them, submit the form and a new image will be generated. Not case sensitive.

Save Preview

Related articles

- **Difference Between Monolithic and Microservices based Architecture (/difference-between-monolithic-and-microservices-based-architecture)**
- **Docker Network Configuration Tutorial (/docker-network-configuration-tutorial)**
- **How To Configure Split Horizon DNS in BIND (/how-to-configure-split-horizon-dns-in-bind)**
- **LVS(Linux Virtual Server) - Different Methods of Load Balancing (/lvs-linux-virtual-server-different-methods-of-load-balancing)**
- **Understanding File System Superblock in Linux (/understanding-file-system-superblock-linux)**

Get in touch with The Authors

Sarath Pillai

Ph: +917303074400

Email: **sarath@slashroot.in (mailto:sarath@slashroot.in)**

Satish Tiwary

Ph: +919509452488

Email: **satish@slashroot.in (mailto:satish@slashroot.in)**

Follow Us

Subscribe to our RSS Feed (rss.xml)

Follow us on Twitter (https://twitter.com/slashrootin)

Be a fan on Facebook (http://www.facebook.com/pages/Slashrootin-A-technical-Blog/101642306662299)

Our News Letter

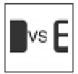
* indicates required

Email Address

*

Subscribe

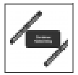
Recent Posts



(/difference-between-monolithic-and-microservices-based-architecture)

Difference Between Monolithic and Microservices based Architecture (/difference-between-monolithic-and-microservices-based-architecture)

Archives (/taxonomy/term/2) - *3 months 1 week* ago



(/docker-network-configuration-tutorial)

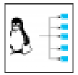
Docker Network Configuration Tutorial (/docker-network-configuration-tutorial)

Archives (/taxonomy/term/2) - *3 months 1 week* ago

(/how-to-configure-split-horizon-dns-in-bind)

How To Configure Split Horizon DNS in BIND (/how-to-configure-split-horizon-dns-in-bind)

Archives (/taxonomy/term/2) - *5 months 2 weeks* ago



(/lvs-linux-virtual-server-different-methods-of-load-balancing)

LVS(Linux Virtual Server) - Different Methods of Load Balancing (/lvs-linux-virtual-server-different-methods-of-load-balancing)

Archives (/taxonomy/term/2) - *5 months 2 weeks* ago

Last Viewed

(/telnet-server-linux) **telnet server on linux (/telnet-server-linux)**

Archives (/taxonomy/term/2) - last view *18 sec* ago

(/how-do-linux-nfs-performance-tuning-and-optimization)

How to do Linux NFS Performance Tuning and Optimization (/how-do-linux-nfs-performance-tuning-and-optimization)

Archives (/taxonomy/term/2) - last view *28 sec* ago

(/puppet-tutorial-introduction-puppet-configuration-management-tool)

Puppet Tutorial: Introduction to Puppet Configuration Management Tool (/puppet-tutorial-introduction-puppet-configuration-management-tool)

Archives (/taxonomy/term/2) - last view *30 sec* ago

(/how-to-use-wkhtmltopdf-to-convert-url-to-pdf)

how to use wkhtmltopdf to convert URL to PDF (/how-to-use-wkhtmltopdf-to-convert-url-to-pdf)

Technical News (/technical-news) - last view *42 sec* ago